Honeywell

Lab Exercise – Code an INFO button to Call Up Related Displays

57311005L
11/99

TotalPlant

Notices and Trademarks

This module supports **TotalPlant** Solution (TPS) system network.

TPS is the evolution of TDC 3000^{X}.

Honeywell Inc.
Industrial Automation and Control
Automation College
2820 West Kelton Lane
Phoenix, AZ 85053-3028
1-800 852-3211

Lab Exercise 5

Introduction

In this section you will create a button that provides an operator additional display support. In the lab exercise, you will add the following functionality to the INFO button:

- The INFO button calls up another GUS display, and if one is not available, a Native Window detail display is called up.
- The INFO button changes color to alert the operator that a problem exists.

Objectives

At the end of the lab exercise, you will be able to do the following:

- Code button script that calls up another display.
- Cause button color changes to alert the operator of a process condition requiring operator response.

Design Criteria – Digital Target

To make the INFO button work in the popup dialog means that you first have to:

- Add two display parameters that define the INFO button color and pathname to the associated GUS display.
- Modify the OnLbuttonUp and OnDataChange scripts you had coded earlier in the digital target to pass the button color and display pathname from the target to the popup dialog.

The display parameters that you will add are the following:

- Info – this parameter is defined as a data type string. It contains the path to an associated GUS display.
- Info_Color - this parameter is defined as a data type long integer. It contains the initial color for the Info button.

Code that is revised in your digital target is shown in **bold**:

```
Global Ptdesc as string
Global pname as string
Dim My_Name as string
dim State0 as string
Dim State1 as string
Dim State2 as string
Dim PntType as string
Dim NoStates as integer
Dim once as boolean
Sub OnLButtonUp() ‘touchscreen and mouse click event
       Tag.external = pname ‘send tagname string to dispdb.ent
       ‘send display.params to popup dialog
       display.params.obj.params.PName = pname
       display.params.obj.params.State0 = State0 ‘Button2
       display.params.obj.params.State1 = State1 ‘Button1
       display.params.obj.params.State2 = State2 ‘Button3
       display.params.obj.params.PtDesc = PtDesc
       display.params.obj.params.PntType = PntType
       display.params.obj.params.NoStates = NoStates
       display.params.obj.params.Info_File = display.params.Info

       ‘other targets check dispb.str01
       me.visible = true ‘target is selected
       dispdb.str01 = My_name
End Sub

Global ALM as string ‘ALM stores alarm status
Sub OnDataChange()
       On Error Goto ODC_Error

       ‘collector uses inline called point
       ‘change color if selected
       ALM = collector("ackstat(\pe(point))") ‘uses inline called point
       If My_Name = dispdb.str01 Then
              me.fillcolor = makecolor(200,200,200)
              me.visible = true
              display.params.obj.params.Alarm = ALM   ‘send status to dialog
              display.params.obj.params.Info_Color = display.params.Info_Color
       Else
              me.visible = false
       End If

‘only need to send these values once
       If not once then
              me.visible = false ‘me invisible at startup
              My_Name = display.name
              once = true ‘performant reset flag
              ‘send the following values once to popup
              pname = point.[name]
              State0 = point.State0
              State1 = point.State1
              PtDesc = point.ptdesc
              PntType = point.pntType
              NoStates = point.NoStates
              If NoStates = 3 Then State2 = point.State2
       End If
       Exit Sub
ODC_Error:
              End Sub
```

Design Criteria – Popup Dialog

To make the INFO button work in the popup dialog means that you have to:

- Define two display parameters that will receive from the digital target the INFO button color and the pathname to the associated GUS display.
- Modify the OnLButtonUp and OnDataChange scripts you had coded earlier in the digital target to receive the button color and display pathname. Also, you need to add script to the INFO button in order to call up the associated display.

The display parameters that you add to the popup are the following:

- Info_File – this parameter is defined as a data type string. It contains the path to an associated GUS display.
- Info_Color - this parameter is defined as a data type long integer. It contains the initial color for the Info button.

To update the popup dialog with a color change means that you have to modify the OnLButtonUp scripts you had coded earlier for the buttons with the following **boldfaced** statement to the button's error handler.

```
'state1 button, button1
Sub OnLButtonUp()
      On Error Goto Button_error

      display.params.Tag.op = display.params.State1

      Exit Sub
      Button_error:
            INFO.fillcolor = tdc_red 'INFO button goes red
            MsgBox "Error '" & Err & " - " & Error$ & "'"
End Sub
```

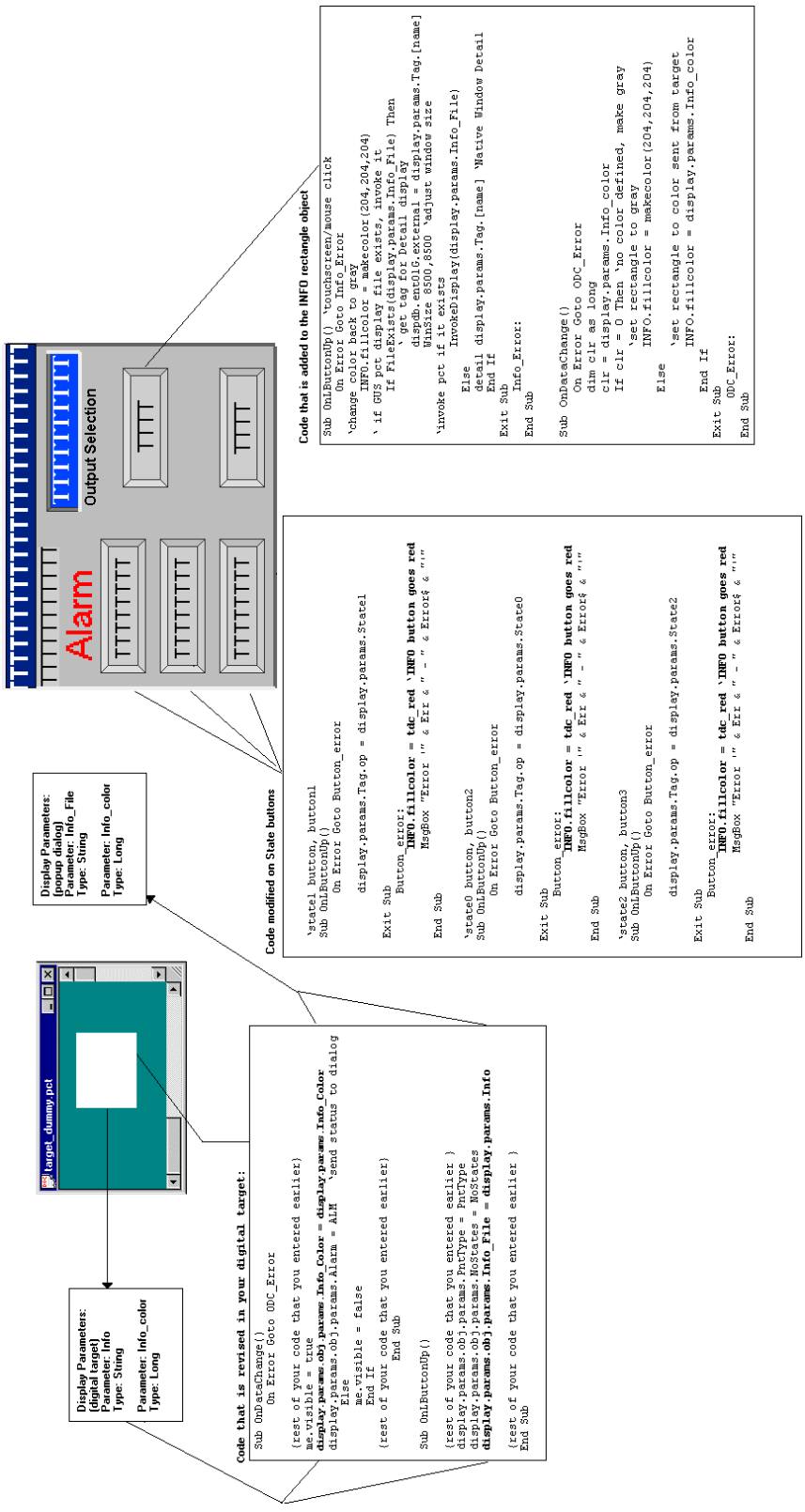
To make the INFO button call up another display means that you have to invoke another GUS display. You do that by identifying its pathname in the display parameter, Info_File. If a related GUS display does not exist for the process element, you can call up the Native Window detail display.

Code needs to be added to your popup dialog's INFO button. Add the following **boldfaced** statements.

```
Sub OnLButtonUp ()
      On Error Goto Info_Error
      INFO.fillcolor = makecolor(204,204,204) 'change color back to gray
      If FileExists(display.params.Info_File) Then ' if GUS pct display exists
            ' get tag for Detail display
            dispdb.ent01G.external = display.params.Tag.[name]
            WinSize 8500,8500 'adjust window size
            InvokeDisplay(display.params.Info_File) 'invoke pct if it exists
      Else
            detail display.params.Tag.[name] 'Native Window Detail
      End If
Exit Sub
      Info_Error:
End Sub


Sub OnDataChange()
      On Error Goto ODC_Error
      dim clr as long
      clr = display.params.Info_color
      If clr = 0 Then 'no color defined, make gray
            'set rectangle to gray
            INFO.fillcolor = makecolor(204,204,204)
      Else
            'set rectangle to color sent from target
            INFO.fillcolor = display.params.Info_color
      End If
Exit Sub
      ODC_Error:
End Sub
```

The following figure summarizes the design changes you need to make.



Lab Procedure – Modify the Digital Target to Send Color and Pathname

Step	Action
1.	From the EmbedLab4 folder, open the digital target display from the earlier Lab 4 exercise called target4.pct.
2.	Save this target as target5.pct into your EmbedLab5 folder.
3.	Add the following display parameters to your digital target (Choose Display>Define Parameters) : • Parameter: Info_Color • Data Type: Long • Initial value: 50331647 (Note: Initial value integer represents the color white. Initial value port appears on R120 and later systems) • Prompt: Enter an integer for the color of the INFO button. Example: 50331647 (Note: This parameter stores an integer representing a color to send to the popup dialog's INFO button. Assume that an associated GUS display exists for the tag. To alert the operator that another display is available, send an initial color of white.) • Parameter: Info • Data Type: string • Initial value: "c:\info.pct" (Note: Initial value port appears on R120 and later systems) • Prompt: Enter the pathname of the associated display. Example: "c:\info.pct" (Note: The pathname must be enclosed with quotes, and the display must be a validated display.)

4.	Add script to your rectangle target's OnDataChange code so that Info_color can be transferred to the dialog. The script can follow the design criteria scripting example; the code fragment that you need to add is listed below in **bold**:
	
	`{rest of your code that you entered earlier}`
	`Sub OnDataChange()`
	`  On Error Goto ODC_Error`
	
	`{rest of your code that you entered earlier}`
	`me.visible = true`
	`display.params.obj.params.Info_Color =`
	`display.params.Info_Color`
	`display.params.obj.params.Alarm = ALM  'send status to dialog`
	`  Else`
	`  me.visible = false`
	`  End If`
	`{rest of your code that you entered earlier}`
	`  End Sub`

5.	Check your syntax. (Ignore errors against inline data types.)

6.	Add additional script to your rectangle target's OnLButtonUp subroutine so that the Info's pathanme can be transferred to the dialog. The script can follow the design criteria scripting example shown earlier; the code fragment that you need to add is listed below in **bold**:
	`Sub OnLButtonUp()`
	
	`{rest of your code that you entered earlier }`
	`display.params.obj.params.PntType = PntType`
	`display.params.obj.params.NoStates = NoStates`
	`display.params.obj.params.Info_File = display.params.Info`
	
	`{rest of your code that you entered earlier }`
	`End Sub`

7.	Check your syntax. (Ignore errors against inline data types.)
8.	Validate the target5 display. (Note: you may get false validation errors against the inline data type.
9.	Save the display as target5.pct in your EnbedLab5 folder.

Lab Procedure – Modify the Popup Dialog's INFO button

Step	Action
1.	From your EmbedLab4 folder, open up your popup dialog , dig_dialog4.pct, from the previous exercise.
2.	Save the display as dig_dialog5.pct into your EmbedLab5 folder.
3.	Add the following display parameters to your popup dialog to receive values from the target: • Parameter: Info_Color • Data Type: Long • Parameter: Info_File • Data Type: string (Note: No prompts are required)
4.	Modify the INFO text object and INFO rectangle object so that it can support operator interactivity. Modify the INFO text object so that has the following properties: • Text object name: Info_text • Selectable: false • Fill: none • Text: INFO • Alignment: center • Type: string • Expression: none Modify the INFO rectangle object so that has the following properties: • Name: Info • Selectable: true

5.	Add the following OnLButtonUp script to the INFO rectangle object. The script responds to an operator selection and invokes another GUS display if the display exists. If the GUS display does not exist, the Native Window detail display is called up.
	
	`Sub OnLButtonUp()  'touchscreen/mouse click`
	`  On Error Goto Info_Error`
	`  'change color back to gray`
	`  INFO.fillcolor = makecolor(204,204,204)`
	`  ' if GUS pct display file exists, invoke it`
	`  If FileExists(display.params.Info_File) Then`
	`  ' get tag for Detail display`
	`  dispdb.ent01G.external = display.params.Tag.[name]`
	`  WinSize 8500,8500 'adjust window size`
	`  'invoke pct if it exists`
	`  InvokeDisplay(display.params.Info_File)`
	`  Else`
	`  detail display.params.Tag.[name]  'Native Window Detail`
	`  End If`
	`Exit Sub`
	`  Info_Error:`
	`End Sub`

6.	Also, add an OnDataChange script to the INFO rectangle object. The script causes the rectangle object's color to change to the one defined in the display parameter.
	
	`Sub OnDataChange()`
	`  On Error Goto ODC_Error`
	`  dim clr as long 'color variable`
	`  clr = display.params.Info_color`
	`  If clr = 0 Then 'no color defined, make gray`
	`  'set rectangle to gray`
	`  INFO.fillcolor = makecolor(204,204,204)`
	`  Else`
	`  'set rectangle to color sent from target`
	`  INFO.fillcolor = display.params.Info_color`
	`  End If`
	`Exit Sub`
	`  ODC_Error:`
	`End Sub`

7.	Check your syntax.
8.	Save the display as dig_dialog5.pct in your Embed Lab5 folder.

9.	Modify the OnLButtonUp scripts for **all** of the rectangle objects used as State change buttons. When that button is selected, the state defined in the script will be written to an inline parameter (Tag) that will be linked to an LCN tag through the "Enter Parameter" definition when this display is inserted into another display. If the digital point should have a runtime error, it causes the INFO rectangle object to change to a red color.
	
	`'state1 button, button1`
	`Sub OnLButtonUp()`
	`  On Error Goto Button_error`
	
	`  display.params.Tag.op = display.params.State1`
	
	`Exit Sub`
	`  Button_error:`
	`  **INFO.fillcolor = tdc_red 'INFO button goes red**`
	`  MsgBox "Error '" & Err & " - " & Error$ & "'"`
	`End Sub`

10.	Check your syntax.
11.	Save the display as dig_dialog5.pct in your Embed Lab5 folder.
12.	Validate the dig_dialog5.pct display. (Note: You may get false validation errors against the inline data type. Ignore them.)
13.	Save the display as dig_dialog5.pct in your Embed Lab5 folder.

Lab Procedure – Modify the Display

Step	Action
1.	From your EmbedLab4 folder, open your embed4.pct display from the previous lab exercise.
2.	Save the display as embed5.pct into your EmbedLab5 folder.
3.	Replace your previous target4.pct(s) from the embed5 display and insert the new target5.pct. (Choose Edit>Replace Embedded Displays)
4.	Be sure to: • Review the previous parameter entries Enter the values for • Info_color : 50331647 (Note: Enter this in the Initial Value expression text port) • Info: Enter any pct file for one target, leave blank for the other target so that it calls up the Native Window Detail Display. (Note: Be sure to enclose the pct file pathname in quotes).
5.	Replace your previous dig_dialog4.pct from the embed4 display and insert the new dialog, dig_dialog5.pct.
6.	Validate your embed5.pct display.
7.	Save the display as embed5.pct display into your EmbedLab5 folder.
8.	Run the display.
9.	Select a digital target. Note that the INFO button is set to white.
10.	Call up a Native Window Detail display for one of your digital composite points and set the point's attribute to PROGRAM (P-MAN). (This sets up the conditions for a runtime error. The runtime error will cause the INFO button color change to red.)
11.	From your embed5 display, select the digital composite point that is in P-MAN mode and try to change the state. Expected result: The INFO button changes to red because a runtime error occurred.
12.	Close the runtime error message dialog (Choose OK).
13.	Choose the INFO button. Expected result: The INFO button changes from red to gray. Either another pct display or the Native Window detail display is called up. Summary of INFO button behavior: When the popup dialog is first called, the INFO button is set to white. When you select the INFO button, it is set (through script) to gray and a display is called based on the INFO button's script. If you then select another rectangle target on your display, the INFO button will be set back to white because the new target that has been selected passes new data to the INFO button. If a runtime error occurs, the INFO button changes to red.

Last Page