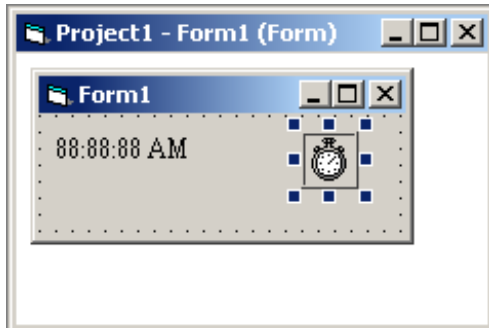


VB & Comm – Introduction

Example 1-Clock

Control	Name
Form	Form1
Timer	Timer1
Label	Label1



```
Private Sub Form_Load()  
    Form1.Caption = "Clock"  
    Timer1.Interval = 1000  
    Timer1.Enabled = True  
    Label1.FontSize = 22  
    Label1.FontBold = True  
    Label1.Font = "Times New Roman"  
End Sub
```

```
Private Sub Timer1_Timer()  
    Label1.Caption = Time  
End Sub
```

VB & Comm – Introduction

Example 2-NotePad

Control	Name
Form	Form1
Textbox	TxtPad
	Multilane = true
	Scrollbar = Vertical
CommandButton	CmdCut
CommandButton	CmdPaste
CommandButton	CmdClear

```
Dim sClipboard As String
```

```
Private Sub Form_Load()
```

```
    Form1.BorderStyle = 1 'Fixed single = 1  
    Form1.Caption = "Note Pad"
```

```
    txtPad.Text = ""
```

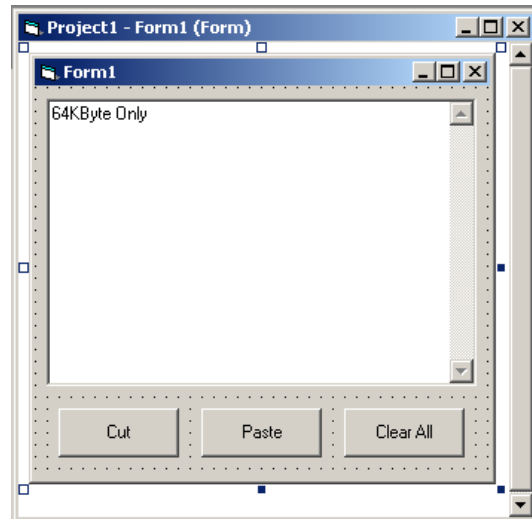
```
End Sub
```

```
Private Sub cmdCut_Click()  
    sClipboard = txtPad.SelText  
    txtPad.SelText = "" ' clear selected text  
    cmdPaste.Enabled = True  
    txtPad.SetFocus  
End Sub
```

```
Private Sub cmdPaste_Click()  
    txtPad.SelText = sClipboard 'Paste to cursor position  
    txtPad.SetFocus  
End Sub
```

```
Private Sub cmdClear_Click()  
    txtPad.Text = ""  
    txtPad.SetFocus  
End Sub
```

```
Private Sub txtPad_Change()  
    cmdCut.Enabled = True  
    cmdClear.Enabled = True  
End Sub
```



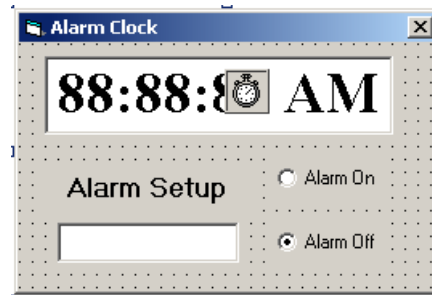
VB & Comm – Introduction

Example 3-Alarm

Control	Name
Form	Form1
TextBox	txtTime
TextBox	txtSetTime
Label	Label1, Content = “Alarm Setup”
OptionButton	OptAlarm()

Dim AlarmOn As Boolean

```
Private Sub Form_Load()  
    AlarmOn = False  
    txtTime.Text = Time  
    txtTime.Enabled = False  
    optAlarm(0).Value = False  
    optAlarm(1).Value = True  
End Sub
```



```
Private Sub optAlarm_Click(Index As Integer)  
    Select Case Index  
        Case 0  
            AlarmOn = True  
        Case 1  
            AlarmOn = False  
    End Select  
End Sub
```

```
Private Sub Timer1_Timer()  
    txtTime.Text = Time  
    If (AlarmOn) And IsDate(txtSetTime.Text) Then  
        If Time = CDate(txtSetTime.Text) Then  
            Beep  
            MsgBox Str(CDate(txtSetTime.Text))  
        End If  
    End If  
End Sub
```

```
Private Sub txtSetTime_KeyPress(KeyAscii As Integer)  
    Dim sKey As String  
    sKey = Chr$(KeyAscii)  
    'If ((sKey < "0" Or sKey > "9") And (sKey <> ":")) Then  
    ' KeyAscii = 0  
    'End If  
End Sub
```

VB & Comm – Introduction

Example 4-Notepad /w menu

```
Dim sClipboard As String
```

```
Private Sub Form_Load()
```

```
    Form1.BorderStyle = 1 'Fixed single = 1
```

```
    Form1.Caption = "Note Pad"
```

```
    txtPad.Text = ""
```

```
End Sub
```

```
Private Sub cmdCut_Click()
```

```
    sClipboard = txtPadSelText
```

```
    txtPadSelText = "" ' clear selected text
```

```
    cmdPaste.Enabled = True
```

```
    txtPad.SetFocus
```

```
End Sub
```

```
Private Sub cmdPaste_Click()
```

```
    txtPadSelText = sClipboard 'Paste to cursor position
```

```
    txtPad.SetFocus
```

```
End Sub
```

```
Private Sub cmdClear_Click()
```

```
    txtPad.Text = ""
```

```
    txtPad.SetFocus
```

```
End Sub
```

```
Private Sub txtPad_Change()
```

```
    cmdCut.Enabled = True
```

```
    cmdClear.Enabled = True
```

```
End Sub
```

```
Private Sub MnuClear_Click()
```

```
    cmdClear_Click
```

```
End Sub
```

```
Private Sub MnuCut_Click()
```

```
    cmdCut_Click
```

```
End Sub
```

```
Private Sub MnuPaste_Click()
```

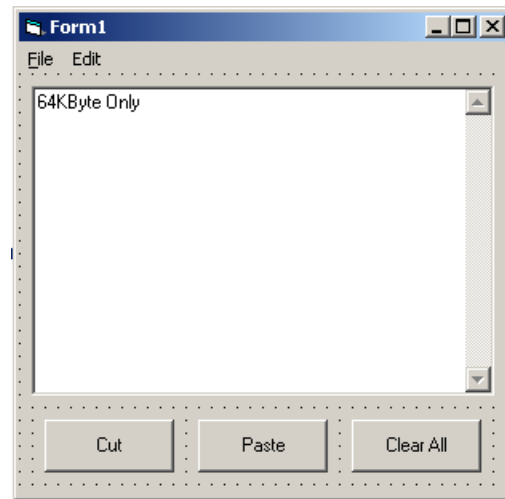
```
    cmdPaste_Click
```

```
End Sub
```

```
Private Sub MnuExit_Click()
```

```
    End
```

```
End Sub
```



VB & Comm – Introduction

OPEN STATEMENT

Open *pathname* **For mode** [**Access** *access*] [*lock*] **As** [#]*filenumber* [**Len=***reclength*]

The **Open** statement syntax has these parts:

Part	Description
<i>pathname</i>	Required. String expression that specifies a file name — may include directory or folder, and drive.
<i>mode</i>	Required. Keyword specifying the file mode: Append, Binary, Input, Output, or Random. If unspecified, the file is opened for Random access.
<i>access</i>	Optional. Keyword specifying the operations permitted on the open file: Read, Write, or Read Write.
<i>lock</i>	Optional. Keyword specifying the operations restricted on the open file by other processes: Shared, Lock Read, Lock Write, and Lock Read Write.
<i>filenumber</i>	Required. A valid file number in the range 1 to 511, inclusive. Use the FreeFile function to obtain the next available file number.
<i>reclength</i>	Optional. Number less than or equal to 32,767 (bytes). For files opened for random access, this value is the record length. For sequential files, this value is the number of characters buffered.

Remarks - You must open a file before any I/O operation can be performed on it. Open allocates a buffer for I/O to the file and determines the mode of access to use with the buffer.

If the file specified by *pathname* doesn't exist, it is created when a file is opened for **Append**, **Binary**, **Output**, or **Random** modes.

If the file is already opened by another process and the specified type of access is not allowed, the **Open** operation fails and an error occurs.

The **Len** clause is ignored if *mode* is **Binary**.

Important In **Binary**, **Input**, and **Random** modes, you can open a file using a different file number without first closing the file. In **Append** and **Output** modes, you must close a file before opening it with a different file number.

VB & Comm – Introduction

Example 5-Order Form

Control	Name
Form	Form1
Combobox	cboItem
Textbox	txtDesc, txtQty
List	List1
Command Button	CmdAdd, cmdDel, cmdSave

```
Dim sItem(4, 1) As String  
Const Filename = "temp.txt"
```

```
Private Sub Form_Load()
```

```
    'initialization
```

```
    lblTitle.Caption = "VB List / Combo Example"
```

```
    lblTitle.FontName = "Times New Roman"
```

```
    lblTitle.FontSize = 28
```

```
    lblTitle.FontBold = True
```

```
    sItem(0, 0) = "DJ200  "
```

```
    sItem(0, 1) = "DeskJet 200    "
```

```
    sItem(1, 0) = "DJ400  "
```

```
    sItem(1, 1) = "DeskJet 400    "
```

```
    sItem(2, 0) = "DJ880c  "
```

```
    sItem(2, 1) = "DeskJet 880 Color"
```

```
    sItem(3, 0) = "LJ4040TN"
```

```
    sItem(3, 1) = "Laserjet 4050TN "
```

```
    sItem(4, 0) = "LJ5000  "
```

```
    sItem(4, 1) = "Lastejet 5000  "
```

```
    For i = 0 To UBound(sItem, 1)
```

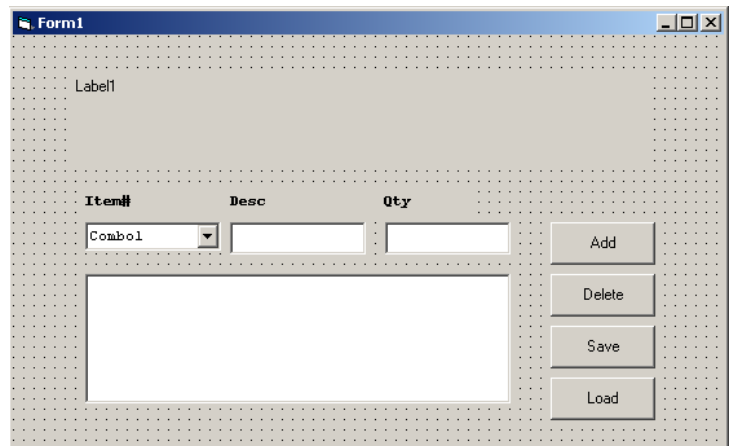
```
        cboItem.AddItem sItem(i, 0)
```

```
    Next i
```

```
    cboItem.ListIndex = 0
```

```
    cmdDel.Enabled = False
```

```
End Sub
```



VB & Comm – Introduction

```
Private Sub cmdAdd_Click()
```

```
    If Len(txtQty) > 0 Then
```

```
        List1.AddItem sItem(cboItem.ListIndex, 0) & " " & _  
            sItem(cboItem.ListIndex, 1) & " " & _  
            txtQty
```

```
    Else
```

```
        MsgBox "Please input correct Qty.", vbCritical
```

```
    End If
```

```
    txtDesc.Text = ""
```

```
    txtQty.Text = ""
```

```
    cboItem.SetFocus
```

```
End Sub
```

```
Private Sub cmdDel_Click()
```

```
    i = List1.ListIndex
```

```
    If i >= 0 Then
```

```
        List1.RemoveItem (List1.ListIndex)
```

```
    End If
```

```
    If i > 0 Then
```

```
        List1.ListIndex = i - 1
```

```
    End If
```

```
End Sub
```

```
Private Sub cmdSave_Click()
```

```
    Dim i, j As Integer
```

```
    j = List1.ListCount
```

```
    Open Filename For Append Lock Write As #1
```

```
    For i = 0 To j - 1
```

```
        List1.ListIndex = i
```

```
        Write #1, List1.Text
```

```
    Next i
```

```
    Close #1
```

```
    MsgBox "File has been saved.", vbOKOnly
```

```
    cboItem.SetFocus
```

```
End Sub
```

VB & Comm – Introduction

```
Private Sub cmdLoad_Click()  
    Dim sTextLine As String  
    Open Filename For Input As #2  
  
    Do While Not EOF(2) ' Loop until end of file.  
        Line Input #2, sTextLine ' Read line into variable.  
        List1.AddItem Mid(sTextLine, 2, Len(sTextLine) - 2) 'remove ""  
    Loop  
  
    Close #2  
    MsgBox "File has been loaded.", vbOKOnly  
    cboItem.SetFocus  
  
End Sub  
  
Private Sub cboItem_GotFocus()  
    cmdDel.Enabled = False  
End Sub  
  
Private Sub cboItem_LostFocus()  
    txtDesc.Text = sItem(cboItem.ListIndex, 1)  
End Sub  
  
Private Sub List1_GotFocus()  
    cmdDel.Enabled = True  
End Sub
```


VB & Comm – Introduction

Example 6-Function, Procedure and scope of Variables

Filename - Ex6.bas

```
Dim abc(100) As Integer
Dim e As String
```

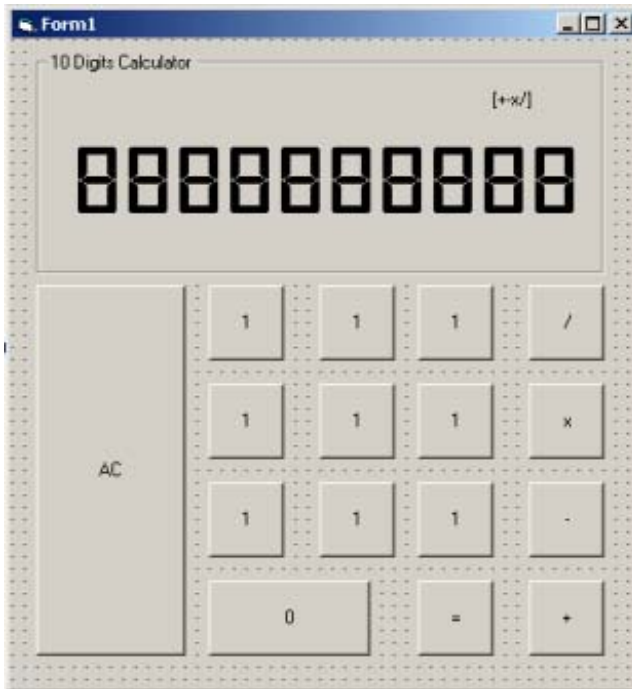
```
Private Sub main()
    Dim a, b, c As Integer
    e = "OK"
    MsgBox "e = " & e
    'Add Example
    a = 2
    b = 3
    'Variables a,b are passed to Function Add by Value
    c = Add(a, b)
    MsgBox "A = " & a & "; B = " & b & "; A + B = " & c
    'Subtract Example
    a = 4
    b = 5
    'Variables c is passed to subroutine by Reference
    Call Subtract(a, b, c)
    MsgBox "A = " & a & "; B = " & b & "; A - B = " & c
End Sub
```

```
'Function Add
Private Function Add(ByVal x As Integer, _
                    ByVal y As Integer) As Integer
    'Declare private variables a, b in Function Subtract
    Dim a, b As Integer
    a = 10: b = 20
    MsgBox "A = " & a & "; B = " & b & "; X = " & x & "; Y = " & y
    MsgBox "e = " & e
    'Place the answer to Add
    Add = x + y
End Function
```

```
'Subroution Subtract
Private Sub Subtract(ByVal u As Integer, _
                    ByVal v As Integer, _
                    ByRef w As Integer)
    w = u - v
    u = 50: v = 40
    'a and b are undefined in Subroution Subtract'
    MsgBox "A = " & a & "; B = " & b & "; U = " & u & "; V = " & v
    MsgBox "e = " & e
End Sub
```

VB & Comm – Introduction

Example 7-Calculator Demo



```
Const cOperators = "[+-x/]"
Dim iAns, iRegA, iRegB As Long
Dim sAns, sRegA, sRegb As String
Dim iOperator As Integer
Dim bNumPress, bEqualPress, bOprPress As Boolean
Dim bRegB, bRegA As Boolean
```

```
Private Sub cmdNum_Click(Index As Integer)
    If Len(lblLED.Caption) < 10 Then
        If bNumPress And bRegA Then
            sRegA = sRegA & Trim(Str(Index))
            iRegA = Val(sRegA)
            lblLED.Caption = Trim(Str(iRegA))
            bRegA = True
        ElseIf bRegB Then
            sRegb = sRegb & Trim(Str(Index))
            iRegB = Val(sRegb)
            lblLED.Caption = Trim(Str(sRegb))
            bRegB = True
        End If
    End If
    bNumPress = True: bEqualPress = False
    bOprPress = False
End Sub
```

VB & Comm – Introduction

Example 7- Demo Calculator – Cont...

```
Private Sub Form_Load()  
    'Variables Initialization  
    For i = 0 To 9  
        cmdNum(i).Caption = Trim(Str(i))  
    Next i  
    iAns = 0  
    lbooperator.Caption = cOperators  
    iRegA = 0: iRegB = 0: iAns = 0  
    sRegA = "": sRegb = "": sAns = ""  
    bRegA = True  
    lblLED.Caption = "0"  
    bNumPress = True  
    bEqualPress = False  
    bOprPress = False  
End Sub  
  
Private Sub cmdAC_Click()  
    lbooperator.Caption = cOperators  
    iRegA = 0: iRegB = 0: iAns = 0  
    sRegA = "": sRegb = "": sAns = ""  
    lblLED.Caption = "0"  
    bNumPress = True  
    bEqualPress = False  
    bOprPress = False  
End Sub  
  
Private Sub cmdOperator_Click(Index As Integer)  
    iOperator = Index  
    Select Case Index  
        Case 0  
            lbooperator.Caption = "[+]"  
        Case 1  
            lbooperator.Caption = "[-]"  
        Case 2  
            lbooperator.Caption = "[x]"  
        Case 3  
            lbooperator.Caption = "[/]"  
    End Select  
    bNumPress = False: bEqualPress = False  
    bOprPress = True  
    bRegA = Not bRegA: bRegB = Not bRegB  
End Sub
```

VB & Comm – Introduction

Example 7- Demo Calculator – Cont...

```
Private Sub cmdEqual_Click()  
    If Not bEqualPress Then  
        Select Case iOperator  
            Case 0  
                iAns = iRegA + iRegB  
            Case 1  
                iAns = iRegA - iRegB  
            Case 2  
                iAns = iRegA * iRegB  
            Case 3  
                iAns = iRegA / iRegB  
        End Select  
        lblLED.Caption = iAns  
        lbloperator.Caption = cOperators  
        iRegA = 0: iRegB = 0  
        sRegA = "": sRegb = ""  
    End If  
    bNumPress = False: bEqualPress = True  
    bOprPress = False  
    bRegA = Not bRegA: bRegB = Not bRegB  
End Sub
```

VB & Comm – Introduction

Visual Basic - Controls

* Object , Properties , Events , Methods

Example Control Object : Textbox



Properties : Text , BackColor ...

Events : Change, Click , KeyPress ...

Methods : Move, setFocus, ShowWhatsthis

Label



Properties	
Events	
Methods	

Button



Properties	
Events	
Methods	

Timer



Properties	
Events	
Methods	

Picture



Properties	
Events	
Methods	

Shape



Properties	
Events	
Methods	

Frame



Properties	
Events	
Methods	

Reference :- Standard Controls - Properties, Events, Methods

<http://www.thebestweb.com/vbarticles/VBPropertiesEventsMethods.htm>