

INPUTBOX AND MSGBOX FUNCTION IN VBSCRIPT

M.ABBAS MANTHIRI MCA.,SET.,

ASSISTANT PROFESSOR

PG DEPARTMENT OF COMMERCE (CA)

HAJEE KARUTHA ROWTHER HOWDIA COLLEGE

UTHAMAPALAYAM

Introduction to Vbscript Functions

Functions are similar to normal procedures but the main purpose of the functions is to accept certain inputs and pass them on to the main program to finish the execution. They are two types of function, the built-in functions(or internal functions) and the functions created by the programmers.

The general format of a function is

```
functionName(arguments)
```

where arguments are values that are passed on to the functions.

In this lesson, we are going to learn two very basic but useful internal functions, i.e. the MsgBox() and InputBox () functions.

MsgBox () Function

The objective of MsgBox is to produce a pop-up message box and prompt the user to click on a command button before he /she can continues. This message box format is as follows:

```
yourMsg=MsgBox(Prompt, Style Value, Title)
```

The first argument, Prompt, will display the message in the message box. The Style Value will determine what type of command buttons appear on the message box, please refer Table 1.1 for types of command button displayed. The Title argument will display the title of the message board.

Table1: Style Values

Style Value	Named Constant	Buttons Displayed
0	vbOkOnly	Ok button
1	vbOkCancel	Ok and Cancel buttons
2	vbAbortRetryIgnore	Abort, Retry and Ignore buttons.
3	vbYesNoCancel	Yes, No and Cancel buttons
4	vbYesNo	Yes and No buttons
5	vbRetryCancel	Retry and Cancel buttons

We can use named constant in place of integers for the second argument to make the programs more readable. Infact, VB6 will automatically shows up a list of names constant where you can select one of them.

```
example: yourMsg=MsgBox( "Click OK to Proceed", 1, "Startup Menu")
```

```
and yourMsg=Msg("Click OK to Proceed". vbOkCancel,"Startup Menu")
```

are the same.

yourMsg is a variable that holds values that are returned by the MsgBox () function. The values are determined by the type of buttons being clicked by the users. It has to be declared as Integer data type in the procedure or in the general declaration section. Table 1.2 shows the values, the corresponding named constant and buttons.

Table 1.2 : Return Values and Command Buttons

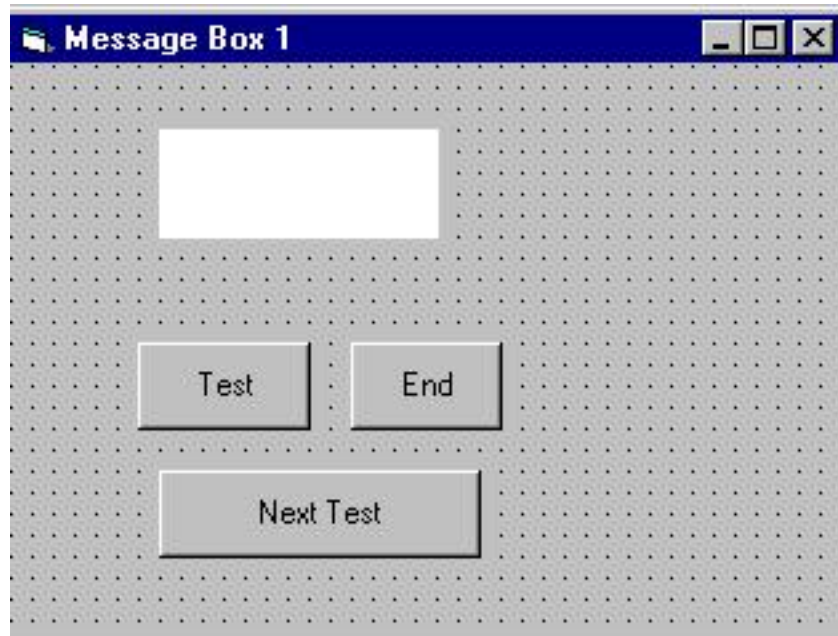
Value	Named Constant	Button Clicked
1	vbOk	Ok button
2	vbCancel	Cancel button
3	vbAbort	Abort button
4	vbRetry	Retry button
5	vbIgnore	Ignore button
6	vbYes	Yes button
7	vbNo	No button

Example 1.1

i. The Interface:

You draw three command buttons and a label as shown in Figure 1.1

Figure 1.1



ii. The procedure for the test button:

```
Private Sub Test_Click()  
Dim testmsg As Integer  
testmsg = MsgBox("Click to test", 1, "Test message")  
If testmsg = 1 Then  
Display.Caption = "Testing Successful"  
Else  
Display.Caption = "Testing fail"  
End If  
  
End Sub
```





When a user click on the test button, the image like the one shown in Figure 1.2 will appear. As the user click on the OK button, the message "Testing sucessful" will be diplayed and when he/she clicks on the Cancel button, the message "Testing fail" will be displayed.

Figure 1.2



To make the message box look more sophisticated, you can add an icon besides the message. There are four types of icons available in VB as shown in Table 1.3

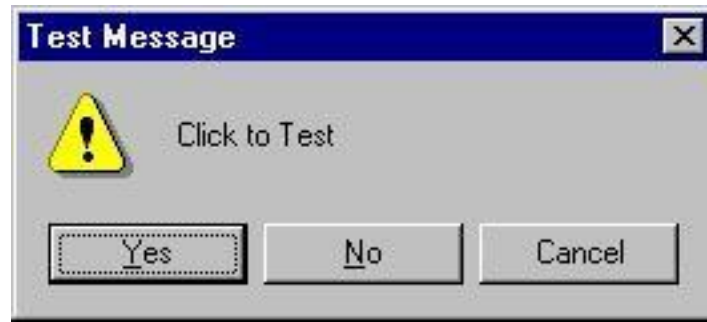
Table 1.3

Value	Named Constant	Icon
16	vbCritical	
32	vbQuestion	
48	vbExclamation	
64	vbInformation	

Example 1.2

In this example, the following message box will be displayed:

Figure 1.3



You could draw the same Interface as in example 1.1 but modify the codes as follows:

```
Private Sub test2_Click()  
Dim testMsg2 As Integer  
testMsg2 = MsgBox("Click to Test", vbYesNoCancel + vbExclamation, "Test Message")  
If testMsg2 = 6 Then  
display2.Caption = "Testing successful"  
ElseIf testMsg2 = 7 Then  
display2.Caption = "Are you sure?"  
Else  
display2.Caption = "Testing fail"  
End If  
  
End Sub
```

The InputBox() Function

An InputBox() function will display a message box where the user can enter a value or a message in the form of text. The format is

```
myMessage=InputBox(Prompt, Title, default_text, x-position, y-position)
```

myMessage is a variant data type but typically it is declared as string, which accept the message input by the users. The arguments are explained as follows:

- Prompt - The message displayed normally as a question asked.
- Title - The title of the Input Box.
- default-text - The default text that appears in the input field where users can use it as his intended input or he may change to the message he wish to key in.
- x-position and y-position - the position or the coordinate of the input box.

Example 1.3

i. The Interface

Figure 1.4

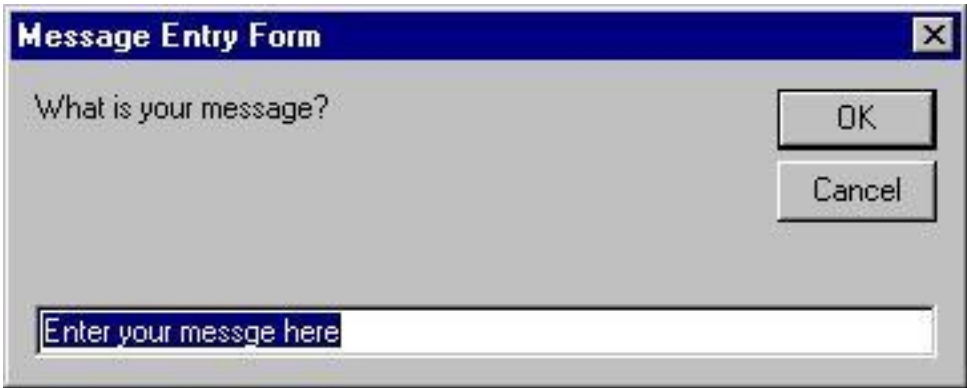


ii. The procedure for the OK button

```
Private Sub OK_Click()  
Dim userMsg As String  
userMsg = InputBox("What is your message?", "Message Entry Form", "Enter your  
messge here", 500, 700)  
If userMsg <> "" Then  
message.Caption = userMsg  
Else  
message.Caption = "No Message"  
End If  
  
End Sub
```

When a user click the OK button, the input box as shown in Figure 1.5 will appear. After user entering the message and click OK, the message will be displayed on the caption, if he click Cancel, "No

message" will be displayed.



A screenshot of a Windows-style dialog box titled "Message Entry Form". The dialog has a blue title bar with a close button (X) in the top right corner. The main area is light gray and contains the text "What is your message?". To the right of this text are two buttons: "OK" and "Cancel". At the bottom of the dialog is a text input field with the placeholder text "Enter your message here".

Thank you