Honeywell

Lab Exercise – Use Collectors for Alarm Status

57310803L
8/00

TotalPlant

Notices and Trademarks

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

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

Honeywell

Industrial Automation and Control

Automation College

2500 W. Union Hills Drive

Phoenix, AZ 85027

1-800-852-3211

Lab Exercise

Introduction

The following example is based upon a real world example that demonstrates how to use more than one collector in an embedded display. A good display building practice in operator interface design is to provide visible changes, in this case color changes, to an operator-viewable object. For example, if a point is in alarm, you could show its acknowledged status through the use of the ACKSTAT collector and the point status collector. The point status would show whether the alarm priority is a HIGH, LOW, or EMERGENCY priority.

Objectives

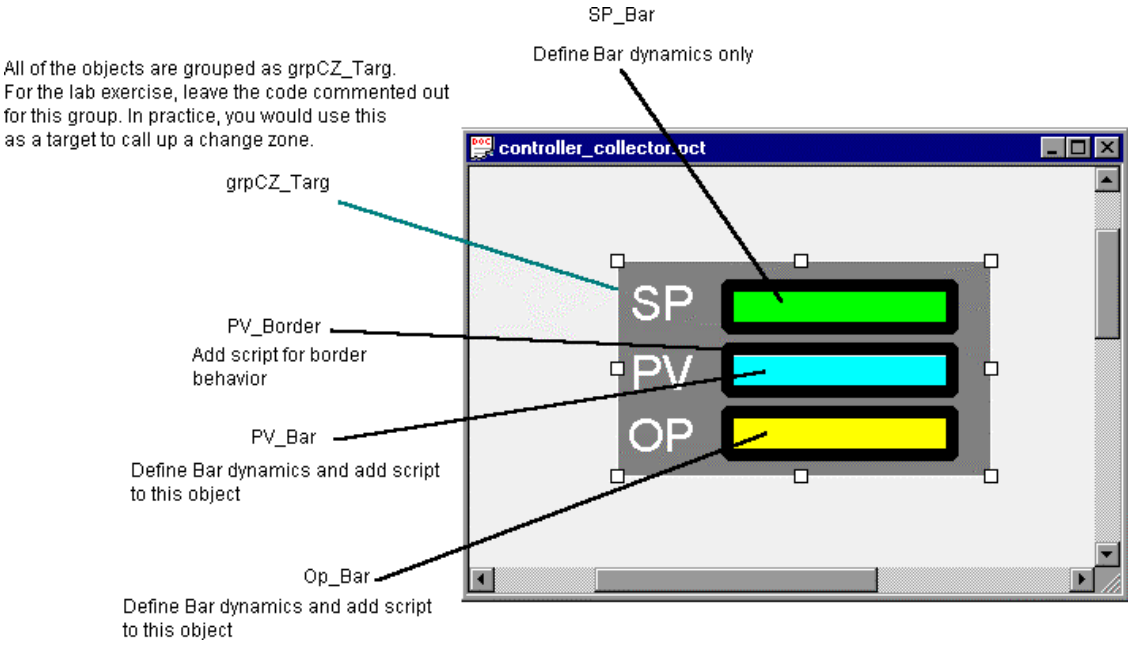
At the end of this lab exercise, you will be able to do the following:

- Use more than one collector function to provide alarm status.
- Provide a visual cue to the operator of alarm status.

Design Criteria

A controller .pct file, based upon a real world example, is provided for you. All you have to do is script the collector function(s) for the PV and MODE to indicate alarm status.

You will have to define a display parameter of type INLINE in order to use the collector functions, and add bar dynamics as shown in the following figure.



Add the following code to your grouped objects, but leave it commented out for now. In the real world, you could apply this object as a target that invokes the standard change_zone. (The rest of the code that follows for objects other than GrpCZ_Targ does not need to be commented out.)

GrpCZ_Targ

```
' THIS CODE IS USED TO CALL THE REGULATORY CONTROL
' POINT THE CHANGE ZONE (ENTITY)
' Code provided by:      [Enter your name & date]
' uncomment if needed

'Sub OnLButtonUp
' Call Change Zone and Activate it the
' point identified under ENTITY.
 'DispDB.[$CZ_ENTY].External = "\pe(ENTITY)"
'End Sub
```

Add the following dynamics to your OP_Bar object from its properties dialog. Then add the script that follows to the object OP_Bar.

```
GrpCZ_Targ:OP_Bar
-----------------------------------------------------------------------
      Bar Dynamic:
            Right: ENTITY.OPHILM
            Left: ENTITY.OPLOLM
            Origin: ENTITY.OPLOLM
            Variable: ENTITY.OP
```

Script Text:

```
' THIS CODE IS USED TO CHANGE THE COLOR OF THE
' OUTPUT BAR BASE ON THE CONTROLLER MODE OR IF
' THE CONTROLLER IS INITIALIZED EXTERNALLY.
' Code provided by:      [Enter your name & date]
Dim PntMode As String

Sub OnDataChange
On error goto Trap

' Collect Data
PntMode = ENTITY.MODE

' Conditional Behavior
IF (ENTITY.INITMAN = TRUE) THEN
'Controller is being Initialized
      Me.FillColor = MAKECOLOR(255,255,255)      'White
      Me.LineColor = MAKECOLOR(255,255,255)      'White
ELSEIF (PntMode = "AUTO") THEN
'Controller MODE is in AUTOMATIC
      Me.FillColor = MAKECOLOR(0,255,255)       'Cyan
      Me.LineColor = MAKECOLOR(0,255,255)       'Cyan
ELSEIF (PntMode = "MAN") THEN
'Controller MODE is in MANUAL
      Me.FillColor = MAKECOLOR(54,191,0)         'Green
      Me.LineColor = MAKECOLOR(54,191,0)         'Green
ELSEIF (PntMode = "CAS") THEN
'Controller MODE is in CASCADE
      Me.FillColor = TDC_YELLOW                 'Yellow
      Me.LineColor = TDC_YELLOW                 'Yellow
ELSE
'Controller MODE parameter is showing
'a bad signal read from field.
      Me.FillColor = MAKECOLOR(224,224,224)      'Gray
      Me.LineColor = MAKECOLOR(224,224,224)      'Gray
End If

Exit Sub
Trap:
      Me.FillColor = TDC_MAGENTA                'MAGENTA
      Me.LineColor = TDC_MAGENTA          'MAGENTA
End Sub
```

Add the following dynamics to your PV_Bar object from its properties dialog. Then add the script that follows to the object PV_Bar.

```
GrpCZ_Targ:PV_Bar
------------------------------------------------------------------------
      Bar Dynamic:
            Right: ENTITY.PVEUHI
            Left: ENTITY.PVEULO
            Origin: ENTITY.PVEULO
            Variable: ENTITY.PV
```

Script Text:

```
' THIS CODE IS USED TO CHANGE THE COLOR OF THE
' PV BAR BASE ON THE CONTROLLER ALARM CONDITION.
' Code provided by:      [Enter your name & date]
Dim alrmStat as String
Dim PntStat as String

Sub OnDataChange
On error goto Trap

'Collect Data
alrmStat = COLLECTOR("ACKSTAT(\pe(ENTITY))")
PntStat = COLLECTOR("$PNTSTS(\pe(ENTITY))")

' Conditional Behavior
IF (alrmStat = "NOALARM") THEN
'Bar controller is in NORMAL operation conditions
      Me.FillColor = MAKECOLOR(0,255,255)     'Bar Fill color is Cyan
      Me.LineColor = MAKECOLOR(0,255,255)     'Bar Fill color is Cyan
ELSEIF (PntStat = "UNACKEM") THEN
'Bar controller is in UNACKNOWLEDGED and PRIORITY is EMERGENCY
      Me.FillColor = TDC_RED                  'Bar Fill color is Red
      Me.LineColor = TDC_RED                  'Bar Line color is Red
ELSEIF (PntStat = "ACKEM") THEN
'Bar controller is in ACKNOWLEDGE and PRIORITY is EMERGENCY
      Me.FillColor = TDC_RED                  'Bar Fill color is Red
      Me.LineColor = MAKECOLOR(0,48,191)      'Bar Line color is Blue
ELSEIF (PntStat = "UNACKHI") THEN
'Bar controller is in UNACKNOWLEDGED and PRIORITY is HIGH
      Me.FillColor = TDC_YELLOW               'Bar Fill color is Yellow
      Me.LineColor = TDC_YELLOW               'Bar Line color is Yellow
ELSEIF (PntStat = "ACKHI") THEN
'Bar controller is in ACKNOWLEDGE and PRIORITY is HIGH
      Me.FillColor = TDC_YELLOW               'Bar Fill color is Yellow
      Me.LineColor = MAKECOLOR(0,48,191)      'Bar Line color is Blue
ELSE
'Bar controller PRIORITY is different that EMERGENCY or HIGH
      Me.FillColor = MAKECOLOR(0,48,191)      'Bar Fill color is Blue
      Me.LineColor = MAKECOLOR(0,48,191)      'Bar Line color is blue
End If
```

```
Exit Sub
Trap:
      Me.FillColor = TDC_MAGENTA        'Bar Fill color is MAGENTA
      Me.LineColor = TDC_MAGENTA        'Bar Line color is MAGENTA
End Sub
```

Add the following script to your **PV_Border** object (this is the thick line that surrounds the PV_bar).

```
GrpCZ_Targ:PV_Border
------------------------------------------------------------------------
```

Script Text:

```
' THIS CODE IS USED TO CHANGE THE COLOR OF THE
' PV BAR BORDER BASED ON THE WHETHER THE ALARM IS
' BEEN ACKNOWLEDGED OR UNACKNOWLEDGED.
' Code provided by:      [Enter your name & date]

Dim alrmStat as String
Dim PntStat as String

Sub OnDataChange
On error goto Trap

' Collect Data
alrmStat = COLLECTOR("ACKSTAT(\pe(ENTITY))")
PntStat = COLLECTOR("$PNTSTS(\pe(ENTITY))")

' Conditional Behavior
IF (alrmStat = "NOALARM") THEN
'Bar controller is in NORMAL operation conditions
      Me.LineColor = MAKECOLOR(0,255,255)  'Border Line color is cyan
ELSEIF (PntStat = "UNACKEM") OR (PntStat = "ACKEM")  THEN
'Bar controller is in ACKNOWLEDGE or UNACKNOWLEDGE and PRIORITY is
EMERGENCY
      Me.LineColor = TDC_RED               'Border Line color is Red
ELSEIF (PntStat = "UNACKHI") OR (PntStat = "ACKHI") THEN
'Bar controller is in ACKNOWLEDGE or UNACKNOWLEDGE and PRIORITY is HIGH
      Me.LineColor = TDC_YELLOW            'Border Line color is Yellow
ELSE
'Bar controller PRIORITY is different that EMERGENCY or HIGH
      Me.LineColor = MAKECOLOR(0,48,191)   'Border Line color is blue
End if

Exit Sub
Trap:
      Me.LineColor = TDC_MAGENTA
End Sub
```

Add the following dynamics to your SP_Bar object from its properties dialog. (There is no script to add to the SP Bar.).

```
GrpCZ_Targ:SP_Bar
------------------------------------------------------------------------
      Bar Dynamic:
            Right: ENTITY.SPHILM
            Left: ENTITY.SPLOLM
            Origin: ENTITY.SPLOLM
            Variable: ENTITY.SP
```

Lab Prerequisites

Lab prerequisites are the following:

- GUS Display Builder R120 or later.
- One off process LCN control point.
- Pre-built display controller_collector.pct in your student folder

Lab Procedure

Step	Action
1.	From the Display Builder, open the controller_collector.pct from your student folder.
2.	Define a display parameter of type INLINE called ENTITY for your controller_collector.pct. For the prompt, enter: Enter tagname. Example: LCN.FIC, LCN.[3MC]
3.	Add bar dynamics to your SP, PV, and OP bars as shown in the design criteria. Because you are working with a grouped object, you will want to access the script and property windows for those objects from the object browser. (Note: Depending on the LCN point type, you may have to choose different parameters if you decide to use this display in the future for more than one box/point type.)
4.	Enter code as shown in the design criteria section of the lab for your controller objects. (Note: There is a controller_collector script "text" file in your Student folder. You can copy code from this text file for your controller objects.)
5.	Syntax check your scripts.
6.	Validate your display. (Ignore any false validation errors against inlines, such as a syntax error against pntmode=entity.mode).
7.	Save your display in your student folder as controller_collector.pct.
8.	Open a new display and save it as controller_test.pct into your student folder.
9.	Insert your controller_collector.pct display into your controller_test display.
10.	Enter a control point from your partition when prompted for one. (Yes, this small object is the actual size being used in the real world!)
11.	Run your controller_test display.
12.	From the Native Window, call up a detail display of your control point and cause an alarm condition to occur. Expected result: Your controller_test display should show corresponding color changes.
13.	When satisfied that your display works correctly, save your controller_test display.

End of Lab

Last Page