

***OptionButton* Sample Help**

Sample Description: [OptionButton](#)

Points of Interest

[Creating an OptionButton Group](#)

[Clearing a TextBox](#)

[Converting Decimal To Hexadecimal](#)

[Converting Decimal to Octal](#)

Control

OptionButton

For Help on Help, Press F1

OptionButton

The OptionButton control provides a technique for presenting a group of choices where only one choice may be selected. An OptionButton appears as a small circle with text to one side of it. When the OptionButton is selected, a black dot appears inside the circle. When this happens, the OptionButton's **Value** property is set to True. When the circle is not selected, no black dot will appear and the Value property is set to False.

An OptionButton displays a choice that the user may select at run time. By clicking the OptionButton, the user toggles the Value between True and False. Typically, several OptionButtons are grouped together to represent mutually exclusive choices. One button in the group is always selected. When the user selects a different OptionButton in the group, Envelop automatically deselects the previously selected button. OptionButtons that are grouped together are often called *radio* buttons.

This sample application consists of three OptionButtons that are used to convert various number formats. To run the application, click the "Use Decimal Number" OptionButton and then enter a number in the TextBox at the top of the form. Now click the "Use Octal Number" or "Use Hexadecimal Number" options to convert the decimal number into the corresponding format.

This application works by having a **CurrentNumber** property added to the Form's property list. This property can be found in the Property Editor. A **KeyPress** event is assigned to the TextBox to prevent non-numbers from being entered. The following Change method on the TextBox **txtValueBox** examines the **Value** property of each of the OptionButtons, then converts the number to a corresponding value and saves the results in the CurrentNumber property of the form.

```
Sub txtValueBox_Change()  
    If txtValueBox.Text <> "" Then  
        ' Val function interprets numbers beginning with &O as octal;  
        ' numbers beginning with &H as hexadecimal.  
        If optOctal.Value == 1 Then  
            CurrentNumber = Val("&O" & LTrim(txtValueBox.Text) & "&")  
        ElseIf optHex.Value == 1 Then  
            CurrentNumber = Val("&H" & LTrim(txtValueBox.Text) & "&")  
        Else  
            CurrentNumber = Val(LTrim(txtValueBox.Text) & "&")  
        End If  
    Else  
        CurrentNumber = ""  
    End If  
End Sub
```

If the contents of the txtValueBox TextBox are empty, the CurrentNumber property is cleared.

Creating an OptionButton Group

When OptionButtons are created in a group, only one option may be toggled on or True at a time. However, your form application may have more than one group of option buttons. To create an OptionButton group, you simply need to create the OptionButtons in sequence, or one after another. OptionButtons that are created in sequence are automatically joined to the same group.

To create two separate OptionButton groups, simply add the first group of OptionButtons, then add another control such as a Label. This will "break" the creation sequence for the OptionButtons. Then add the second group of OptionButtons.

Clearing a TextBox

A Clear Button has been added to this application to clear the contents of the number input box located at the top of the form. This button executes a click event as shown below:

```
Sub btnClear_Click()  
    txtValueBox.Text = ""  
    txtValueBox.SetFocus  
End Sub
```

Setting the **Text** property to a blank string (i.e., "") will automatically clear the contents of the TextBox. Whenever text is typed into the TextBox, the value of the Text property is automatically updated. There is a direct link between the contents of the TextBox and the value of the Text property.

Once the TextBox is cleared, the **SetFocus** method is issued to place the insertion point back into the TextBox. This is the same as setting the TextBox to have the current "focus."

Converting Decimal To Hexadecimal

The **Hex** function converts a decimal numeric expression to a string that represents the value of the numeric expression in hexadecimal format. The following method is executed when the "Use Hexadecimal Number" OptionButton is clicked.

```
Sub optHex_Click()  
    If txtValueBox.Text <> "" Then  
        txtValueBox.Text = Hex(CurrentNumber)  
    End If  
End Sub
```

Hexadecimal notation is a way of counting using 16 digits. The digits in hexadecimal include 0-9 and A-F. Hexadecimal "A" equals a decimal 10, "B" equals 11, and so on. Hexadecimal is often used to display memory addresses because it easily converts back and forth from binary.

Converting Decimal to Octal

The **Oct** function converts a decimal numeric expression to a string that represents the value of the numeric expression in Octal notation (base 8). The following method is executed when the "Use Octal Number" OptionButton is clicked.

```
Sub optOctal_Click()  
    If txtValueBox.Text <> "" Then  
        txtValueBox.Text = Oct(CurrentNumber)  
    End If  
End Sub
```

Octal notation is a method of counting using only eight digits. This is sometimes used to work with memory addresses because it converts back and forth from binary more easily than decimal.

