

Lab Exercise – Introduction to the PMK Object

57311303L

3/00

Notices and Trademarks

**Copyright 2000 by Honeywell International Inc.
Revision 02 Date 3/22/00**

Honeywell International courseware is subject to change without notice.

FLEXTRAINING courseware is copyrighted and all rights are reserved by Honeywell International Inc. These materials are intended solely for use in conjunction with Honeywell International Inc 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 International Inc.

Honeywell and **TotalPlant** are U.S. registered trademarks of Honeywell International 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^X.

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

Lab Exercise

Introduction

The Point Manipulation Key (PMK) object is a built-in GUS scripting object that implements the point manipulation keys (ramp keys) found on the Integrated keyboard (IKB) and other Honeywell engineering keyboards. This lab exercise is a concept lab exercise in that the behavior of the PMK object is introduced. The PMK object's behavior must be understood before you begin applying PMK in a plant application. Those behaviors are the objectives for this lab exercise.

Objectives

Upon completing this lab exercise, you will have an understanding of the PMK object's behavior in terms of

- Registering the PMK object using an OnGotFocus event.
- Deactivating the PMK object using an OnLostFocus event.
- Enabling PMK error handling.

Design Criteria

To demonstrate PMK behavior, you will create a display with two text objects. These text objects will display the SP value for two regulatory control points. You will then add PMK related scripts to each object so that when you click on the SP value (which has the event “OnGotFocus”) that point will be registered with the PMK object and the IKB single arrow ramp key will then be directed to that data point. The scripts that you add to the text objects are similar to the following examples:

'Registering the PMK object:

```
Sub OnGotFocus()  
    set pmk.errorhandler = me          'enable PMK error handling (e.g., invalid mode)  
    set pmk.entity = lcn.fc_01###      'must register the entity  
    pmk.key pmk_sp                     'can ramp SP, OP; change modes from IKB  
End Sub
```

'Deactivating the PMK object

```
Sub OnLostFocus()  
    pmk.clear                          'deactivate PMK for this point  
End Sub
```

'PMK error handling

```
Sub OnPMKError(ErrCode As Long, ErrString As String)  
    msgbox errstring                  'see errors like invalid mode  
End Sub
```

'update display through script or by using the value tab in the properties of the text object

```
Sub OnDataChange()  
    me.text = lcn.fc_01###.sp         'update point (OnGotFocus does not update)  
End Sub
```

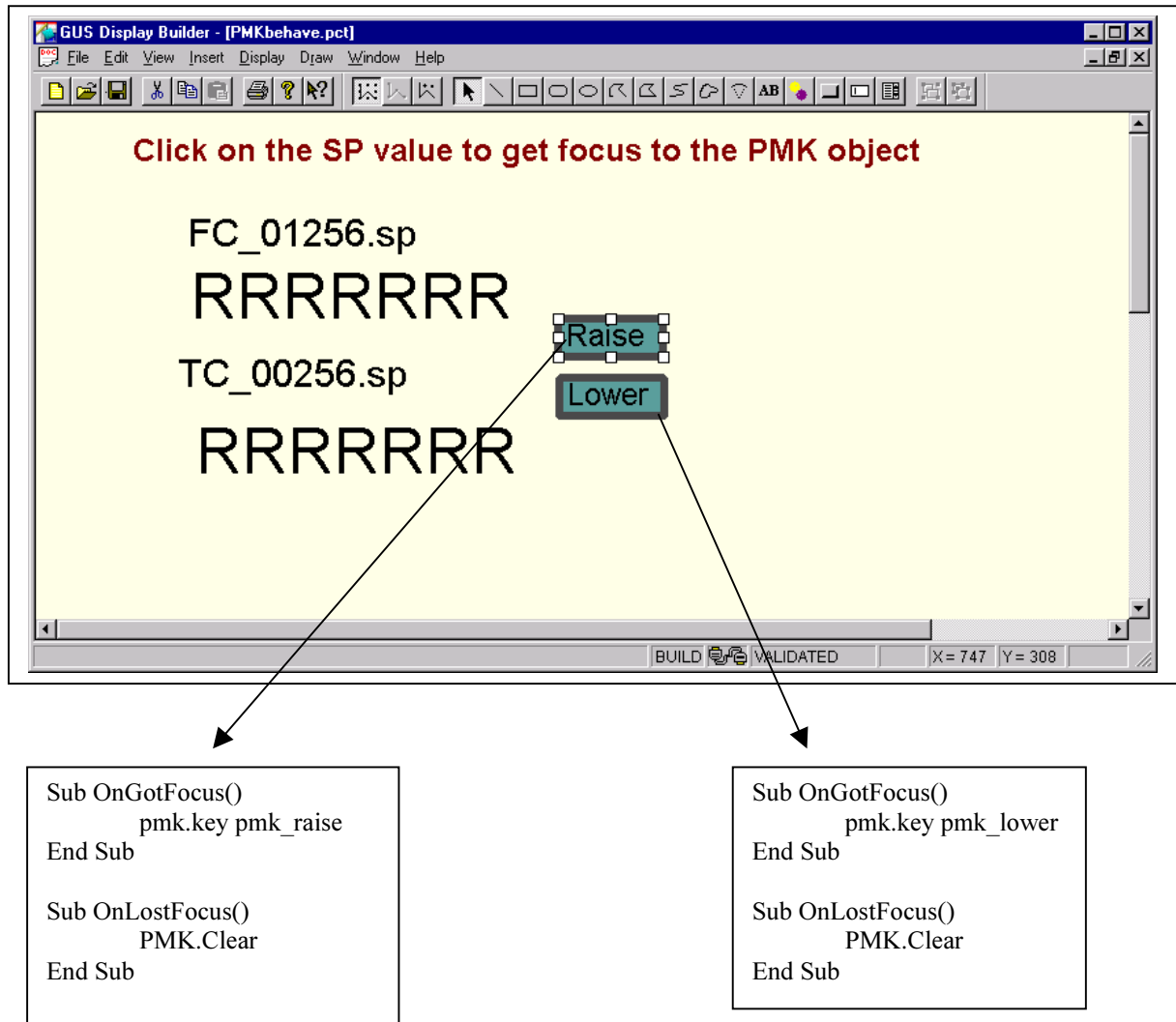
Lab Prerequisites

In order to complete the lab exercise, you will need the following:

- GUS display Builder at Release 120 or later.
- Native Window loaded with access to the setpoints of two process network points.

Note: The above Design Criteria “assumes” the user has an IKB. If the GUS you are using does not have an integrated keyboard, see the following display example. This will demonstrate how to implement a “virtual” raise and lower ramp key through objects added on a graphic display.

The Raise / Lower buttons will have to be added if you do not have an IKB.



NOTE: You can use either the “OnLButtonUp” event or the “OnGotFocus” event to implement the raise and lower functions of the PMK object. Also, if you code each of your RAISE and LOWER buttons with an “OnLostFocus” event, as shown in the above example, you must then re-register the PMK entity each time you want to use either button. However, if you want to be able to use both buttons once an entity has been registered (to raise and/or lower the value), then put the raise and lower functions on the buttons, and do **not** put the “OnLostFocus” event on each button. Instead, group the buttons together and put an “OnExitFocus” event on the button group to clear the PMK object.

Lab Procedure

Step	Action
1.	Open a new GUS display.
2.	<p>Add a text object to the GUS display that will be used to represent a setpoint value. (Note: The setpoint value can be displayed either by script <u>or</u> by entering the tag.sp on the value tab in the properties of the text object. (i.e. LCN.FC_01###)</p> <p>'update SP value on display using script</p> <pre> Sub OnDataChange() me.text = lcn.fc_01###.sp End Sub </pre>
3.	Save this display as PMKbehave.pct into your PMKLab folder.
4.	<p>Add script to your text object that enables access of the PMK keys of an Integrated keyboard for ramping. Your script will look similar to the following:</p> <p>'Registering the PMK object:</p> <pre> Sub OnGotFocus() set pmk.entity =lcn.fc_01### 'register the entity pmk.key pmk_sp 'register SP parameter for ramping End Sub </pre> <p>Note: The pmk.key line of code defines the <u>default</u> parameter for the PMK object. However, once a point has been registered, you can also use the IKB to change the mode as well as select the OP key and ramp the output value.</p>
5.	Syntax check your code.
6.	Validate the display.
7.	Save this display in your PMKbehave folder.
8.	Run the display.
9.	Open a Native Window detail display for your fc_01### point.
10.	Select the text object on your PMKbehave display to register your data point.
11.	<p>From the Integrated keyboard use the single-arrow ramp keys to ramp the setpoint value.</p> <p>Expected result: The setpoint does ramp, as observed in the Native Window detail display and on your GUS graphic (SP value).</p>
12.	<p>Now select the “MAN” mode key on the IKB. Observe in the Native Window that the mode has changed, as requested. Select the OP key on the IKB and ramp the value. The PMK object is now directed to change to the OP parameter of the registered entity.</p> <p>(Note: Although your script originally accessed the setpoint parameter, you can also make output and mode changes while the PMKbehave display has focus and your entity is still registered.)</p>
13.	From the Native Window detail display, set up an error condition, such as placing the point in the wrong mode attribute so that the setpoint or output <u>cannot</u> be changed from the keyboard. (For example, place the point in PROGRAM mode attribute.)
14.	<p>From your PMKbehave display, use the ramp keys again.</p> <p>Expected result: No changes are occurring in the Native Window, and no runtime error is displayed in the PMKbehave display. PMK error trapping needs to be enabled in your PMKbehave display.</p>

Step	Action
15.	Close your runtime PMKbehave display.
16.	<p>Add the following boldfaced statements to your text object so that the display can trap errors. Your script can look similar to the following:</p> <pre> Sub OnGotFocus() set pmk.errorhandler = me 'enable error handling set pmk.entity = lcn.fc_01### 'register the entity pmk.key pmk_sp 'ramp SP value End Sub 'PMK error handling Sub OnPMKError(ErrCode As Long,ErrString As String) msgbox errstring 'see errors End Sub </pre>
17.	Syntax check your code.
18.	Validate the display.
19.	Save this display in your PMKbehave folder.
20.	Run the display.
21.	Make sure an error condition still exists on your point, such as giving the point a PROGRAM mode attribute so that the setpoint <u>cannot</u> be changed from the keyboard.
22.	<p>Select your text object to register your data point and create a link between the PMK object and the IKB. Use the ramp key on the IKB to change the SP value.</p> <p>Expected result: The PMKbehave display now has a message box indicating the error.</p>
23.	Close your runtime PMKbehave display.
24.	<p>Add another text object to your display that represents the setpoint for another control point (i.e. LCN.tc_00###)</p> <p>Copy the code from the previous object to the new text object and change the tagname reference. Your code can look similar to the following:</p> <pre> Sub OnGotFocus() set pmk.errorhandler = me enable error handling set pmk.entity = lcn.tc_00### 'register the entity pmk.key pmk_sp ' ramp SP value End Sub 'PMK error handling Sub OnPMKError(ErrCode As Long,ErrString As String) msgbox errstring 'see errors End Sub 'update display using script or properties Sub OnDataChange() me.text = lcn.tc_00###.sp 'update SP value End Sub </pre>
25.	Syntax check your code.
26.	Validate the display.
27.	Save this display into your PMKbehave folder.
28.	Run the display.

Step	Action
29.	From the PMKbehave display, select the first text object and ramp its setpoint. Correct any error conditions that may still exist on this point. Result: The first point's SP value changes.
30.	From the PMKbehave display, select the second text object and ramp its setpoint. Result: The newly selected point is now registered with the PMK object and <u>its</u> SP value is the one being changed. Caution: If you had selected another object in your display that didn't re-register with the PMK object, then the object that was last registered with the PMK would still have IKB focus and any ramping done would affect that data point. You <u>must</u> deactivate the PMK object when your data point is no longer using it.
31.	Close the running PMKbehave display.
32.	For each text object in your PMKbehave display, add an "OnLostFocus" event to clear the entity that has registered with the PMK object. Your script may look similar to the following: <pre>'deactivate the registered PMK entity Sub OnLostFocus() PMK.Clear End Sub</pre>
33.	Syntax check your code.
34.	Validate the display.
35.	Save this display in your PMKbehave folder.
36.	Run the display.
37.	Select your first point to register it with the PMK object. Using the IKB, ramp the SP value. Now select another object in your display (you can click on the display "background") so that your first point no longer has focus. Use the ramp keys on the IKB to change your first point's SP value. Expected result: The SP value is not changing because the "OnLostFocus" event cleared the entity name from the PMK object.

Last Page