Lecture 2

Lecture 2 – VB Part 2

Component: Labels (lblVariableName)
Code Programming: Procedures and String handling with str(someNumber)

Objective: Simple calculator

Label Control Properties:

- Conventional name prefix is “lbl” e.g. lblResult
- Caption font size, type and text color can be assigned from Properties box
- Typical uses: caption components like text boxes

Procedures and str

```
Private Sub cmdCalculate_Click()
    Multiply 2, 3
End Sub

Public Sub Multiply(X As Integer, Y As Integer)
    Dim Z
    Z = X * Y
    txtResult.Text = Str(Z)
End Sub
```

- When Calculate button is clicked, the Multiply procedure is instantiated.
- Multiply’s procedure is added as follows:
  1. Double click on the form, code will show up
  2. Tools-Add Procedure with Name: Multiply, Type: Sub, Scope: Public
  3. Add code as given above

- Dim is used to declare a variable e.g. Dim Z
- Can type cast variables with Dim Z as Integer
- Good practice: Option Explicit in General Declarations segment of code
- str(someNumber) is converted to string. Can be assigned to text field.
Component: Pop up message boxes

Code Programming: MsgBox “messageString”, DialogType, “titleString”

Objective: GUI’s should provide users a way to navigate a program. A pop up message box does this, which command buttons like YES, NO, OK, EXIT etc.

GUI Properties:
- One form (use frm prefix)
- Two command buttons (use cmd prefix)
- Message box does not have its own component

Message Box Programming:

```vbscript
Private Sub cmdMessage_Click()
    Dim Message As String
    Dim DialogType As Integer
    Dim Title As String

    'What message box should contain
    Message = "Hey! This is a sample message..."

    'want dialog box to have an OK button and
    'an exclamation icon
    DialogType = vbExclamation + vbOKOnly

    'Title of dialog box
    Title = "Dialog Demo"

    'display the dialog box
    MsgBox Message, DialogType, Title
End Sub
```
Lecture 2

Button Constants:

<table>
<thead>
<tr>
<th>Displayed Button</th>
<th>Constant Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>vbOKOnly</td>
</tr>
<tr>
<td>OK, Cancel</td>
<td>vbOKCancel</td>
</tr>
<tr>
<td>Abort, Retry, Ignore</td>
<td>vbAbortRetryIgnore</td>
</tr>
<tr>
<td>Yes, No, Cancel</td>
<td>vbYesNoCancel</td>
</tr>
<tr>
<td>Yes, No</td>
<td>vbYesNo</td>
</tr>
<tr>
<td>Retry, Cancel</td>
<td>vbRetryCancel</td>
</tr>
</tbody>
</table>

Icon Constants:

<table>
<thead>
<tr>
<th>Displayed Icon</th>
<th>Constant Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Critical icon</td>
<td>vbCritical</td>
</tr>
<tr>
<td>Warning query icon</td>
<td>vbQuestion</td>
</tr>
<tr>
<td>Warning message icon</td>
<td>vbExclamation</td>
</tr>
<tr>
<td>Information icon</td>
<td>vbInformation</td>
</tr>
</tbody>
</table>

Exercise:

- Add a YES/NO pop up message box for the Exit command button. The message should say "Are you sure you want to quit?" The title bar can be anything you want.
- If the user clicks YES, then end the program, if NO, then do nothing. Note the following:

<table>
<thead>
<tr>
<th>Clicked Button</th>
<th>Value</th>
<th>Constant Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>1</td>
<td>vbOK</td>
</tr>
<tr>
<td>Cancel</td>
<td>2</td>
<td>vbCancel</td>
</tr>
<tr>
<td>Abort</td>
<td>3</td>
<td>vbAbort</td>
</tr>
<tr>
<td>Retry</td>
<td>4</td>
<td>vbRetry</td>
</tr>
<tr>
<td>Ignore</td>
<td>5</td>
<td>vbIgnore</td>
</tr>
<tr>
<td>Yes</td>
<td>6</td>
<td>vbYes</td>
</tr>
<tr>
<td>No</td>
<td>7</td>
<td>vbNo</td>
</tr>
</tbody>
</table>

Response = MsgBox(Message, DialogType, Title)

'evaluate user response
If Response = vbYes Then
  Beep
End
End If
Component: Text boxes for input
Code Programming: txtTextbox.Text and str(someNumber)

Objective: User enters numbers in Input text box. The multiply command button then generates products.

Code Programming:

```vbnet
Private Sub Form_Load()
    result = 1
    txtInput.Text = Str(0)
End Sub

Private Sub cmdMultiply_Click()
    result = result * txtInput.Text
    txtInput.Text = "" ' clear input textbox
    txtResult.Text = result
End Sub
```

- Control extensions (e.g. Text) are fields with pertinent information
- Ultimately should check for numerical input
- Not intuitive for user
Component: Input Dialog Box

Code Programming: InputBox("Dialog message", "Title bar caption") andLblSomeLabel.Caption

GUI: Two command buttons and one label

Code:

Private Sub cmdEnterNumber_Click()
Dim firstNumber, secondNumber As Integer

firstNumber = InputBox("Message: Enter first integer", "Title bar: First Number")
secondNumber = InputBox("Message: Enter second integer", "Title bar: Second Number")

If firstNumber = secondNumber Then
    lblMessage.Caption = firstNumber & " is = " & secondNumber
ElseIf firstNumber > secondNumber Then
    lblMessage.Caption = firstNumber & " is greater than " & secondNumber
Else: lblMessage.Caption = firstNumber & " is less than " & secondNumber
End If

End Sub

Note:
- Label’s have a Caption field
- InputBox has two strings and can return data
- & is used to connect numbers in strings
Relays

- Electromagnetic switch. Packaged like SPST, SPDT, DPST, DPDT
- Advantages: widely available, wide dynamic range, multiple options
- Disadvantages: limited cycles and response time

SPDT Relay:

- Relay's switch is activated when coil is energized
- Buffer should be used in computer interfacing applications
- Integrated circuit chips sink more current than source