

## ***Message Dialog*** Sample Help

**Sample Description:** [Message Dialog](#)

### **Points of Interest**

[Adding a Message Dialog to your Application](#)

[Handling a Message Dialog's Return Code](#)

### **Control**

MessageBox

AbortRetryIgnoreBox

InfoBox

OKCancelBox

RetryCancelBox

YesNoBox

YesNoCancelBox

For Help on Help, Press F1

## Message Dialog

The Message Dialog displays a message and an optional icon. All message dialogs in Envelop are derived from the "MessageBox" object. The following types of message dialogs are available, each with their own response button configuration:

<u>Object</u>	<u>Buttons</u>
AbortRetryIgnoreBox	Abort, Retry, Ignore
InfoBox	OK
OKCancelBox	OK, Cancel
RetryCancelBox	Retry, Cancel
YesNoBox	Yes, No
YesNoCancelBox	Yes, No, Cancel

The primary purpose of each message dialog type is to display a dialog box with a message. Depending on the type of message box, a different button configuration is available. The message boxes are modal, meaning that the user must click one of the command buttons to close it.

Although the message dialogs do not return any text, the method does return a numeric value. This value indicates which button the user selected when closing the message dialog.

Message dialogs are most often used to display short messages, and to obtain quick feedback from the user. For example, suppose your user tries to exit a program without saving some open files. Your program might display a dialog box to ask the user whether the open files should be saved.

### To test the Message Dialog sample:

1. Enter a title sting to appear in the dialog's banner
2. Then, enter a specific message to display inside the dialog.
3. Click the desired options for button and icon configuration.
4. Finally, click the **Show** button. Your dialog should be displayed in the center of the screen.

### **Adding a Message Dialog to your Application**

Adding a Message Dialog to your application can be done through program code. None of the available MessageBox objects need to be embedded into your application form. The following code example will give you an idea of how to add a message dialog to your application.

### **Example**

```
' Set the title message of the InfoBox
InfoBox.title = "Warning!"

' Present a warning message
InfoBox.Msg("You have not entered any text to be printed.")
```

Both examples use the method "Msg," which accepts a message string. In these examples, the title string was set in a previous operation. Alternatively, both the title and message could be set in the same operation using the "Message" method shown below:

```
InfoBox.Message("Warning!", "You have not entered any text to be printed.")
```

### Handling a Message Dialog's Return Code

Both the `Msg` and `Message` methods for all Message Dialog's return an integer value to indicate which key the user clicked to close the dialog. Below are a list of those return codes:

Buttons	Return Code Value
OK	1
Cancel	2
Abort	3
Retry	4
Ignore	5
Yes	6
No	7

The following example code shows how you can use return code to perform the specific application function of exiting an application.

### Example

```
' METHODS for leaving an application
Sub ExitApplication_Click()

' Set the contents of the titlebar of the YesNoBox object
YesNoBox.title = "Quit?"

' Set the message of the YesNoPrompt object
YesNoBox.Msg("Ok to quit application?")

' If the Yes entry was clicked, hide the textedit form
If YesNoBox.result == 6 Then
    Hide
End If
```

At this point, try various button and icon configurations in the sample Message Dialog demonstration. When you click the **Show** button on the sample form, the corresponding message dialog will be displayed. When you click on one of the message dialog's buttons, the corresponding value will be presented at the bottom of the sample form.

