Pir Mehr Ali Shah

Arid Agriculture University Rawalpindi

University Institute of Information Technology

Lab Exercise: MessageBox Implementation

Perform the following:

1. Simple MessageBox
2. MessageBox with Title
3. MessageBox with Buttons
    a. OK
    b. OKCancel
    c. AborRetryIgnore
    d. YesNo
    e. RetryCancel
4. MessageBox with Icon
    a. None
    b. Hand
    c. Question
    d. Exclamatory
    e. Asterisk
    f. Stop
    g. Error
    h. Warning
    i. Information
5. MessageBox with message
    a. Service Notification
    b. Default Desktop only
    c. Right Align
    d. Rtl Reading
6. MessageBox with Help button

Google: MessageBox
Weblink: C-sharpcorner.com

C# MessageBox (Solution)

MessageBox class has an overloaded static Show method that displays a message box with a message and action buttons. The action buttons can be OK and Cancel, Yes and No etc. Here are some of the options that can be used in C# message box.

Simple MessageBox

The simplest form of a MessageBox is a dialog with a text and OK button. When you click OK button, the box disappears.

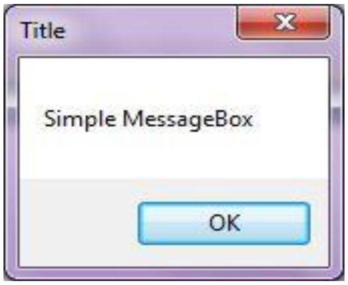
The following code snippet creates a simple Message Box.

```
string message = "Simple MessageBox";
MessageBox.Show(message);
```

MessageBox with Title

The following code snippet creates a simple MessageBox with a title.

```
string message = "Simple MessageBox";
string title = "Title";
MessageBox.Show(message, title);
```



Simple MessageBox

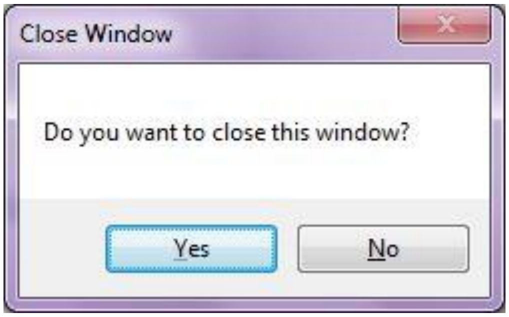
MessageBox with Buttons

A MessageBox can have different button combinations such as YesNo and OKCancel. The MessageBoxButtons enumeration represents the buttons to be displayed on a MessageBox and has following values.

- OK
- OKCancel
- AbortRetryIgnore
- YesNoCancel
- YesNo
- RetryCancel

The following code snippet creates a MessageBox with a title and Yes and No buttons. This is a typical MessageBox you may call when you want to close an application. If the Yes button is clicked, the application will be closed. The Show method returns a DialogResult enumeration.

```
string message = "Do you want to close this window?";
string title = "Close Window";
MessageBoxButtons buttons = MessageBoxButtons.YesNo;
DialogResult result = MessageBox.Show(message, title, buttons);
if (result == DialogResult.Yes) {
    this.Close();
} else {
    // Do something
}
```



Close Window

Do you want to close this window?

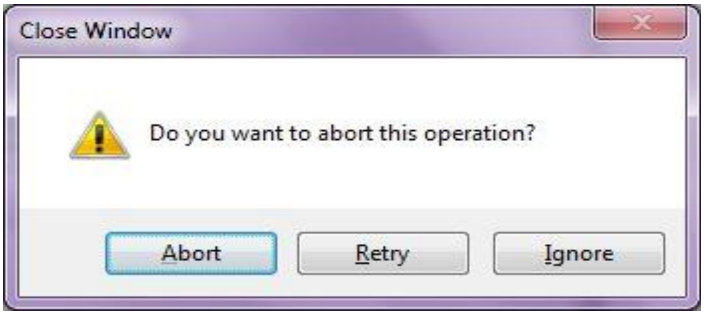
MessageBox with Icon

A MessageBox can display an icon on the dialog. A MessageBoxIcons enumeration represents an icon to be displayed on a MessageBox and has the following values.

- None
- Hand
- Question
- Exclamation
- Asterisk
- Stop
- Error
- Warning
- Information

The following code snippet creates a MessageBox with a title, buttons, and an icon.

```
string message = "Do you want to abort this operation?";
string title = "Close Window";
MessageBoxButtons buttons = MessageBoxButtons.AbortRetryIgnore;
DialogResult result = MessageBox.Show(message, title, buttons, MessageBoxIcon.Warning);
if (result == DialogResult.Abort) {
    this.Close();
}
elseif(result == DialogResult.Retry) {
    // Do nothing
}
else {
    // Do something
}
```



Do you want to abort this operation?

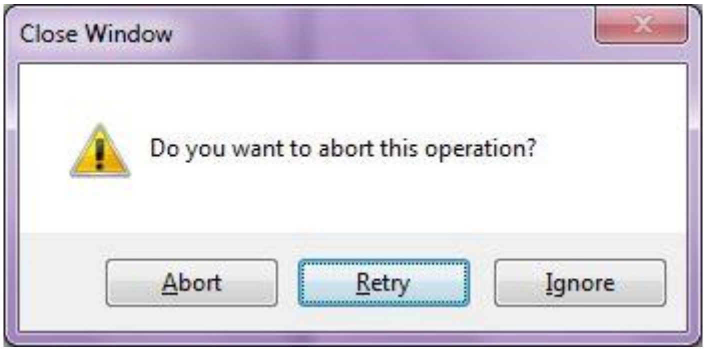
MessageBox with Default Button

We can also set the default button on a MessageBox. By default, the first button is the default button. The MessageBoxDefaultButton enumeration is used for this purpose and it has the following three values.

- Button1
- Button2
- Button3

The following code snippet creates a MessageBox with a title, buttons, and an icon and sets the second button as a default button.

```
string message = "Do you want to abort this operation?";
string title = "Close Window";
MessageBoxButtons buttons = MessageBoxButtons.AbortRetryIgnore;
DialogResult result = MessageBox.Show(message, title, buttons, MessageBoxIcon.Warning, MessageBoxDefaultButton.Button2);
if (result == DialogResult.Abort) {
    this.Close();
}
elseif(result == DialogResult.Retry) {
    // Do nothing
}
else {
    // Do something
}
```



Do you want to abort this operation?

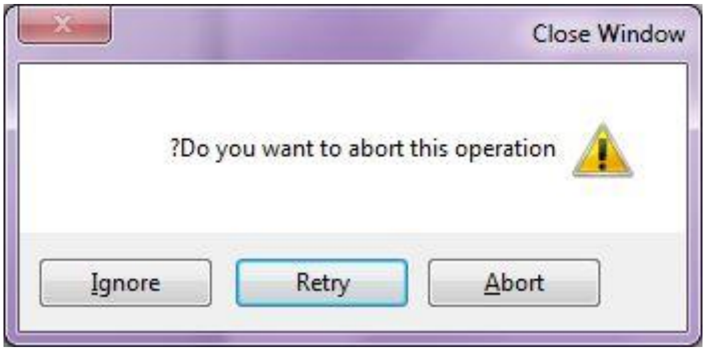
MessageBox with Message Options

MessageBoxOptions enumeration represents various options and has the following values.

- ServiceNotification
- DefaultDesktopOnly
- RightAlign
- RtlReading

The following code snippet creates a MessageBox with various options.

```
DialogResult result = MessageBox.Show(message, title, buttons,
MessageBoxIcon.Warning, MessageBoxDefaultButton.Button2,
MessageBoxOptions.RightAlign|MessageBoxOptions.RtlReading);
```

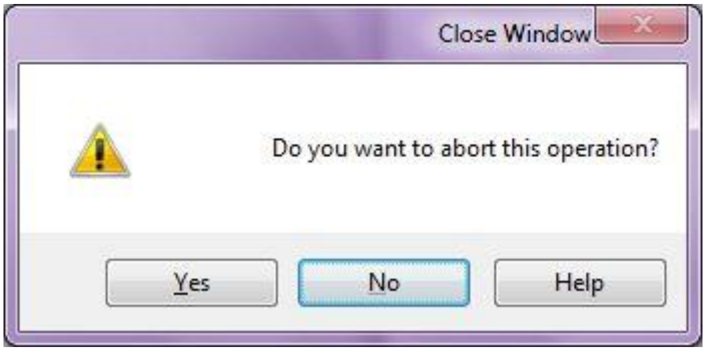


?Do you want to abort this operation

MessageBox with Help Button

A MessageBox can have an extra button called Help button. This is useful when we need to display a help file. The following code snippet creates a MessageBox with a Help button.

```
DialogResult result = MessageBox.Show(message, title, buttons,
MessageBoxIcon.Warning, MessageBoxDefaultButton.Button2,
MessageBoxOptions.RightAlign, true );
```



Close Window

Do you want to abort this operation?

We can also specify a help file when the Help button is clicked. The following code snippet references a help file.

```
DialogResult result = MessageBox.Show(message, title,
buttons, MessageBoxIcon.Question, MessageBoxDefaultButton.Button1, 0, "helpfile.chm");
```