

## ***OptionButton Array*** Sample Help

**Sample Description:** [OptionButton Array](#)

### **Points of Interest**

[Creating An OptionButton Array](#)

[Removing An Option Button Instance](#)

### **Control**

OptionButton

For Help on Help, Press F1

### OptionButton Array

This OptionButton Array sample is intended to show how to create and delete controls through program code. There may be times in your application, that you may want to create copies of existing controls, and then remove them when they are no longer needed.

Several objects and methods exist in Envelop to help you manipulate objects. This sample will show you how a few of them work.

#### To run this sample:

1. Click the **Add** button to create a copy of the option button displayed. A maximum of (8) copies can be added to the form.
2. Click the **Remove** button to remove the last copy created.
3. Click on any of the option buttons to change the color of the box displayed at the top of the form.

### Creating An OptionButton Array

In the sample, an **Add** button is used to issue the following program code:

```
Sub btnAdd_Click()  
    Dim name As String  
    Dim o As Object  
    If OptionCounter == 8 Then Exit Sub  
    OptionCounter = OptionCounter + 1  
    name = "MyOptionButton" & OptionCounter  
    o = EmbedObject(ArrayOptBtnMasterForm, MyOptionButton, name)  
    o.Caption = name  
    o.Left = 1200  
    o.Width = 2100  
    o.Height = 300  
    o.Top = 1200 + (OptionCounter * 300)  
End Sub
```

The program code above shows how the Option Button control MyOptionButton can be copied. The method **EmbedObject** is defined by **Globals/ObjectTools** as follows:

```
Function EmbedObject(ByVal host As Object, ByVal embed As Object, ByVal name As String) As Object  
    ' Source not available  
End Function
```

A "Counter" property is used on the form to track the number of copies that have been created. Once the new OptionButton is created, it is moved into position and sized. The following Click method has been assigned to the option buttons:

```
Sub Click()  
    Dim red, green, blue As Integer  
    ' If there is no parent, then don't do anything  
    If Not Parent Then Exit Sub  
    red = rnd() * 255  
    green = rnd() * 255  
    blue = rnd() * 255  
    Parent.lblColorDisplay.BackColor = RGB(red, green, blue)  
End Sub
```

The Click method is used to change the color of a label (lblColorDisplay) on the form. A random color is generated.

### Removing An Option Button Instance

Once copies of the OptionButton have been created, they may be removed by clicking the **Remove** button on the sample form. The Remove button invokes the following program code:

```
Sub btnRemove_Click()  
    Dim o As Object  
    Dim name As String  
    name = "MyOptionButton" & OptionCounter  
    o = FindObject(ArrayOptBtnMasterForm, name)  
    DestroyObject(o)  
    OptionCounter = OptionCounter - 1  
End Sub
```

Removing objects in Envelop applications is as easy as creating them. Once again, the method called **DestroyObject**, which is defined by **Globals/ObjectTools**, was used. In order to remove the last option button from the form, the **OptionCounter** property is used to locate the last option button created. This is done by the **FindObject** method from the **Globals/ObjectTools**.

