

## ***Object Reference Sample Help***

**Sample Description:** [Object Reference](#)

### **Points of Interest**

[Creating Object References](#)

[Using Object References on the Form](#)

[Using Object References in Program Code](#)

### **Control**

Object

For Help on Help, Press F1

## Object Reference

The purpose of the Object Reference sample is to illustrate the various ways to create object references in an application, as well as to explain the reasons behind using object references. An object reference is basically a pointer from one object to another. Instead of embedding an object directly into another object, it is sometimes better to create a reference object that points to other objects.

The simplest way to illustrate an object reference is by embedding several fonts in a form and then setting the form's Font reference to one of these embedded font objects as shown below:

+ SmallFont	Font
+ MidFont	Font
+ LargeFont	Font
.	
.	
>> Font	SmallFont

When this is done, all objects on the form that have no embedded font of their own, will inherit the form's font. Changing the font object in the Font reference will change the font for the form. All controls that inherit their font from the form will display the new font.

## Creating Object References

Object references are created using the Property Editor. In this sample application, an object reference named MyRef has been created. It is displayed at the bottom of the ObjectReferenceForm property list along with the other object references.

To create an object reference, enter a name for the object then click the Object... entry from the object type combobox in the Property Editor. A Select Object dialog containing a list of Envelop objects will appear on the screen.

The sample object reference MyRef is an "Untyped" reference. This means that it is a generic object reference to any type of object. To create an untyped reference, simply click the Reference and Untyped checkboxes to True. The object list will be disabled so that a specific object cannot be chosen. Now click the OK button on the Select Object dialog, then the Property Editor. This will embed an object reference into the Property Editor.

>> MyRef                      >>Object

You can also create a specific type of object reference, such as a Font reference. This is done by only clicking the Reference checkbox in the Select Object dialog, and then clicking a specific type of object from the list. Make sure that you do not check the "Untyped" checkbox. Below is an example of a specific type of object reference:

>>MyFont                      >>Font

In this case, only font objects may be assigned as the MyFont reference. If you try to assign another type of object to the reference, an exception will be raised.

### Using Object References on the Form

This sample's form contains several types of controls such as Buttons, TextBoxes, ListBoxes, etc. A generic object reference named MyRef has also been created and added as one of the form's properties. To visually demonstrate the object reference, each of the controls on the form have a GotFocus event handler. For example, Button1 has the following GotFocus method:

```
Sub Button1_GotFocus()  
    MyRef = Button1  
End Sub
```

Each time you click the mouse on one of the controls on the form, its corresponding GotFocus event handler will be executed. This sets the generic object reference to point to itself. Clicking Button1 will change the object reference as follows:

```
>>MyRef          ObjectReferenceForm.Button1
```

To show how object references may be used, the Enable/Disable button at the lower left corner of the form executes the following code when clicked:

```
Sub BtnEnable_Click()  
    MyRef.Enabled = Not MyRef.Enabled  
End Sub
```

In essence, this click handler toggles the Enable property of the object being referenced by MyRef object reference. Other properties, events, and methods may also be accessed in this same manner.

To run the demonstration, click on various controls on the form, then click the Enable/Disable button to see the control "grayed" when it is disabled.

### Using Object References in Program Code

You may also use object references in program code. The Code Example button has a click event handler as shown below. The purpose of this code is to examine each of the controls on the form to determine which ones are buttons. When a button object is found, an InfoBox dialog is posted and the button's Caption is displayed.

```
Sub BtnCodeExample_Click()  
    Dim b As Button  
    Dim c As Control  
    Dim i As Integer  
    For i = 0 To Controls.Count - 1  
        c = Controls(i)  
        ' TypeOf checks to see if object is of a specific type  
        If TypeOf c Is Button Then  
            b = c  
        Else  
            ' This is like a "0" used for object references  
            b = Nothing  
        End If  
        ' Same as saying If Not (Nothing) Then .....  
        If b Then  
            InfoBox.Message("", b.Caption)  
        End If  
    Next i  
End Sub
```

In this example, the object reference named Controls, references a ControlGroup for the form. This control group contains information about the form, such as the number of controls that are currently embedded in the form. The "For" loop shown above cycles through each control on the form. The variable "c" is declared as a Control, which is an object reference to any type of control. The TypeOf command compares each control to the Button object. If the control is a type of Button, the InfoBox is displayed, otherwise the next control is examined.

