

Programming Logic

152-101 152-102

Using the Message Box

Notes	Activity
-------	----------

Quick Links & Text References

- [Message Box Format](#) Page 73 – 75
- [Message Parameter](#) Pages
- [Caption Parameter](#) Pages
- [Buttons Parameter](#) Pages
- [Icon Parameter](#) Pages
- [Default Button Parameter](#) Pages

Message Box

- The Message Box class allows you to quickly create a dialog box with the message of your choice, an icon of your choice and any combination of standard buttons.
- The Message Box can (if necessary) let you know which button the user selected.

Message Box Format

A Message Box has one required and four optional parameters.

```
result = MessageBox.Show("Your Message Here",  
                          "Dialog Box Title Here",  
                          buttons selection here,  
                          icon select here,  
                          default button selection );
```

- Separate parameters with commas
- The first parameter is the message you want displayed in the dialog box (often an error message). This is the only required parameter.
 - To create multi-line message, concatenate `Environment.NewLine` to each line of the message.

```
MessageBox.Show("This is line 1 " + Environment.NewLine +  
               "This is line 2");
```

Notes	Activity
➤ The <u>second parameter</u> allows you to set the text that appears in the dialog box title bar (optional). ➤ The <u>third parameter</u> (optional) allows you to designate which buttons appear in the dialog box. – The list of available choices appears when you type the comma following the second parameter. AbortRetryIgnore OK OKCancel RetryCancel YesNo YesNoCancel ▪ If the list doesn't appear, type "m" to select the MessageBoxButtons enumeration, then a dot (period) to see the list of button options. – Select the button combination you want, then press the comma key to access the fourth parameter – Default (if not provided) is an OK button.	

- The fourth parameter (optional) allows you to designate which icon appears in the dialog box.
 - The list of available choices appears when you type the comma following the third parameter.
 - Select the icon you want, then press the comma key to access the fifth parameter or type a right parenthesis to end the command.
 - If you don't provide this parameter, no icon is displayed.
 - The icon appearance maybe different depending on the operating system your program is running in. The images below are from Windows 7

MessageBoxIcon.Asterisk MessageBoxIcon.Information	
MessageBoxIcon.Error MessageBoxIcon.Hand MessageBoxIcon.Stop (also plays an <i>error</i> sound)	
MessageBoxIcon.Exclamation MessageBoxIcon.Warning	
MessageBoxIcon.Question	
MessageBoxIcon.None	

- If the list doesn't appear, type "m" to select the MessageBoxIcon enumeration, then a dot (period) to see the list of icon options.
- The optional fifth parameter allows you to designate which button in the dialog box is the default button (assuming there's more than one button in the dialog box)
 - The list of available choices appears when you type the comma following the fourth parameter.
 - If the list doesn't appear, press "m", select MessageBoxDefaultButtons, then a dot (period) to see the list of options.

- The MessageBox always returns the ID number of the button the user clicked to close the dialog box.
 - You can ignore it if your code doesn't need it.
 - The variable that holds the button ID number should be declared as a DialogResult
 - You can compare this result (using an If or Select statement) to a value in the DialogResult class.

```
DialogResult result;           //User's dialog box selection

result = MessageBox.Show(~~~~~);
if(result == DialogResult.Yes){
    //(do something)
} //end if
```