Honeywell

Lab Exercise – Send Alarm Status to Popup Dialog

57311004L
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 4

Introduction

In this lab exercise, you use the ACKSTAT collector to send alarm status to your popup dialog. In order to send alarm status, you have to use the inline data type defined in an earlier rectangle target exercise (Lab2). Inlines are used in embedded displays whenever collector data, such as ACKSTAT, is required.

Objectives

At the end of this lab exercise, you will be able to

- Use an inline data type with a collector.
- Pass alarm status data from one embedded display to another using the object data type.

Design Criteria – Digital Target script

To continually update the popup dialog with the latest alarm status means that you will also have to modify the OnDataChange script you coded earlier for the target that will be used with your digital points. In the following lab exercise you will add the **boldfaced** statements listed below to your rectangle target.

```
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
       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

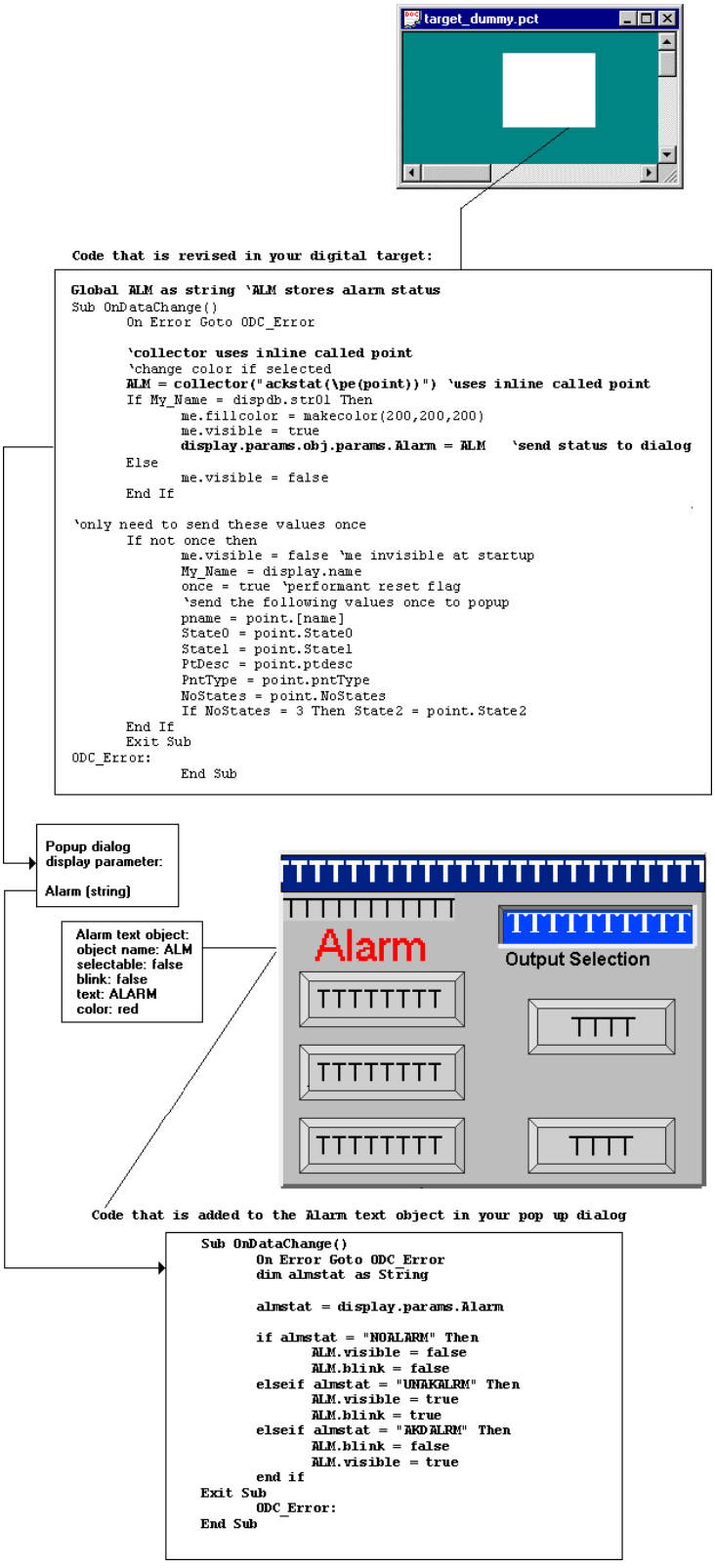
Providing the alarm status requires that an Alarm text object be added to your popup dialog that has the following script.

```
Sub OnDataChange()
       On Error Goto ODC_Error
       dim almstat as String

       almstat = display.params.Alarm

       if almstat = "NOALARM" Then
              ALM.visible = false
              ALM.blink = false
       elseif almstat = "UNAKALRM" Then
              ALM.visible = true
              ALM.blink = true
       elseif almstat = "AKDALRM" Then
              ALM.blink = false
              ALM.visible = true
       end if
Exit Sub
       ODC_Error:
End Sub
```

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



Lab Procedure – Modify the Digital Target

Step	Action
1.	From your EmbedLab2 folder, open the digital target display from the earlier Lab 2 exercise called target2.pct.
2.	Save this target as target4.pct into your EmbedLab4 folder.
3.	Add script to your rectangle target's OnDataChange code so that alarm status can be passed to the popup dialog. The script follows the Design Criteria example. Code fragments that you need to add are listed below in bold: **Global ALM as string 'ALM stores alarm status** Sub OnDataChange() On Error Goto ODC_Error **ALM = collector("ackstat(\pe(point))")** **'collector uses inline called point** **display.params.obj.params.Alarm = ALM** **'send status to dialog** (Note: The target's OnLButtonUp script does not require modification.)
4.	Check your syntax.
5.	Validate the target4 display. (Note: if you get false validation errors against the inline data type, ignore them.)
6.	Save the display as target4.pct into your EmbedLab4 folder.

Lab Procedure – Modify the Popup Dialog

Step	Action
1.	From your EmbedLab3 folder, open your popup dialog display, dig_dialog3.pct.
2.	Save the display as dig_dialog4.pct into your EmbedLab4 folder.

3.	Add a display parameter to the popup dialog that receives alarm data from the target. (Choose Display>Define Parameters).
	`Parameter: Alarm`
	`Data Type: string`
	`Prompt: none`

4.	Add a text object to your popup dialog that represents Alarm status. The text object becomes visible and blinks when an alarm condition occurs. (Display building tip: You may want to build the text object off of the dialog first to avoid selecting the wrong objects. Position the text object on the dialog after you make all of your entries.) Add a text object that is: • Object name: ALM (this name is referenced in your script) • Selectable: false • Blink set to false • Text :ALARM • Font: Arial, 12 pt, Bold • Color: Red

5.	Add script to the alarm text object.
	Sub OnDataChange()
	On Error Goto ODC_Error
	dim almstat as String
	
	almstat = display.params.Alarm
	
	if almstat = "NOALARM" Then
	ALM.visible = false
	ALM.blink = false
	elseif almstat = "UNAKALRM" Then
	ALM.visible = true
	ALM.blink = true
	elseif almstat = "AKDALRM" Then
	ALM.blink = false
	ALM.visible = true
	end if
	Exit Sub
	ODC_Error:
	End Sub
	(Display building tip: Be sure to type in the Alarm enumerations in UPPER CASE and EXACTLY as they are spelled in the Honeywell system. System enum typos are not syntax checked, which can be time consuming to debug.)

6.	Check your syntax.
7.	Validate the dig_dialog4.pct display.
8.	Save the display as dig_dialog4.pct in your Embed Lab4 folder.

Lab Procedure – Modify the Display

Step	Action
1.	From your EmbedLab3 folder, open your embed display, embed3.pct.
2.	Replace your previously inserted popup dialog dig_dialog3.pct from the embed3 display and insert your new dialog, dig_dialog4.pct. (Choose Edit>Replace Embedded Display).
3.	Replace your previously inserted targets with your new target4.pct (Choose Edit>Replace Embedded Display).
4.	Validate your display.
5.	Save the display as embed4.pct into your EmbedLab4 folder.
6.	Run the display.
7.	Cause an alarm on the digital composite point and see the results of the alarm condition on your dialog. (Changing the PVSOURCE to MAN will cause a feedback alarm when you try to open your valve.) Expected result: Your alarm text object becomes visible on the popup dialog and goes to a blinking state. Acknowledging the alarm causes the blinking text to stop blinking. (Troubleshooting tips: If your Alarm text object does not blink or appear when there is an alarm, check your Unit Assignments from the Native Window's Console Status Display. Also, check that the enumerations are typed in UPPER case and EXACTLY as the system requires them.)

Last Page