

***TPS R500—R530  
US Implementation***

***Lab Exercise and  
Evaluation***

***Build Custom Alarm  
Schematic***

**L53435L  
LCN**

# Copyright, Notices, and Trademarks

---

© Copyright 1996, 1998 by Honeywell Inc.

Revision 07– February 11, 1998

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 for use solely 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.

*FLEXTRAINING* and **TotalPlant** Solution (TPS) are trademarks of Honeywell Inc.

TPS is the evolution of TDC 3000 X.

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

Honeywell  
Industrial Automation and Control  
Automation College  
2820 West Kelton Lane  
Phoenix, AZ 85023  
1-800-852-3211

# Table of Contents

---

<b>LAB EXERCISE .....</b>	<b>1</b>
Overview.....	1
\$AL_ENTY.....	2
Composite Alarm Status .....	6
<b>STUDENT PROFICIENCY EVALUATION.....</b>	<b>19</b>
Criterion Test.....	19
Self-Evaluation.....	21
<b>APPENDIX A 25</b>	
Procedure to Read a Text File Into a PE Entry Port.....	27
<b>APPENDIX B 29</b>	
Actors Used In The Lab Exercise .....	31

## Figures and Tables

---

Figure 1	\$AL_ENTY .....	3
Figure 2	Condition if Selected Point is in Alarm .....	5
Figure 3	Area Database—Unit Assignment Configuration Display .....	7
Figure 4	Unit Alarm Detail Schematic .....	13



# Lab Exercise

## Overview

### Objective

Given a description of requirements for a custom alarm schematic, use the Picture Editor to apply the following enhancements to meet the display requirements:

- \$AL\_ENTY DDB variable to show data for an operator-selected point,
- alarm status collector to view a composite alarm status for an area, unit, primmod group, annunciator group, or individual point,
- alarm count collector to view the total number of alarms for a given alarm status category, such as unacknowledged emergency alarms, and
- Universal Station PSDP parameters to view alarm status information for a unit.

### ATTENTION

ATTENTION—The following steps apply only to those taking this training at a Honeywell training center:

1. The schematic named COOKY### is in directory ALM for use in this exercise.

Copy the display to your student directory.

```
CP NET>ALM>COOKY###.DS NET>S###>=
```

2. To save time, you may choose to not do all of the keystroking included in this exercise. Instead, you can use *prebuilt text files* that contain the required syntax. The .text files are in directory ALM and have the .XX suffix.

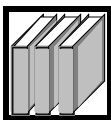
Copy the text files to your student directory.

```
CP NET>ALM>*.XX NET>S###>=
```

Refer to Appendix A as needed during the lab exercise for the procedure on how to read a text file into a Picture Editor port.

### Actors used in this exercise

Appendix B contains a description of the Picture Editor actors used in this lab exercise for targets and initial action.



REFERENCE—If you want more details about the actors, refer to the *Actors Manual*.

# \$AL\_ENTY

## Introduction

In this part of the exercise, you will add the new DDB variable \$AL\_ENTY to a schematic to enable the schematic to do the following:

- display detailed information for a selected point, and
- change the display appearance if the selected point is in alarm.

## Requirements

For use in this exercise, choose two alarmable analog points on the system you are using that have the same primary module point configured in their PRIMMOD parameter.

For future reference, record the point names in the space provided below.

	<u>Point Name</u>	<u>Primary Module Point</u>
1.	_____	_____
2.	_____	_____

## Build schematic

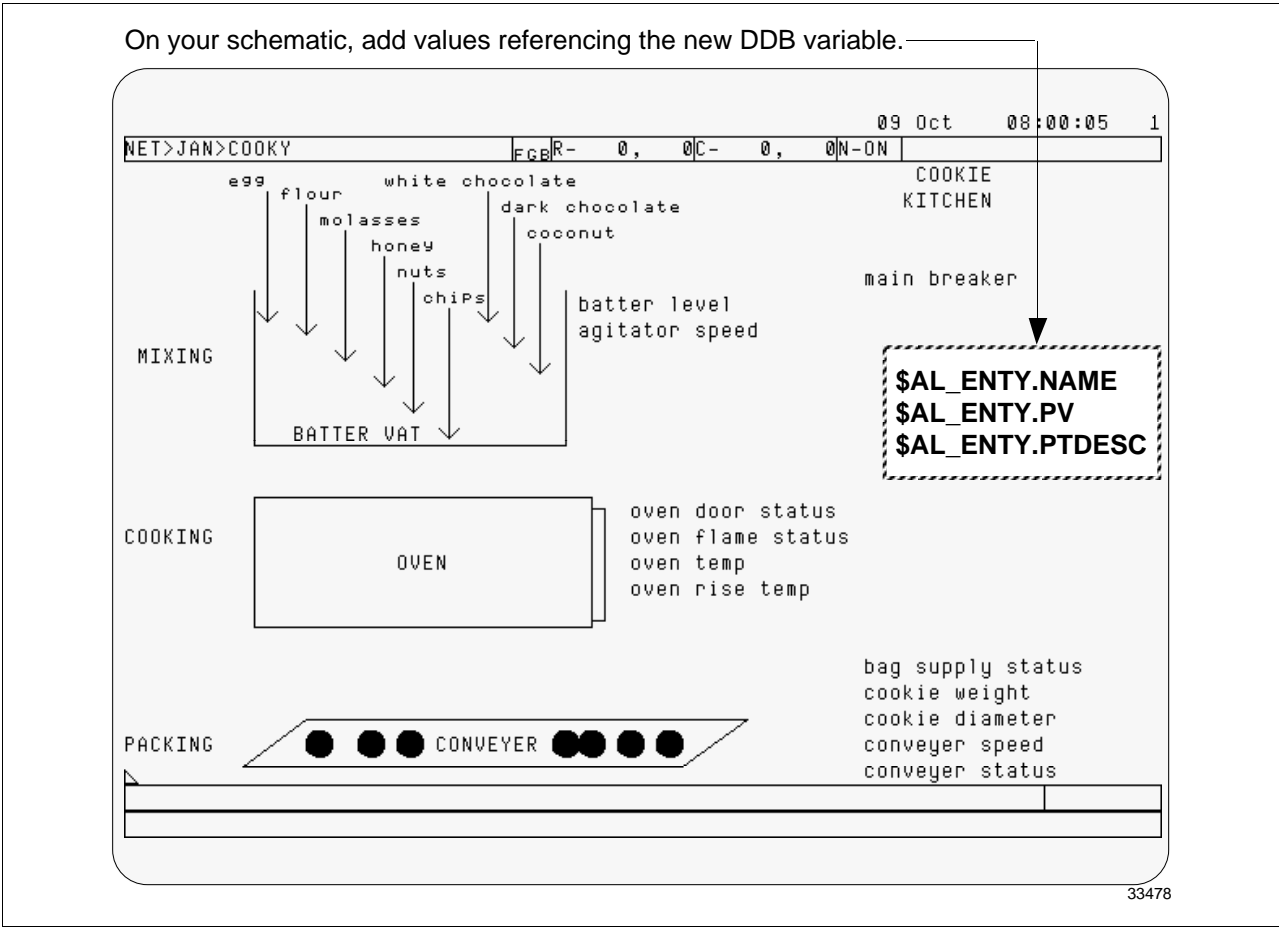
Perform these steps to add \$AL\_ENTY to a schematic.

Step	Action												
1	<p>Add these values to a schematic.</p> <p>If you are at a Honeywell training center, add the values to your schematic named COOKY#### as shown in Figure 1.</p> <p>ADD VAL</p> <p>\$AL_ENTY.NAME</p> <p>\$AL_ENTY.PV</p> <p>\$AL_ENTY.PTDESC</p> <table><tr><th><u>Variable</u></th><th><u>Variable Type</u></th><th><u>Format</u></th></tr><tr><td>NAME</td><td>STRING</td><td>TEXTL1:16</td></tr><tr><td>PV</td><td>REAL</td><td>R-ZZ9.9</td></tr><tr><td>PTDESC</td><td>STRING</td><td>TEXTL1:24</td></tr></table>	<u>Variable</u>	<u>Variable Type</u>	<u>Format</u>	NAME	STRING	TEXTL1:16	PV	REAL	R-ZZ9.9	PTDESC	STRING	TEXTL1:24
<u>Variable</u>	<u>Variable Type</u>	<u>Format</u>											
NAME	STRING	TEXTL1:16											
PV	REAL	R-ZZ9.9											
PTDESC	STRING	TEXTL1:24											

*Continued on next page*

**\$AL\_ENTY,** Continued

Figure 1      \$AL\_ENTY



*Continued on next page*

## \$AL\_ENTY, Continued

### Build schematic, continued

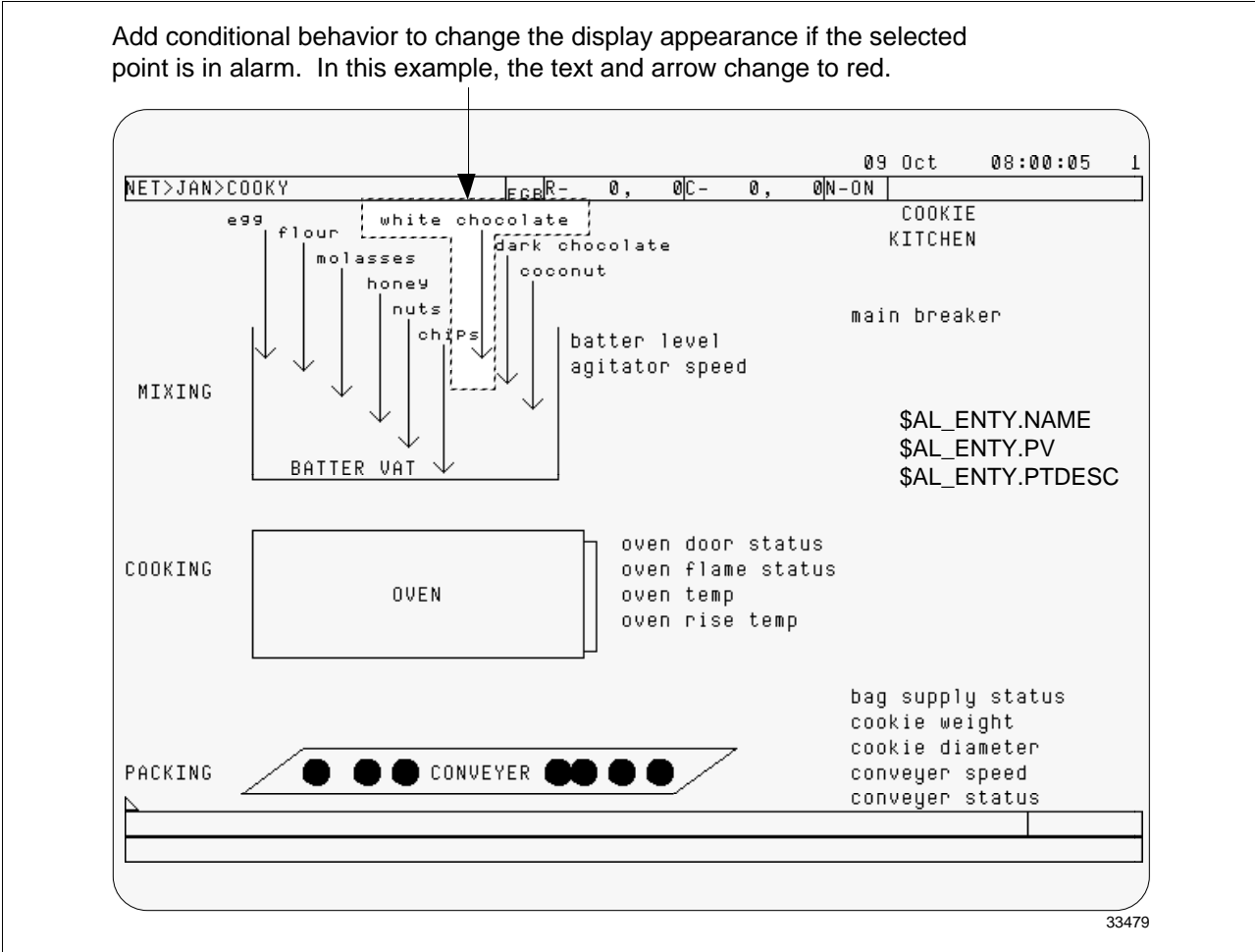
Step	Action				
2	Enter the Define Initial Action command.  DEF INIT				
3	<p>Configure the display's initial action to do the following:</p> <ul style="list-style-type: none"> <li>Convert the \$AL_ENTY entity name to a string, then store it to a DDB variable.</li> <li>Store the names of predefined points to String DDB variables.</li> </ul> <pre>S_STR(STRING01, IE_ENT(\$AL_ENTY));</pre> <pre>S_STR(STRING02, "NNNNNNNN");</pre> <pre>S_STR(STRING03, "NNNNNNNN");</pre> <p>where NNNNNNNN = names of two predefined alarmable points</p> <p>NOTE: It is necessary to use upper case letters for the point names.</p> <p>ATTENTION: If you are in a Honeywell training center, the above syntax is in the text file named 1INIT.XX. Refer to Appendix A for the procedure.</p>				
4	<p>As shown in Figure 2, configure conditions that cause a display object to change appearance when a point is selected and in alarm.</p> <p>STRING01 will contain the name held in \$AL_ENTY. STRING02 will contain the name of a predefined point STRING03 will contain the name of another predefined point.</p> <p>ATTENTION: If you are in a Honeywell training center, the following syntax is in the text file named 1COND.XX.</p> <pre>ADD COND</pre> <p>Bad Value:           RED NO BLINK NO REVERSE FULL</p> <p>Initial Behavior:    CYAN NO BLINK NO REV H</p> <pre>IF \$AL_ENTY.PTINAL AND (STRING01=STRING02) THEN SET RED FULL;</pre> <pre>IF \$AL_ENTY.PTINAL AND (STRING01=STRING03) THEN SET MAG FULL</pre> <table> <tr> <td><u>Variable</u></td><td><u>Variable Type</u></td></tr> <tr> <td>PTINAL</td><td>LOGICAL</td></tr> </table>	<u>Variable</u>	<u>Variable Type</u>	PTINAL	LOGICAL
<u>Variable</u>	<u>Variable Type</u>				
PTINAL	LOGICAL				
5	Compile the display.				
6	Copy the object file to a directory in the Area pathname catalog. If you are at a Honeywell Training Center, copy it to the PICT directory.				

*Continued on next page*



Build schematic, continued

Figure 2 Condition if Selected Point is in Alarm



Test your schematic

Perform these steps to select a point then view your display.

Step	Action
1	Select one of your two predefined points from one of these displays: <ul style="list-style-type: none"><li>• Area Alarm Summary or Unit Alarm Summary</li><li>• Area Annunciator (points appear at top of display)</li><li>• Organizational Summary display</li></ul>
2	Call up your schematic. RESULT: The point information for the selected point is displayed.
3	If it is not currently in alarm, cause an alarm on the selected point. RESULT: The schematic reflects the alarm condition.

# Composite Alarm Status

## Introduction

---

Next, you will build two schematics to show composite unit alarm status:

- a unit alarm status overview schematic, and
- a detailed unit alarm status schematic.

The unit alarm overview schematic should contain a box for each unit. Different colors will indicate the alarms at each priority level. The configured Unit ID will be displayed in the box, such as “FE.” Each box will be a target to invoke a detailed unit alarm schematic.

The unit detail alarm schematic should show all the parameters for a selected unit. Different colors will indicate the alarms at each priority level.

You will use Universal Station PSDP parameters to display unit alarm status information.

---

*Continued on next page*

# Composite Alarm Status, Continued

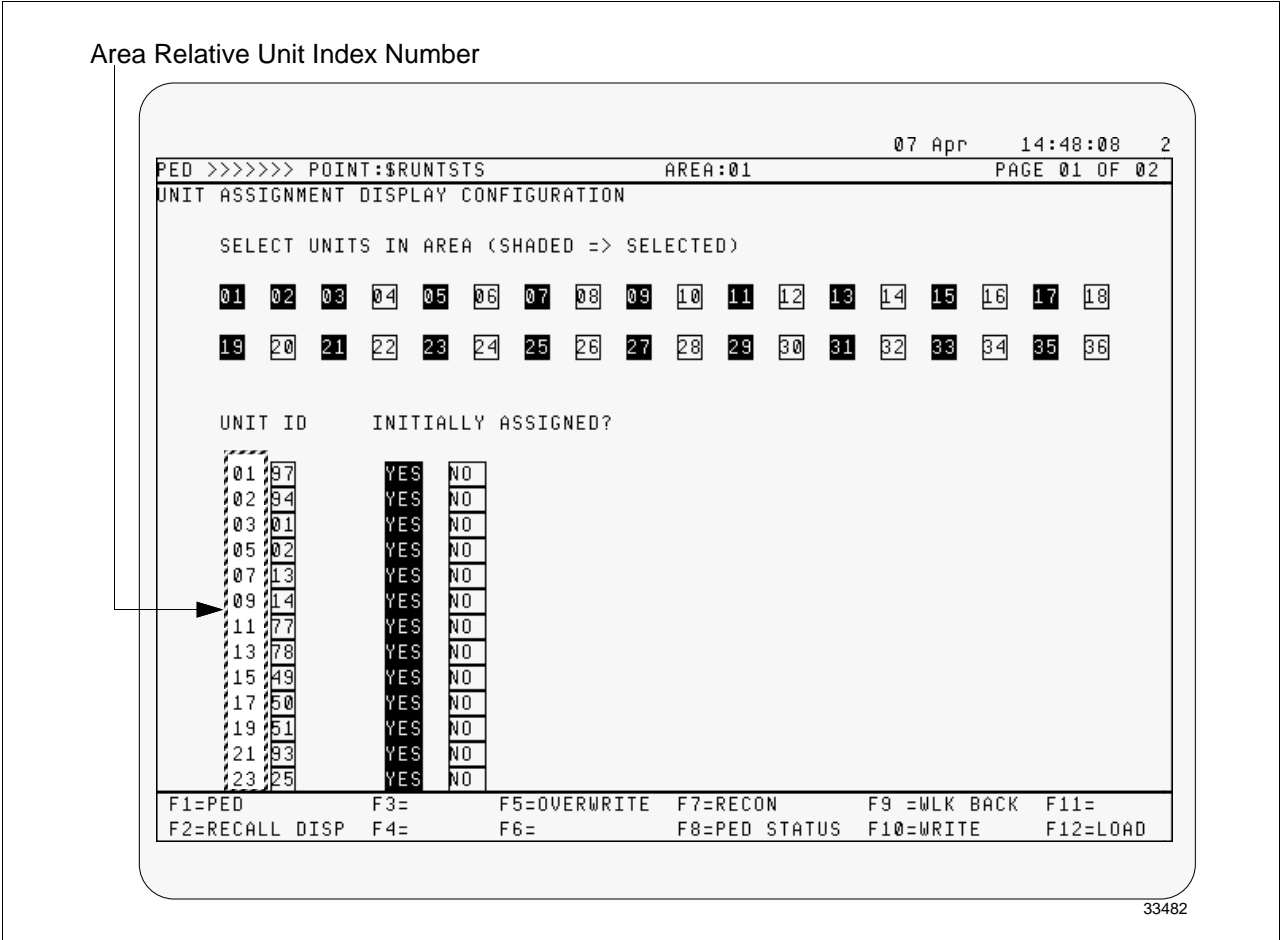
**Area-relative unit number**

The new R500 PSDP parameters require the Area-relative unit number. For this exercise, you will need to know the Area-relative unit numbers of at least two units on the system you are using. To get this information, reconstitute the Unit Assignment display for the Area (see Figure 3).

Record the information in the space provided below:

<u>Area-Relative</u> <u>Unit Index Number</u>	<u>Unit ID</u>
_____	_____
_____	_____

Figure 3      Area Database—Unit Assignment Configuration Display



## Composite Alarm Status, Continued

### Build unit alarm subpicture

Perform these steps to build subpicture to use as a unit alarm status box similar to the one shown for unit FE in Figure 3.

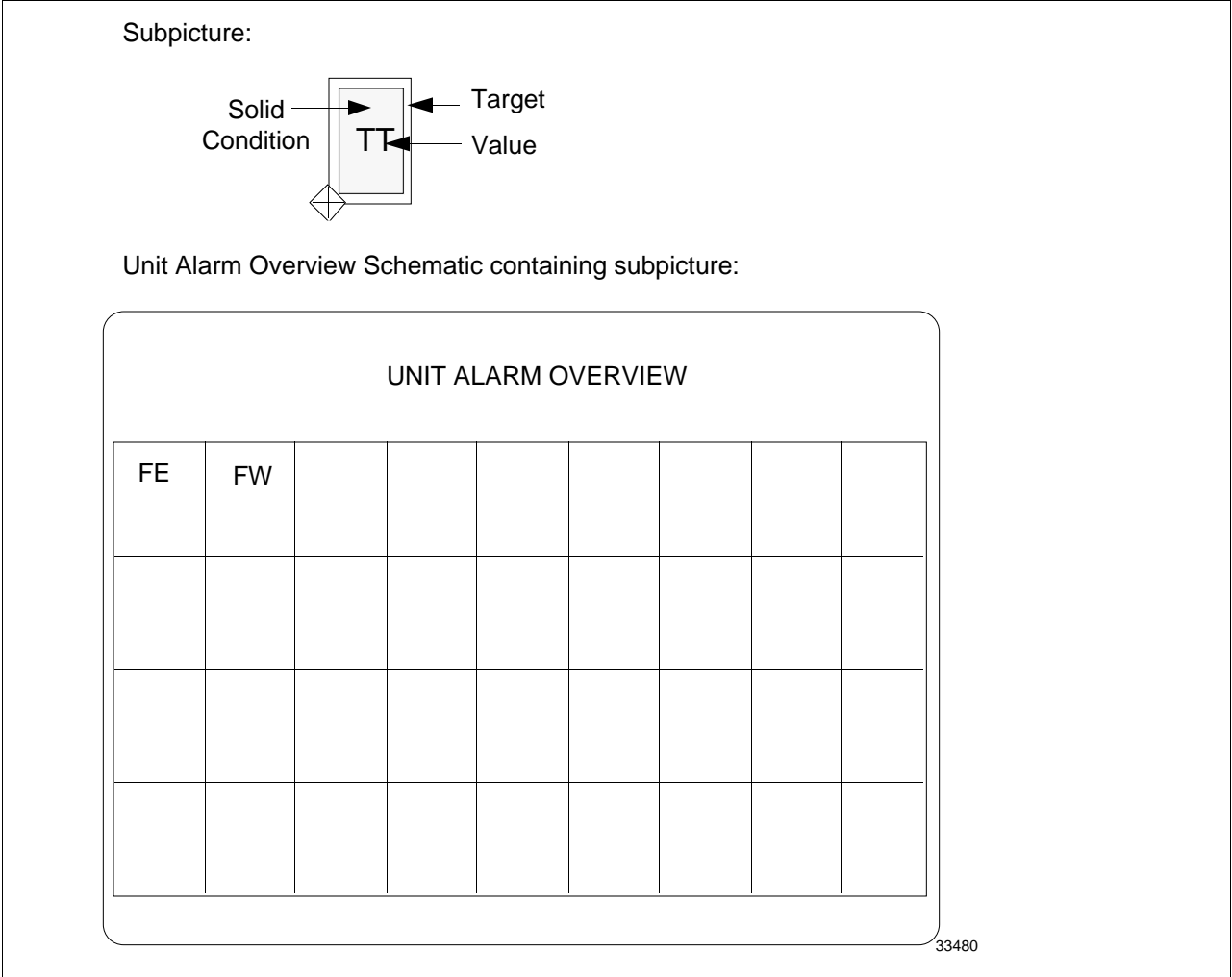
Step	Action				
1	<p>Define the unit alarm subpicture as follows:</p> <pre>ADD TARGET    S_INT( INT01G, &amp;UT_INX );    IF( CMP_S( GS_ENM( C_VAR( G_ENT( ENT01G ), 0, .UNITSTAT,     &amp;UT_INX ) ), NEQ, "NOTCONFIG" ) );    SCHEM( "UDET###" ); ENDIF</pre> <p>where: ###=your student number</p> <p>ATTENTION: If you are in a Honeywell training center, the above syntax is in the text file named <b>1TAR.xx</b>. See Appendix A for the procedure.</p> <table> <tr> <td><u>Variable</u></td><td><u>Variable Type</u></td></tr> <tr> <td>&amp;UT_INX</td><td>INTEGER</td></tr> </table> <p>NOTES:</p> <p><b>C_VAR</b>—Combine Variable actor  C_VAR (entity, index, parameter, index)</p> <p>entity = ENT01G contains the entity. The initial action of the schematic in which the subpicture is used will store the PSDP parameter entity name (\$PRSTSnn) to the DDB variable ENT01G.</p> <p>index = a value of zero references just the entity</p> <p>parameter = .UNITSTAT is the PSDP parameter that returns the composite unit alarm status.</p> <p>index = &amp;UT_INX is a parameterized value representing the Area-relative unit index number. You will enter this number when you add the subpicture to a schematic. The number you enter will be stored to the DDB variable INT01G.</p> <p>\$PRSTSnn.UNITSTAT(ii): Returns an enumerated value indicating the current composite alarm status of the specified unit.</p>	<u>Variable</u>	<u>Variable Type</u>	&UT_INX	INTEGER
<u>Variable</u>	<u>Variable Type</u>				
&UT_INX	INTEGER				
3	<p>ADD VALUE (see Figure 3 for location of value in subpicture)</p> <pre>ENT01G.UNIT_ID( &amp;UT_INX )</pre> <p>NOTE: \$PRSTSnn.UNIT_ID(ii): Returns 2-character string Unit ID.</p>				

*Continued on next page*

# Composite Alarm Status, Continued

## Build unit alarm subpicture, continued

Figure 3 Unit Alarm Overview Subpicture and Schematic



Continued on next page

## Composite Alarm Status, Continued

### Build unit alarm subpicture, continued

Step	Action				
4	<p>ADD SOLID (see Figure 3 for location of solid in subpicture)</p> <p>Build the solid box inside the perimeter of the target one pixel smaller than the target</p>				
5	<p>Add the following condition to the solid:</p> <pre>ADD COND IF ENT01G.UNITSTAT(&amp;UT_INX)=NOALARM THEN S BLACK NO BLINK ELSE IF ENT01G.UNITSTAT(&amp;UT_INX)=ACKEM THEN S RED NO BLINK ELSE IF ENT01G.UNITSTAT(&amp;UT_INX)=ACKHI THEN S YELLOW NO BLINK ELSE IF ENT01G.UNITSTAT(&amp;UT_INX)=ACKLO THEN S CYAN NO BLINK ELSE IF ENT01G.UNITSTAT(&amp;UT_INX)=UNACKEM THEN S RED BLINK ELSE IF ENT01G.UNITSTAT(&amp;UT_INX)=UNACKHI THEN S YELLOW BLINK ELSE IF ENT01G.UNITSTAT(&amp;UT_INX)=UNACKLO THEN S CYAN BLINK ELSE S WHITE NO BLINK</pre> <p>ATTENTION: If you are in a Honeywell training center, the above syntax is in the text file named 2COND.XX.</p> <table><tr><td><u>Variable</u></td><td><u>Variable Type</u></td></tr><tr><td>UNITSTAT</td><td>ENUM:\$ALRMSTS</td></tr></table>	<u>Variable</u>	<u>Variable Type</u>	UNITSTAT	ENUM:\$ALRMSTS
<u>Variable</u>	<u>Variable Type</u>				
UNITSTAT	ENUM:\$ALRMSTS				
6	<p>Set the origin to the lower left-hand corner of the box.</p>				
7	<p>Save (Write) the subpicture to your student directory as file BOX###, where ### is your student number.</p> <p>Enter this prompt for &amp;UT_INX:</p> <p>“Enter Area-relative unit index”</p> <p>NOTE: At the time you add your subpicture to a schematic, the prompt will appear reminding you to enter an index number. You recorded your index numbers earlier in this lab exercise.</p>				

*Continued on next page*

## Composite Alarm Status, Continued

### Build unit alarm overview schematic

Perform the following steps to build a unit alarm overview schematic using your new subpicture.

Step	Action										
1	<p>Add your subpicture box to a new schematic named UOVER###.</p> <p>where ### is your student number.</p> <p>In response to the prompt, enter the <i>first</i> Area-relative unit number you recorded earlier.</p> <table> <tr> <td><u>Variable</u></td><td><u>Variable Type</u></td></tr> <tr> <td>ENT01G.UNIT_ID(ii)</td><td>STRING</td></tr> <tr> <td>ENT01G.UNITSTAT(ii)</td><td>ENUM:\$ALRMSTS</td></tr> </table>	<u>Variable</u>	<u>Variable Type</u>	ENT01G.UNIT_ID(ii)	STRING	ENT01G.UNITSTAT(ii)	ENUM:\$ALRMSTS				
<u>Variable</u>	<u>Variable Type</u>										
ENT01G.UNIT_ID(ii)	STRING										
ENT01G.UNITSTAT(ii)	ENUM:\$ALRMSTS										
2	Add your subpicture again. In response to the prompt, enter the <i>second</i> Area-relative unit number you recorded earlier.										
3	<p>Enter the Define Initial Action command: DEF INIT</p> <p>Define the initial action of the schematic as follows:</p> <pre>S_STR(STRING02,CNV_I(G_INT(\$MY_PNA),"I99")); S_STR(STRING01,CONCAT("\$PRSTS",G_STR(STRING02))); S_ENT(ENT01G,EI_ENT(G_STR(STRING01)));</pre> <p><b>ATTENTION:</b> If you are in a Honeywell training center, the above syntax is in the text file named <b>2INIT.XX</b>.</p> <p><b>NOTES:</b></p> <table> <tr> <td>\$MY_PNA</td><td>System DDB variable of type integer that represents the US node number on which schematic is running.</td></tr> <tr> <td>CNV_I</td><td>Converts a DDB integer to a string.</td></tr> <tr> <td>CONCAT</td><td>Concatenates (links) two strings</td></tr> <tr> <td>EI_ENT</td><td>Converts a string to an internal entity ID. In this case, the entity ID \$PRSTSnn is generated, where nn is the US node number.</td></tr> <tr> <td>S_ENT</td><td>Stores \$PRSTSnn into ENT01G</td></tr> </table>	\$MY_PNA	System DDB variable of type integer that represents the US node number on which schematic is running.	CNV_I	Converts a DDB integer to a string.	CONCAT	Concatenates (links) two strings	EI_ENT	Converts a string to an internal entity ID. In this case, the entity ID \$PRSTSnn is generated, where nn is the US node number.	S_ENT	Stores \$PRSTSnn into ENT01G
\$MY_PNA	System DDB variable of type integer that represents the US node number on which schematic is running.										
CNV_I	Converts a DDB integer to a string.										
CONCAT	Concatenates (links) two strings										
EI_ENT	Converts a string to an internal entity ID. In this case, the entity ID \$PRSTSnn is generated, where nn is the US node number.										
S_ENT	Stores \$PRSTSnn into ENT01G										
4	Add a title to your display: UNIT OVERVIEW										
5	Compile the schematic as file UOVER###.										
6	Copy the object file to a directory specified in the Area pathname catalog. If you are at a Honeywell training center, copy it to directory PICT.										

*Continued on next page*

## Composite Alarm Status, Continued

### Test unit alarm overview schematic

Perform these steps to test your schematic.

Step	Action
1	Call up your schematic.
2	Cause an alarm on a point in one of the units. RESULT: Schematic reflects the alarm with color and blink behavior.

### Build unit alarm detail schematic

Now perform the following steps to build a Unit Detail schematic, similar to Figure 4, that displays alarm status information for one selected unit.

Step	Action						
1	Start a new schematic named UDET###. where ### is your student number						
2	Add the text as shown in Figure 4. Make EMERGENCY, HIGH, and LOW <i>separate</i> text objects.						
3	Add the following condition to the text EMERGENCY:  ADD COND  IF ENT01G.UNACKEM(INT01G)>0 THEN S RED BLINK ELSE IF ENT01G.ACKEM(INT01G)>0 THEN S RED NO BLINK ELSE S BLACK NO BLINK  ATTENTION: If you are in a Honeywell training center, the above syntax is in the text file named 3COND.XX.  <table><tr><td><u>Variable</u></td><td><u>Variable Type</u></td></tr><tr><td>ENT01G.UNACKEM(INT01G)</td><td>INTEGER</td></tr><tr><td>ENT01G.ACKEM(INT01G)</td><td>INTEGER</td></tr></table>	<u>Variable</u>	<u>Variable Type</u>	ENT01G.UNACKEM(INT01G)	INTEGER	ENT01G.ACKEM(INT01G)	INTEGER
<u>Variable</u>	<u>Variable Type</u>						
ENT01G.UNACKEM(INT01G)	INTEGER						
ENT01G.ACKEM(INT01G)	INTEGER						

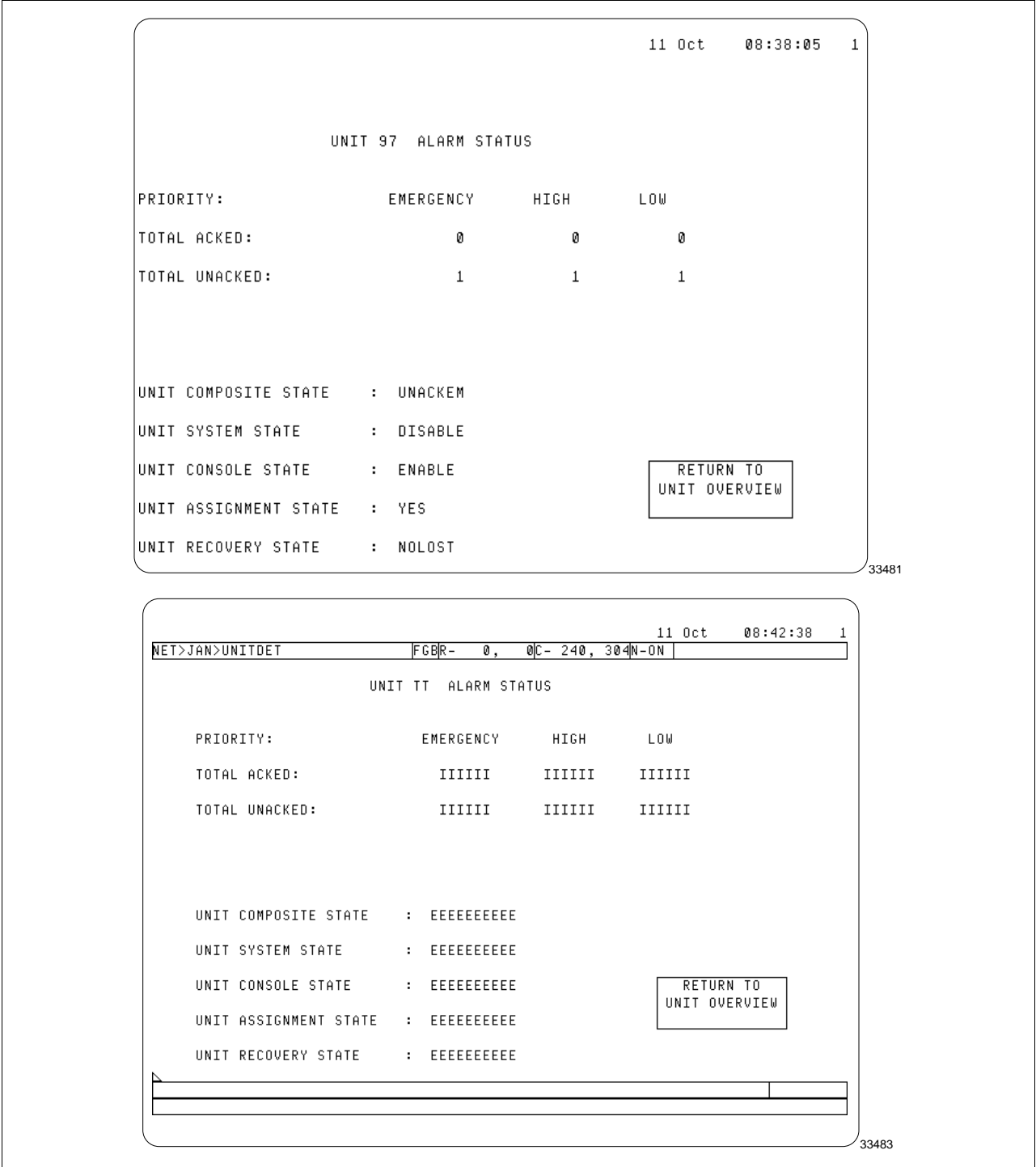
*Continued on next page*



# Composite Alarm Status, Continued

## Build unit alarm detail schematic, continued

Figure 4 Unit Alarm Detail Schematic



Continued on next page

## Composite Alarm Status, Continued

### Build unit alarm detail schematic, continued

Step	Action												
4	<p>Add a similar condition to the text HIGH:</p> <pre>ADD COND IF ENT01G.UNACKHI(INT01G)&gt;0   THEN S YELLOW BLINK   ELSE IF ENT01G.ACKHI(INT01G)&gt;0     THEN S YELLOW NO BLINK     ELSE S BLACK NO BLINK</pre> <p>ATTENTION: If you are in a Honeywell training center, the above syntax is in the text file named <b>4COND.XX</b>.</p>												
5	<p>Add a similar condition to the text LOW:</p> <pre>ADD COND IF ENT01G.UNACKLO(INT01G)&gt;0   THEN S CYAN BLINK   ELSE IF ENT01G.ACKLO(INT01G)&gt;0     THEN S CYAN NO BLINK     ELSE S BLACK NO BLINK</pre> <p>ATTENTION: If you are in a Honeywell training center, the above syntax is in the text file named <b>5COND.XX</b>.</p>												
6	<p>Add counters of alarms at each priority level:</p> <pre>ADD VAL      ENT01G.ACKEM(INT01G) ADD VAL      ENT01G.ACKHI(INT01G) ADD VAL      ENT01G.ACKLO(INT01G) ADD VAL      ENT01G.UNACKEM(INT01G) ADD VAL      ENT01G.UNACKHI(INT01G) ADD VAL      ENT01G.UNACKLO(INT01G)</pre>												
7	<p>Add these unit states:</p> <ul style="list-style-type: none"> <li>• composite alarm status of the unit,</li> <li>• alarms annunciated system-wide</li> <li>• alarms annunciated in local console</li> <li>• unit controlled by this Area database</li> <li>• unit alarm recovery state</li> </ul> <pre>ADD VAL      ENT01G.UNITSTAT(INT01G) ADD VAL      ENT01G.UNITSYS(INT01G) ADD VAL      ENT01G.UNITCONS(INT01G) ADD VAL      ENT01G.UNITASSG(INT01G) ADD VAL      ENT01G.UNITRECV(INT01G)</pre> <table> <thead> <tr> <th><u>Variable</u></th><th><u>Variable Type</u></th></tr> </thead> <tbody> <tr> <td>UNITSTAT</td><td>ENUM:\$ALRMSTS</td></tr> <tr> <td>UNITSYS</td><td>ENUM:ALENBST</td></tr> <tr> <td>UNITCONS</td><td>ENUM:ALENBST</td></tr> <tr> <td>UNITASSG</td><td>ENUM:UNITASGN</td></tr> <tr> <td>UNITRECV</td><td>ENUM:\$UNITREC</td></tr> </tbody> </table>	<u>Variable</u>	<u>Variable Type</u>	UNITSTAT	ENUM:\$ALRMSTS	UNITSYS	ENUM:ALENBST	UNITCONS	ENUM:ALENBST	UNITASSG	ENUM:UNITASGN	UNITRECV	ENUM:\$UNITREC
<u>Variable</u>	<u>Variable Type</u>												
UNITSTAT	ENUM:\$ALRMSTS												
UNITSYS	ENUM:ALENBST												
UNITCONS	ENUM:ALENBST												
UNITASSG	ENUM:UNITASGN												
UNITRECV	ENUM:\$UNITREC												

*Continued on next page*

## Composite Alarm Status, Continued

### Build unit alarm detail schematic, continued

---

Step	Action
8	Add a title to your display: <code>UNIT ALARM STATUS</code> Add a value for the unit ID between the words <code>UNIT</code> and <code>ALARM</code> : <code>ADD VAL ENT01G.UNIT_ID( INT01G)</code>
9	Add a target to allow you to return to the unit overview schematic. <code>ADD TAR SCHEM( "UOVER###" )</code>
10	Compile the schematic as file <code>UDET###</code> . where <code>###</code> is your student number.
11	Copy the object file to the appropriate directory.

---

### Test unit alarm detail schematic

Perform these steps to test your schematic.

Step	Action
1	Call up your schematic <code>UOVER###</code> . Select a unit target box.  RESULT: The schematic <code>UDET###</code> appears. The display reflects the current status of the selected unit.
2	Cause an alarm on a point in the unit.  RESULT: The appropriate alarm counter reflects the additional alarm.

---

*Continued on next page*

## Composite Alarm Status, Continued

### Display point and PRIMMOD status

Perform the following steps to display the composite alarm status of a point and a PRIMMOD using new Picture Editor collectors.

Step	Action
1	<p>Add these values to your schematic COOKY### to display the composite alarm statuses of the two points you selected earlier:</p> <pre>ADD VAL      \$PNTSTS (nnnnnnnnn ) ADD VAL      \$PNTSTS (nnnnnnnnn )</pre> <p>where nnnnnnnnn = name of point</p>
2	Add text to describe the values.
3	<p>Add these values to display the number of unacknowledged emergency alarms for two points:</p> <pre>ADD VAL      \$PNTCNT (nnnnnnnnn , \$ALRMSTS : UNACKEM ) ADD VAL      \$PNTCNT (nnnnnnnnn , \$ALRMSTS : UNACKEM )</pre> <p>where nnnnnnnnn = name of point</p>
4	Add text to describe the values.
5	<p>Add this value to display the composite alarm status for the PRIMMOD of the two points:</p> <pre>ADD VAL      \$PRIMSTS ( "nnnnnnnnn" )</pre> <p>where nnnnnnnnn = name of Primary Module point</p>
6	Add text to describe the value.
7	Compile the display, then call it up to view the values.
8	<p>Cause the points to go into alarm.</p> <p>Verify that the following appears on your schematic.</p> <ul style="list-style-type: none"> <li>The point status values indicate the highest alarm condition existing for each point: UNACKEM, UNACKHI, UNACKLO, ACKEM, ACKHI, ACKLO</li> <li>The counter values indicate the number of unack'd emergency alarms for each point.</li> <li>The PRIMMOD status value indicates the highest alarm condition of the two points. UNACKEM, UNACKHI, UNACKLO, ACKEM, ACKHI, ACKLO, or NOALARM</li> </ul>

*Continued on next page*

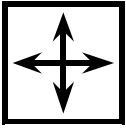
## Composite Alarm Status, Continued

### Optional—Enter Unit ID to view unit status

In this optional exercise, you will create a target that allows the operator to display detailed unit alarm status information by entering the desired Unit ID into a text input port.

Step	Action				
1	<p>Add a target box to your schematic UOVER###:</p> <p><b>ADD TAR</b></p> <pre>S_STR(STRING03,R_STR(31,1,2,"Enter the configured Unit ID.",t,1));</pre> <pre>S_INT(INT01,CNV_UID(G_STR(STRING03)))</pre> <p><b>ATTENTION:</b> If you are in a Honeywell training center, the above syntax is in the text file named <b>2TAR.XX</b>.</p>				
2	<p>Add these values below the target:</p> <p><b>ADD VAL</b>            STRING03</p> <p><b>ADD VAL</b>            ENT01G.UNITASSG( INT01 )</p> <div><p>Target → <div>UNIT INFO</div></p><p>Values → <table><tr><td><u>UNIT</u></td><td><u>ASSIGNED?</u></td></tr><tr><td>TT</td><td>EEEEEEEEEE</td></tr></table></p></div>	<u>UNIT</u>	<u>ASSIGNED?</u>	TT	EEEEEEEEEE
<u>UNIT</u>	<u>ASSIGNED?</u>				
TT	EEEEEEEEEE				
3	Recompile the schematic and copy it to the appropriate directory.				
4	<p>Call up your schematic, then select the target. Enter a valid Unit ID.</p> <p>RESULT: The current unit assignment state appears.</p>				

## Directions



---

**DIRECTIONS**—This is the end of the lab exercise. Discuss questions concerning the study material or lab activities with a colleague or your course manager.

If you are satisfied that you have achieved the objective of the course module, continue with the Student Proficiency Evaluation.

---

# Student Proficiency Evaluation

## Criterion Test

---

### Instructions

Completion of the lab exercise satisfies the test requirements for this course module.

Be prepared to show your completed displays to your course manager:

- Demonstrate that your COOKY### schematic containing \$AL\_ENTY shows detailed information for a selected point, and that it changes appearance if the selected point is in alarm.
  - Demonstrate that your unit alarm status *overview* schematic indicates different colors depending on the highest priority level of alarm in the units.
  - Demonstrate that your schematic COOKY### contains new alarm status collectors for a specified PRIMMOD group and specified points.
  - Demonstrate that your *detailed* unit alarm status schematic contains all of the new US PSDP parameters for a selected unit.
-





# Self-Evaluation

## Evaluation

Completion of the lab exercise satisfies the test requirements for this course module.

Be prepared to show your completed configuration to your course manager:

- Demonstrate that your COOKY### schematic containing \$AL\_ENTY displays detailed information for a selected point, and that it changes appearance if the selected point is in alarm.

*Cause an alarm on a point or choose a point that already has an alarm.*

*Select the point from the alarm summary display or from any of the organizational summary displays, then call up your schematic. The \$AL\_ENTY values you added to the schematic (point name, point description, and PV) should reflect those of the selected point.*

*The condition you added to the schematic (IF \$AL\_ENTY.PTINAL then...) will cause the schematic to change color to indicate the alarm.*

- Demonstrate that your unit alarm status *overview* schematic indicates different colors depending on the highest priority level of alarm in the units.

*The boxes on the schematic should change colors.*

*The schematic should indicate the composite alarm status for a unit by the following behavior:*

*Acknowledged emergency alarm —RED NO BLINK*

*Acknowledged high alarm—YELLOW NO BLINK*

*Acknowledged low alarm—CYAN NO BLINK*

*Unacknowledged emergency alarm —RED BLINK*

*Unacknowledged high alarm—YELLOW BLINK*

*Unacknowledged low alarm—CYAN BLINK*

*Continued on next page*

- Demonstrate that your schematic COOKY### contains new alarm status collectors for a specified PRIMMOD group and specified points.

*Your schematic should contain these collectors, indicating the count of point alarms and the alarm status of a point and the PRIMMOD grouping:*

*Point Alarm Status:*

*\$PNTSTS ("nnnnnnnn")*

*Point Alarm Count:*

*\$PNTCNT (nnnnnnnn, \$ALRMSTS:UNACKEM)*

*PRIMMOD Alarm Status:*

*\$PRIMSTS ("nnnnnnnn")*

- Demonstrate that your *detailed* unit alarm status schematic contains all of the new US PSDP parameters for a selected unit.

*Your schematic should contain parameters to show these unit statuses:*

*\$PRSTSnn.UNITSTAT(unit number)*

*\$PRSTSnn.UNITSYS(unit number)*

*\$PRSTSnn.UNITCONS(unit number)*

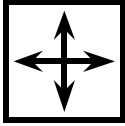
*\$PRSTSnn.UNITASSG(unit number)*

*\$PRSTSnn.UNITRECV(unit number)*

*\$PRSTSnn.UNITSYS(unit number)*

---

## Directions



---

**DIRECTIONS**—This is the end of this course module.

Use your course map to

- Get your course manager to sign off this course module.
- Choose your next eligible course module.

If you have a question

- Ask your course manager.
-

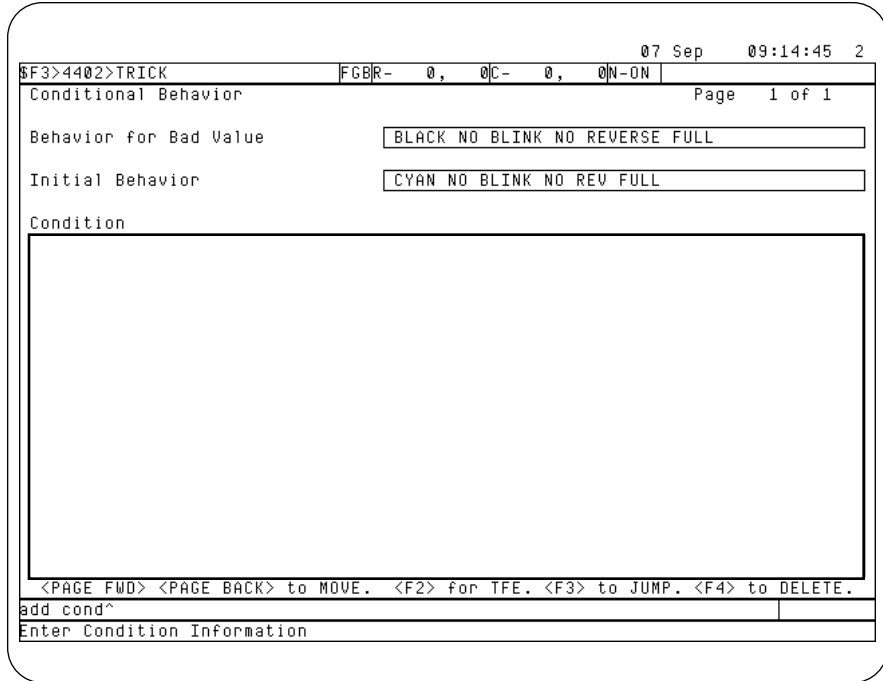



## ***Appendix A***

# ***How to Import a Text File into a PE Entry Port***



## Procedure to Read a Text File Into a PE Entry Port

Step	Action
1	<p>Display the entry port of the object you want to create.</p> <p>Notice the function codes at the bottom of the screen.</p>  <p style="text-align: right;">10526</p>
2	<p>Press [CTL] [F2].</p> <p>RESULT: Text Editor is accessed.</p>
3	<p>From the Text File Editor, press [CTL] [F7] to access the FILE commands.</p> <p>RESULT: The file commands appear at the bottom of the screen.</p> 
4	<p>Press [CTL] [F1] to DEFINE the file</p> <p>RESULT: Message appears ENTER TEMPORARY FILE PATHNAME</p>
5	<p>Type in the pathname of the text file you want to read into the PE port, then press [ENTER].</p> <p>RESULT: Entry is accepted.</p>
6	<p>Press [CTL] [F2] to GET the file.</p> <p>RESULT: File is read into the Text File Editor. Message appears LINES READ = n</p>
7	<p>Press [CTL] [F1] to QUIT, then [CTL] [F2] to EXIT the Text Editor and return to the Picture Editor.</p> <p>RESULT: Picture Editor entry port appears with text in the port.</p>





## ***Appendix B***

# ***Actors Used In The Lab Exercise***



## Actors Used In The Lab Exercise

Actor	Description
C_VAR	Combines an indexed entity and an indexed parameter to form a Variable ID, such as Point.Parameter
CMP_S	Compare Strings. Actor uses comparison operator, such as NEQ (not equal to).
CNV_I	Convert Integer to String
CNV_UID	Convert Unit ID to Unit Index Number
CONCAT	Concatenate Strings
EI_ENT	Convert Internal Entity ID to String
ENDIF	End of IF statement
G_INT	Get Integer Data From Local Display Database
G_STR	Get String Data From Local Display Database
GS_ENM	Get Enumeration Data From System Database
IE_ENT	Convert Internal DDB Entity ID to External Entity ID
IF	Used to create conditional statement
R_STR	Read String Data Through Text Input Port
S_ENT	Store Entity ID to Local Display Database
S_INT	Store Integer Data to Local Display Database
S_STR	Store String Data to Local Display Database
SCHEM	Call Up Schematic

**Last Page**