

# InputBox, MsgBox and TextBox

Interaction with user via InputBox, MsgBox

Location: <http://www.mvps.org/dmccritchie/excel/inputbox.htm>

Home page: <http://www.mvps.org/dmccritchie/excel/excel.htm>

[\[View without Frames\]](#)

## Example using both InputBox and MsgBox

```
Option Explicit
Sub test01()
    Dim a As Long, response As Long
    a = Application.InputBox( _
        Prompt:="Enter the Rownumber", _
        Title:="Delete rownumber:", Type:=1)
    If a <> False Then
        response = MsgBox("Are you sure.", vbYesNo)
        If response = vbYes Then Rows(a).Delete
    End If
End Sub
```

Perhaps a little more convention is Ok or Cancel on the MsgBox

```
dim response as long
response = msgbox(prompt:="ok or cancel",buttons:=vbokcancel)
if response = vbok then ...
```

`MsgBox(prompt[, buttons] [, title] [, helpfile, context])`

### Simple MsgBox Example(s)

worksheets(1) — for first sheet  
worksheets("sheet1") – for named sheet  
vbYesCancel – might be used instead

```
Sub stay_OR_Sheet1()
    MsgBox "Do you want to go sheet1", vbYesNo
    On Error Resume Next
    If vbYes Then Worksheets("sheet4").Activate
    If Err.Number = 9 Then
        MsgBox "Sorry but sheet4 no longer exists, so staying here anyway"
    End
    On Error GoTo 0
    Rem rest of code
End Sub
```

## More Explicit Example for InputBox (#explicit)

InputBox allows you to set a default, and to place a title on the dialog, as well as the prompt. I prefer to specify each of those things. See HELP for specific parameter information:

*expression*.**InputBox**(*Prompt, Title, Default, Left, Top, HelpFile, HelpContextId, Type*)

Warning the following example defaults to deleting all rows in the initial selection range, if only one cell is selected that would be the row of the active cell.

```
Sub test01()
    Dim a As Variant, response As Variant
    a = Selection(1).row & ":" & Selection(1).SpecialCells(xlLastCell).row
```

```

a = Application.InputBox( _
    "Enter the Rownumber range to be deleted", _
    "Delete rownumber:", a, , , , 8) '8 for a range
'MsgBox VarType(a) '8=string, 11=Boolean
If a <> False Then ' use False if a is defined as a number
    response = MsgBox("Are you sure, you want do delete row " & a, vbYesNo)
    If response = vbYes Then Rows(a).Delete
End If
End Sub

```

## Multiple Range Selection in InputBox (#multi)

```

Sub InputBox_MultiRange()
Dim varRange As Range, subArea As Range, AreasStr As String
'-- initial selection area(s) will be used as suggestion
On Error Resume Next
Set varRange = _
    Application.InputBox("Select single or multiple ranges:", _
        "MultiRange test", Selection.Address(0, 0), Type:=8)
On Error GoTo 0
If varRange Is Nothing Then Exit Sub
For Each subRange In varRange.Areas
    AreasStr = AreasStr & Chr(10) & subRange.Rows.Count & _
        & " rows in " & subRange.Address(0, 0)
Next subRange
MsgBox "Areas: " & varRange.Areas.Count & ", "& _
    "rows: " & varRange.Rows.Count & AreasStr
End Sub

```

John Walkenbach has an example, [Excel Developer Tip: Copying a Multiple Selection \(Tip 36\)](#), that obtains a multiple range without the inputbox, with a msgbox asking you to select range(s), and then does a multiple range paste that is not available in Excel.

## PopUp or Warning Box, short duration (#popup)

Jim Rech posted code ([2000-09-20](#)) to put a warning box up for two seconds and then disappear.  
 Thanks to Dave Peterson for mentioning this in newsgroups

```

Sub SelfClosingMsgBox()
CreateObject("WScript.Shell").Popup "Hello", _
    2, "This closes itself in 2 seconds"
End Sub

```

## Parameters for InputBox and then for MsgBox (#parameters)

### Type parameter for InputBox (#type)

*expression*.InputBox(prompt[, title] [, default] [, xPos] [, yPos] [, helpfile, type])

xPos optional parameter from Left in points  
 yPos optional parameter from Top in points

#### Value Meaning for Type

- |   |   |  |
|---|---|--|
| 0 | A formula                                       | The type information from HELP   |
| 1 | A number  |  |
| 2 | Text (a string) -- default                      | Type Optional Variant. Specifies the return data type. If this argument is omitted, the dialog box returns text. Can be one or |
| 4 | A logical value ( <b>True</b> or <b>False</b> ) | a sum of the type values.  |
|   | A cell reference, as a <b>Range</b>             |  |

8 object  
 16 An error value, such as #N/A  
 64 An array of values

You can use the sum of the allowable values for *Type*. For example, for an input box that can accept both text and numbers, set *Type* to 1 + 2.

### Button parameter for MsgBox (#msgbox)

**MsgBox(prompt[, buttons] [, title] [, helpfile, context])**

Constant for Buttons	Value	Button Description(s)
<b>vbOKOnly</b>	0	Display OK button only.
<b>vbOKCancel</b>	1	Display OK and Cancel buttons.
<b>vbAbortRetryIgnore</b>	2	Display Abort, Retry, and Ignore buttons.
<b>vbYesNoCancel</b>	3	Display Yes, No, and Cancel buttons.
<b>vbYesNo</b>	4	Display Yes and No buttons.
<b>vbRetryCancel</b>	5	Display Retry and Cancel buttons.
<b>vbCritical</b>	16	Display Critical Message icon.
<b>vbQuestion</b>	32	Display Warning Query icon.
<b>vbExclamation</b>	48	Display Warning Message icon.
<b>vbInformation</b>	64	Display Information Message icon.
<b>vbDefaultButton1</b>	0	First button is default.
<b>vbDefaultButton2</b>	256	Second button is default.
<b>vbDefaultButton3</b>	512	Third button is default.
<b>vbDefaultButton4</b>	768	Fourth button is default.
<b>vbApplicationModal</b>	0	Application modal; the user must respond to the message box before continuing work in the current application.
<b>vbSystemModal</b>	4096	System modal; all applications are suspended until the user responds to the message box.
<b>vbMsgBoxHelpButton</b>	16384	Adds Help button to the message box
<b>VbMsgBoxSetForeground</b>	65536	Specifies the message box window as the foreground window
<b>vbMsgBoxRight</b>	524288	Text is right aligned
<b>vbMsgBoxRtlReading</b>	1048576	Specifies text should appear as right-to-left reading on Hebrew and Arabic systems

### VarType Constants (#vartype)

```
dim a as variant a = "abc"
msgbox vartype(a) -- would display 8 for string
```

The following constants can be used anywhere in your code in place of the actual values:

Constant	Value	Description
<b>vbEmpty</b>	0	Uninitialized (default)
<b>vbNull</b>	1	Contains no valid data
<b>vbInteger</b>	2	Integer
<b>vbLong</b>	3	Long integer
<b>vbSingle</b>	4	Single-precision floating-point number
<b>vbDouble</b>	5	Double-precision floating-point number
<b>vbCurrency</b>	6	Currency
<b>vbDate</b>	7	Date
<b>vbString</b>	8	String
<b>vbObject</b>	9	Object
<b>vbError</b>	10	Error
<b>vbBoolean</b>	11	Boolean
<b>vbVariant</b>	12	Variant (used only for arrays of variants)
<b>vbDataObject</b>	13	Data access object
<b>vbDecimal</b>	14	Decimal
<b>vbByte</b>	17	Byte
<b>vbUserDefinedType</b>	36	Variants that contain user-defined types
<b>vbArray</b>	8192	Array

## TextBox

An [Event macro](#) that will change all text boxes on the worksheet to caps upon selection of the worksheet. (install with right on sheet tab, view code, ...)

```
Private Sub Worksheet_Activate()  
' change all textboxes on the page to caps  
' upon worksheet activation. DMcR 2004-07-15  
' also see kbid=  
  Dim tbox As TextBox  
  For Each tbox In ActiveSheet.TextBoxes  
    tbox.Text = UCase(tbox.Text)  
  Next tbox  
End Sub
```

kbid=152379 - How to Copy the Text Within a Text Box to a Cell,  
<http://support.microsoft.com/?kbid=152379> ( you can Google kbid=152379)

## Speech, Voice message alternative (#speech)

In the VBE, Tools, References check -- Microsoft Direct Speech Synthesis, then  
code: `Application.Speech.Speak "I am done"`

## Related (#related)

- [Getting Started with Macros](#) for help in installing and running a macro.
- [Cell Comment](#) boxes
- [Event Macros](#), more information on Event macros and how to install them. They are installed differently than regular macros.
- [Font size in comboboxes](#), depends on where you created combobox from.
- [NameBox](#), increase the size of the name box, based on ...
- [Proper, and other Text changes -- Use of Special Cells](#)
- [Shapes](#), a textbox is a shape, also see [collections](#).

---

This page was introduced on May 11, 2002. (last update: 07/21/2007 18:53:57)

[\[My Excel Pages -- home\]](#)   [\[INDEX to my site and the off-site pages I reference\]](#)  
[\[Site Search -- Excel\]](#)   [\[Go Back\]](#)   [\[Return to TOP\]](#)

Please send your comments concerning this web page to: David McRitchie [send email comments](#)

---

[Copyright © 1997 - 2004](#), F. David McRitchie, All Rights Reserved