



Specifying dialog properties

The features of a **MsgBox** dialog can be determined by adding a comma and specification value after the message string within its parentheses. This can specify which buttons the dialog will display and what graphic icon, if any, will appear on the dialog.

Button constant	Value
vbOkOnly	0
vbOkCancel	1
vbAbortRetryIgnore	2
vbYesNoCancel	3
vbYesNo	4
vbRetryCancel	5

The dialog button combinations can be specified using the Visual Basic constant values, or their numeric equivalents, in this table. For example, to have the dialog display Yes, No, and Cancel buttons, specify the **vbYesNoCancel** constant or its numeric equivalent **3**.



Always specify a graphic icon when calling a **MsgBox** dialog to help the user easily understand the nature of the message.

Icon constant	Value
vbCritical	 16
vbQuestion	 32
vbExclamation	 48
vbInformation	 64

The dialog icon can be specified using the Visual Basic constant values, or their numeric equivalents, in this table. For example, to have the dialog display the question mark icon, specify the **vbQuestion** constant or its numeric equivalent **32**.

In order to have the **MsgBox** display both a particular button combination and a certain graphic icon, the specification can add the button constant and the icon constant together using the addition + operator. For example, the specification to display Yes, No, and Cancel buttons along with a question icon would be **vbYesNoCancel + vbQuestion**. Alternatively, specify the sum total of their numeric equivalents – in this case it's **35** (3 + 32).

...cont'd

The buttons in a **MsgBox** dialog each return a specific numeric value to the application when they are clicked. This can be assigned to a property in much the same way as the value returned from the **InputBox** dialog in the previous example.

- 1 Click on **File, New, Project**, then create a new **Windows Forms App** and name it "MsgBoxDialog"
- 2 From the ToolBox add a **Button**, a **Label**, and a **TextBox** to the default Form and arrange them to your liking
- 3 Set the Label's **Text** property to "Button Value :." and name the TextBox **TxtValue**
- 4 Double-click on the **Button** to open the **Code** editor within its **Click** event-handler, and add the statement `TxtValue.Text = MsgBox("Click any button" , _vbYesNoCancel + vbQuestion)`



- 5 Press the **Start** button to run the application, then click any button and note the value it returns to the **TextBox**



MsgBoxDialog



Don't confuse the button return values with the Visual Basic constant values used to specify the button combinations.



Try changing the **MsgBox** combination specification using constant or numeric values – and make a note of the value returned by each button.