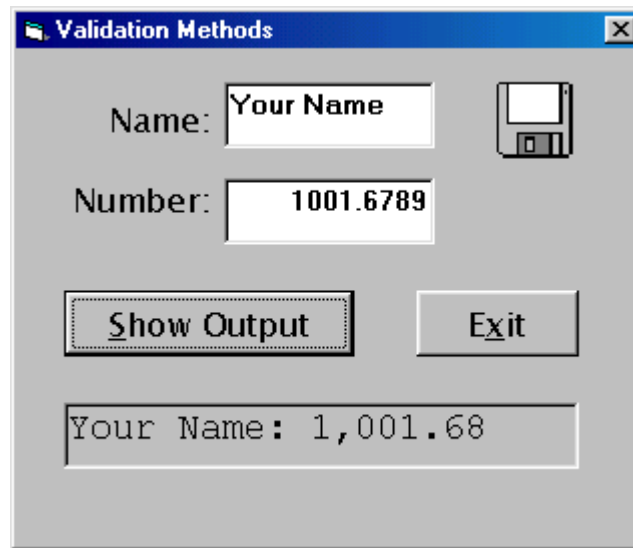


Validation of User Input in Text Boxes Using Visual Basic 6.0

By C. Gribble for hitmill.com

For the Validation code, this form will be used. The objects are txtName, txtNumber, lblOutput, cmdCompute, and cmdExit.

For users needing keyboard access methods, set the cmdCompute **Default property** to true, so that a user may use the Enter key to trigger the click event for the Show Output button (cmdCompute).
Set the cmdExit **Cancel property** to true, so that users may use the Esc key to trigger the click event of the Exit button.
Only one button per form may have the default property set to true and only one button per form may have the cancel property set to true.



A good programmer will take steps to prevent the user from entering invalid data, incomplete data, or missing data. Code has to be execute each time the user leaves a blank text box, or enters incorrect characters or invalid values into a text box. If the text box is empty, leave a message for the user and reset the focus back to the text box. Exit the Sub with the **Exit Sub** statement. Else, if the text box has the wrong type of date, leave a message for the user, then use the Sel (selection) method to "highlight" the incorrect text, and exit the sub, so that processing will stop, allowing time for the user to correct the faulty input.

Using the Sel method

```
txtName.SelStart = 0           'this begins highlighting at the first character
txtName.SelLength = Len(txtName.Text)
'SelLength will highlight the full length of the characters in the text box
'Reset the focus:
txtName.SetFocus
'Exit the Sub, to stop the processing. Do not confuse this with the END Sub
statement.
Exit Sub
```

If you use the With and End With structure, you can shorten this coding:

```
With txtName
    .SelStart=0
    .SelLength=Len(txtName.Text)
    .SetFocus
End With
Exit Sub
```

Here is the full validation code for the txtName and txtNumber text boxes.

```
Private Sub cmdCompute_Click()
    Dim strName As String
    Dim sngNumber As Single
    ' Validation handles an empty text box or number box,
    ' a number in the name box, a non-numeral in the number box,
    ' negative numbers in the number box, sel method
    If txtName.Text = "" Then
        MsgBox "Enter your name", , "Missing Name data"
        txtName.SetFocus
        Exit Sub 'stops processing to enter a name
    ElseIf IsNumeric(txtName.Text) Then
        MsgBox "Numerals not allowed in" & vbCrLf & _
            "the text box. Enter your name.", , "Invalid Data"
        With txtName
            .SelStart = 0 'begins highlighting at first character
            .SelLength = Len(txtName.Text)
            .SetFocus
        End With
        Exit Sub 'stops processing to correct invalid data
    'If txtName is not empty and is not a number, accept the input
    'advanced programmers may also use a KeyPress event
    Else 'the name box is correct
        strName = txtName.Text
        'Students validate the number box if the name box is ok
        If txtNumber.Text = "" Then
            MsgBox "Enter a number greater" & vbCrLf & _
                "than zero.", , "Missing number"
            txtNumber.SetFocus
            Exit Sub 'stops processing to enter a number
        ElseIf Not IsNumeric(txtNumber.Text) Then
            'a number was not entered
            MsgBox "Enter numerals and not" & vbCrLf & _
                "any other characters", , "Invalid data"
            'Use sel method and set focus
            With txtNumber
                .SelStart = 0
                .SelLength = Len(txtNumber.Text)
                .SetFocus
            End With
            Exit Sub 'stops processing; user to correct input error
        'Do not accept a negative number or zero
        ElseIf IsNumeric(txtNumber.Text) And _
            Val(txtNumber.Text) <= 0 Then
            MsgBox "Number must be greater" & vbCrLf & _
                "than 0.", , "Number out of accepted range"
            With txtNumber
```

```
        .SelStart = 0
        .SelLength = Len(txtNumber.Text)
        .SetFocus
    End With
    Exit Sub 'stops processing: user to correct input error
Else 'accept the user input
    sngNumber = txtNumber.Text
End If
End If
lblOutput.Font = "Courier New"
lblOutput.FontSize = 14
lblOutput.Caption = strName & ": " & FormatNumber(sngNumber, 2)
End Sub
```

Prior to validation, you will dimension your variables, whether they are global, module level, or local. Code your statements for constants before any validation.

Someone has hired the programmer to write a program, specifically designed for a particular end-user of the software. That user may tell the programmer about additional project needs. All projects should be planned on paper before any coding has even begun.