

## **Lab Exercise – Introduction to InLines**

57310202L

11/99

## Notices and Trademarks

**Copyright 1999 by Honeywell Inc.  
Revision 01 Date 11/99**

Honeywell IAC courseware is subject to change without notice.

*FLEXTRAINING* courseware is copyrighted and all rights are reserved by Honeywell Inc. These materials are intended solely for use in conjunction with Honeywell products. The materials comprising the courseware may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without the prior, express written consent of Honeywell Inc.

Honeywell and **TotalPlant** are U.S. registered trademarks of Honeywell, Inc.

Other brand or product names are trademarks of their respective owners.

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

TPS is the evolution of TDC 3000<sup>X</sup>.

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

# Lab Exercise

## Introduction

When building GUS displays, embedded displays are part of your object library that can help minimize your build efforts. Embedded Displays can use either Entity, Variable, or Inline data types to represent process values.

Inline display parameters support a capability that entity and variable data types do not. That is, inline parameters support collector functions. Collector functions can provide the operator with additional process data. One commonly used collector is the ACKSTAT collector, which provides a tagname's alarm and alarm acknowledgement status.

The following exercise is based upon a real world example display that uses an inline parameter.

## Objectives

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

- Interpret the use of inlines as an embedded display data type.
- Describe the format when using an ACKSTAT collector with an inline data type.

## Lab Prerequisites

Lab prerequisites are the following:

- GUS Display Builder
- Native Window is loaded
- One off process REG AM control point
- Pre-built GUS display Level1.pct

## Design Criteria

An example display that is already built for you is in your DCH\_INC folder as Level1.pct and can be used with REG AM points (the other Level.pct is for APM points).

A listing for Level1.pct is at the back of this lab.

## Lab Procedure

Step	Action
1.	Open a new display from the GUS Display Builder.
2.	From your DCH_Inc folder, insert the display level1.pct as an embedded display into your new display.
3.	From your partition sheet, define LC_002## as your point (where ## is your partition number) when the dialog requests an ENTITY. Example : LCN.LC_00252
4.	Validate the display.
5.	Save the display as <b>inline_intro.pct</b> in your Student folder.
6.	Run your display
7.	Call up a detail display of LC_002## in the Native Window and verify that your AM point shows similar data.
8.	Cause LC_002## to go into alarm. Expected Result: The alarm behavior causes the bar to change colors and/or blink.

*End of Lab Exercise*

## Listing (Alt-L) for Level1.pct

GUS Display Builder  
Version: 0.6.1.0 (pictures\_20.21.4)  
January 18, 1999 11:35:36  
C:\Student\DCH\_Inc\level1.pct

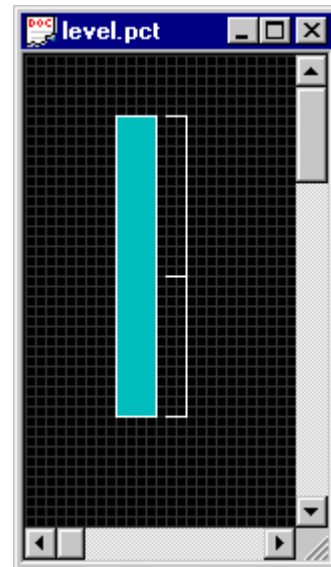
### Display

-----  
Type: MainPicture  
Visible: True  
Blink: False  
Selectable: True  
Angle: 0.000000  
Category: guspicture

Parameter: ENTITY  
Initial Value:  
Data Change Value:  
Type: Inline  
Data Change Event On Assignment: True  
Appears On Enter Parameters: True

### Rectangle1

-----  
Type: Rectangle  
Visible: True  
Blink: False  
Selectable: False  
Angle: 0.000000



## Rectangle1

```

-----
Type: Rectangle
Visible: True
Blink: False
Selectable: False
Angle: 0.000000
Line Width: 1
Line Style: 0
Line Color: 0x2ffffff
Scale X: 1.000000
Scale Y: 1.000000
Trans X: 0.000000
Trans Y: 0.000000
Filled: True
Fill Color: 0x2bfbf00
Fill Pattern: 15
Fill Direction: 3
Fill Percent: 100
Inherit: Disabled
Property Inheritance :
    None
Bar Dynamic:
    Bottom: ENTITY.PVLOTP
    Top: ENTITY.PVHHTP
    Origin: ENTITY.PVLOTP
    Direction: 0
    Expression: ENTITY.PV

Script Text:
'Code provided by:      [Enter your name & date]
'This is used to represent a level indication bar in a vessel
'Its input is PV of a level controller point or a level indicator
'If the point is not in alarm it show TDC_cyan
'If the point is in alarm unacknowledge it shows RED BLINKING
'If the point is in alarm acknowledge it shows RED NOT BLINKING
'If a bad value is received from the field it shows BLUE
Sub onDataChange
On Error Goto catch
'Conditional Behavior
IF ( COLLECTOR("ACKSTAT(\pe(ENTITY))") = "NOALARM") THEN
    Me.FillColor = TDC_CYAN
    Me.Blink = FALSE
ELSEIF ( COLLECTOR("ACKSTAT(\pe(ENTITY))") = "UNAKALRM") THEN
    Me.FillColor = TDC_RED
    Me.Blink = TRUE
ELSEIF ( COLLECTOR("ACKSTAT(\pe(ENTITY))") = "AKDALRM") THEN
    Me.FillColor = TDC_RED
    Me.Blink = FALSE
End If
Exit Sub
catch:
'Bad Value Behavior
    Me.FillColor = TDC_BLUE
    Me.Blink = FALSE
End Sub

```

Polygon1

-----  
Type: Polygon  
Visible: True  
Blink: False  
Selectable: False  
Angle: 0.000000  
Line Width: 1  
Line Style: 0  
Line Color: 0x2ffffff  
Scale X: 1.000000  
Scale Y: 1.000000  
Trans X: 0.000000  
Trans Y: 0.000000  
Filled: False  
Fill Color: 0x2ffff00  
Fill Pattern: 15  
Fill Direction: 3  
Fill Percent: 100  
Inherit: Disabled  
Property Inheritance :  
    None

Data Collection:

Name	Rate	Group
------	------	-------

-----  
Empty

Listing Options:

Properties: True  
Scripts: True  
Dynamics : True  
Embedded Displays: True  
Parameters: True  
Parameter Bindings: True  
Inheritance Flags: True  
Data Collection: True

**Last Page**