

***Advanced Graphics
Building
Discussion Material***

Course 4402

L53198

AG

Copyright, Notices, and Trademarks

© Copyright 1996, 1998 by Honeywell Inc.

Revision 04— July 21, 1998

Honeywell IAC courseware is subject to change without notice.

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 are trademarks of Honeywell Inc.

TPS is the evolution of TDC 3000^X.

Other brand or product names are trademarks of their respective owners

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

Table of Contents

WELCOME TO AUTOMATION COLLEGE	1
COURSE OVERVIEW	2
Course Design	2
Task Oriented Material	3
Tasks	3
Skills Oriented Material	4
Skills.....	4
Bernoulli Cartridge—Volume AGB1—left drive	4
Bernoulli Cartridge—Volume AGB2—right drive	4
LAB SETUP (IF NECESSARY)	8
Copy the contents of your AGB1 cartridge	8
Copy the contents of your AGB2 cartridge	8
COURSE PROCEDURES	9
10 Area databases	9
Normal Operating Procedures.....	13
Area Database Considerations.....	14
Menu Of Our Schematic Examples	15
Exercises	16
MEMORY-RESIDENT DISPLAYS	17
Station Memory Available for .DO and .FO files	17
.DO Block Size	20
PE CONSIDERATIONS	21
ACKSTAT Collector	21
ACKSTAT Limitations—Before R420	22
VARIANTS	26
ACTORS	27
GENERIC SCHEMATICS.....	35
ENUMERATIONS.....	43
SUBPICTURE REVIEW	47
Parameter References Per Subpicture.....	47
User Visible (parameterized) Variables	50
DDB VARIABLES	53
RETURN TO SPECIFIC SCHEMATIC.....	61
PE COMPILER ERROR MESSAGES.....	63
PE RUN-TIME ERROR MESSAGES.....	65

Figures and Tables

Figure 1	Operator's Pathname Catalog—R5xx	10
Figure 2	Pathname Catalog for Area 1	11
Figure 3	Pathname Catalog for Area 6	12
Figure 4	MOOSE	15
Figure 5	Memory-Resident Displays—R5xx	18
Figure 6	Memory-Resident Displays—R4xx	19
Figure 7	Making Text Invisible to the Operator	25
Figure 8	Checking the State of a Digital Composite	26
Figure 9	Actor Syntax	27
Figure 10	IF Actor Syntax	28
Figure 11	NOT Actor	29
Figure 12	Internal Entity to String Conversion	30
Figure 13	Compare String Without IF Actor	31
Figure 14	Reading, Storing and Converting Variable	32
Figure 15	Converting System Variable Type to a String	33
Figure 16	Preventing Operator Contention	34
Figure 17	Generic Picture Examples	35
Figure 18	Instructions for Operating Generic Schematics	36
Figure 19	Target Action from Subpicture GENERIC1	37
Figure 20	Target Action from Subpicture GENERIC2	38
Figure 21	Conditions on the Subpicture GENERIC1	39
Figure 22	Conditions on the Subpicture GENERIC2	40
Figure 23	Values for the Subpicture	41
Figure 24	AM Point Information	42
Figure 25	Standard Enumerations	43
Figure 26	Self-Defined Enumerations	44
Figure 27	Custom Enumerations	45
Figure 28	Enumerations in Generic Subpictures	46
Figure 29	System DDB Variables	53
Figure 30	System DDB Variables Page 2	54
Figure 31	System DDB Variables Page 3	55
Figure 32	System DDB Variables Page 4	56
Figure 33	User Change Zone	57
Figure 34	Variant Using \$CZ_ENTY	58
Figure 35	Example of \$CZ_ENTY in an Actor	58
Figure 36	PE Print Example Showing Honeywell DDB	59
Figure 37	Different Ways to Return to a Specific Schematic	61
Figure 38	Schematic Return Using an Indexed Custom	62
Figure 39	Local Name Type Expected	63
Figure 40	Local Name Type Expected	64
Figure 41	EQ Object Not Used in Abstract	65

WELCOME TO AUTOMATION COLLEGE

- General Information
- General Rules
- Course Materials
- Tour of Facility

COURSE OVERVIEW

Course Design

- Use the US for Job/Training Aids to improve Skills
- Use FlexTraining Material to learn New Tasks
- Use Group Discussions for Questions and Problem Solving

COURSE OVERVIEW, continued

Task Oriented Material

- Tasks
 - Configure Area Pathname Catalog for Custom Displays
 - Update Custom Displays
 - Build Efficient Custom Displays
 - Build Generic Custom Displays
 - Debug Custom Displays
 - Use Picture Editor Slick Tricks
 - Document Custom Displays
 - Design Custom Displays

COURSE OVERVIEW, continued

Skills Oriented Material

- **Skills**
 - Use Custom Display Values
 - Use Custom Display Conditions
 - Use Target Actors
 - Use Custom Display Variants
 - Use Multiple Overlays

- **Bernoulli Cartridge—Volume AGB1—left drive**
 - 10 Area Databases
 - Numerous Directories and Files for Exercises
 - Directory WORK for your own work
 - Directory TMAP containing Topological Schematics

- **Bernoulli Cartridge—Volume AGB2—right drive**
 - Directories Containing Schematic Examples
 - ACT - Directory for Actor Examples
 - DEMO - Directory for Everything Else
 - FUN - Directory for Fun Schematic Examples
 - GENP - Directory for Generic Schematic

Examples

- SYNT - Directory for Syntax Examples
- SYS - Directory for System Schematic

Examples

- TRCK - Directory for Slick Trick Schematics

VOLUME: **AGB1**
 VOLUME DESC: ADV GRAPH BUILDING 1
 MAX FILES: 3084
 TOTAL SECTORS: 83609
 SECTORS IN USE: 54765
 CURRENT DIRECTORIES ASSIGNMENT:
&D01 AREA DATABASE FILES
&D02
&D03
&D04
&D05
&D06
&D07
&D08
&D09
&D10
AMOD SCHEM BAD_C_IF-GO TO ARITHMETIC VALUE-MOD OPERATION OVAL HAS GAP
ARRW NEED TO CORRECT ARROWS AND LINES IN S_BOOLEX
BUG SOME OLDER SCHEMATICS AND THEIR PROBLEMS
BUGS SEE README.XX FILE FOR ALL THE PROBLEMS WITH THE SCHEM FDH872
CDAT COMBINE DATE AND TIME PROBLEM
CERR COMPILE ERRORS
CON1 CONDITIONS WITH SYNTAX ERRORS
CONV TARGET PROBLEM WITH FLOAT ACTOR
CVAR COMBINE VARIABLES PROBLEM
DATA DATABASE FILES FOR SCHEMS (POINTS, CL, AND CUSTOM DATA SEGMENT)
DEBUG THESE FILES ARE USED WITH MODULE "DEBUGGING CUSTOM DISPLAYS"
DDB1 RUN-TIME ERROR "DISPLAY DATA BASE ERROR" FOR SCHEM DDBTEST
DDB2 SCHEMATIC CLDATA HAS A PROBLEM SETTING POINT ACTIVE/INACTIVE
ERRS BUG IN PE, ADD A VARIANT TO BRING IN SUB L TO SCHEM PLOT
GEN REPLACE SUB:HOW DO YOU FIND NAME OF OLD SUB ALREADY IN A SCHEM?
ILLV ILLEGAL VARIABLE ERROR WHEN TRY TO COMPILE - LOOK FOR GHOSTS
ILV1 ILLEGAL VARIABLE ERROR WHEN TRY TO COMPILE GRID
INVD WHEN SCHEMATIC REALX IS CALLED UP, GET ERROR "INVALID DATA TYPE"
KILN SCHEMATIC KILN HAS COMPILE EROR
KLSB KILN SUBPICTURES
OFFD COMPILE ERROR - DRAWING OFF SCREEN "MIXEDEXP"
OVLY OVLAY PROBLEM WITH SCHEMATIC GENERAL1
RUNT RUN-TIME ERRORS
SETC SCHEMATICS TO PRACTICE SET COLLECTION ON
SUB1 SUBPCTURE CIRCLE HAS A PROBLEM
SUBS SUBPICTURES AND THEIR PROBLEMS
SYNE PROBLEM TRYING TO MODIFY A VARIANT IN COND_IF TO ADD S_MIXED
TARG BAD TARGET ACTIONS
TEXT SCHEMATICS WITH TEXT EXERCISES AND OR PROBLEMS
TMAP SCHEMS FOR A TOPOLOGICAL MAP OF THE LCN, SEE FILE ZTMAP_Z1.XI
VAR1 VARIANT PROBLEM: SELECTING TARGET CAUSES TEXT ABOVE TO DISAPPEAR
VAR2 VARERR2 CONTAINS A VARIANT WHICH WIPES OUT PART OF SCHEMATIC
VAR3 VARIANT PROBLEM: SUBS WIPE OUT PART OF SCHEM - PROBLEM IN SUBS

VAR4 WANT TO ADD 2 LINES TO SUB S_STRNGX, BUT AFFECTS OTHER AREAS
WORK STUDENT DIRECTORY FOR LAB WORK
VOLUME: **AGB2**
VOLUME DESC: ADV GRAPH BUILDING 2
MAX FILES: 3084
TOTAL SECTORS: 83609
SECTORS IN USE: 71050
CURRENT DIRECTORIES ASSIGNMENT:
ACT ACTORS EXAMPLES SEE SCHEMATIC ACTMENU
DEMO SEE SCHEMATIC MOOSE AND OTHERS OF THE SAME ILK (ELK)
FUN THESE SCHEMATICS HAVE A BIT OF WHIMSY SEE SCHEMATIC FUNMENU
GENP GENERIC PICTURES EXAMPLES SEE SCHEMATIC GENPMENU
SYNT SYNTAX DIAGRAM SCHEMATICS SEE SCHEMATIC SYNTAX
SYS EXAMPLES OF LCN SYSTEM LEVEL SCHEMATICS SEE SCHEMATIC
SYSTEMENU
TRCK SLICK TRICKS AND SLEIGHT OF HAND SEE SCHEMATIC TRCKMENU

LAB SETUP (if necessary)

OBJECTIVE: Create copies of the source cartridges.

Copy the contents of your AGB1 cartridge.

1. Insert your cartridge in the right drive.
2. Type `CV $Fnn>T` (create a new volume)
3. Deselect the right drive.
4. Re-insert your cartridge into the right drive.
5. Insert AGB1 master into left drive.
6. Type `FCP $Fmm $Fnn` (copy left to right)

Copy the contents of your AGB2 cartridge.

1. Insert your cartridge in the right drive.
2. Type `CV $Fnn>T` (create a new volume)
3. Deselect the right drive.
4. Re-insert your cartridge into the right drive.
5. Insert AGB2 master into left drive.
6. Type `FCP $Fmm $Fnn` (copy left to right)

COURSE PROCEDURES

10 Area databases

- Area Items of Particular Interest for This Course
 - Button file
 - Memory-Resident Schematics
 - Pathname Catalog

COURSE PROCEDURES, continued

Figure 1 Operator's Pathname Catalog—R5xx

SELECT SUMMARY08 Aug 96 17:36:441

ORGANIZATIONAL SUMMARY MENU

UNIT POINT SUMMARY

HIWAY BOX POINT SUMMARY

POINT USAGE LIST

UNIT POINT ATTRIBUTE SUMMARY

AREA POINT ATTRIBUTE SUMMARY

LCN NODE POINT SUMMARY

UCN DEVICE POINT SUMMARY

UCN DEVICE MODULE PT SUMMARY

AREA TITLES

UNIT TITLES

GROUP TITLES

UNIT TREND TITLES

SCHEMATIC/FFL TITLES

CANCEL

PRINT

22 Aug 96 11:52:391

TITLE SUMMARY - SCHEMATIC/FFL

LAST PAGE 1

DISPLAY TIME 11:52:16

PATHNAMES FOR SCHEMATIC SEARCH

\$F1>WORK

\$F2>TRCK

\$F2>ACT

\$F2>FUN

\$F2>DEMO

NET>PICS

\$F2>SYNT

\$F2>GENP

\$F2>SYS

34500

SCHEMATIC/FFL TITLES (Yellow means memory-resident)

MOOSE

ARITHEXP

IFMENU

TRCKMENU

SYNTAX

APPLMENU

RSDDDB1

COND_IF

STRNGEXP

SYSTEMENU

VALUMENU

CONDSYN

CONDSYN2

IFSYN

MIXEDEXP

ENUMEXP

SYSTEMENU

HELP_EXP

COLOR TIP

CONDSYN3

BOOLEXP

GENPMENU

FUNMENU

NUM_PAR1

CONDMENU

VARMENU

ACTRULES

ACTMENU

DDBMENU

NUM_PAR2

TIPSMENU

GRID1

34566

COURSE PROCEDURES, continued

Figure 2 Pathname Catalog for Area 1

22 Aug 96 11:54:13 1

PED >>>>> POINT:\$OABSTR
AREA:01
PAGE 01 OF 04

PATHNAME CATALOG CONFIGURATION

BUTTON CONFIGURATION FILE: VOLUME ID 001
 FILE NAME BUTTON

SCHEMATIC OR FFL PATHNAMES (DEVICE>VOLUME)
(SEARCHED LEFT TO RIGHT, TOP TO BOTTOM):

\$F1>WORK	\$F2>ACT	\$F2>DEMO	\$F2>SYNT
\$F2>SYS	\$F2>TRCK	\$F2>FUN	NET>PICS
	\$F2>GENP		

MEMORY RESIDENT SCHEMATIC OR FFL FILE NAMES
(LOADED LEFT TO RIGHT, TOP TO BOTTOM):

MOOSE	FFL	COND_IF	FFL	MIXEDEXP	FFL	BOOLEXP	FFL	ACTRULES	FFL
ARITHEXP	FFL	STRNGEXP	FFL	ENUMEXP	FFL	GENPMENU	FFL	ACTMENU	FFL
IFMENU	FFL	SYSTEMENU	FFL	SYSTEMENU	FFL	FUNMENU	FFL	DDBMENU	FFL
TRCKMENU	FFL	VALUMENU	FFL	HELP_EXP	FFL	NUM_PAR1	FFL	NUM_PAR2	FFL
SYNTAX	FFL	CONDSYN	FFL	COLORTIP	FFL	CONDMENU	FFL	TIPSMENU	FFL
APPLMENU	FFL	CONDSYN2	FFL	CONDSYN3	FFL	VARMENU	FFL	GRID1	FFL
RSO0B1	FFL	IFSYN	FFL		FFL		FFL		FFL

F1=PED	F3=	F5=OVERWRITE	F7=RECON	F9 =WLK BACK	F11=TAB
F2=RECALL DISP	F4=	F6=	F8=PED STATUS	F10=WRITE	F12=LOAD

34567

COURSE PROCEDURES, continued

Figure 3 Pathname Catalog for Area 6

22 Aug 96 11:55:21 1

PED >>>>> POINT:\$0ABSTRT
AREA:06
PAGE 01 OF 04

PATHNAME CATALOG CONFIGURATION

BUTTON CONFIGURATION FILE: VOLUME ID 006
 FILE NAME BUTTON

SCHEMATIC OR FFL PATHNAMES (DEVICE>VOLUME)
 (SEARCHED LEFT TO RIGHT, TOP TO BOTTOM):

\$F11>WORK	\$F12>ACT	\$F12>DEMO	\$F12>SYNT
\$F12>SYS	\$F12>TRCK	\$F12>FUN	NET>PICS
	\$F12>GENP		

MEMORY RESIDENT SCHEMATIC OR FFL FILE NAMES
 (LOADED LEFT TO RIGHT, TOP TO BOTTOM):

MOOSE	FFL	COND_IF	FFL	MIXEDEXP	FFL	BOOLEXP	FFL	ACTRULES	FFL
ARITHEXP	FFL	STRNGEXP	FFL	ENUMEXP	FFL	GENPMENU	FFL	ACTMENU	FFL
IFMENU	FFL	SYSTEMU	FFL	SYSTEMU	FFL	FUNMENU	FFL	OGBMENU	FFL
TRCKMENU	FFL	VALUMENU	FFL	HELP_EXP	FFL	NUM_PAR1	FFL	NUM_PAR2	FFL
SYNTAX	FFL	CONDSYN	FFL	COLORTIP	FFL	CONDMENU	FFL	TIPSMENU	FFL
APPLMENU	FFL	CONDSYN2	FFL	CONDSYN3	FFL	VARMENU	FFL	GRID1	FFL
RSODB1	FFL	IFSUN	FFL		FFL		FFL		FFL

F1=PED
F3=
F5=OVERWRITE
F7=RECON
F9 =WLK BACK
F11=TAB

F2=RECALL DISP
F4=
F6=
F8=PED STATUS
F10=WRITE
F12=LOAD

34568

COURSE PROCEDURES, continued

Normal Operating Procedures

- Your Operating Area = Station Number
 - Select Left Drive as Alternate Source
 - Loads Area from left drive
 - Loads Memory-Resident Schematics from Right
- When you leave for lunch, take your disks with you.
- Unfortunately Your AGB1 Needs a Little Work
 - Area 1 through Area 8 are correct.
 - Area 9 and Area 10 need to have their pathname catalogs updated.
 - Your task is to Correct This Problem

COURSE PROCEDURES, continued

Area Database Considerations

- If you modify a display but the old one still comes up, there are four possible reasons:

- Directory containing new display is not in Area search path.

Action: Copy display object to directory that is searched.

- Display is memory-resident.

Action: Perform Area Change to see new version.

- Another schematic with the same name is in a directory specified on the search path before the directory containing your schematic.

Action: Delete or replace the schematic in the first directory.

- R510 and later: The schematic exists as the prior display in station memory and the station checked that buffer before going to the the Area search path media.

Action: Clear the screen to clear the prior display from station memory.

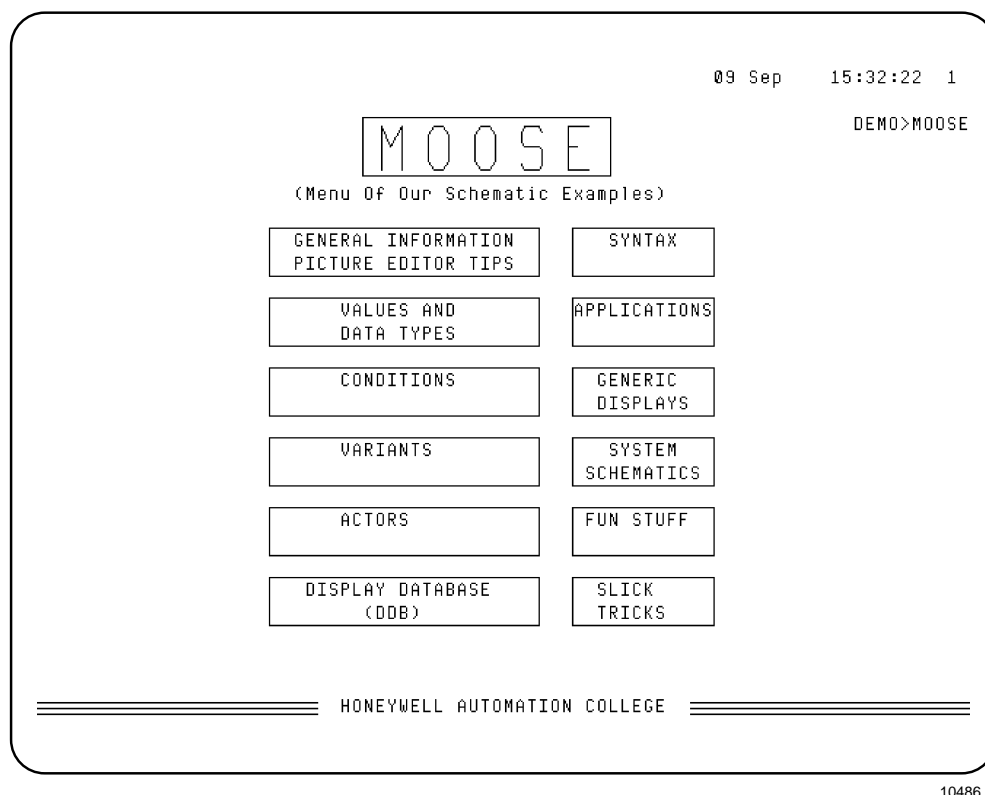
(For convenience, you may want to configure a Clear Screen button on the US used for display building.)

COURSE PROCEDURES, continued

Menu Of Our Schematic Examples

- Schematics available for your viewing pleasure
- To call up MOOSE:
 - press [J] on alphabetic operator keyboard, or
 - press [P] on qwerty operator keyboard.
- If pathname appears in upper-right corner, then source is included on your course cartridges.
- For a Picture Editor overview, select GENERAL INFORMATION on MOOSE.

Figure 4 MOOSE



COURSE PROCEDURES, continued

Exercises

- **Overall Plan**

- Work through a set of schematics, following MOOSE.
- Take notes if you find something new or interesting
- After finishing a set of schematics, then do the lab exercise from a directory on the left drive.

- **Guidelines**

- If the file is a .DS, set your path from the Command Processor, enter PE, read in the file, and follow instructions.
- If the file is a .XX, this means it is a text file. You can print the text file to the screen by using a PR command and try to find the syntax errors.
- The real fun is letting the PE tell where the syntax errors are. Instructions are in each course module.

MEMORY-RESIDENT DISPLAYS

Station Memory Available for .DO and .FO files

- In R4xx, 72 Kwords (144 Kbytes), of station memory is available for Custom Schematics and Free Format Log object files.

In R500 and later, you may increase the space available for US-resident schematics and logs through an NCF configuration entry (in kilowords) called External Custom Schematic Memory.

	Required	Maximum	Maximum Available
for	<u>Memory</u>	<u>Usable Memory</u>	<u>Additional Space</u>
OPR w/EPDG	4 Mw	up to 7.25 Mw	3.25 Mw (3250 Kw)
OPR w/PDG	4 Mw	up to 7.50 Mw	3.50 Mw (3500 Kw)
UNP w/EPDG	6 Mw	up to 7.25 Mw	1.25 Mw (1250 Kw)
UNP w/PDG	6 Mw	up to 7.50 Mw	1.50 Mw (1500 Kw)
UXS w/TPDG	8 Mw	up to 7.50 Mw	0 Mw

- To determine the remaining available memory, list the files that are currently memory-resident:

R4xx: LS \$MEMORY>MRnn>*.*

R5xx: LS \$EMEMORY>MRnn>*.*

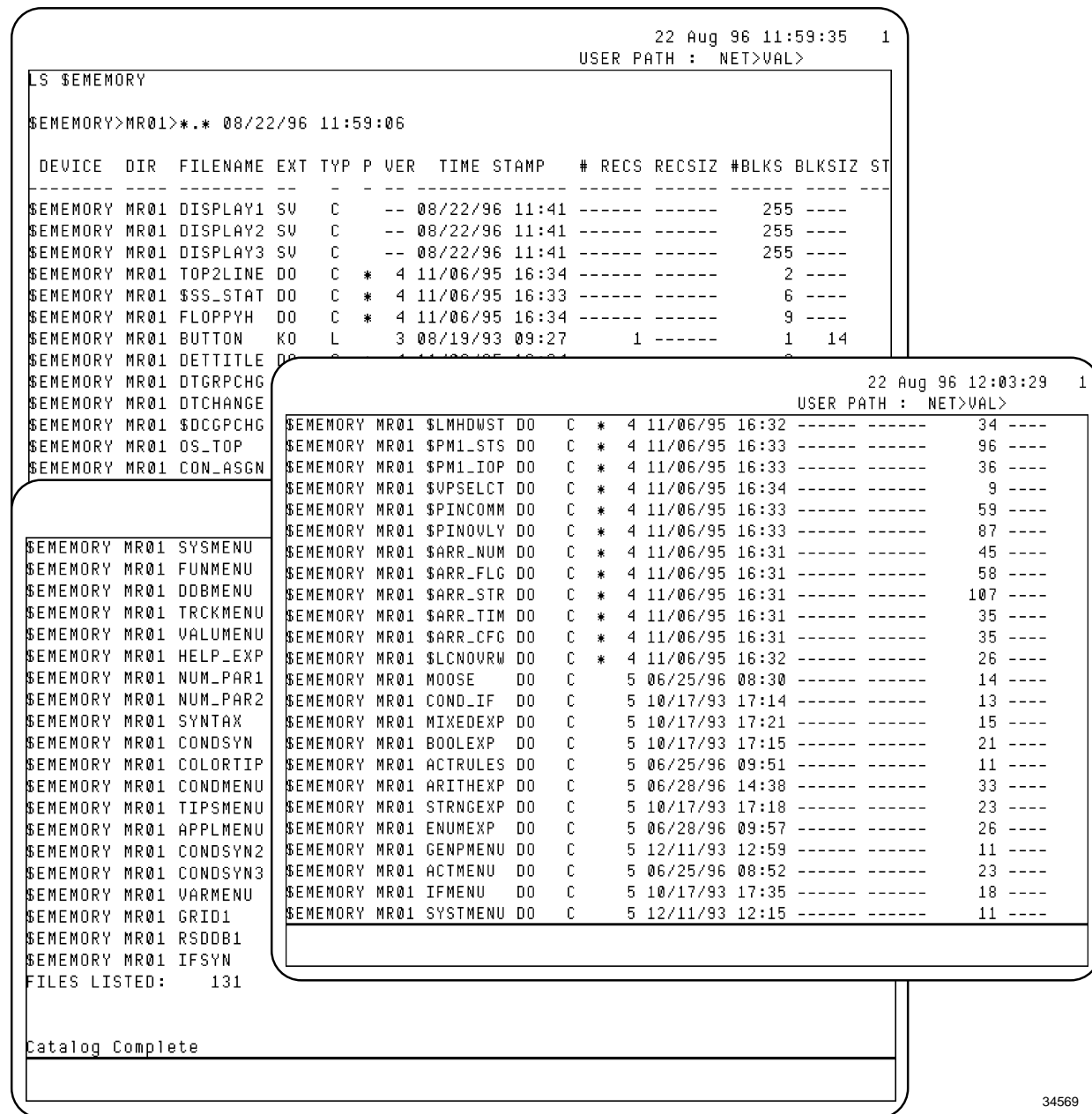
nn = station node number

- Figures 5 and 6 show examples of the results.

At the end of the listing are the memory-resident user .DO files.

MEMORY-RESIDENT DISPLAYS, continued

Figure 5 Memory-Resident Displays—R5xx



34569

MEMORY-RESIDENT DISPLAYS, continued

Figure 6 Memory-Resident Displays—R4xx

16 Aug 14:16:49 6
USER PATH : NET>GENE>

```
LS $MEMORY>MR44>*.*
```

```
$MEMORY>MR44>*. 08/16/93 14:03:31
```

DEVICE	DIR	FILENAME	EXT	TYP	P	VER	TIME	STAMP	# RECS	RECSIZ	#BLKS
\$MEMORY	MR44	AREA06	DA	C		7	08/11/93	11:34	-----	-----	1152
\$MEMORY	MR44	\$ARR_TIM	DO	C	*	4	05/03/93	13:56	-----	-----	35
\$MEMORY	MR44	\$ARR_CFG	DO	C	*	4	05/03/93	13:56	-----	-----	33
\$MEMORY	MR44	\$HMDSKST	DO	C	*	4	05/03/93	13:56	-----	-----	39
\$MEMORY	MR44	MOOSE	DO	C		5	08/11/93	11:27	-----	-----	10
\$MEMORY	MR44	COND_IF	DO	C		5	08/09/93	14:49	-----	-----	13
\$MEMORY	MR44	MIXEDEXP	DO	C		5	08/03/93	15:53	-----	-----	15
\$MEMORY	MR44	BOOLEXP	DO	C		5	08/13/93	09:17	-----	-----	21
\$MEMORY	MR44	ACTRULES	DO	C		5	07/21/93	17:07	-----	-----	7
\$MEMORY	MR44	ARITHEXP	DO	C		5	08/03/93	10:04	-----	-----	28
\$MEMORY	MR44	STRNGEXP	DO	C		5	08/03/93	10:05	-----	-----	23
\$MEMORY	MR44	ENUMEXP	DO	C		5	08/03/93	10:05	-----	-----	23
\$MEMORY	MR44	GENPMENU	DO	C		5	08/07/93	16:37	-----	-----	7
\$MEMORY	MR44	ACTMENU	DO	C		5	08/15/93	16:54	-----	-----	12
\$MEMORY	MR44	IF_CMP	DO	C		5	08/15/93	12:50	-----	-----	10
\$MEMORY	MR44	SYSTEMMENU	DO	C		5	06/25/93	14:55	-----	-----	7

16 Aug 14:17:59 6
USER PATH : NET>GENE>

\$MEMORY	MR44	SYSTEMMENU	DO	C		5	07/31/93	10:42	-----	-----	6
\$MEMORY	MR44	FUNMENU	DO	C		5	08/01/93	10:49	-----	-----	9
\$MEMORY	MR44	DOBMENU	DO	C		5	07/28/93	16:03	-----	-----	4
\$MEMORY	MR44	TRCKMENU	DO	C		5	08/02/93	11:10	-----	-----	6
\$MEMORY	MR44	VALUMENU	DO	C		5	08/12/93	14:17	-----	-----	8
\$MEMORY	MR44	HELP_EXP	DO	C		5	08/09/93	15:40	-----	-----	6
\$MEMORY	MR44	NUM_PAR1	DO	C		5	08/09/93	13:43	-----	-----	6
\$MEMORY	MR44	NUM_PAR2	DO	C		5	08/09/93	13:44	-----	-----	5
\$MEMORY	MR44	SYNTAX	DO	C		5	08/10/93	08:11	-----	-----	9
\$MEMORY	MR44	CONDSYN	DO	C		5	07/10/93	14:24	-----	-----	36
\$MEMORY	MR44	COLORTIP	DO	C		5	08/09/93	16:11	-----	-----	3
\$MEMORY	MR44	CONDMENU	DO	C		5	08/10/93	09:57	-----	-----	5
\$MEMORY	MR44	TIPSMENU	DO	C		5	08/15/93	14:02	-----	-----	7
\$MEMORY	MR44	APPLMENU	DO	C		5	08/15/93	13:56	-----	-----	3

FILES LISTED: 178

10489

MEMORY-RESIDENT DISPLAYS, continued

.DO Block Size

- The maximum size of a .DO file (display or overlay) is
 - R4xx = •17 K words (150 blocks)
 - R5xx = •32 K words (250 blocks)
- As shown in Figures 5 and 6, .DO files are sized in blocks.

256 bytes = 1 block

8 blocks = 1 K words

1000 K words = 1 Mw

1 Mw = 8000 blocks

8000 blocks = 32 schematics at maximum R5xx size

PE CONSIDERATIONS

ACKSTAT Collector

1. Tagnames with remote PIN IDs should not be used with ACKSTAT; that is, do not use ACKSTAT through a Network Gateway.

IF...a tagname with a remote PIN ID is used as the input parameter to an ACKSTAT collector,

THEN...the schematic will compile and can be invoked, but with the following results:

- ACKSTAT references with remote PIN IDs will return indicators for “unavailable data”. The displayed value is a string of @ signs.
2. For R420 and later, there is only one ACKSTAT limitation—a picture cannot have more than 20 different Collection Rate/Group ID combinations containing ACKSTAT references.

PE CONSIDERATIONS, continued

ACKSTAT Limitations—Before R420

1. Before R420, you can reference the ACKSTAT collector on a main picture or on an overlay picture, but not on both.

If referenced on both, the ACKSTAT on the overlay picture function properly, but not on the main picture.

2. Before R420, all references to the ACKSTAT collector on a picture must be contained in the **same collection group** and must be the **same collection rate**.

WARNING—Before R420

IF...references to the ACKSTAT collector are divided between collection groups and/or are given differing collection rates,

THEN...the references will not function properly; furthermore the references will ultimately cause the US to crash because of a lack of heap memory.

IF... multiple collection groups are specified,

THEN...the schematic will compile, it can be invoked, and ACKSTAT references within the first specified collection group will display proper values,

BUT...references within other groups will return indicators for “unavailable data”. The displayed value is a string of @ signs.

PE CONSIDERATIONS, continued

ACKSTAT Limitations—Before R420, *continued*

3. Before R420, each picture or overlay is limited to 100 ACKSTAT collector references.

WARNING—Before R420

IF...the ACKSTAT limit is exceeded,

THEN...the following consequences could occur:

- The operator's keyboard could be locked out for an excessive length of time (over 15 seconds) during the schematic invocation.
- Alarm displays and indicators (LEDs, annunciator horns) could become unsynchronized across US nodes in the same console and with the same Area Database.

CONDITIONS

Figure 7 Making Text Invisible to the Operator

02 Nov14:52:566

\$F12>GENP>DDBLAB4FGBR- 0, 2C- 0, 0N-ON

GENP>DDBLAB4

REQUIRES LOADING A USER DDBFILE L DDBLAB4

THIS DISPLAY USES THE USER GLOBAL DDB ONLY

UNIT 1

UNIT 2

TTTTTTTT RRRRRRRR

TTTTTTTT RRRRRRRR

TTTTTTTT RRRRRRRR

TTTTTTTT RRRRRRRR

HELP AND ASSOC DISP ARE CONFIGURED FOR THIS DISPLAY

02 Nov14:53:346

\$F12>GENP>DDBLAB4FGBR- 0, 2C- 48, 352N-ON

Page 1 of 1

Conditional Behavior

Behavior for Bad ValueBLACK NO BLINK NO REVERSE FULL

Initial BehaviorRED NO BLINK NO REV FULL

Condition

IF ON THEN SET BLACK

<PAGE FWD> <PAGE BACK> to MOVE. <F2> for TFE. <F3> to JUMP. <F4> to DELETE.

M

Enter Condition Information

10693

VARIANTS

Figure 8 Checking the State of a Digital Composite

02 Nov14:55:016

NET>GENE>VARIANT1FGBR- 0, 0C- 80, 224N-ON

Variant At 0, 0Page 1 of 1

Subpicture Or Text For Bad Value "Bad Value"

Variant Body

IF EXTERNAL(FVL21851.PV) = FVL21851.STATE2 THEN SUB SUB1
ELSE IF EXTERNAL(FVL21851.PV) = FVL21851.STATE1 THEN ""

<PAGE FWD> <PAGE BACK> to MOVE. <F2> for TFE. <F3> to JUMP. <F4> to DELETE.

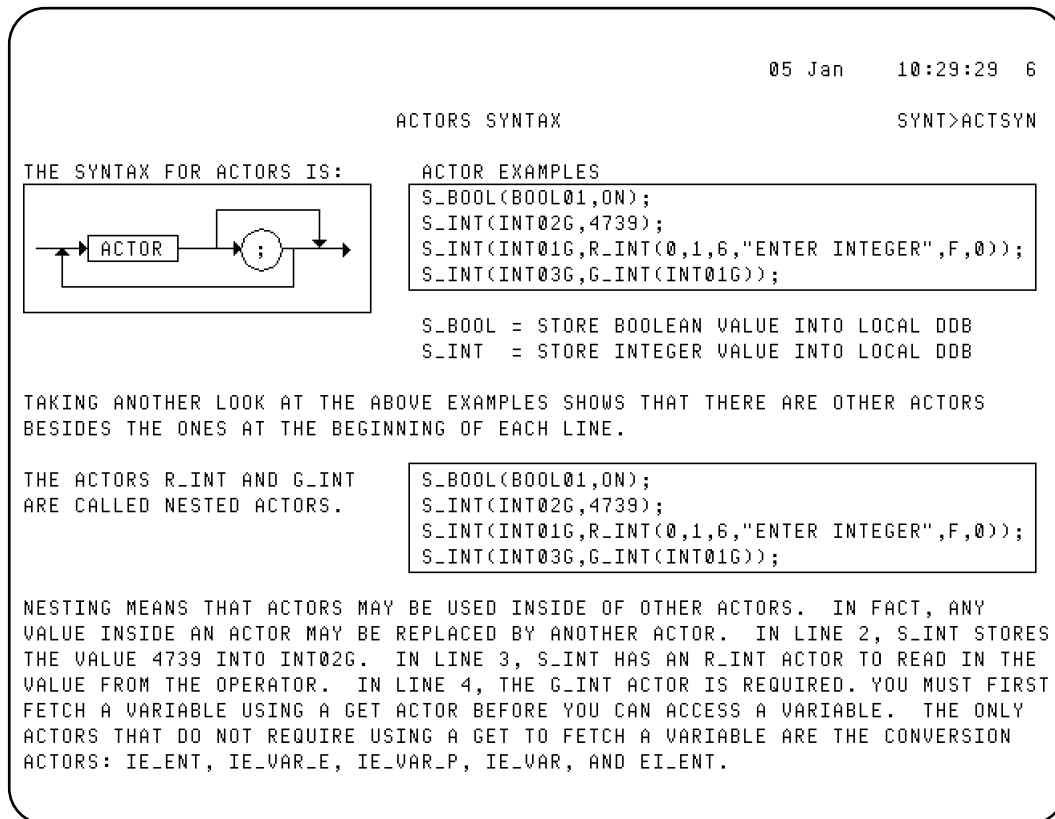
M

Enter Variant Information

10694

ACTORS

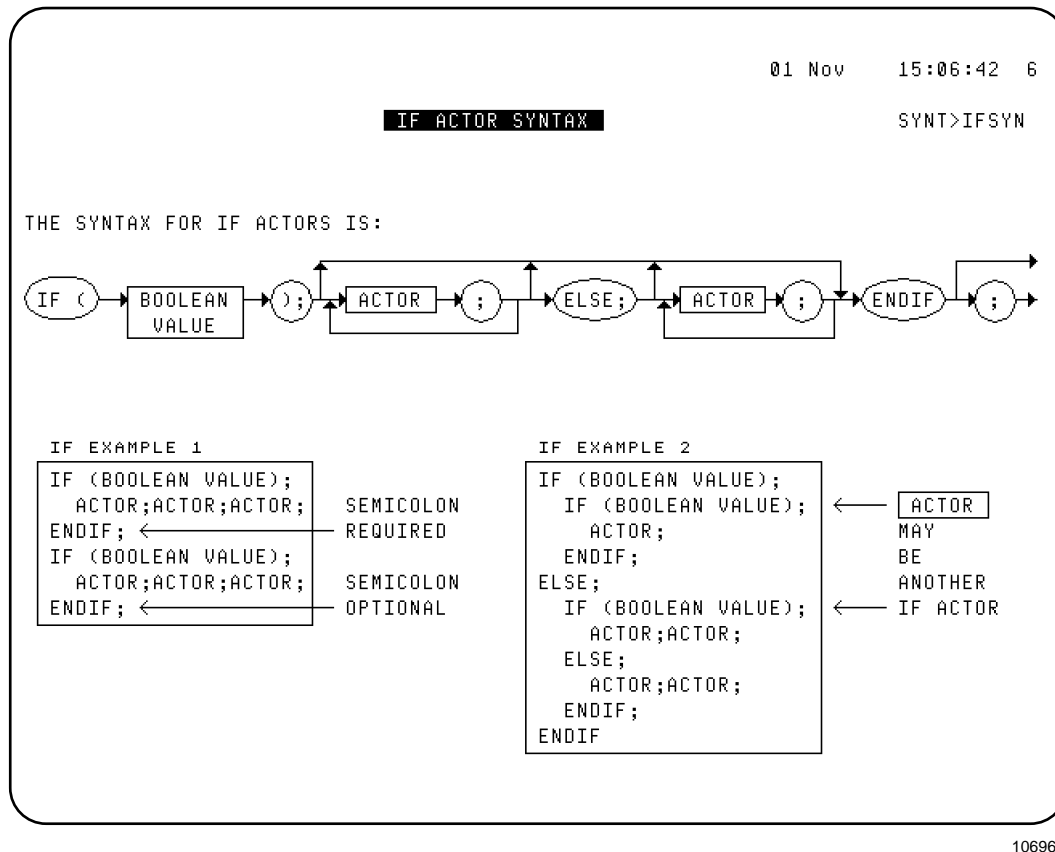
Figure 9 Actor Syntax



10695

ACTORS, continued

Figure 10 IF Actor Syntax



ACTORS, continued

Figure 11 NOT Actor

01 Nov15:08:276

NET>GENE>BOOLEAN

FGBR-0,0C-48,304N-ON

Target At48,304

Page1 of 1

Solid/Box/Invisible

Solid

Action

S_BOOL(B00L01G,NOT(G_BOOL(B00L01G)));
UPDATE(-1,0)

<PAGE FWD> <PAGE BACK> to MOVE. <F2> for TFE. <F3> to JUMP. <F4> to DELETE.

M

Enter Target Specifications

10697

ACTORS, continued

Figure 12 Internal Entity to String Conversion

01 Nov 88 15:09:44 6

NET>GENE>DETAIL	FCBR- 0, 0C- 112, 304N-ON
-----------------	---------------------------

Target At 112, 304Page 1 of 1

Solid/Box/Invisible

Action

DETAIL(IE_ENT(ENT02))

<PAGE FWD> <PAGE BACK> to MOVE. <F2> for TFE. <F3> to JUMP. <F4> to DELETE.

M

Enter Target Specifications

10698

ACTORS, continued

Figure 13 Compare String Without IF Actor

02 Nov14:44:146

NET>GENE>TARGETS

FGB|R-0,0|C-240,304|N-ON

Target At240, 304Page1 of 1

Solid/Box/InvisibleSolid

Action

S_BOOL(B00L01G,CMP_S(GS_ENM(TIC21841.MODE),EQ,"MAN"));
UPDATE(-1,0)

<PAGE FWD> <PAGE BACK> to MOVE. <F2> for TFE. <F3> to JUMP. <F4> to DELETE.

M

Enter Target Specifications

10699

ACTORS, continued

Figure 14 Reading, Storing and Converting Variable

02 Nov 14:48:30 6

NET>GENE>TARGETS	FGB R- 0, 0C- 432, 304N-0N
Target At 432, 304	Page 1 of 1
Solid/Box/Invisible <input type="text" value="Solid"/>	
Action	
<div><div><A TARGET FROM PEEKPOKE - USER ENTERS AN ENTITY.PARAMETER></div><div>S_INT(INT01,4);UPDATE(-1,2); S_VAR(VAR04,R_VAR(3,23,26,"ENTER VARIABLE NAME",T,0)); S_STR(STRING04,IE_VAR(VAR04)); UPDATE(4,2)</div></div>	
<PAGE FWD> <PAGE BACK> to MOVE. <F2> for TFE. <F3> to JUMP. <F4> to DELETE.	
M	
Enter Target Specifications	

10700

ACTORS, continued

Figure 15 Converting System Variable Type to a String

02 Nov14:43:326

NET>GENE>TARGETSFGBR-0,0C-176,304N-0N

Target At176, 304Page1 of 1

Solid/Box/InvisibleSolid

Action

S_STR(STR01G,GS_VAR_S(RAMP301.CODSTN(1)));
UPDATE(-1,0)

<PAGE FWD> <PAGE BACK> to MOVE. <F2> for TFE. <F3> to JUMP. <F4> to DELETE.

M

Enter Target Specifications

10701

ACTORS, continued

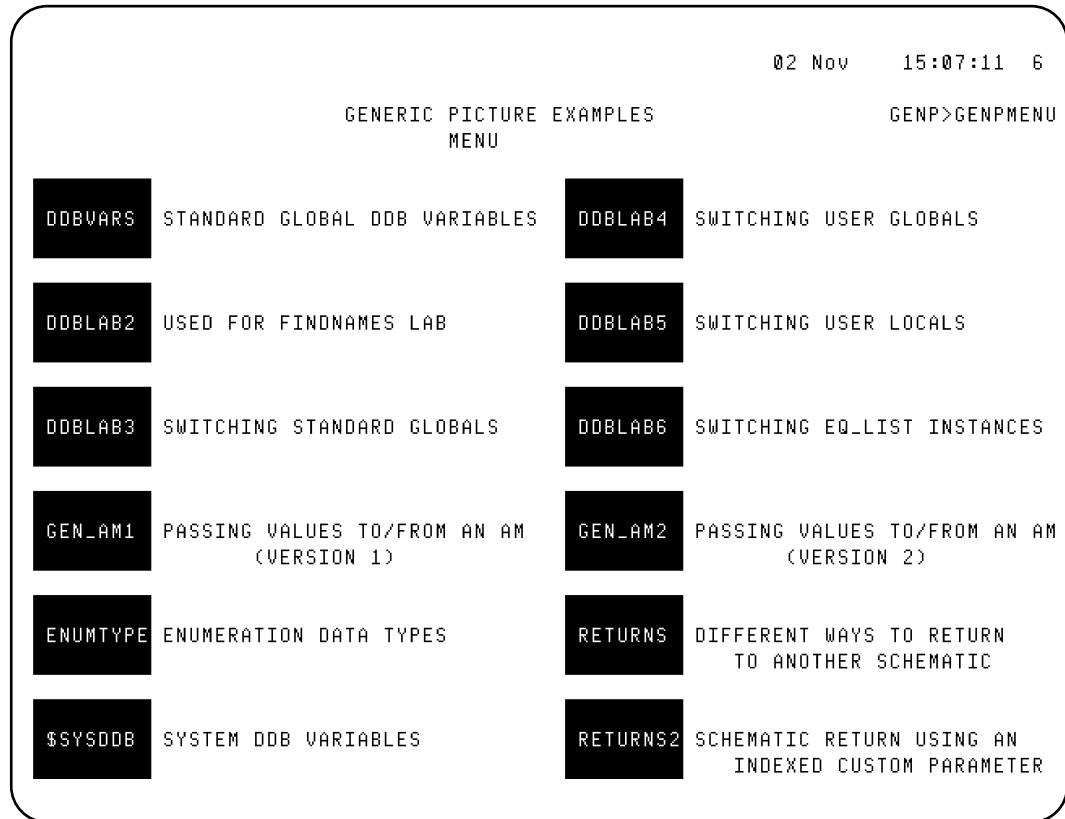
Figure 16 Preventing Operator Contention

02 Nov 14:46:44 6	
NET>GENE>TARGETS	FGBR- 0, 0C- 368, 304N-ON
Target At 368, 304	Page 1 of 1
Solid/Box/Invisible <input type="text" value="Solid"/>	
Action	
<pre>ENT_EXEC("PRESS ENTER: ALLOCATE FOR TEST"); IF (AND(CMP_R(GS_REAL(ENT_SEPR.NODE_NUM),EQ,0.0), CMP_S(GS_ENM(ENT_SEPR.ALOCAT),EQ,"IDLE"))); SS_REAL(ENT_SEPR.NODE_NUM,FLOAT(G_INT(\$MY_PNA))); DELAY(0,0,500); IF (CMP_R(GS_REAL(ENT_SEPR.NODE_NUM),EQ,FLOAT(G_INT(\$MY_PNA)))); SS_ENT(ENT_SEPR.WELL_TAG,G_ENT(ENT_WLL)); SS_BOOL(ENT_SEPR.ENTRY,ON); SS_BOOL(ENT_SEPR.PPS,ON); ELSE; PROMPT("REQUESTED SEPARATOR BUSY"); ENDIF; ENDIF;</pre>	
<PAGE FWD> <PAGE BACK> to MOVE. <F2> for TFE. <F3> to JUMP. <F4> to DELETE.	
M	
Enter Target Specifications	

10702

GENERIC SCHEMATICS

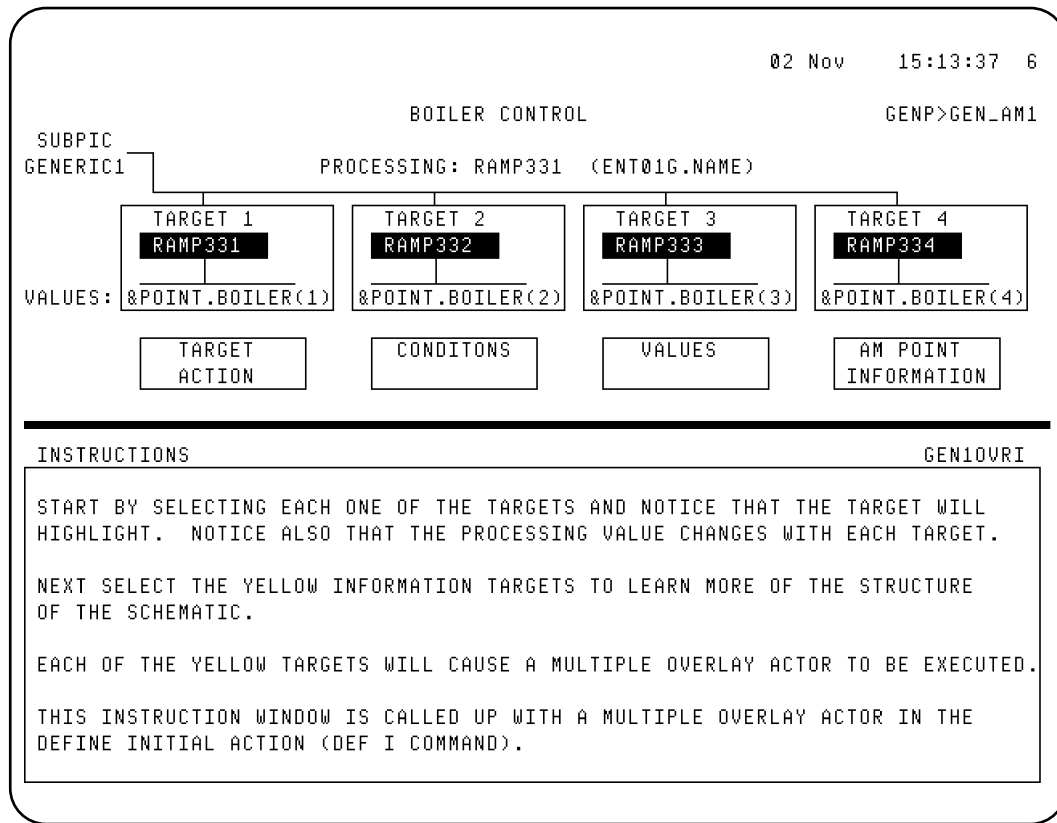
Figure 17 Generic Picture Examples



10703

GENERIC SCHEMATICS, continued

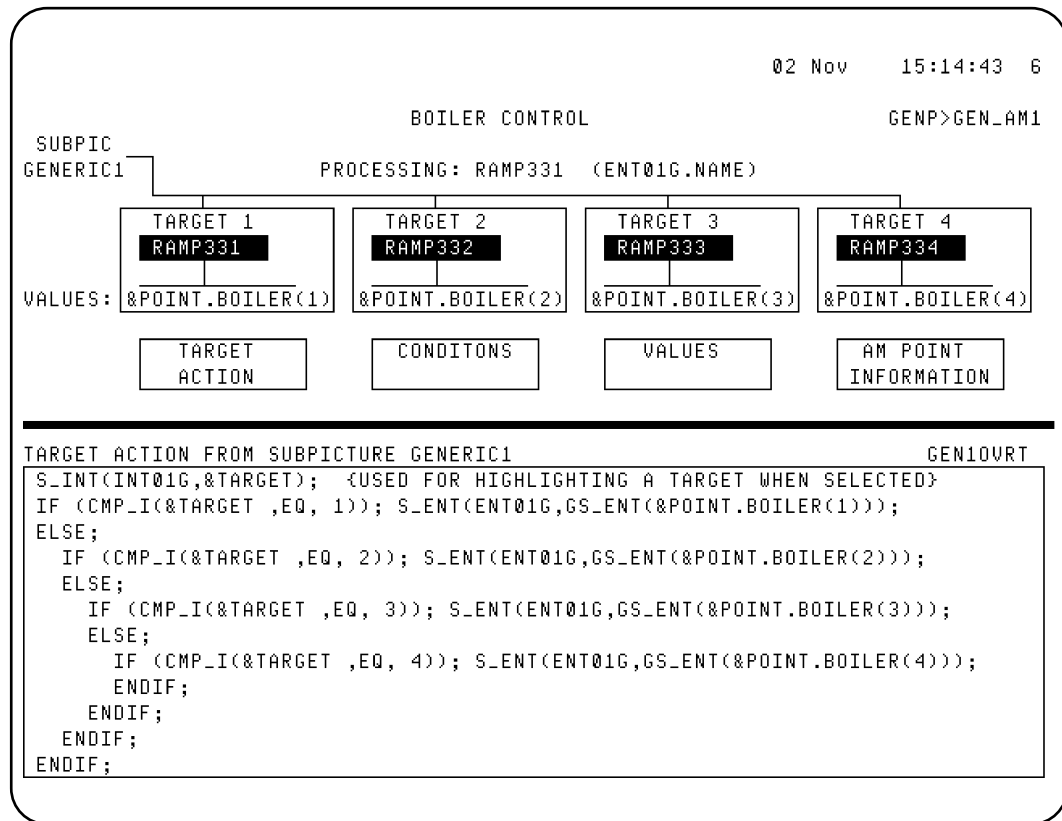
Figure 18 Instructions for Operating Generic Schematics



10704

GENERIC SCHEMATICS, continued

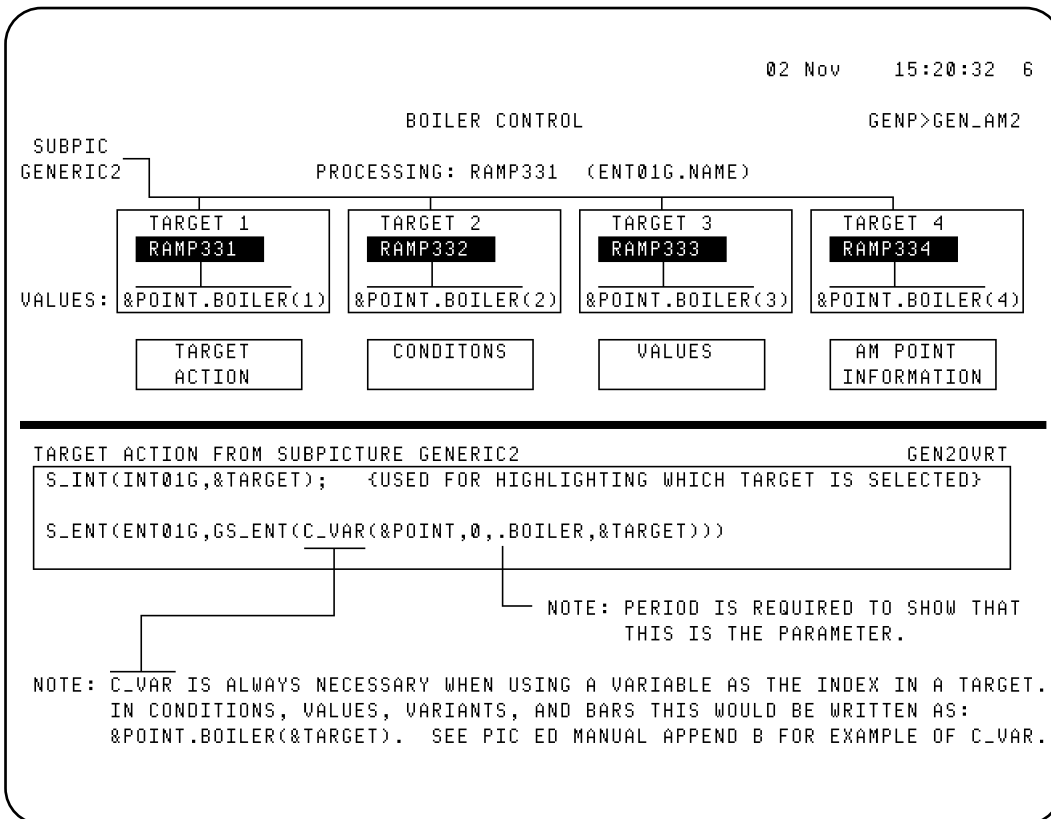
Figure 19 Target Action from Subpicture GENERIC1



10705

GENERIC SCHEMATICS, continued

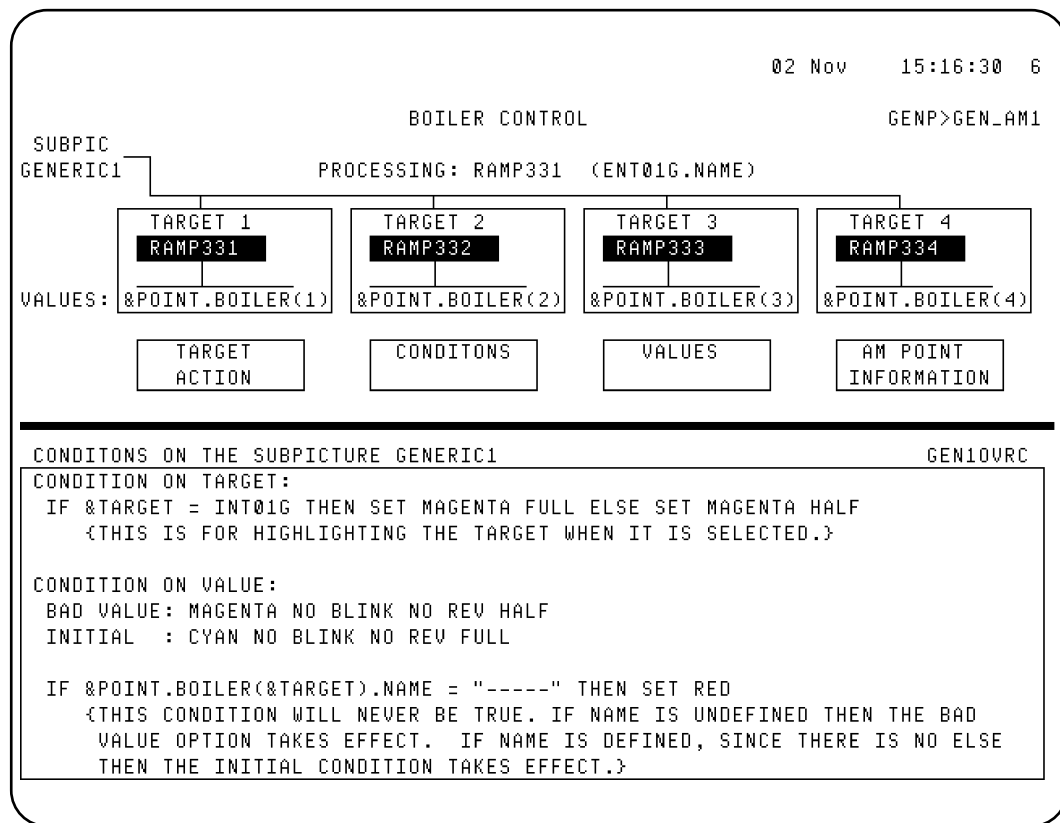
Figure 20 Target Action from Subpicture GENERIC2



10706

GENERIC SCHEMATICS, continued

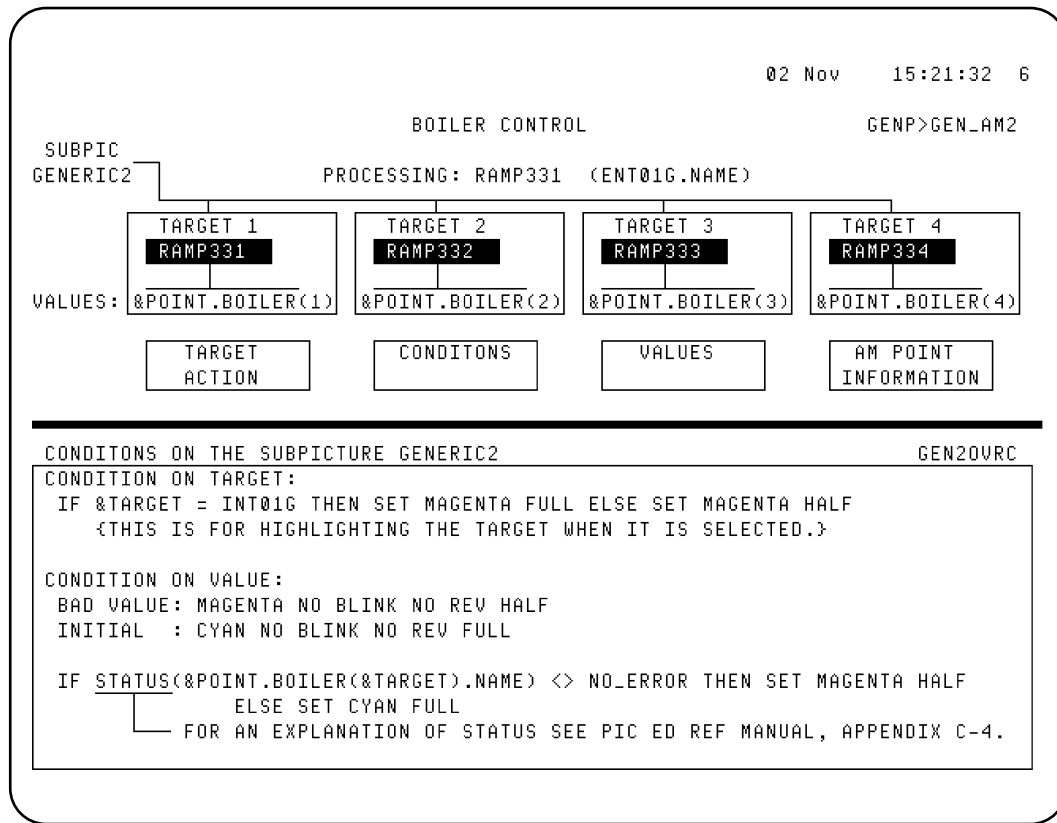
Figure 21 Conditions on the Subpicture GENERIC1



10707

GENERIC SCHEMATICS, continued

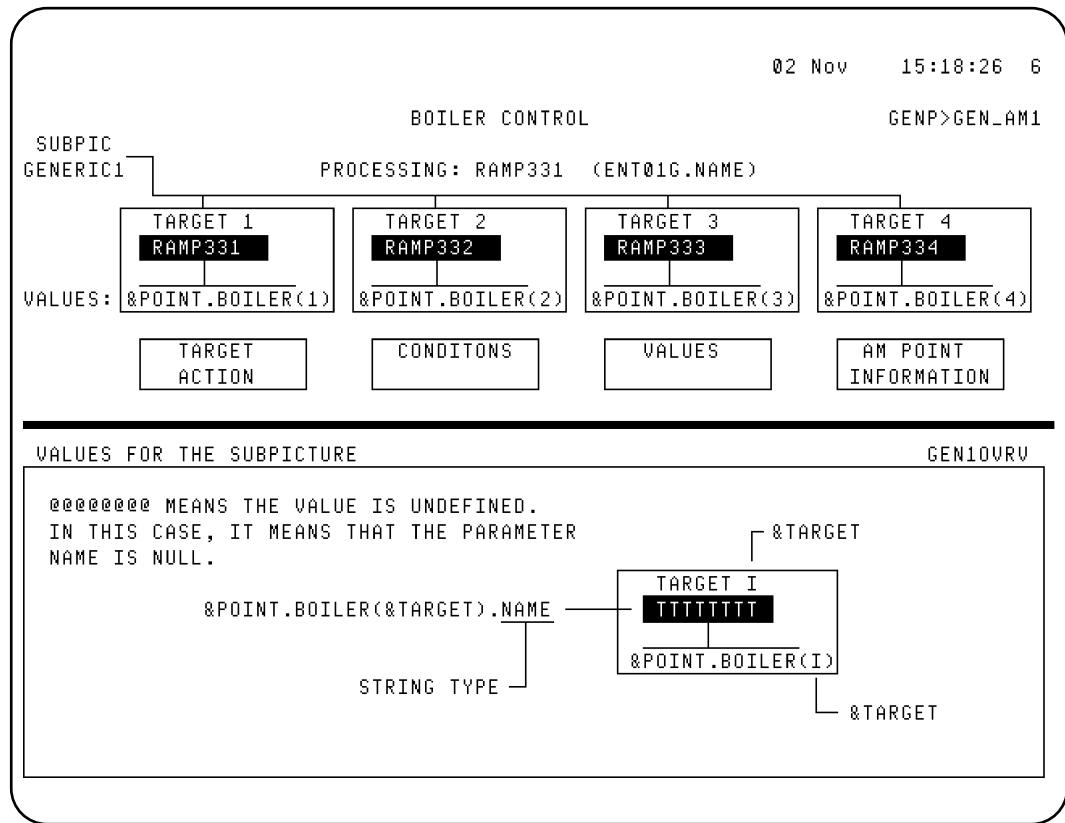
Figure 22 Conditions on the Subpicture GENERIC2



10708

GENERIC SCHEMATICS, continued

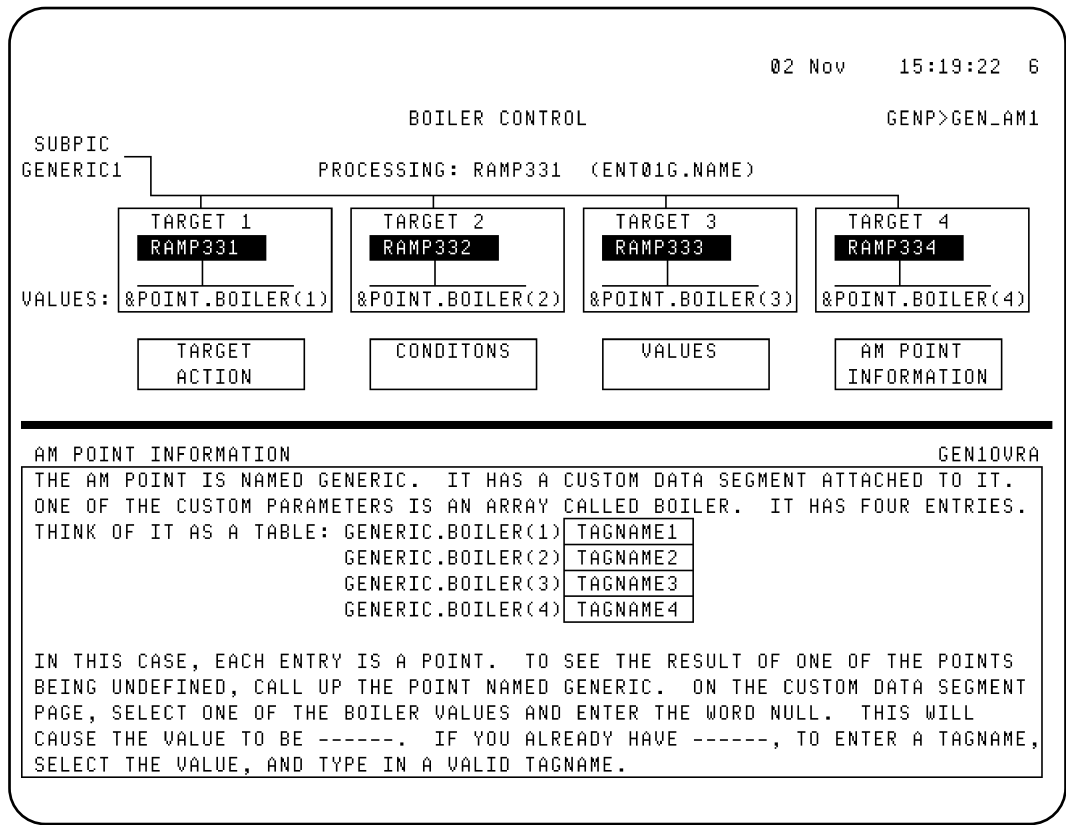
Figure 23 Values for the Subpicture



10709

GENERIC SCHEMATICS, continued

Figure 24 AM Point Information



10710

ENUMERATIONS

Figure 25 Standard Enumerations

02 Nov15:22:486

ENUMERATION TYPES

DEMO>ENUMTYPE

DEFINITION: AN ENUMERATION IS A LIST OF STATES: FORWARD / STOPPED / REVERSE,
OPEN / CLOSED OK / IDLE / ERROR / OFF MAN / AUTO / CAS / BCAS

THERE ARE THREE DIFFERENT TYPES OF ENUMERATIONS:

1. HONEYWELL STANDARD ENUMERATIONS: MODE, REDTAG, STATE, ETC.

2. SELF-DEFINED ENUMERATIONS: STATES OF DIGITAL COMPOSITES, FLAGS, DI's, & DO'S.

3. CUSTOM ENUMERATIONS: CUSTOM DATA SEGMENT PARAMETERS OF TYPE ENUMERATION.

THESE LAST TWO ARE ALSO CALLED USER-DEFINED ENUMERATIONS.

STANDARD
ENUMERATIONS

SELF-DEFINED
ENUMERATIONS

CUSTOM
ENUMERATIONS

GENERIC
SUBPICTURES

THE PARAMETER REFERENCE DICTIONARIES USE DIFFERENT NOTATION TO INDICATE A
STANDARD PARAMETER OF TYPE ENUMERATION. REGARDLESS OF THE NOTATION USED IN A
DICTIONARY, TO THE PICTURE EDITOR, THE TYPE OF MODE IS E:MODE OR ENUM:MODE.
ANOTHER EXAMPLE: THE PE TYPE FOR THE AM PARAMETER PSTS IS E:PVVALST. SEE AM PRO.
VARIOUS PARAMETER REFERENCE DICTIONARIES ENTRIES FOR THE PARAMETER MODE:
PM/APM: PARAMETER MODE TYPE- E:MODE R400 VERSION, OLDER VERSIONS E(MODE)
AM : PARAMETER MODE TYPE- MODE ENUMERATION
HG : PARAMETER MODE VALUE TYPE- ENUM.

10711

ENUMERATIONS, continued

Figure 26 Self-Defined Enumerations

02 Nov15:24:066

ENUMERATION TYPESDEMO>ENUMTYPE

DEFINITION: AN ENUMERATION IS A LIST OF STATES: FORWARD / STOPPED / REVERSE,
OPEN / CLOSED OK / IDLE / ERROR / OFF MAN / AUTO / CAS / BCAS

THERE ARE THREE DIFFERENT TYPES OF ENUMERATIONS:

1. HONEYWELL STANDARD ENUMERATIONS: MODE, REDTAG, STATE, ETC.
2. SELF-DEFINED ENUMERATIONS: STATES OF DIGITAL COMPOSITES, FLAGS, DI's, & DO's.
3. CUSTOM ENUMERATIONS: CUSTOM DATA SEGMENT PARAMETERS OF TYPE ENUMERATION.

THESE LAST TWO ARE ALSO CALLED USER-DEFINED ENUMERATIONS.

STANDARD
ENUMERATIONS

SELF-DEFINED
ENUMERATIONS

CUSTOM
ENUMERATIONS

GENERIC
SUBPICTURES

IN THE PARAMETER REFERENCE DICTIONARIES, A PARAMETER OF TYPE SELF_DEFINING
ENUMERATION WILL BE INDICATED DIFFERENT WAYS. REGARDLESS OF THE NOTATION, TO
THE PICTURE EDITOR, THE TYPE IS SD_ENM OR SD_ENUM.

PM/APM: PARAMETER PV TYPE - SD_ENM:PUSTATES OR SD_ENM:STATETXT R400 VERSION
E(PUSTATES) OR E(STATETXT) OLDER VERSIONS

AM : PARAMETER PV TYPE - ENUM. FOR FLAG

HG : PARAMETER PV TYPE - SELF-DEFINING ENUM.

10712

ENUMERATIONS, continued

Figure 27 Custom Enumerations

```
02 Nov    15:25:04  6
                                ENUMERATION TYPES          DEMO>ENUMTYPE
DEFINITION: AN ENUMERATION IS A LIST OF STATES: FORWARD / STOPPED / REVERSE,
              OPEN / CLOSED  OK / IDLE / ERROR / OFF  MAN / AUTO / CAS / BCAS
THERE ARE THREE DIFFERENT TYPES OF ENUMERATIONS:
1. HONEYWELL STANDARD ENUMERATIONS: MODE, REDTAG, STATE, ETC.
2. SELF-DEFINED ENUMERATIONS: STATES OF DIGITAL COMPOSITES, FLAGS, DI's, & DO's.
3. CUSTOM ENUMERATIONS: CUSTOM DATA SEGMENT PARAMETERS OF TYPE ENUMERATION.
   THESE LAST TWO ARE ALSO CALLED USER-DEFINED ENUMERATIONS.
   STANDARD  SELF-DEFINED  CUSTOM  GENERIC
   ENUMERATIONS  ENUMERATIONS  ENUMERATIONS  SUBPICTURES
PARAMETERS DEFINED IN CUSTOM DATA SEGMENTS OF TYPE ENUMERATION ARE CREATED
USING THE CL COMPILER.  A SAMPLE DEFINITION:
  ENUMERATION NODETYPE = US/NIM/AM/CG/HG/PLCG/HM
  CUSTOM
    PARAMETER NODE : NODETYPE
  END CUSTOM
THIS MEANS NODE IS A VARIABLE THAT HAS A TYPE OF E:NODETYPE.
```

10713

ENUMERATIONS, continued

Figure 28 Enumerations in Generic Subpictures

02 Nov15:25:576

ENUMERATION TYPESDEMO>ENUMTYPE

DEFINITION: AN ENUMERATION IS A LIST OF STATES: FORWARD / STOPPED / REVERSE,
OPEN / CLOSED OK / IDLE / ERROR / OFF MAN / AUTO / CAS / BCAS

THERE ARE THREE DIFFERENT TYPES OF ENUMERATIONS:

1. HONEYWELL STANDARD ENUMERATIONS: MODE, REDTAG, STATE, ETC.
2. SELF-DEFINED ENUMERATIONS: STATES OF DIGITAL COMPOSITES, FLAGS, DI's, & DO's.
3. CUSTOM ENUMERATIONS: CUSTOM DATA SEGMENT PARAMETERS OF TYPE ENUMERATION.

THESE LAST TWO ARE ALSO CALLED USER-DEFINED ENUMERATIONS.

STANDARD
ENUMERATIONS

SELF-DEFINED
ENUMERATIONS

CUSTOM
ENUMERATIONS

GENERIC
SUBPICTURES

WHEN YOU ARE CREATING GENERIC SUBPICTURES, IF THE PICTURE EDITOR CANNOT TELL THE
TYPE OF A PARAMETER, IT WILL ASK YOU FOR THE TYPE IN THE UPPER LEFT CORNER OF
OF THE SCREEN. YOU MUST BE CAREFUL BECAUSE THE PE ACCEPTS WHATEVER YOU TYPE.

1. &TAG.MODE --> THIS IS A HONEYWELL ENUMERATION SO THE TYPE IS E:MODE
2. &DC.PV --> THIS IS SELF-DEFINED ON A POINT SO THE TYPE IS SD_ENM
3. &A.UNITS --> FROM AM/CL ENUMERATION DESU301 = DOL_DAY/DOL_HR/DOL_MIN
PARAMETER UNITS : DESU301 --> THE TYPE IS E:DESU301

10714

SUBPICTURE REVIEW

Parameter References Per Subpicture

- A subpicture can contain
 - **16 User Visible** generic references (64 on R510)
and
 - **16 Formal** parameter references (64 on R510)

User Visible → &A

An error appears when you add a reference that exceeds the maximum count.

Formal → &A.PV

If maximum count is exceeded, an error appears when you write the subpicture.

10487-A

Type	Definition	Examples
User Visible	A User Visible generic reference: <ul style="list-style-type: none">• instructs user to enter a prompt when the subpicture is written,• always begins with an ampersand (&)• does not include a dot (.)	&A
Formal	A Formal parameter reference: <ul style="list-style-type: none">• includes a dot (.) if of type entity or parameter• does not include a dot (.) if of type real• contains an ampersand (&)	&A.PV RAMP301.&B

- The PE displays `Max Sub Parameters Exceeded` if maximum count is exceeded.
- Multiple references to the same User Visible reference count as only one.
- Slick Trick:
Use the PE Set Collection command to list all references, including User Visible and Formal.

SUBPICTURE REVIEW, continued

- **Written Exercise**

If a subpicture has the following references:

Symbol	Data Type of Parameterized Reference	User Visible Reference	Formal Reference
&P1	REAL		
&P2.PV	ENTITY		
TC3000.&P3	PARAMETER		
&P2.&P3			

How many User Visible references are there? ____

How many Formal parameter references are there? ____

Answers: Three user-visible (&P1, &P2, &P3) and
four formal (&P1, &P2.PV, TC3000.&P3, &P2.&P3)

SUBPICTURE REVIEW, continued

User Visible (parameterized) Variables

- A compile error can occur in the Picture Editor when using user visible (parameterized) variables. The error manifests itself in the following situation:
 - A parameterized (user visible) value is used as an index in a variable *and* as a “stand-alone” variable in the same picture editor object.
 - A subpicture is written that contains the object.
 - Attempting to compile the main picture containing the subpicture results in a “compile error”.

EXAMPLE:

This is an arithmetic expression in a subpicture that causes the above problem. The expression may appear in a Condition, Variant, Value, or Bar.

`&A + PT.PAR (&A)`

In this example, the user visible value used by itself is not flagged to be a formal parameter; therefore, there is no value returned to the subpicture. The compiler fails because it is not able to retrieve the value.

SUBPICTURE REVIEW, continued

User visible (parameterized) Variables, *continued*

SOLUTION:

This solution must be used any time you reference one variable as both an index and a stand-alone variable in the same object in the Picture Editor.

1. Use two different user visible variable names, giving them the same value.

`&A1 + PT.PAR(&A2)`

2. Make a subpicture using the first parameter, then add the second parameter to it. WRITE the subpicture to a file (SUB1). It asks for the prompts for each value. Enter a prompt requesting entry of &A in each:

Prompt for &A1: **ENTER &A**

Prompt for &A2: **ENTER &A**

3. ADD this subpicture (SUB1) to another subpicture (SUB2). This ADD command uses the prompts from step 2 above, requesting that &A be entered for each variable (&A1 and &A2). Enter &A.
4. WRITE SUB2. It now asks for the prompts you want for &A. This is the prompt you really wanted back when the compile error occurred.

Prompt for &A: **ENTER INDEX VALUE**

5. Now ADD subpicture SUB2 to your main schematic. It will use the prompt from step 4 above, requesting an index value be entered. Enter the value.

The compile now shows the two parameters as you want them. This resolves the compile error.

DDB VARIABLES

Figure 29 System DDB Variables

22 Aug 96 12:10:461

SYSTEM DDB VARIABLES

SYS>\$SYS00B

THESE VARIABLES ARE AVAILABLE FOR READ-ONLY USE.

\$ARAI0 = NORTH

NAME OF THE CURRENT AREA (8 CHARACTERS)

\$ARAI001 = NORTH

THE NAME OF AREA 1 (8 CHARACTERS) AVAILABLE FOR ALL AREAS

\$ARAI002 = EAST

THE NAME OF AREA 2

\$ARAI010 = POWER_2

THE NAME OF AREA 10

\$ARADSC = NORTH MANUFACTURING

DESCRIPTION OF THE CURRENT AREA (24 CHARS)

\$ARADS01 = NORTH MANUFACTURING

DESCRIPTION OF AREA 1 (24 CHARS) ALL AREAS

\$ARADS02 = EAST MANUFACTURING

DESCRIPTION OF AREA 2

\$ARADS10 = POWER HOUSE TWO

DESCRIPTION OF AREA 10

\$CONDSC = MANUFACTURING

DESCRIPTION OF CURRENT CONSOLE (24 CHARS)

\$CONDS01 = MANUFACTURING

DESCRIPTION OF CONSOLE 1 (24 CHARS) ALL CONS

\$CONDS02 = POWER HOUSE

DESCRIPTION OF CONSOLE 2

\$CONDS10 = WATER TREATMENT

DESCRIPTION OF CONSOLE 10

\$CONNUM = 1

THE CURRENT CONSOLE (INTEGER)

RETURN
BACK

PAGE
2

PAGE
3

PAGE
4

10715-A

7/98

Advanced Graphics Building—Discussion Material L53198.04

53

DDB VARIABLES, continued

Figure 30 System DDB Variables Page 2

				13 May 98 10:26:27		1
\$F2>SYS>\$SYSDDDB2		FGBR- 0, 2C- 0, 0N-ON				
SYSTEM DDB VARIABLES PAGE 2				SYS>\$SYSDDDB2		
BEFORE COMPILING YOU MUST LOAD A USER DB FILE: L ACT>AG						
ALL OF THESE VARIABLES EXCEPT \$KEYLEVL AND \$STRT_UP ARE READ ONLY.						
\$GRPBASE = AN INTERNAL NAME USED TO ACCESS POINT GROUP DATA.						
\$KEYLEVL = EEEE		AN ENUMERATION STATE OF THE KEYLOCK (VIEW,OPR,SUP,ENGR) YOU CAN STORE TO THIS VARIABLE USING THE S_SENM ACTOR.				
\$MY_AREA = II		THE NUMBER OF THE CURRENT AREA (INTEGER)				
\$MY_PNA = II		THE NUMBER OF THE CURRENT NODE (INTEGER)				
\$REDYEL = EEEEEEEE		AN ENUMERATION CONTAINING THE MINIMUM ALARM PRIORITY INDICATED BY THE COLOR RED (LOW,HIGH,EMERGNCY)				
\$STNNUM = II		THE STATION NUMBER OF THE US WITHIN THE CONSOLE (INTEGER)				
\$DT_ENTY = AN ENTITY ID WHICH CONTAINS THE NAME OF THE LAST POINT FOR WHICH A DETAIL DISPLAY WAS INVOKED. IT IS INITIALIZED TO NULL_ID.						
\$STRT_UP = BBB		THIS BOOLEAN VALUE IS SET TO FALSE WHEN THE US IS STARTED UP AND LOADED.				

DDB VARIABLES, continued

Figure 31 System DDB Variables Page 3

03 Sep 12:26:13 1

SYSTEM DDB VARIABLES PAGE 3 SYS>\$SYSDDB3

\$CZ_ENTY = RAMP331

THE POINT NAME USED BY A CHANGE ZONE. THE TAGNAME IS STORED HERE EITHER BY INVOKING THE STANDARD CHANGE ZONE USING THE ACTOR CHGZONE (CHG_ZONE) OR BY EXECUTING THE ACTOR USER_CZ. FOR MORE ON USER_CZ SELECT THE USER CHANGE ZONE TARGET AT THE BOTTOM.

\$CZ_ACT = OFF

ON IF THE STANDARD CHANGE ZONE HAS BEEN CALLED UP. IT WILL STAY ON UNTIL THE CLEAR_CZ ACTOR EXECUTES OR THE

CLR

 TARGET IS SELECTED IN THE CHANGE ZONE.

STANDARD
CHANGE ZONE

TARGET ACTION
CHGZONE(RAMP331,0)

CLEAR STD
CHANGE ZONE

TARGET ACTION
CLEAR_CZ

USER CHANGE
ZONE EXAMPLE

RETURN
MENU

PAGE
1

PAGE
2

PAGE
4

10717-A

DDB VARIABLES, continued

Figure 32 System DDB Variables Page 4

03 Sep 96 12:27:47 1

SYSTEM DDB VARIABLES PAGE 4SYS>\$SYSDDb4

ALL OF THESE VARIABLES ARE READ ONLY

\$ALMCLR = 1 AN INTEGER THAT SPECIFIES THE ALARM PRIORITY COLOR OPTION
SELECTED IN THE NCF; 0 = TWO COLOR OPTION (RED, YELLOW),
1 = 3 COLOR OPTION (USER SELECTED COLORS).

\$EALMCLR = 1 AN INTEGER THAT SPECIFIES THE EMERGENCY ALARM PRIORITY
COLOR OPTION SELECTED IN THE NCF. SEE TABLE A-5 IN THE
ACTORS MANUAL.

\$HALMCLR = 3 AN INTEGER THAT SPECIFIES THE HIGH ALARM PRIORITY COLOR
OPTION SELECTED IN THE NCF. SEE TABLE A-5 IN THE ACTORS
MANUAL.

\$FALMCLR = 6 AN INTEGER THAT SPECIFIES THE LOW ALARM PRIORITY COLOR
OPTION SELECTED IN THE NCF. SEE TABLE A-5 IN THE ACTORS
MANUAL.

\$AL_ENTY = AN ENTITY VARIABLE THAT PROVIDES THE NAME OF A POINT THAT IS SELEC-
TED ON THE ALARM SUMMARY, UNIT SUMMARY, ALARM ANNUNCIATION, OR
ORGANIZATIONAL SUMMARY DISPLAYS.

RETURN
MENU

PAGE
1

PAGE
2

PAGE
3

34572

DDB VARIABLES, continued

Figure 33 User Change Zone

05 Jan 10:23:48 6

USER CHANGE ZONE ACT>CHNGZONE

THIS USER CHANGE ZONE IS CONTROLLED BY A VARIANT WHICH TESTS BOOL10. BOOL10 IS SET ON WITH THE SELECT TARGET AND OFF WITH THE CLEAR TARGET IN THE CHANGE ZONE. IF YOUR SCHEMATIC IS COMPLEX WITH MANY VALUES, YOU SHOULD CONTROL THE CHANGE ZONE WITH THE "MULT_OV" ACTOR INSTEAD OF A VARIANT. USING A VARIANT IS LESS EFFICIENT BECAUSE EACH LIMB OR OPTION OF THE VARIANT IS EVALUATED EVERY CYCLE. THE MULT_OV ACTOR EXECUTES ONLY WHEN YOU SELECT THE TARGET.

\$CZ_ENTY = RAMP331 THE POINT NAME USED BY A CHANGE ZONE. THE TAGNAME IS STORED HERE EITHER BY INVOKING THE STANDARD CHANGE ZONE USING THE ACTOR CHGZONE (CHG_ZONE) OR BY EXECUTING THE ACTOR USER_CZ. THE BENEFITS OF USING USER_CZ ARE:

SELECT
USER
CHANGE ZONE

1. YOU CAN REFERENCE PARAMETERS BY USING \$CZ_ENTY.PV.
THIS IS FASTER ACCESS THAN USING ENT01 - ENT20G.

2. THE MAN, AUTO, NORM, SP, OP, AND RAMP BUTTONS CAN BE
USED WITH THE CUSTOM CHANGE ZONE.

RAMP331	SP	65.0	MAN	AUTO	CAS
ADV GRAPHICS BUILDING	PV	50.0	CLEAR		

10718

DDB VARIABLES, continued

Figure 34 Variant Using \$CZ_ENTY

02 Nov 14:56:32 6

NET>GENE>VARIANT2 FGBR- 0, 0C- 32, 128N-ON

Variant At 0, 0 Page 1 of 1

Subpicture Or Text For Bad Value "Bad Value"

Variant Body

IF \$CZ_ENTY.PTEXECST = INACTIVE THEN "INACTIVE" ELSE
IF \$CZ_ENTY.ALENBST = DISABLE THEN "DISABLED" ELSE
IF \$CZ_ENTY.ALENBST = INHIBIT THEN "INHIBITED" ELSE
IF \$CZ_ENTY.HIGHAL = NOALARM THEN "" ELSE S HIGHAL

<PAGE FWD> <PAGE BACK> to MOVE. <F2> for TFE. <F3> to JUMP. <F4> to DELETE.

M

Enter Variant Information

10719-A

Figure 35 Example of \$CZ_ENTY in an Actor

02 Nov 14:45:05 6

NET>GENE>TARGETS FGBR- 0, 0C- 304, 304N-ON

Target At 304, 304 Page 1 of 1

Solid/Box/Invisible Solid

Action

DETAIL(GS_STR(\$CZ_ENTY.NAME))

<PAGE FWD> <PAGE BACK> to MOVE. <F2> for TFE. <F3> to JUMP. <F4> to DELETE.

M

Enter Target Specifications

10720-A

DDB VARIABLES, continued

Figure 36 PE Print Example Showing Honeywell DDB

Print of Picture NET>GENE>VALUES1 November 01, 1993 21:49

```
VARIABLES
BEGIN
  VARIABLE    1
  NAME       : INT01
  TYPE       : INTEGER
  COLLECTION RATE:    1
  GROUP ID   :    0
  DDB LOCATION : -5001
  DDB REFERENCE : LOCAL DDB

  VARIABLE    2
  NAME       : INT20G
  TYPE       : INTEGER
  COLLECTION RATE:    1
  GROUP ID   :    0
  DDB LOCATION : -5020
  DDB REFERENCE : GLOBAL DDB

  VARIABLE    3
  NAME       : INT50
  TYPE       : INTEGER
  COLLECTION RATE:    1
  GROUP ID   :    0
  DDB LOCATION : 101
  DDB REFERENCE : GLOBAL HW USER DDB

  VARIABLE    4
  NAME       : INT51
  TYPE       : INTEGER
  COLLECTION RATE:    1
  GROUP ID   :    0
  DDB LOCATION : 102
  DDB REFERENCE : GLOBAL HW USER DDB
```

DDB VARIABLES, continued

Figure 36 PE Print Example Showing Honeywell DDB,
continued

```
VARIABLE    5
NAME        : REAL50
TYPE        : REAL
COLLECTION RATE:    1
GROUP ID    :    0
DDB LOCATION :    101
DDB REFERENCE : GLOBAL HW USER DDB
```

```
VARIABLE    6
NAME        : REAL51
TYPE        : REAL
COLLECTION RATE:    1
GROUP ID    :    0
DDB LOCATION :    102
DDB REFERENCE : GLOBAL HW USER DDB
```

```
END
```


RETURN TO SPECIFIC SCHEMATIC

Figure 37 Different Ways to Return to a Specific Schematic

02 May 13:56:30 6

DIFFERENT WAYS TO RETURN TO A SPECIFIC SCHEMATIC GENP>RETURNS1

RIGID ALWAYS RETURN TO THE SAME PLACE.

TARGET TARGET
SCHEM("MOOSE") OR SCHEM("GENPMENU")

FLEXIBLE

1. STORE THE NAME OF A SCHEMATIC IN A GLOBAL US DDB OR USER DDB.
THIS IS WITHIN THE UNIVERSAL STATION'S CONTROL.

TARGET TARGET
S_STR(STR01G,"MOOSE") OR S_STR(STR01G,"GENPMENU")

WHEN YOU ARE READY TO RETURN, USE THE STORED NAME.
YOU MUST FIRST "GET" THE STRING NAME.

TARGET
SCHEM(G_STR(STR01G)) STR01G = MOOSE

2. STORE THE NAME OF A SCHEMATIC IN A CUSTOM STRING PARAMETER.
THIS IS A SYSTEM WIDE NAME THAT COULD BE CHANGED BY ANYBODY.

TARGET TARGET
SS_STR(GENERIC.RETURN,"MOOSE") OR SS_STR(GENERIC.RETURN,"GENPMENU")

WHEN YOU ARE READY TO RETURN, USE THE STORED NAME.

TARGET
SCHEM(GS_STR(GENERIC.RETURN)) GENERIC.RETURN = GENPMENU

10721

RETURN TO SPECIFIC SCHEMATIC, continued

Figure 38 Schematic Return Using an Indexed Custom Parameter

```
04 Jan 11:53:38 6

SCHEMATIC RETURN USING AN INDEXED CUSTOM PARAMETER      GENP>RETURNS2

IF YOU HAVE A CUSTOM ARRAY PARAMETER THEN YOU CAN USE INDEXING TO CONTROL THE
RETURN SCHEMATIC. WE HAVE A CUSTOM STRING ARRAY NAME_ARR(1) ... NAME_ARR(6).
THIS COULD BE PREBUILT TO CONTAIN THE SCHEMATIC NAMES. IN THIS CASE, WE WILL
INITIALIZE THE ARRAY HERE. THEN BY SETTING AN INDEX WE CAN CHOOSE WHICH
SCHEMATIC TO RETURN TO. IF THE VALUE OF INT01 IS 0 THEN YOU WILL SEE AN ERROR:
INVALID DATA TYPE.

TARGET TO INITIALIZE NAME_ARR
SS_STR(GENERIC.NAME_ARR(1),"MOOSE");
SS_STR(GENERIC.NAME_ARR(2),"GENPMENU");
SS_STR(GENERIC.NAME_ARR(3),"RETURNS");
SS_STR(GENERIC.NAME_ARR(4),"ACTMENU");
SS_STR(GENERIC.NAME_ARR(5),"MOOSE");
SS_STR(GENERIC.NAME_ARR(6),"FUNMENU")

TARGETS TO SET INDEX
1 S_INT(INT01,1)
2 S_INT(INT01,2)
3 S_INT(INT01,3)
4 S_INT(INT01,4)
5 S_INT(INT01,5)
6 S_INT(INT01,6)

TARGET FOR A VARIABLE RETURN
SCHEM(GS_STR(C_VAR(GENERIC,0,.NAME_ARR,G_INT(INT01)))) INT01 = 2
```

10722

PE COMPILER ERROR MESSAGES

Figure 39 Local Name Type Expected

02 Nov 15:35:13 6

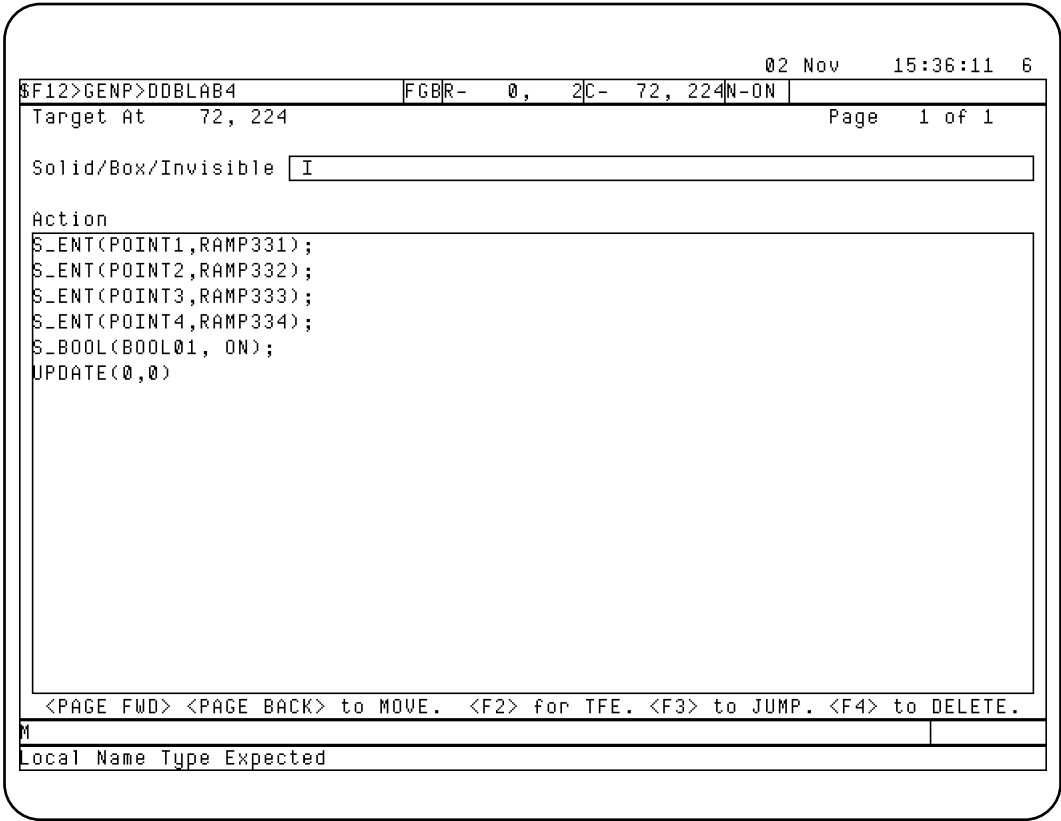
PICTURE EDITOR	FGDR- 0, 0C- 104, 256N-ON
Target At 56, 288	Page 1 of 1
Solid/Box/Invisible <input type="text" value="Solid"/>	
Action	
<div>S_REAL(RAMP331.SP,12.0)</div>	
<PAGE FWD> <PAGE BACK> to MOVE. <F2> for TFE. <F3> to JUMP. <F4> to DELETE.	
A TAR	
Local Name Type Expected	

10723

See the module “Debug Custom Displays,” Appendix A, for other compile-time error messages.

PE COMPILER ERROR MESSAGES, continued

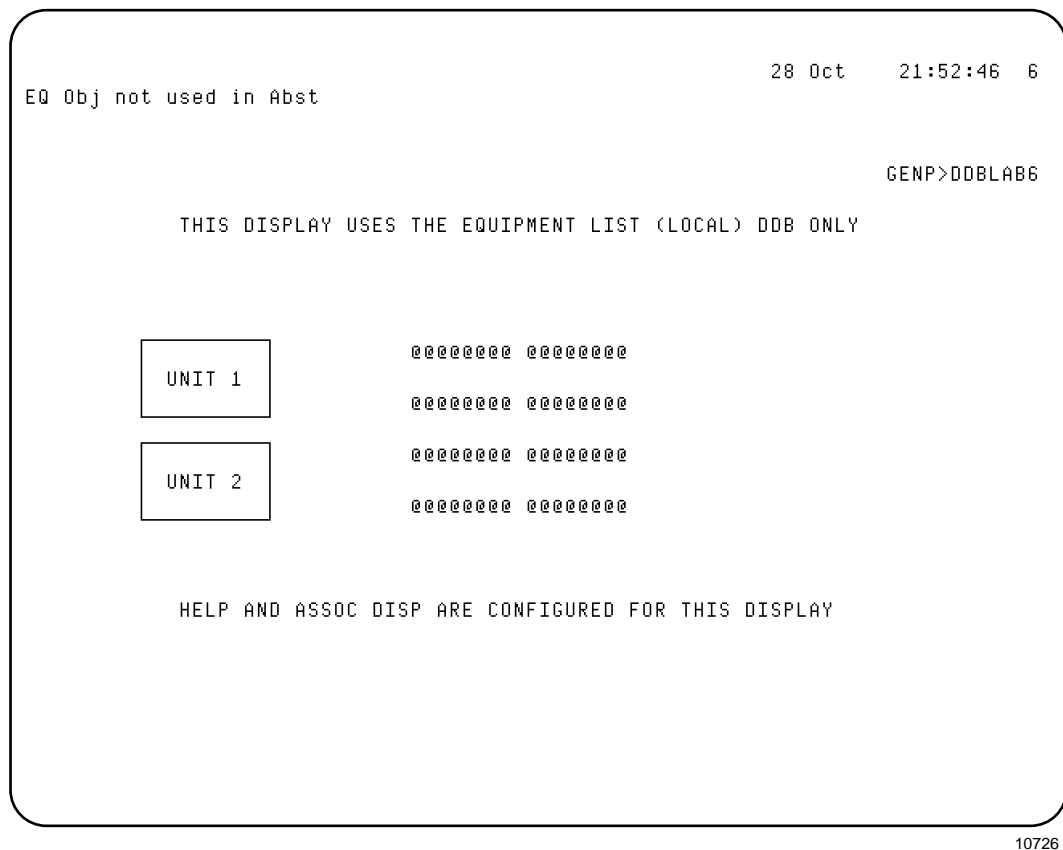
Figure 40 Local Name Type Expected



10724

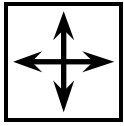
PE RUN-TIME ERROR MESSAGES

Figure 41 EQ Object Not Used in Abstract



See Appendix B in the module “Debug Custom Displays” for other run-time error messages.

Directions



DIRECTIONS—This is the end of this discussion material.

Last Page

