

MsgBox Function (Visual Basic)

Visual Studio 2008 36 out of 69 rated this helpful

Displays a message in a dialog box, waits for the user to click a button, and then returns an integer indicating which button the user clicked.

```
Public Function MsgBox( _
    ByVal Prompt As Object, _
    Optional ByVal Buttons As MsgBoxStyle = MsgBoxStyle.OKOnly, _
    Optional ByVal Title As Object = Nothing _
) As MsgBoxResult
```

Parameters

- Prompt**
Required. **String** expression displayed as the message in the dialog box. The maximum length of *Prompt* is approximately 1024 characters, depending on the width of the characters used. If *Prompt* consists of more than one line, you can separate the lines using a carriage return character (**Chr(13)**), a line feed character (**Chr(10)**), or a carriage return/linefeed character combination (**Chr(13) & Chr(10)**) between each line.
- Buttons**
Optional. Numeric expression that is the sum of values specifying the number and type of buttons to display, the icon style to use, the identity of the default button, and the modality of the message box. If you omit *Buttons*, the default value is zero.
- Title**
Optional. **String** expression displayed in the title bar of the dialog box. If you omit *Title*, the application name is placed in the title bar.

Settings

The **MsgBoxStyle** enumeration values are listed in the following table.

Member	Value	Description
OKOnly	0	Displays OK button only.
OKCancel	1	Displays OK and Cancel buttons.
AbortRetryIgnore	2	Displays Abort, Retry, and Ignore buttons.
YesNoCancel	3	Displays Yes, No, and Cancel buttons.
YesNo	4	Displays Yes and No buttons.
RetryCancel	5	Displays Retry and Cancel buttons.
Critical	16	Displays Critical Message icon.
Question	32	Displays Warning Query icon.
Exclamation	48	Displays Warning Message icon.
Information	64	Displays Information Message icon.
DefaultButton1	0	First button is default.
DefaultButton2	256	Second button is default.
DefaultButton3	512	Third button is default.
ApplicationModal	0	Application is modal. The user must respond to the message box before continuing work in the current application.
SystemModal	4096	System is modal. All applications are suspended until the user responds to the message box.
MsgBoxSetForeground	65536	Specifies the message box window as the foreground window.
MsgBoxRight	524288	Text is right-aligned.

MsgBoxRtlReading	1048576	Specifies text should appear as right-to-left reading on Hebrew and Arabic systems.
-------------------------	---------	---

The first group of values (0–5) describes the number and type of buttons displayed in the dialog box. The second group (16, 32, 48, 64) describes the icon style. The third group (0, 256, 512) determines which button is the default. The fourth group (0, 4096) determines the modality of the message box, and the fifth group specifies whether or not the message box window is the foreground window, along with the alignment and direction of the text. When adding numbers to create a final value for the *Buttons* argument, use only one number from each group.

Return Value

Constant	Value
OK	1
Cancel	2
Abort	3
Retry	4
Ignore	5
Yes	6
No	7

Exceptions

Exception type	Error number	Condition
ArgumentException	5	<i>Prompt</i> is not a String expression, or <i>Title</i> is invalid.
InvalidOperationException	5	Process is not running in User Interactive mode.
InvalidEnumArgumentException	5	One or more parameters not a member of MsgBoxResult or MsgBoxStyle enumeration.

See the "Error number" column if you are upgrading Visual Basic 6.0 applications that use unstructured error handling. (You can compare the error number against the [Number Property \(Err Object\)](#).) However, when possible, you should consider replacing such error control with [Structured Exception Handling Overview for Visual Basic](#).

Remarks

If the dialog box displays a **Cancel** button, pressing the ESC key has the same effect as clicking **Cancel**. If the dialog box contains a **Help** button, context-sensitive Help is provided for the dialog box. However, no value is returned until one of the other buttons is clicked.

Note:
To specify more than the first argument, you must use the MsgBox function in an expression. If you omit any positional arguments, you must retain the corresponding comma delimiter.
Note:
The MsgBox function requires UIPermission at the SafeTopLevelWindows level, which may affect its execution in partial-trust situations. For more information, see Requesting Permissions and UIPermission .

Example

This example uses the **MsgBox** function to display a critical-error message in a dialog box with Yes and No buttons. The No button is specified as the default response. This is done by combining the **MsgBox** constant values into one numeric expression. In this case, adding 4 (the Yes/No button combination) and 16 (the **Critical Message** window) and 256 (the second button as default button) gives a total of 276. The value returned by the **MsgBox** function depends on the button chosen by the user: Yes returns a value of 6; No returns a value of 7.

VB

```
' The following example requires that Option Infer be set to On.

' Define the message you want to see inside the message box.
Dim msg = "Do you want to continue?"
```

```
' Display a simple message box.
MsgBox(msg)

' Define a title for the message box.
Dim title = "MsgBox Demonstration"

' Add the title to the display.
MsgBox(msg, , title)

' Now define a style for the message box. In this example, the
' message box will have Yes and No buttons, the default will be
' the No button, and a Critical Message icon will be present.
Dim style = MsgBoxStyle.YesNo Or MsgBoxStyle.DefaultButton2 Or _
    MsgBoxStyle.Critical

' Display the message box and save the response, Yes or No.
Dim response = MsgBox(msg, style, title)

' Take some action based on the response.
If response = MsgBoxResult.Yes Then
    MsgBox("YES, continue!!", , title)
Else
    MsgBox("NO, stop!!", , title)
End If
```

Requirements

- Namespace:** [Microsoft.VisualBasic](#)
- Module:** **Interaction**
- Assembly:** Visual Basic Runtime Library (in Microsoft.VisualBasic.dll)

See Also

- Reference**
- [InputBox Function \(Visual Basic\)](#)

Community Additions

See Also

See aslo

- WPF MessageBox class (System.Windows.MessageBox) <http://msdn.microsoft.com/en-us/library/system.windows.messagebox.aspx>
- WinForms MessageBox class (System.Windows.Forms.MessageBox) <http://msdn.microsoft.com/en-us/library/system.windows.forms.messagebox.aspx>

 **Donny**
5/7/2008

Prompting a user.

If you want to get a response from the user use the **MsgBox Function**

Ex:

If MsgBox(" Prompt?", MsgBoxStyle.YesNo) = Windows.Forms.DialogResult.Yes Then

MsgBox is in the Visual Basic Runtime Library. Thus if you are in a different language you will need to import the library or alternately use

System.Windows.Forms.MessageBox.Show which is in System.Windows.Forms and language neutral.

Internally Msgbox eventually winds up doing something like this:

ReturnDirectCast(**MessageBox.Show**([text], titleFromAssembly, (DirectCast(Buttons, MessageBoxButtons) AndDirectCast(15, MessageBoxButtons)), (DirectCast(Buttons, MessageBoxIcon) AndDirectCast(240, MessageBoxIcon)), (DirectCast(Buttons, MessageBoxDefaultButton) AndDirectCast(&HF00, MessageBoxDefaultButton))), MsgBoxResult)

Msgbox is a wrapper to MessageBox.Show and is a holdover from classic VB.

