***Inheritance* Sample Help**

**Sample Description: Inheritance**

**Points of Interest**

- Operation Instructions
- Copying a Form through Program Code
- Deleting an Object through Program Code

**Control**

- Form
- Button

For Help on Help, Press F1

**Inheritance**

When objects are copied or abstracted in Envelop, their properties, methods, and events are passed on to the new object. This is referred to as "inheritance" in object-oriented terms. The new object automatically inherits all the characteristics of its parent. In addition, a new object may define it's own methods, properties, and events, or override the inherited characteristics of the original class.

The benefits of using inheritance are:

- ease of maintenance
- code reuse
- extensibility

When changes are made to a parent object, all child objects derived from that parent will automatically inherit the changes. Child objects can also extend the function of the parent by adding new methods, properties, and events.

The best way to understand inheritance in Envelop is to look at the Object Viewer in the "Hierarchical" view. The (+) sign indicates that additional child objects are derived from the parent object located to the right of the (+) sign. The (o) donut sign indicates that the child object has been modified from its parent object. When an object is first copied, the center of the donut is filled and looks like a round disc. When a new property, method, or event has been defined, the object is symbolized with a donut with a hollow center.

**Operation Instructions**

1. The first step in running this inheritance sample is to make sure that the Form Editor is disabled and you are able to click the Copy Form button on the sample form. This will automatically create a copy of the form and change the caption of the copied form to indicate that it is a "child" form.
2. The next step is to enable the Form Editor on the parent form. This may be done by dragging the "Finger" icon over the Titlebar of the parent form, then toggling the Form Editor icon into edit mode. A grid should be displayed on the form indicating that it has edit focus.
3. Then, click the Button icon in the Control Palette and add a button to the parent form in the area indicated. You will notice that the same button will automatically appear on the child form in the same area.
4. Select the added button and move it down to the designated area. Notice that the button on the child form also moves to the same area.
5. At this point, you can enable the Form Editor on the child form by dragging the "Finger" icon over the title bar of the child form. Now add another button in the designated area of the child form and notice that it has not been added to the parent form. Parent objects cannot inherit any changes from their child object.
6. Now enable the Form Editor on the parent form and add a second button to the parent form in the designated area. Notice how the button is also added to the child form, but also that the button number accounts for the button that had been added to the child.
7. The last step is to click the Delete Child button on the child form. This will remove the child object from the screen as well as from the object hierarchy. You may reset the sample demonstration by clicking the File|Reset menu entry.

**Copying a Form through Program Code**

In this Inheritance sample, we are copying the form that appears on the screen. This is accomplished by the CopyObject method shown in the sample code below. A unique form name is assigned by the UniqueObjectName method. These methods can be found in the Globals/ObjectTools object.

```
Sub BtnCopyForm_Click()
    Dim unique_name As String
    Dim new_form As Object
    unique_name = UniqueObjectName(InheritanceForm)
    new_form = CopyObject(InheritanceForm, unique_name)
    new_form.Show
    new_form.Move Left + (Width / 10), Top + (Height / 10), Width, Height
    new_form.Caption = unique_name & " (Child)"
End Sub
```

The last step of this routine is to display and move the new form into position on the screen and to change the caption of the form to indicate it is a child form.

**Deleting an Object through Program Code**

In order to remove the child form, the BtnDeleteForm method is executed as shown below.

```
Sub BtnDeleteForm_Click()
        ' If the form is a copy of InheritanceForm, then destroy it
        If Me = InheritanceForm Then
          ' Do nothing
        Else
          DestroyObject(Me)
        End If
End Sub
```

This code first makes sure that you can't delete the parent form (InheritanceForm), only a child form. Then it passes the reserved word "Me" on to the DestroyObject method.