Honeywell

Lab Exercise: Introduction to GUS Solution Pack

57310901L
07/00

TotalPlant

Notices and Trademarks

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

TPS is the evolution of TDC 3000X.

Honeywell Inc.
Industrial Automation and Control
Automation College
2500 W. Union Hills Drive
Phoenix, AZ 85027
1-800 852-3211

Lab Exercise

Introduction to GUS Solution Pack

Introduction

The following lab exercise provides practice in the basics of building an operator interface using the GUS Solution Pack.

Objectives

After completing this lab exercise, you will be able to

- Build an operator interface using the GUS Solution Pack.
- Interpret design strategies using GUS Solution Pack in terms of the following:
  - Applying GSP to an interface
  - Evaluating the design strategy of GSP components such as the display template(s), task panel, and SafeView templates.

Duration

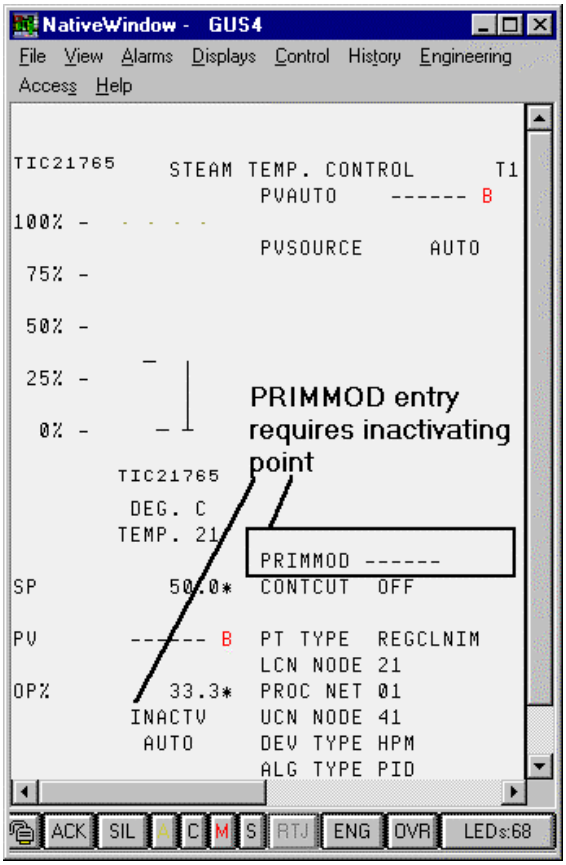
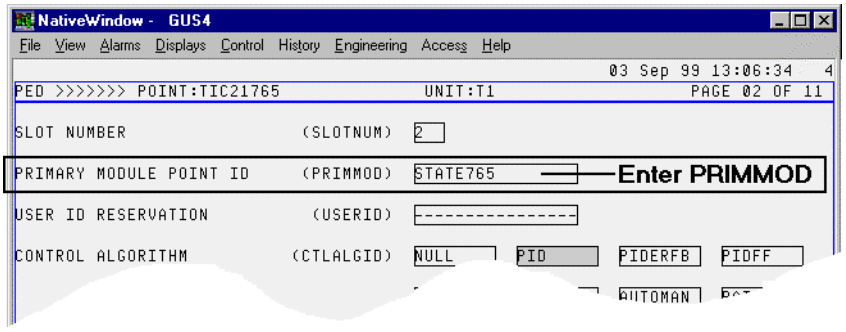
Estimated time to complete this lab is 60 minutes.

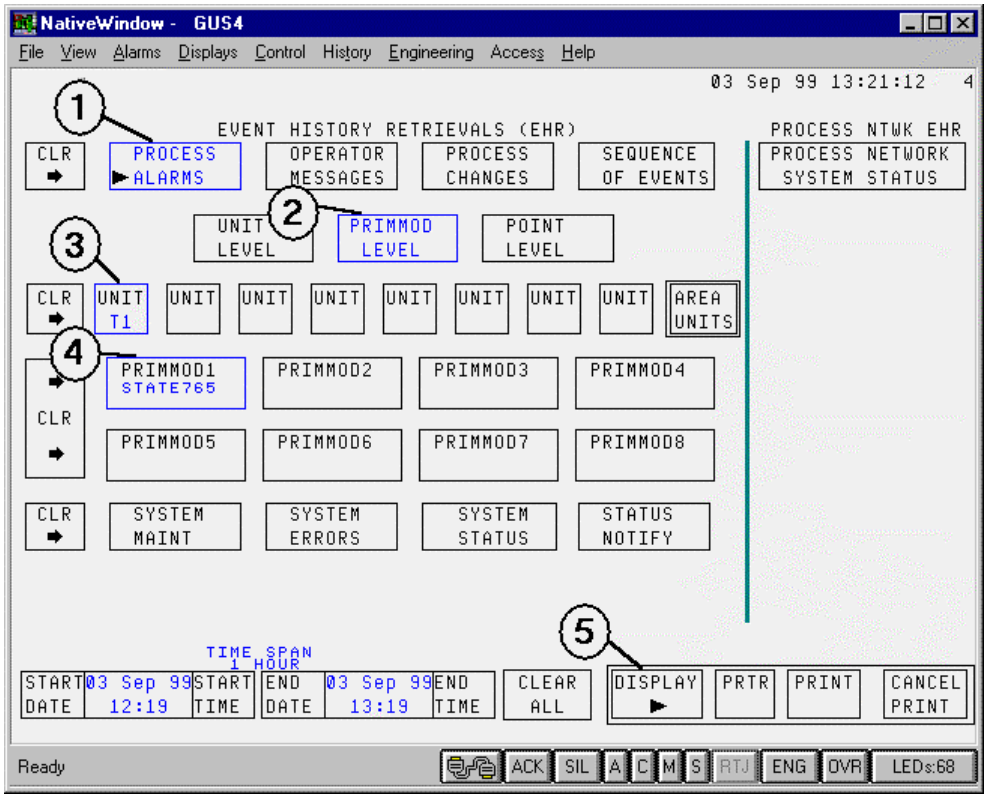
Lab Overview

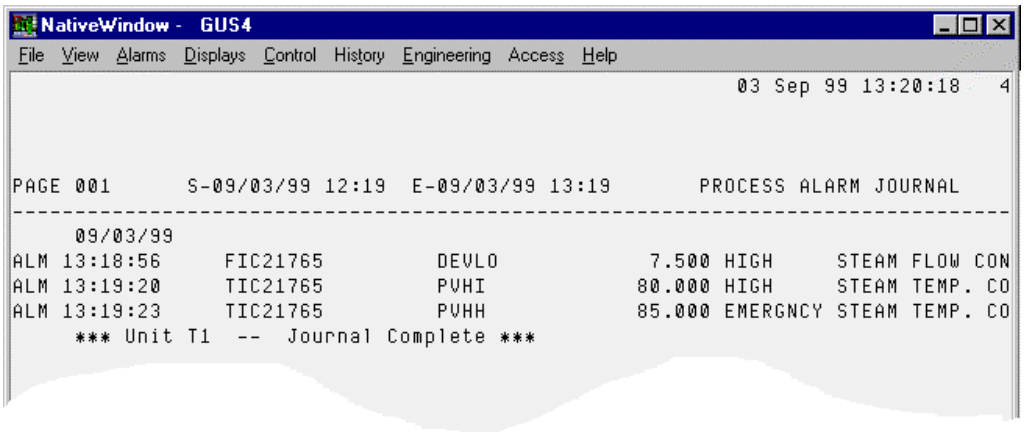
In the following lab exercise, you begin using the GUS Solution Pack to build an operator interface. Your primary focus should be on how to deploy GSP, do not worry about how aesthetically pleasing your display appears. For example, your static vessel shapes can simply be rounded rectangles. An overview of the lab exercise procedures follows:

1. Define PRIMMOD collectors. In this procedure, you specify a PRIMMOD that is used later in your task panel to annunciate alarms.
2. Copy GSP templates over to a folder. You DO NOT want to overwrite the GSP templates; the GSP templates are your master copies for future display builds.
3. Build several process displays. The static portion of these displays can be relatively simple, your main focus is to become familiar with the dynamics of GSP display components. The displays you build will be invoked from the task panel as high and low level displays.
4. Build a simple task panel. Recall that the task panel is the display that provides navigation to other displays.
5. Configure SafeView. You use a GSP SafeView template to manage your task panel, process displays, and faceplate

Define PRIMMOD collectors

Step	Action
1	From your class project partition, choose a flag to become a PRIMMOD collector. Example: STATE###. Write the name of your PRIMMOD point here_____.
2	Inactivate control point TIC21### 
3	Reconstitute control point TIC21###.
4	Define its PRIMMOD parameter to reference STATE###. 
5	Load the control point TIC21###.
6	Repeat steps 2 through 5 for FIC21###.

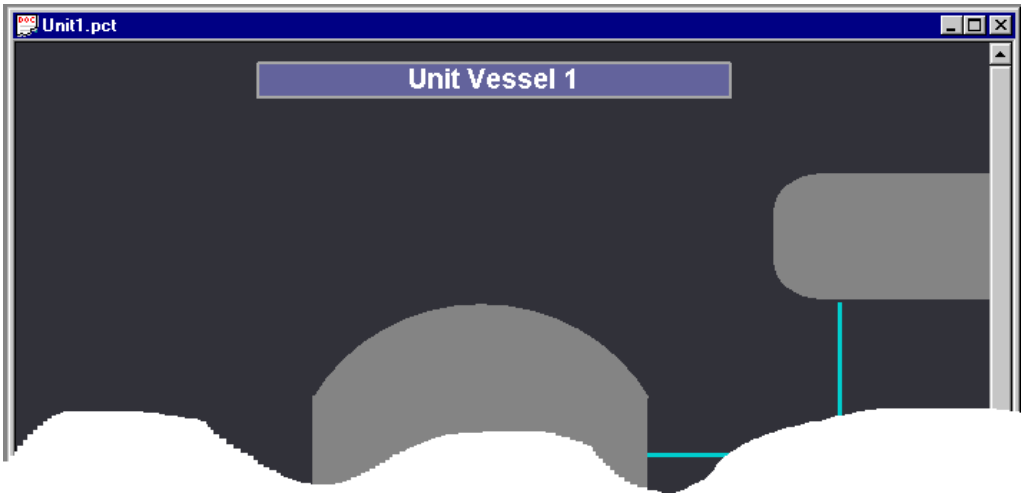
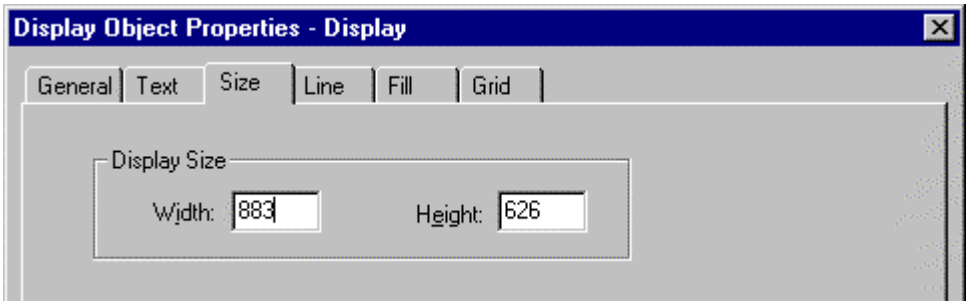
Step	Action
7	Make your control points active. Result: Both control points have STATE### enabled as it PRIMMOD collector.
8	(Optional steps 8 to 10) To verify that your PRIMMOD collectors work, place FIC21### and TIC21### into an alarm condition.
9	Call up the Event Retrieval display from the Native Window's System Menu. 
10	Select the PRIMMOD target and enter the tagname STATE### assigned as your PRIMMOD.

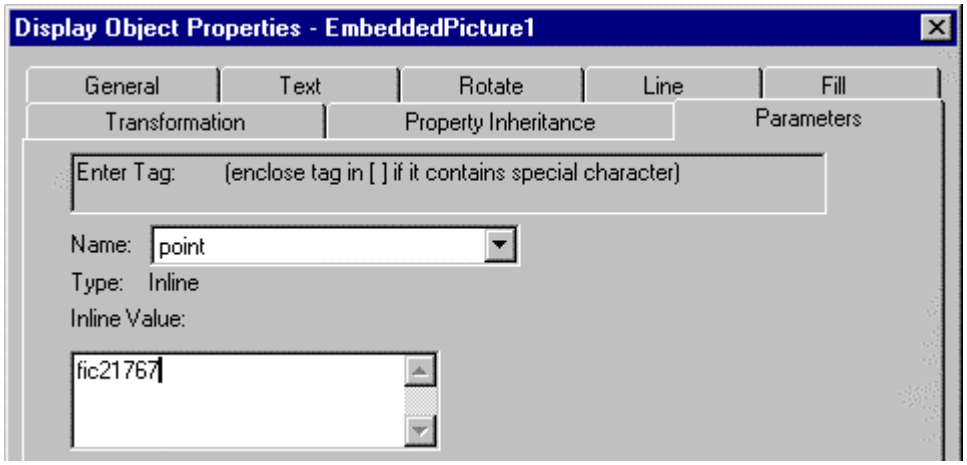
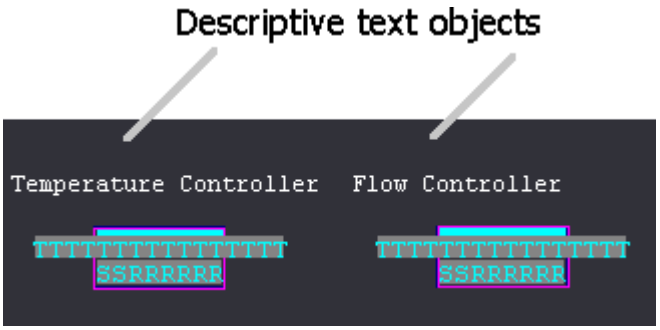
Step	Action
11	Select the DISPLAY target. Result: The PRIMMOD displays the alarm events. 

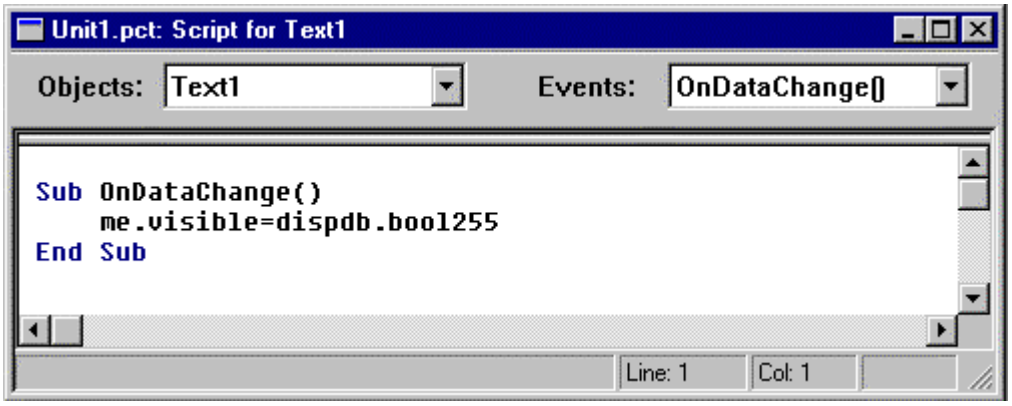
Copy GSP templates

Step	Action
1	Open Explorer and locate the folder GSP (path Program Files>Honeywell>Tps >Gsp)
2	Open the folder labeled Template.
3	Copy (Ctrl+C) the file DisplayTemplate_GSP100_2.pct.
4	Open your student folder and do the following: • Paste (Ctrl +V) the DisplayTemplate_GSP100_2.pct into your student folder. • Rename the file to Unitx.pct (You can rename the file by right clicking on the file and selecting rename from the context menu.) Result: You now have a “master” display template to build your process displays.
6	Return to the GSP folder and open the Workspaces folder.
7	Copy the file Gsp1024x768_fp2_1screen_GSP100_2.wdl.
8	Return to your student folder and do the following: • Paste the Gsp1024x768_fp2_1screen._GSP100_2.wdl into your student folder. • Rename the file to Safeview_Unit.wdl Result: You now have a “master” SafeView template to manage your process displays. Later in the lab exercise you will specify its operating parameters.
9	Return to the folder labeled Template.
10	Copy the file TaskPanelTemplate_GSP100_2.pct.
11	Return to your student folder and do the following: • Paste the TaskPanelTemplate_GSP100_2.pct into your student folder. • Rename the file to TaskPanelUnit.pct Result: You now have a TaskPanel template that can provide navigation to your process displays. Later in the lab exercise you will add buttons to the Task Panel that call up operating displays.
12	Open the folder Display (path Program Files>Honeywell>Tps >Gsp).
13	From the folder labeled Displays, copy the file GSPplate_fp2_123x626_GSP100_2.pct
14	Return to your student folder and do the following: • Paste the GSPplate_fp2_123x626_GSP100_2.pct into your student folder. • Rename the file to GSPplate.pct (Note: the file must be renamed as GSPplate because pre-scripted displays connect to the faceplate referenced as GSPplate.) Result: You now have a faceplate that will appear in your SafeView configuration whenever a tag is selected.
15	While in Explorer, select the properties of **each** of the files you have copied and renamed and remove the “read-only” attribute checkmark.
16	Minimize Explorer.

Build a Process Display

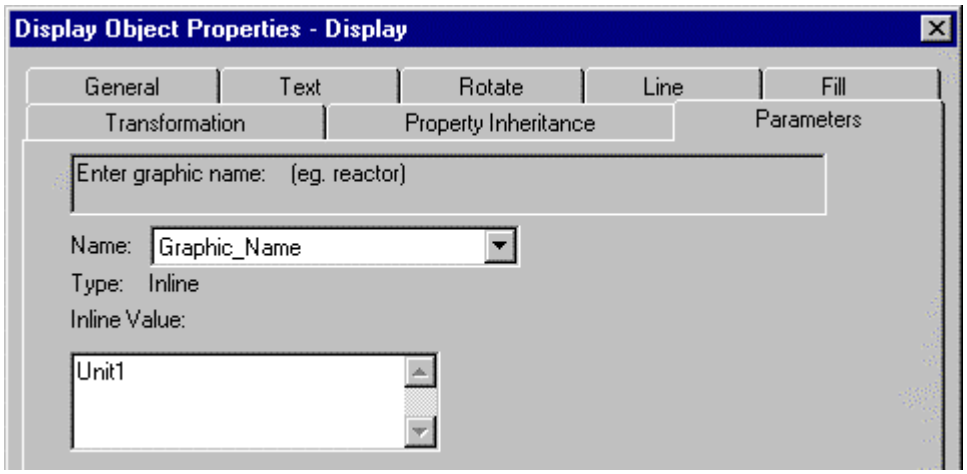
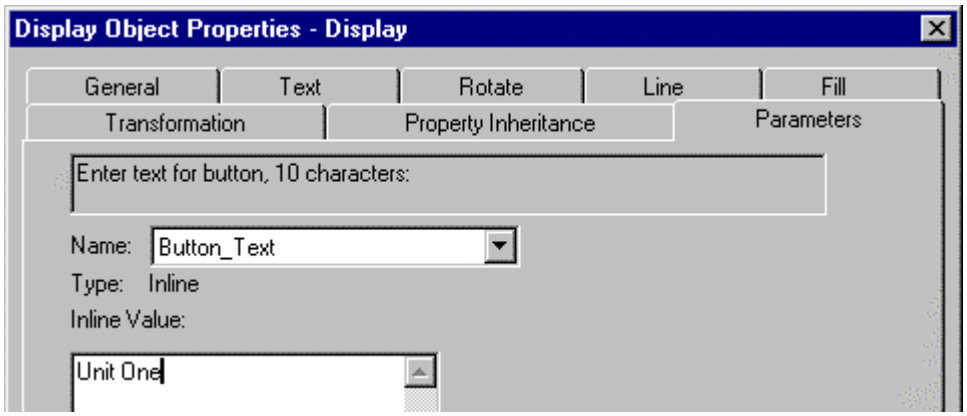
Step	Action
1	Invoke Display Builder and open the file you had named earlier as Unitx.pct.
2	Change the text descriptor of your display from “GUS Display Template” to “Unit Vessel 1”.
3	Insert one or more static objects into your display that can represent vessels. Do not be too concerned about how realistic your shapes appear. 
4	If necessary, modify the size of your display’s width and height so that it will fit into the SafeView template. 
5	Save the display in your student folder as Unit1.pct.
6	In order to view the display components that you insert in the next steps, enable **View>Invisible** Objects.
7	In the following steps, add several display components that provide process data. First, choose **Insert>Display**. Locate the folder DisplayComponent (path Program Files>Honeywell>Tps >Gsp).
8	Select and open the component that represents a large control box. The component is labeled cntl_box_lg_16_GSP100_2.pct Note: GUS Solution Pack components use a naming convention that describes the component. For example, a large control box supporting 16 characters is represented as cntl_box_lg_16.

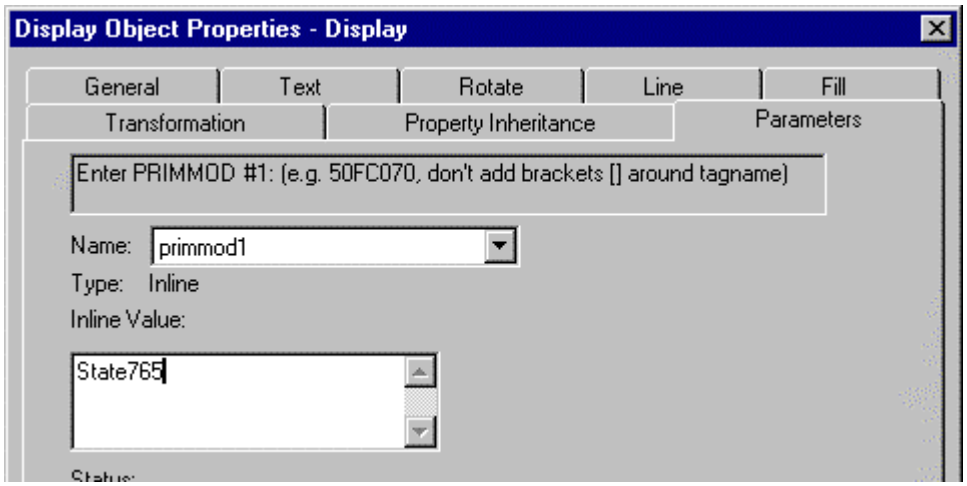
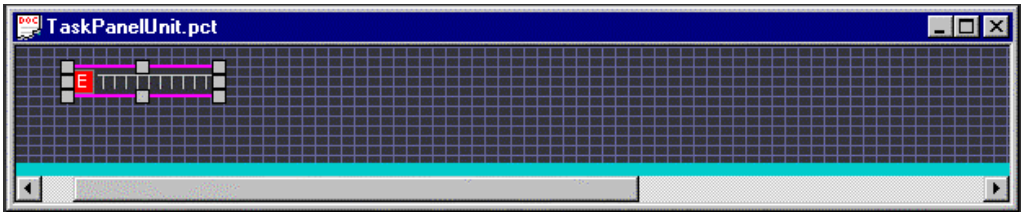
Step	Action
9	Insert the component into your process display. Respond to the first parameter prompt requesting a Point with the following data: Enter a control point from your partition, FIC21###. Note that you do not need to enter an LCN prefix. The remaining prompts can be left at their default values. 
10	Repeat the previous steps 7 through 9 for another control point, TIC21###.
11	Add text objects that describe your controllers. Example text: Temperature controller, flow controller. 

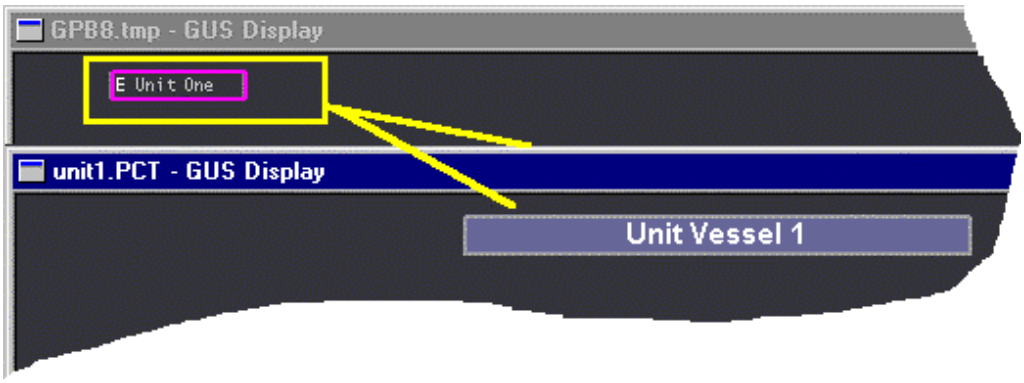
Step	Action
12	Add the following script ONLY to your descriptive text objects for the 2 controllers.  (Note: the script works with a Task Panel's Show/Hide button that you use later in the lab. DispDB.bool255 represents a boolean value that is toggled true/false to Show/Hide text.)
13	Insert a component that represents a horizontal control valve. You can use the component labeled cntl_vlv_hor_GSP100_2.pct
14	Respond to the screen prompts with the following data: Point: Enter a control point from your partition, FIC21###. Position the component into your process display near the large control box for FIC21###.
15	Run the display to confirm that it operates.
16	Validate and save the display as Unit1.pct.
17	Save the display again as Unit2.pct
18	Re-label the text descriptor on your display from Unit Vessel 1 to Unit Vessel 2. Validate and save this display. Even though your display uses the same control points, the Task Panel that you build later in the lab can have buttons to call up either the Unit1 or Unit 2 display.

Build a Task Panel Display

Step	Action
1	From your student folder, open the file you had named earlier as TaskPanelUnit.pct.
2	Zoom to a viewing size such as 100% or greater. (Choose View>Zoom>100%). Also, set your view selection to View Invisible Objects.
3	From the GSP's DisplayComponent folder, insert a navigation button. Choose the button identified as navbtn_hilvl_aggr_10_GSP100_2.pct. Note: The component naming convention represents a high level navigation button with 10 characters.

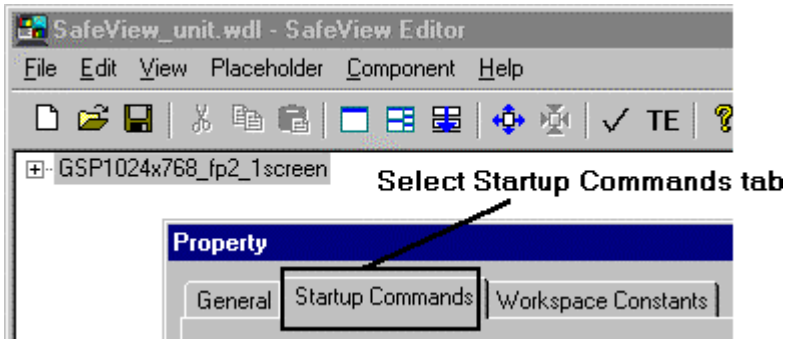
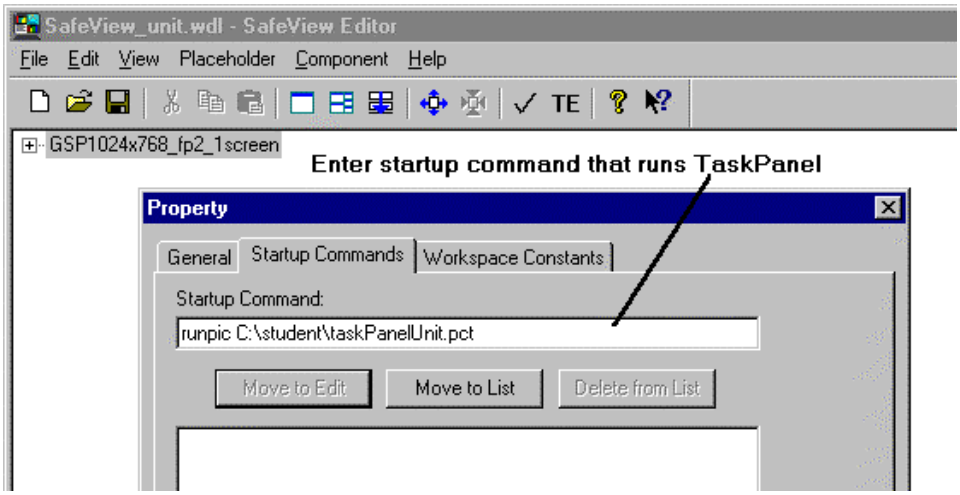
Step	Action
4	Respond to the first parameter prompt by entering just the name of the display that you want the button to invoke. Example entry: Unit 1. It is not necessary to enter a pathname. However, the folder you store the file into **MUST** be part of your Area Database's pathname catalog. 
5	Choose the Next button in the dialog.
6	Respond to the second parameter prompt by entering a label for your navigation button. Example entry: Unit One. 
7	Choose the Next button in the dialog.

Step	Action
8	Respond to the third parameter prompt by entering your PRIMMOD name. Example entry: State**###**, where **###** is your partition number. 
9	Choose OK. (It is not necessary to enter the remaining 22 parameters as they refer to additional PRIMMODs that you can add.) Click on the location in your display where you would like the button inserted. Result: Your button appears in the Task Panel at the inserted location. 
10	Validate and save your Task Panel Unit display.
11	Run your TaskPanelUnit display.

Step	Action
12	Choose the Unit One button. It should invoke your Unit1 display.  Note: If your Unit1 display does not appear, verify that your Unit1.pct is closed in Display Builder. The Unit1 display also must be located in a folder that is referenced by your pathname catalog.
13	After verifying that the Unit 1 display is invoked, close the runtime versions of the Task Panel and Unit1.
14	Repeat the above steps 3 through 12 for a second button to invoke a Unit2 display.
15	Verify that your TaskPanel can invoke both the Unit1 and Unit2 display.
16	Close all displays and the Display Builder.

Enable a SafeView configuration

Step	Action
1	Start the SafeView Workspace editor. (Choose Start>Program Files>Honeywell TPS>SafeView Workspace Editor).
2	From your student folder, open the SafeView template file named SafeView_Unit.wdl.
3	Double click on the Workspace Title. 

Step	Action
4	Select the Startup Commands tab. 
5	Add a startup command that will run the TaskPanelUnit.pct you had created earlier in the lab exercise. (Note: Be sure to reference the Task Panel you had created earlier in the lab exercise and not your process overview display.) Example entry: RUNPIC C:\student\TaskPanelUnit.pct 
6	Save your SafeView configuration.
7	To verify that your SafeView template operates correctly, load the SafeView workspace with the SafeView_unit.wdl file in your student directory. (Choose Start>Programs>Honeywell TPS>SafeView. Click on Load Workspace from the SafeView Control Panel, then open your SafeView file.) Note: if your SafeView configuration does not appear to load properly, ask your course manager for assistance. Likely causes of not loading include referencing non-validated displays, or not referencing your Task Panel in the startup command properly, or a need to setup the Environment Variables. The Environment Variables can be accessed from Control Panel>System>Environment, then scroll to verify the path for your GUS applications).
8	(Optional step) Hide the SafeView control panel.
9	Click on any of the navigation buttons that appear in the TaskPanel. Result: The referenced process display is called up.

Step	Action
10	Click on any display component in your process display. Result: The selected component border changes and becomes highlighted. The Faceplate appears and updates to show the tag of the selected component.
11	Click on the Show and Show Tag/Setpoint buttons in the Task Panel. Result: The Show button toggles the text descriptor visibility. The Show Hide button for setpoint and tag toggles the visibility of the setpoint and tag for the control box components.

End of Lab

Last Page