Honeywell

Use Picture Editor Slick Tricks

L5212

AG

TotalPlant®

Copyright, Notices, and Trademarks

Revision 03 – April 21, 1998

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

FLEXTRAINING and **TotalPlant** are US registered 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

Figures and Tables

Acronyms

AM ..... Application Module
APM ..... Advanced Process Manager
CDS ..... Custom Data Segment
DDB ..... Display Database
IEEE ..... Institute of Electrical and Electronics Engineers
LCN ..... Local Control Network
PE ..... Picture Editor
PM ..... Process Manager
TDC ..... Total Distributed Control
UCN ..... Universal Control Network
US ..... Universal Station
UWS ..... Universal Work Station
U^XS ..... Universal Station^X

Parameters

NaN ..... Not a Number
PV ..... Process Variable
PVEUHI ..... Process Variable High Engineering Value
PVEULO ..... Process Variable Low Engineering Value
PVFORMAT ..... Display Format
PVHITP ..... Process Variable High Trip Point
SP ..... Setpoint

References

Publication Title	Publication Number	Binder Title	Binder Number
Picture Editor Reference Manual	SW09-550	Implementation/Engineering Operations-2	TPS 3032-2
Actor's Manual	SW09-555	Implementation/Engineering Operations-2	TPS 3032-2

Introduction

Module Overview

About this module

This course module describes ways to accomplish certain Picture Editor tasks using “slick tricks” discovered by various TPS users.

Objectives

Given a Universal Station, the Picture Editor manuals, and this course module, be able to perform the following Picture Editor “slick tricks”:

- use text editing keys, such as [CTL] [DEL LINE] and [INSERT LINE] to modify text in a entry port of targets, conditions, or variants,
- use the Text File Editor to write the entry port of a target, variant, or condition to a text file, then read the contents of the text file into a different target, variant, or condition.
- create a target that uses the Combine Date/Time actor to store to either a date or a time,
- create a “tick mark” using the Add Bar command,
- create a target, variant, or condition that tests for a bad value,
- create a target that performs a series of keystrokes,
- create a subpicture that determines the PV format (decimal place) of a value when it is displayed, and
- create a target that types default data into an entry port displayed to the operator.

Test items

1. Build a custom display that demonstrates *at least three* slick tricks:
   a. target that uses the Combine Date/Time actor to store either a date or time,
   b. “tick mark” by using Add Bar command (The tick mark should represent one value in relation to another.),
   c. target, variant, or condition that tests for NaN,
   d. target that performs a keystroke macro,
   e. subpicture that displays a PV value in its currently configured PVFORMAT, and
   f. target that puts text or values into an input port displayed to the operator

   When you have completed your display, demonstrate to your course manager that it successfully performs as defined.

Text Editing Keys

Description

When adding or modifying Values, Conditions, Targets, or Variants, several text editing keys can be used to significantly improve your productivity.

The main concept associated with the editing keys is a *capture buffer* that automatically saves text for later use. For example, text can be marked, deleted, then reinserted.

The contents of the capture buffer remain saved as long as you remain in the Picture Editor. This means blocks of text can be "borrowed" from one custom display and pasted into another, for example.

Text editing keys

Table 1 describes how the text editing keys work.

Table 1 Text Editing Keys

IF you press...	THEN...
[CTL] [DEL LINE]	Marks the beginning of a block of text.
[DEL LINE]	**If…** / **Then this key…** Beginning of text is marked, / Marks end and deletes block of text and saves it into capture buffer. No mark at beginning of text, / Deletes line and saves it into capture buffer.(no matter where cursor is)
[CLR ENT]	Deletes the entire text input port, such as a Target expression, and saves it into the capture buffer.
[INS LINE]	Inserts the capture buffer contents into the text input port (before the cursor).

CAUTION

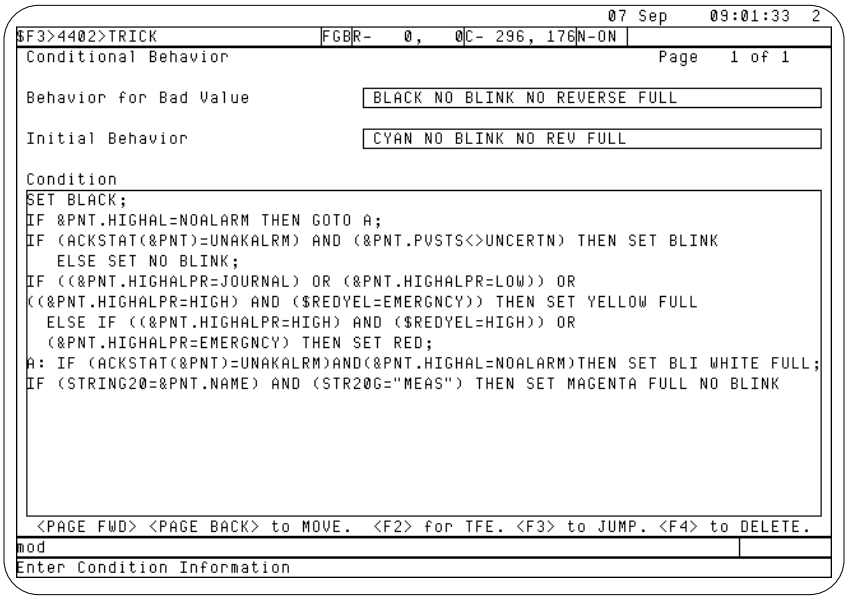
CAUTION—You can use the Add Text command to paste the capture buffer as text, but hidden characters such as carriage returns can do strange things to other display objects.

Write PE Entry Port Data To a Text File,

Procedure Perform the procedure in Table 2 to write entry port data of a Target, Variant, or Condition, then retrieve it for use in another display.

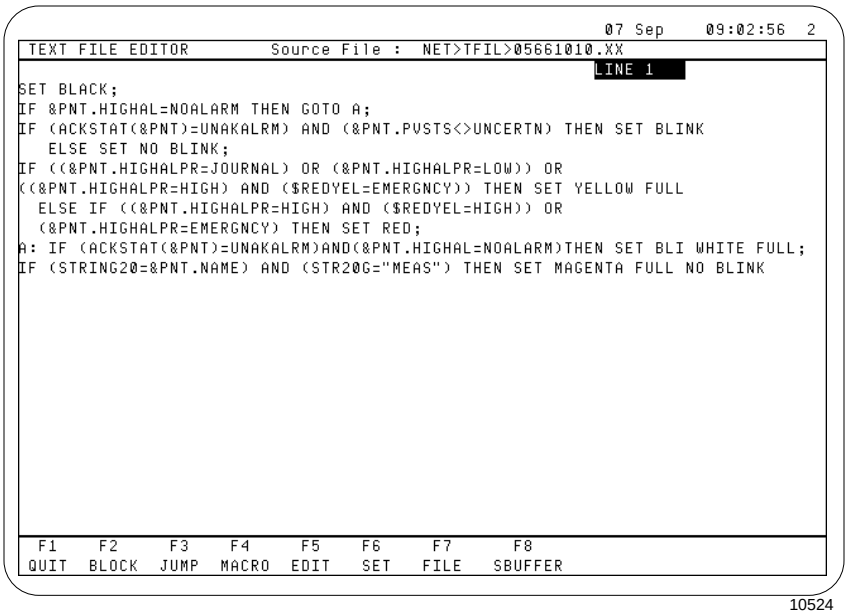
Table 2 Procedure to Write PE Entry Port Data

Step	Action
1	In the Picture Editor, display the entry port that you want to save, then press [CTL] [F2].



10523

RESULT: The entry port data is copied into the Text File Editor (see below).



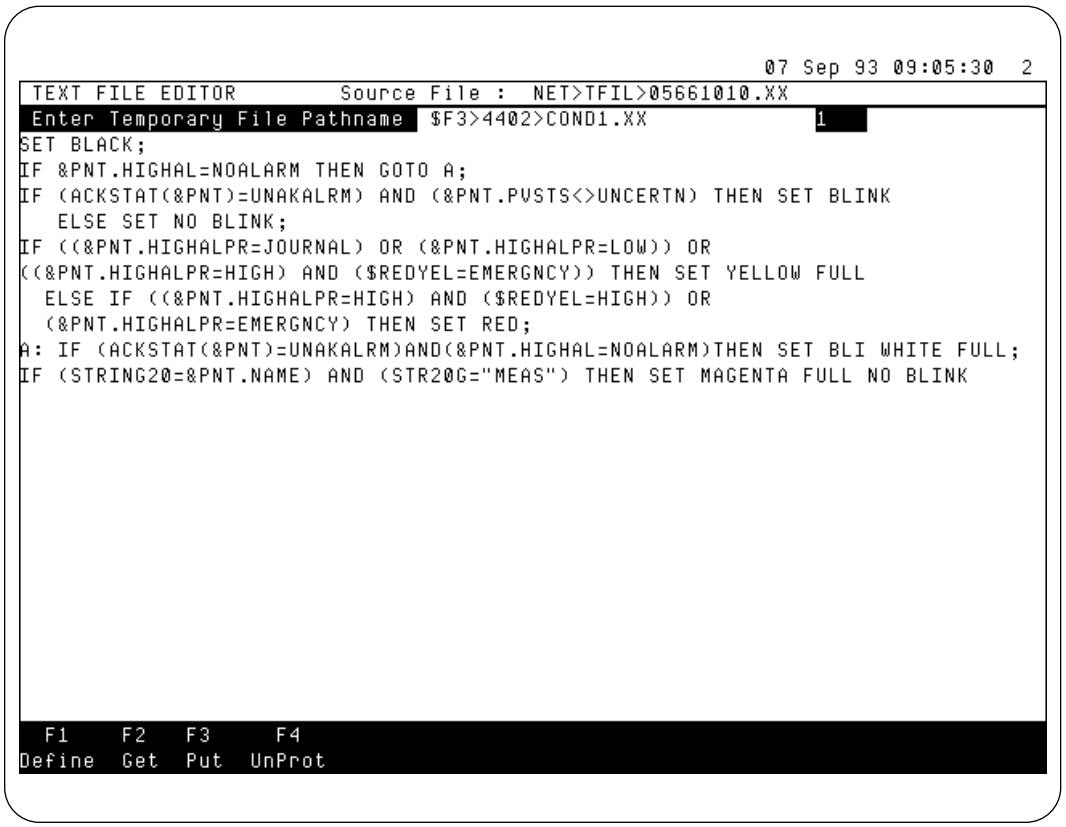
10524

2	From the Text File Editor, press [CTL] [F7]. RESULT: The FILE commands appear at the bottom of the screen. F1 Define F2 Get F3 Put F4 UnProt

Continued on next page

Procedure (continued)

Table 2 Procedure to Write PE Entry Port Data, *continued*

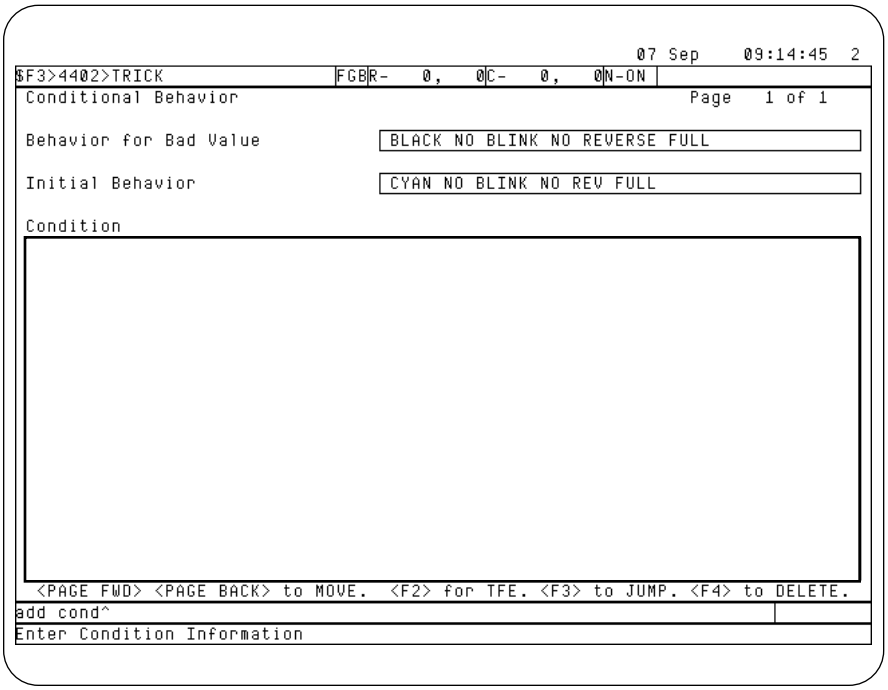
Step	Action
3	Press [CTL] [F1] to define the file. RESULT: Message appears ENTER TEMPORARY FILE PATHNAME  10525
4	Type in the pathname of the text file to which you want to write the data, then press [ENTER]. RESULT: Entry is accepted.
5	Press [CTL] [F3] to PUT the data into the file. RESULT: Message appears HOW MANY LINES TO OUTPUT?
6	Enter a number equivalent to the number of lines you want to write to the named text file. HINT: Enter 999 to ensure all lines are output. RESULT: Message appears Lines written = n to (pathname).

Read Text File Into PE Entry Port

Procedure

Perform the procedure in Table 3 to read a text file into a Picture Editor entry port.

Table 3 Procedure to Read a Text File Into a PE Entry Port

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

Accessing Parameters of the Data-Type TIME

Description

Many TPS parameters are of the data-type Time.

A Time-type parameter has two fields: Date and Time.

Because a Time-type parameter is actually a 32-bit word containing an integer representing the number of seconds since 01-01-79 (the birthday of TDC 3000^X), *a store to either date or time always results in a store to the other, unless you use the Combine Date/Time actor* (C_DATTIM).

Accessing current time

The Picture Editor Collector that you use to fetch the *current* Date/Time is SYS_TIME.

To Get or Store to current time, you must add it to the custom display by using ADD VALUE, before referencing it with a Target actor.

To prevent the current time value from impacting display efficiency, SET its COLLECTION rate to zero (update at invocation only).

If you don't want to see the current time value, add it in the color BLACK.

Actors used

Table 4 lists the actors used to access Time type parameters.

Table 4 Actors That Access Time and Date

Actor	Description
GS_DATIM SS_TIME SS_DATE	Get from or store to a system parameter such as a Custom Data Segment of type Time.
G_DATIME S_TIME S_DATE	Get from or store to a DDB date/time variable such as DATIME1G.
R_DATE R_TIME	Read a date or time from a Text Input Port.

Continued on next page

Accessing Parameters of the Data Type TIME, Continued

Actor examples

Two frequently used parameters of type Time that are available to the Picture Editor are

- Custom Data Segment (CDS) parameters (as in the Application Module), and
- Display Database (DDB) variables:
  - DATIME1 - DATIME4, and
  - DATIME1G - DATIME4G.

Listed below are some examples using Date/Time actors *(Remember, to read or store to Date and Time independently, you must use the Combine Date/Time actor.):*

Example 1:

Read a time (or date) from a Text Input Port and store it to the CDS parameter PT.CDS, without affecting the date (or time):

```
SS_DATE(PT.CDS,C_DATTIM(GS_DATIM(PT.CDS),
R_TIME(0,0,10,"ENTER TIME",TRUE,0));
```

Example 2:

Store system date and time to the DDB parameter DATIME1G:

```
S_DATE(DATIME1G, G_DATIME(SYS_TIME));
```

Example 3:

Read a date from a Text Input Port and store it to the DDB parameter DATIME4:

```
S_DATE(DATIME4, R_DATE(0,0,10,"ENTER DATE",TRUE,0));
```

Example 4:

Read a time from a Text Input Port and store it to the CDS parameter PT.CDS:

```
SS_TIME(PT.CDS, R_TIME(0,0,10,"ENTER TIME",TRUE,0));
```

Example 5:

Read both a time and date from Text Input Ports, and store both to the DDB parameter DATIME1:

```
S_DATE(DATIME1, C_DATTIM(R_DATE(0,0,10,"ENTER
DATE",TRUE,0)), R_TIME(0,0,10,"ENTER TIME",TRUE,0));
```

Creating Tick marks

Description

To add a “tick mark” next to a bar graph (similar to the setpoint mark on the Detail display of a controller point), simply add another bar graph that has its origin equal to its value (expression); that is, the same parameter. This works for both vertical and horizontal bars.

Examples

Listed below are examples of the entries for tick marks:

Setpoint example:

ADD BAR Text Input Port	Example
Expression	Point Name.SP
Solid/Hollow Bar	Solid
Vert/Horiz Bar	Vertical
Left/Bottom Bound	0.0
Right/Top Bound	100.0
Origin	Point Name.SP

High Alarm example:

ADD BAR Text Input Port	Example
Expression	Point Name.PVHITP
Solid/Hollow Bar	Solid
Vert/Horiz Bar	Vertical
Left/Bottom Bound	Point Name.PVEULO
Right/Top Bound	Point Name.PVEUHI
Origin	Point Name.PVHITP

Testing for a NaN

Description

Parameter values of type Real (such as a controller's PV) might occasionally be uninterpretable (such as when a transmitter output is out of range).

When a value is uninterpretable, the display shows dashes (- - - - - - - -) in the value field.

TPS uses the IEEE standard real number data representation for values that are uninterpretable, normally referred to as Not a Number (NaN).

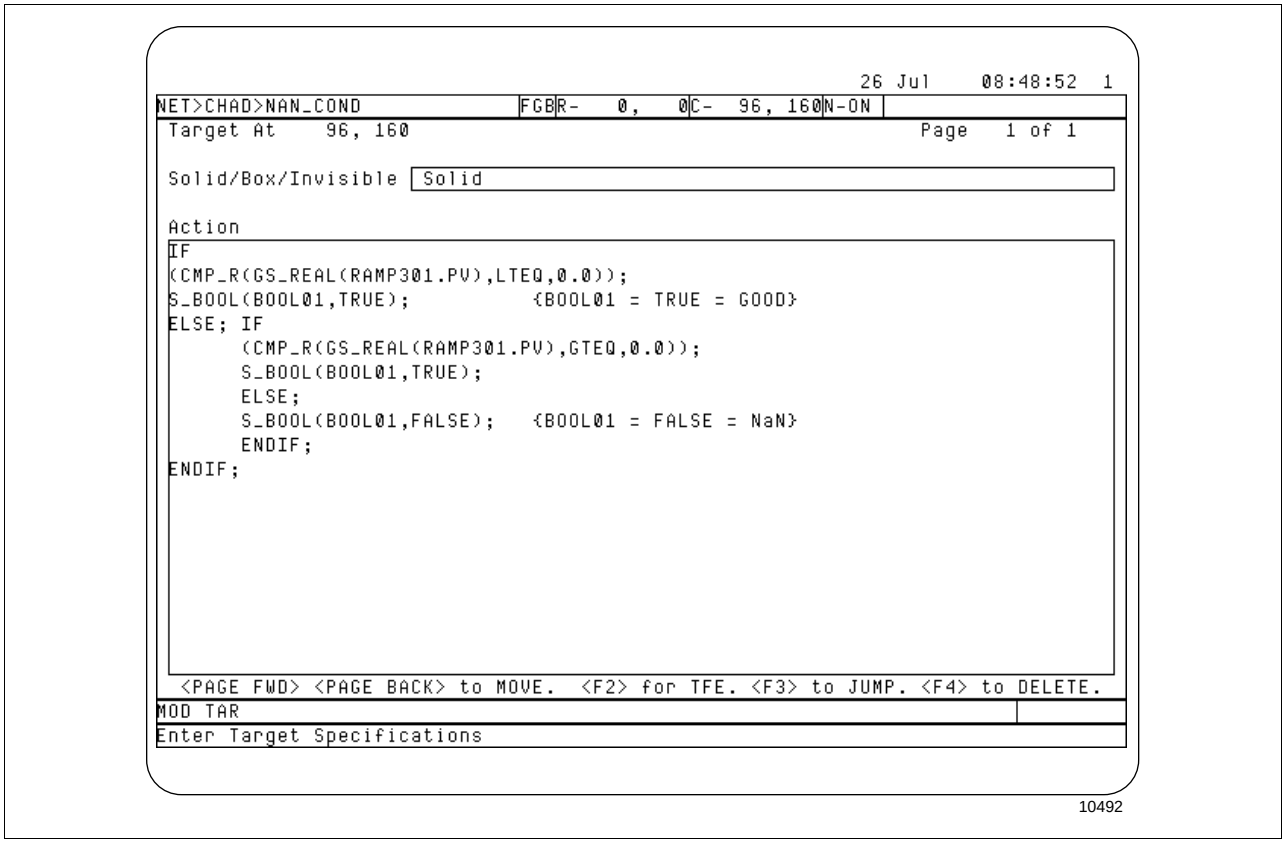
While the exact 32/64 bit representations for NaN are unimportant to us when building custom displays, we might want to test a real number to see if it is NaN. This is possible.

Using a target

To test a real number, test to see that the real number is not greater than or equal to zero and not less than or equal to zero. If it passes both tests, it is Not a Number (NaN). It might help to think of NaN as being equal to -0.0.

Figure 1 is an example of a Target expression that tests for NaN.

Figure 1 Target Test for NaN



Continued on next page

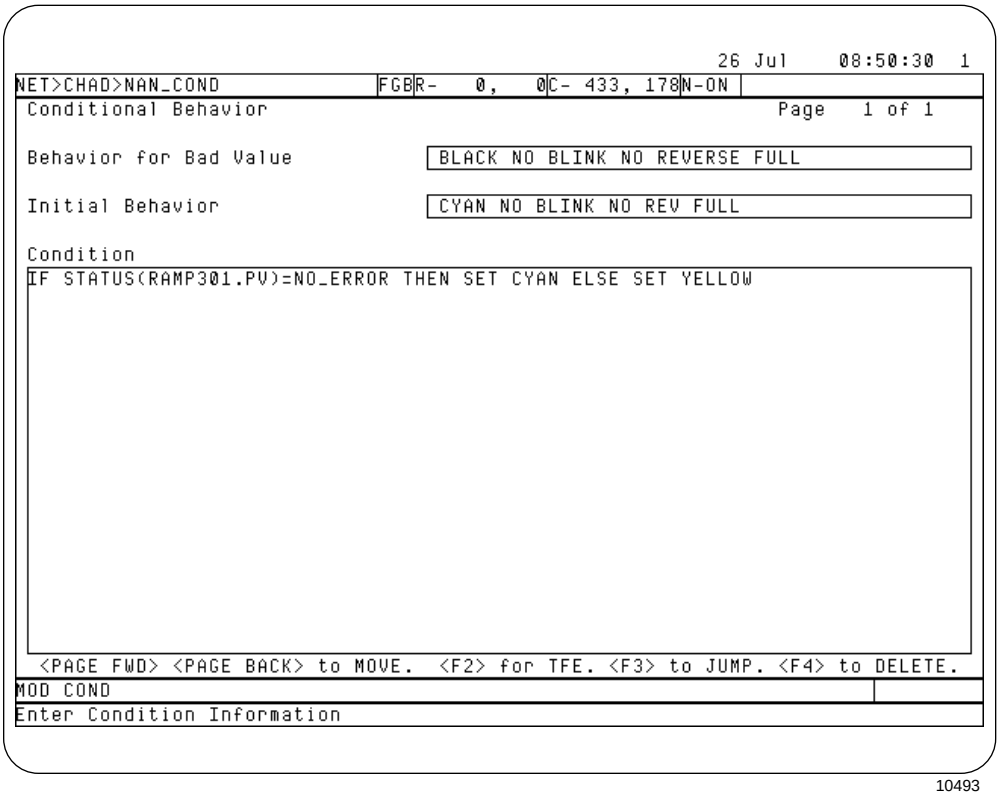
Testing for a NaN, Continued

Using a Value, Condition, or Variant

To test for NaN in a Value, Condition, or Variant, use the intrinsic function, STATUS. Appendix C.2 of the *Picture Editor Reference Manual* describes the STATUS function.

Figure 2 is an example of a Condition expression that tests for NaN.

Figure 2 Test for NaN with Value, Condition, or Variant



Statuses returned

Table 5 lists the statuses returned by the STATUS function.

Table 5 Statuses Returned by STATUS

Status Returned	Description
NO_ERROR	Value is O.K.
VAL_ERR	**Value is NaN**
COMM_ERR	Communication Error
CONF_ERR	Configuration Error

Creating Keystroke Macros

Description

Automating repetitive functions that require an exact series of keystrokes is possible using Target actions and Configurable Button actions.

These actions can be “strung” together serially when the magnitude of the job requires it. This allows you, in a sense, to write custom programs making common procedures more efficient.

Actors

Table 6 lists the primary actors used to accomplish a series of keystrokes.

Table 6 Actors Used To Simulate Keystrokes

Actor	Description
MOVE	Moves the cursor to a target.
KEY	Simulates a keystroke on a specified station (Tab and Select keys are very useful). Also simulates a keystroke on another US in the same console.
QUE_KEY	Simulates pushing a configurable button. In Target actions, KEY is preferred.
REM_MOVE	Moves the cursor on another US in the same console (remote move).
REM_STR	Sends a character string to another US screen in the same console (remote string).

Buttons

Configurable buttons can be used in conjunction with custom displays to automate a procedure.

The Button Configurator has nearly the same capability to use the keystroke actors as the Picture Editor. Pushing configurable buttons can be simulated with the actor QUE_KEY (button name).

Continued on next page

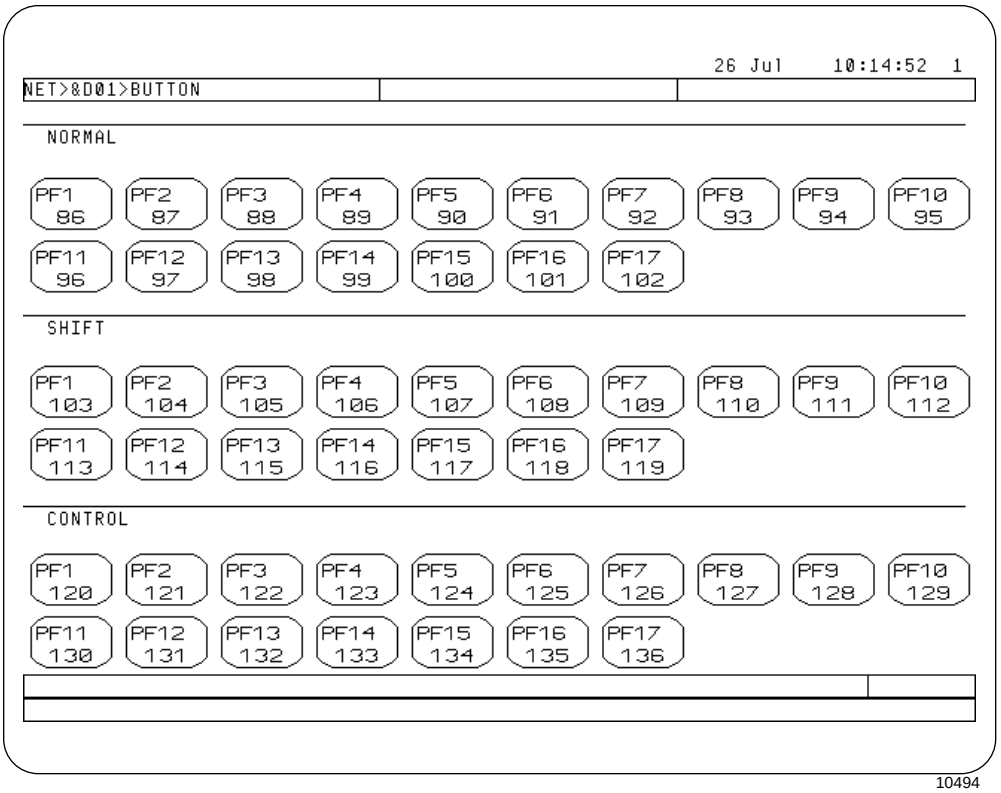
PF1 - PF17

There are 17 programmable function keys (PF1 - PF17) that are available through the Button Configurator (see Figure 3), but do not physically exist on a Universal Station (they exist only on a Universal Work Station (UWS) and U^{X}S).

Although these keys do not physically exist on a US, they can be configured through the Button Configurator and “pressed” through the action of a Custom Display!

Used in conjunction with the [SHIFT] or [CONTROL] keys the total number of programmable function keys available to a US is increased to 51.

Figure 3 Button Configuration - PF Buttons

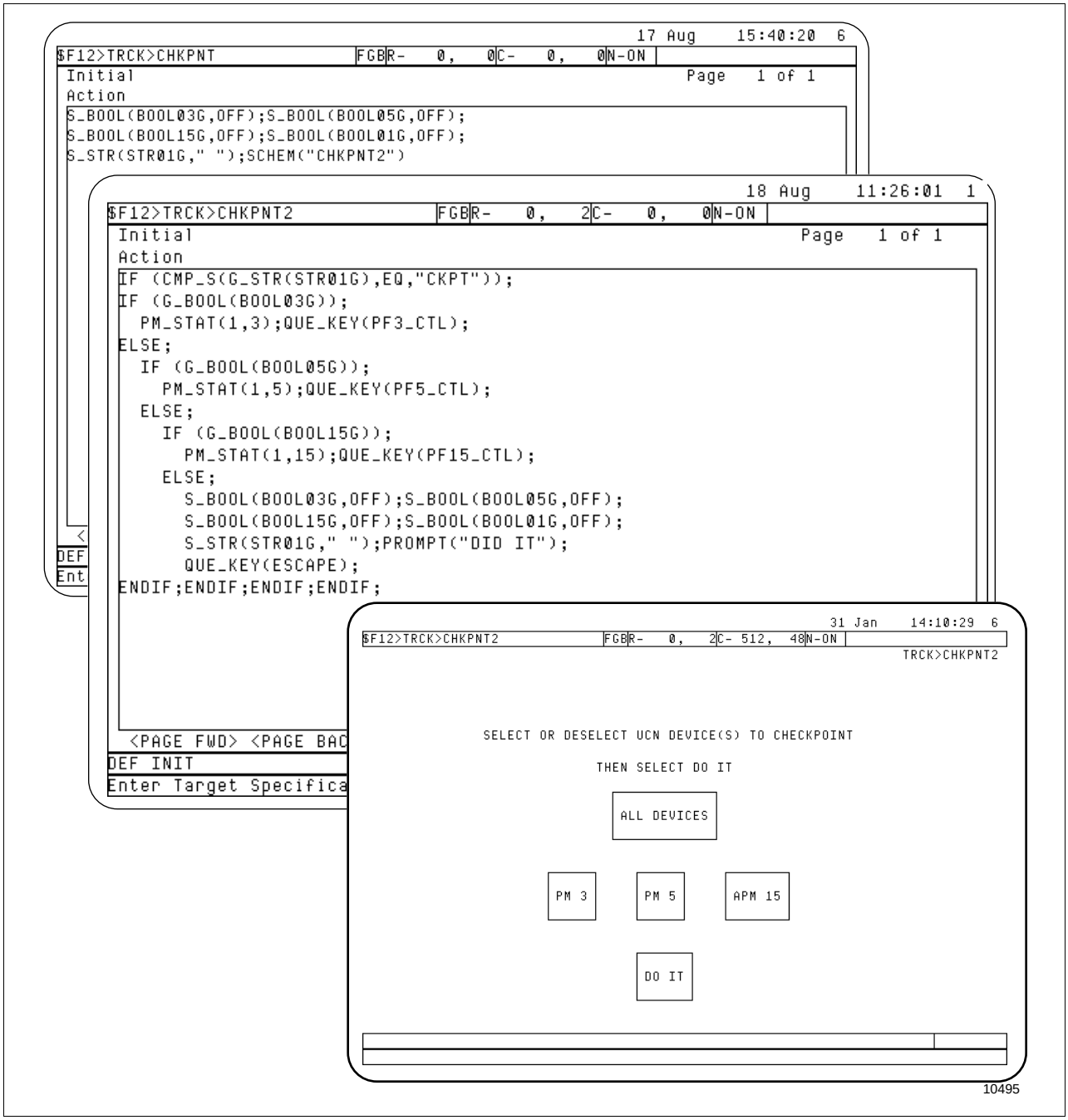


Continued on next page

Example

The display shown at the bottom of Figure 4 (CHKPNT**2**) uses a *combination of targets and configurable buttons* to checkpoint UCN devices. To checkpoint UCN devices, the operator would call up display CHKPNT, which is a display whose only purpose is to initialize global DDB variables, then call up display CHKPNT**2**. The top of Figure 4 shows the Initial Action of both displays.

Figure 4 INITIAL ACTION- Checkpointing UCN



Continued on next page

Example (continued)

Figures 5 through 8 illustrate the *targets* of the custom display:

- Figure 5—Target to Checkpoint All Devices
- Figure 6—Target to Checkpoint APM 15
- Figure 7—Target to Checkpoint PM 3
- Figure 8—Target to DO IT

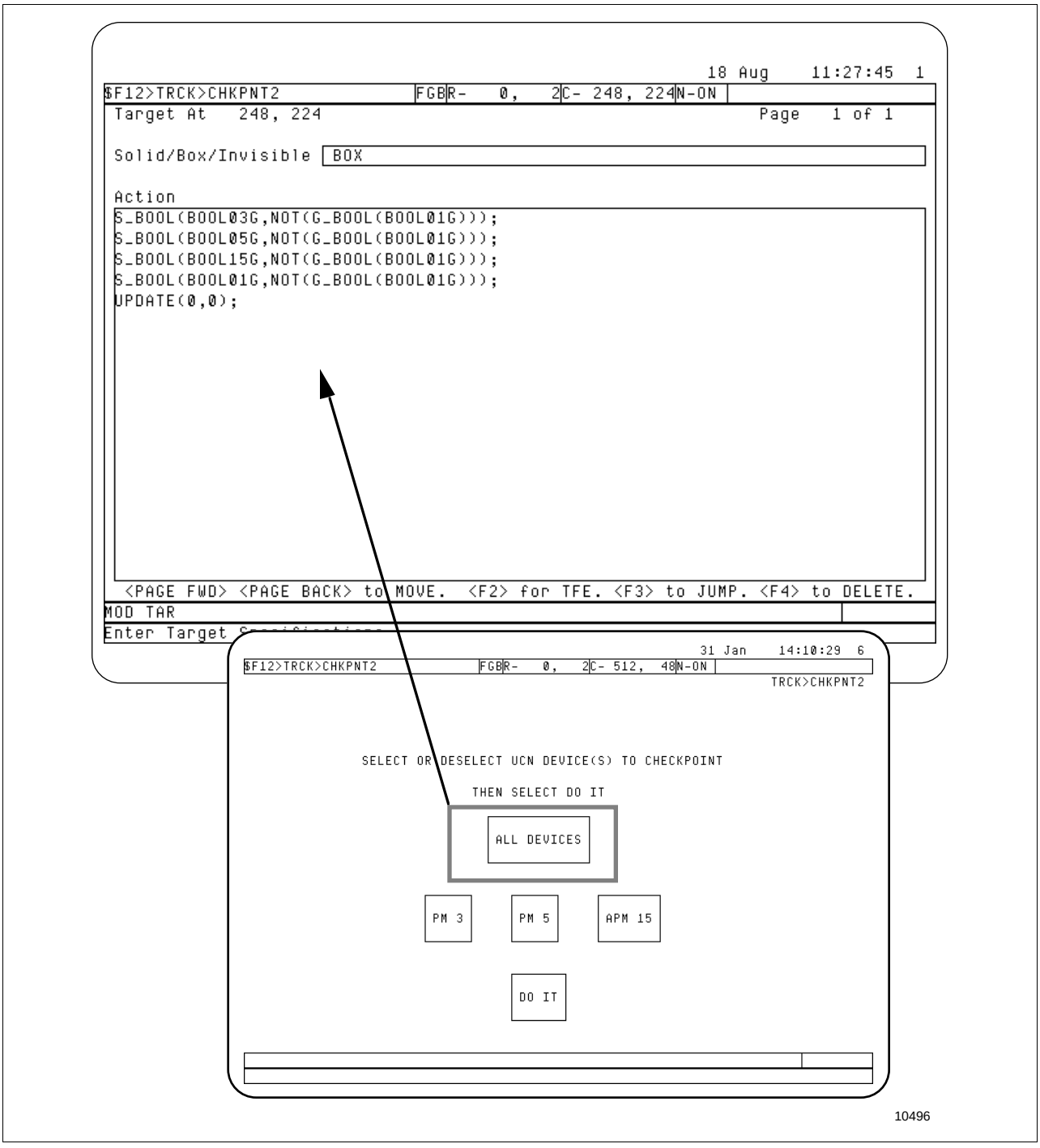
Figures 9 through 12 illustrate the configuration of the buttons that the actors call:

- Figure 9—Button [CTL] [PF3] and [CTL] [PF5]
- Figure 10—Button [CTL] [PF11] and [CTL] [PF12]
- Figure 11—Button [CTL] [PF13] and [CTL] [PF15]

Take some time to analyze the Actor ":strings" in Figures 5 through 11, and you will see how powerful "keystroke macros" are.

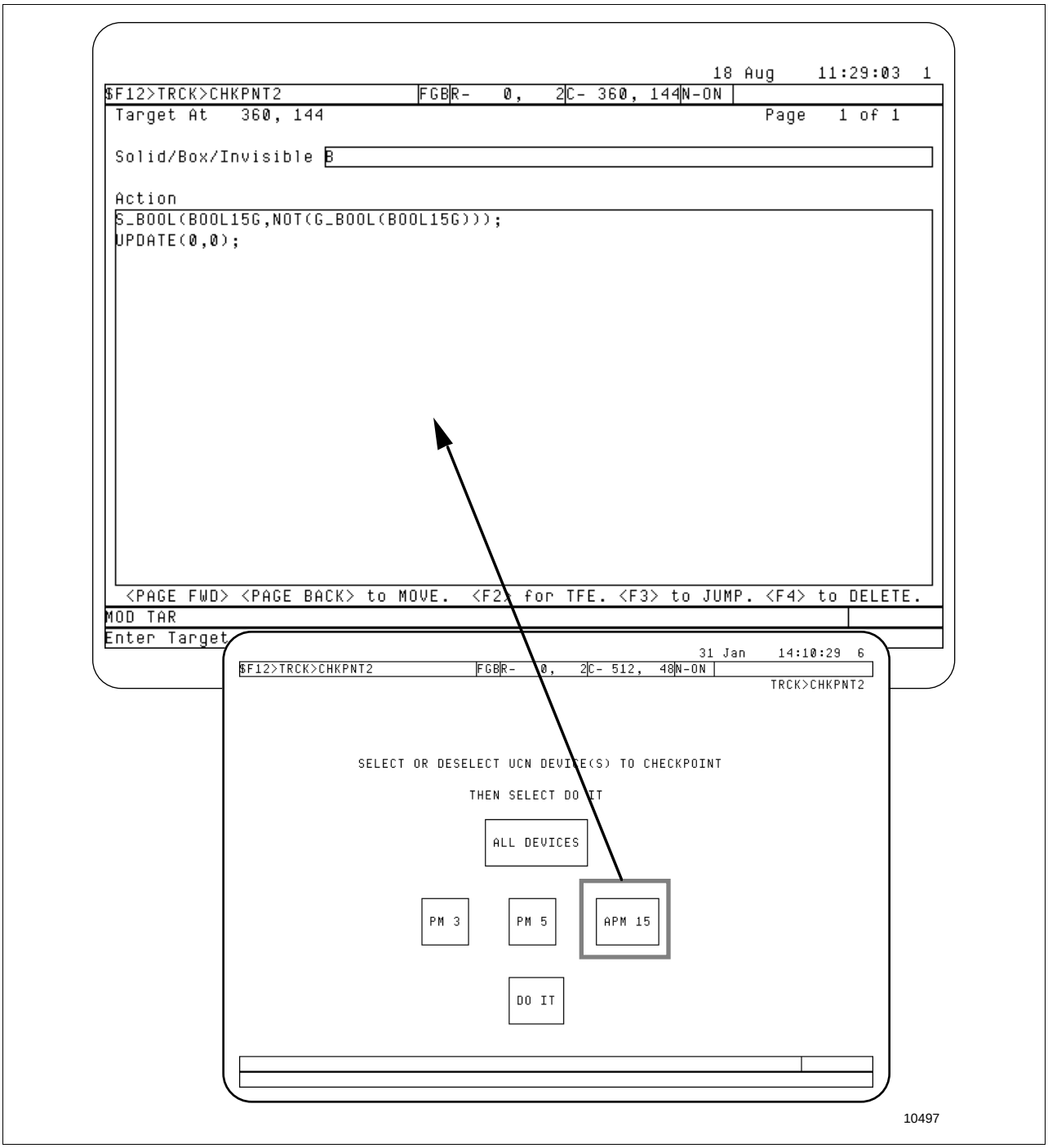
Continued on next page

Figure 5 Target To Checkpoint All Devices



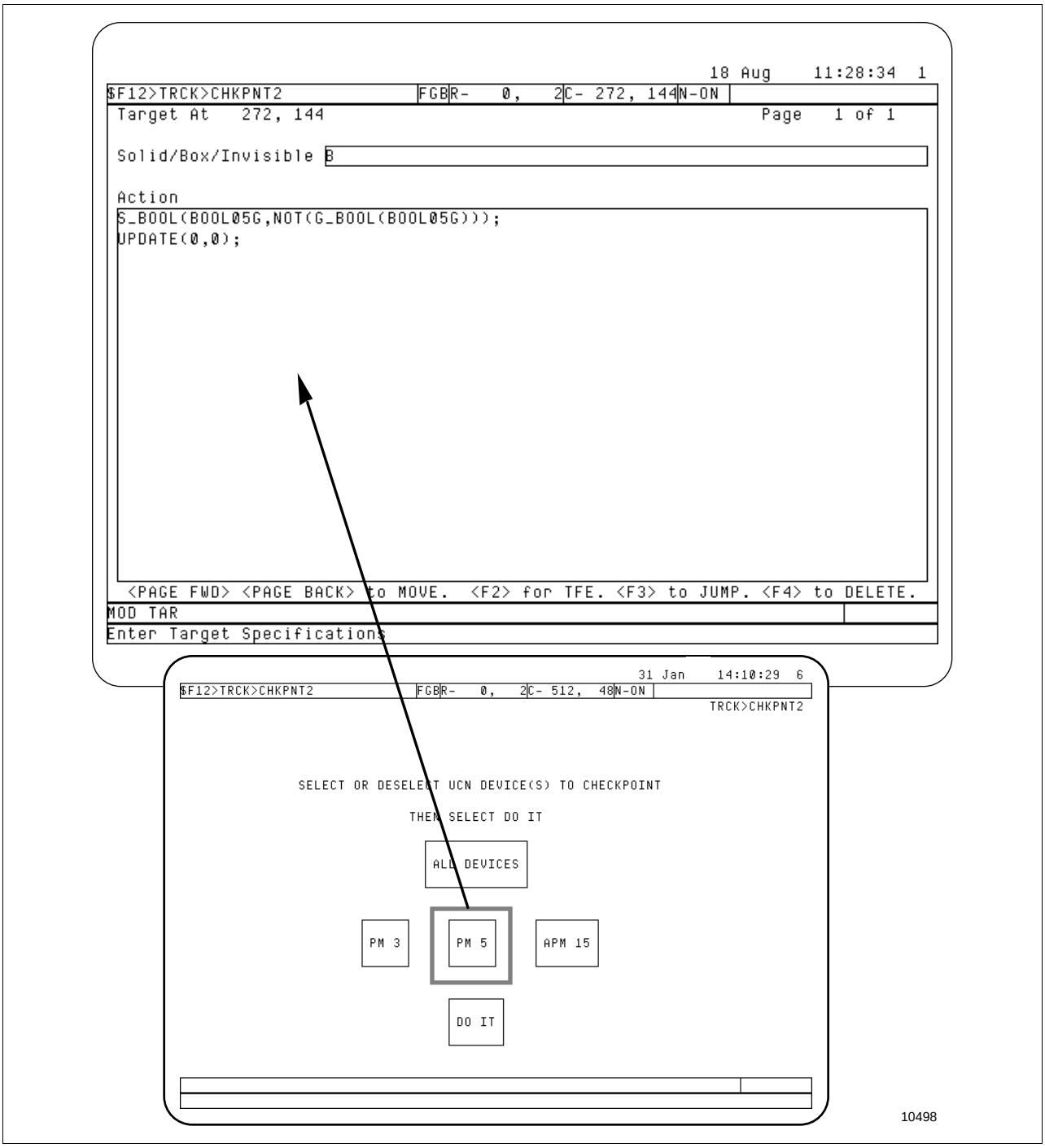
Continued on next page

Figure 6 Target To Checkpoint APM 15



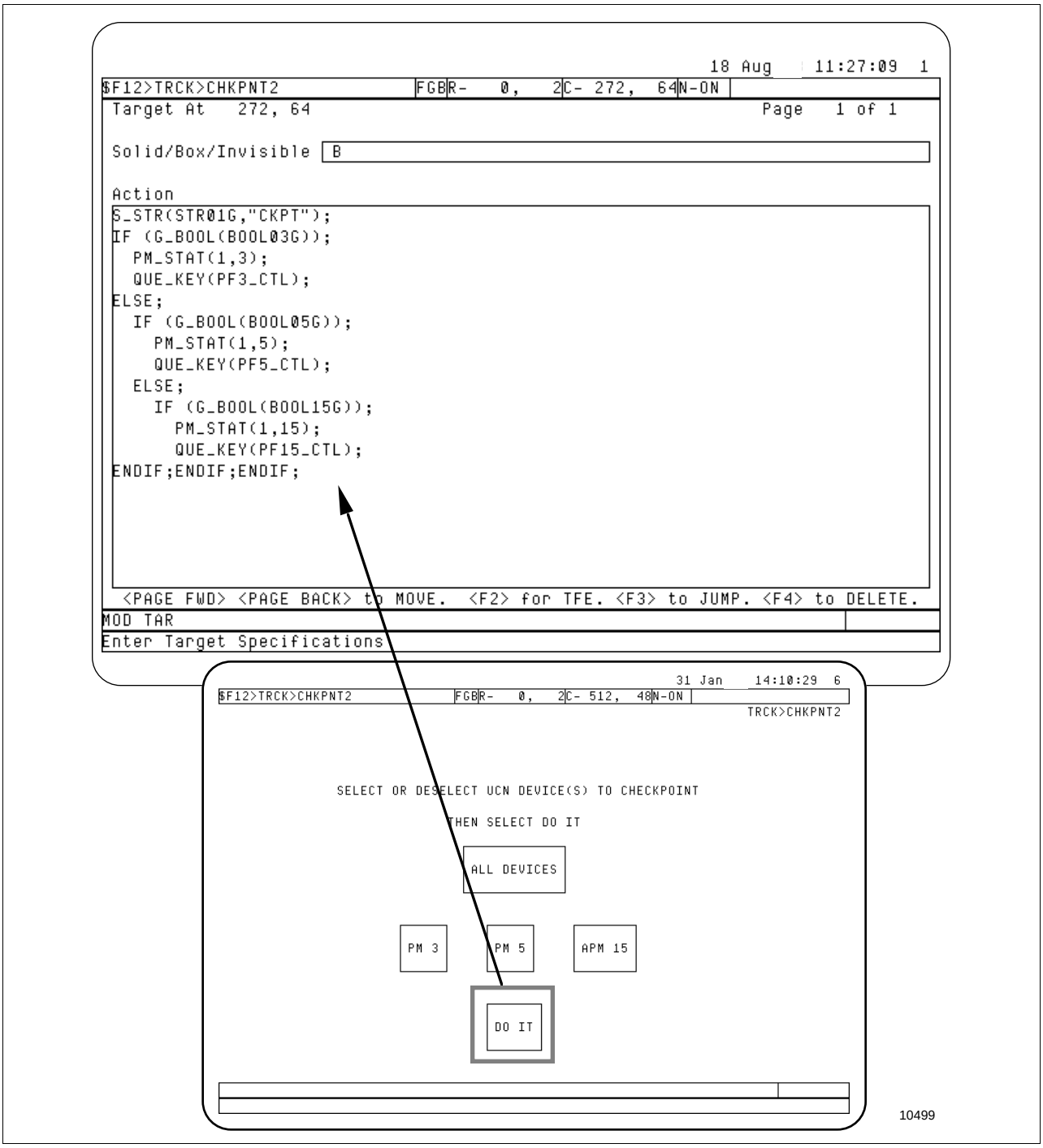
Continued on next page

Figure 7 Target To Checkpoint PM 5



Continued on next page

Figure 8 Target To DO IT

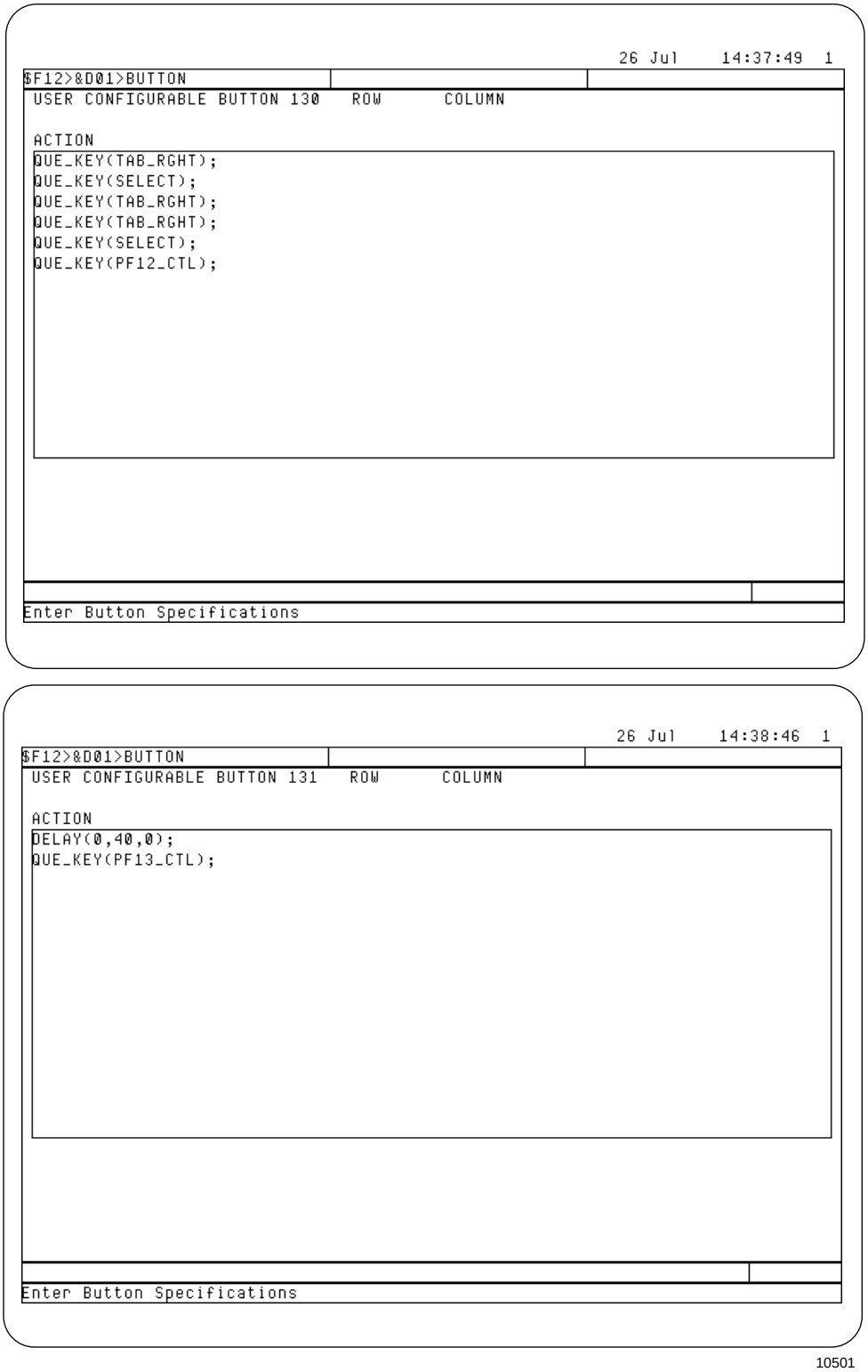


Continued on next page

Figure 9 CTL PF3(122) and CTL PF5(124) Button Configuration

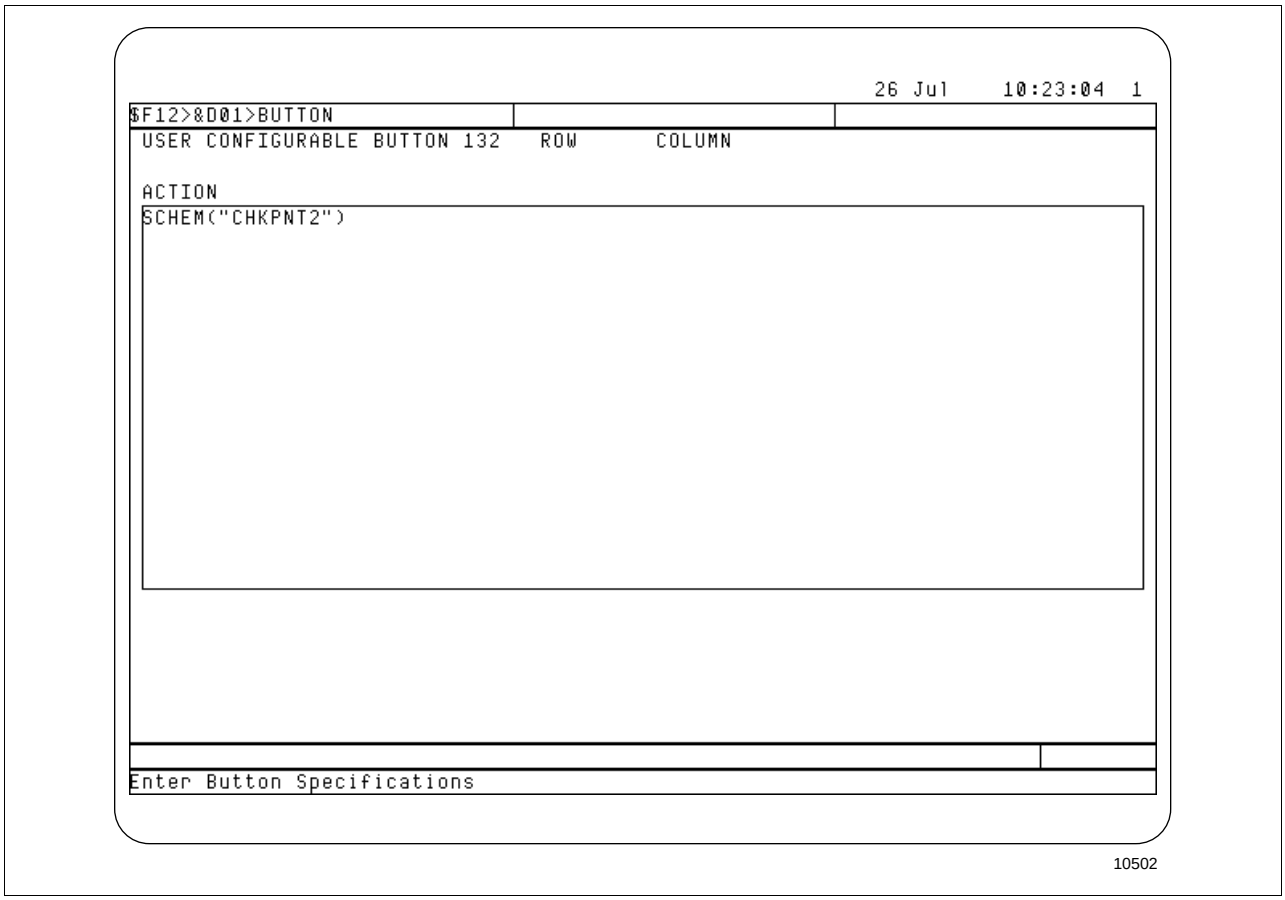
Continued on next page

Figure 10 CTL PF11(130) and CTL PF12(131) Button Configuration



Continued on next page

Figure 11 CTL PF13



Remoting Keystroke Macros

Description

If you have another station available in the console, you can simulate keystrokes without *using configurable buttons*; for example, you can simulate the same actions as shown in the previous example.

With release 400 came the addition of actors that manipulate displays and targets on another US in the same console. These actors are

- KEY,
- REM_MOVE, and
- REM_STR.

Example

Figure 12 shows the target of a "robot" custom display that can be used to initiate checkpointing of UCN devices, just as in the previous example, but does not require configurable buttons. The "trade off" is that it requires another station in the same console to be the "robot."

The display shown in Figure 12 (REMOTECP) is used to specify the station that is to be the robot.

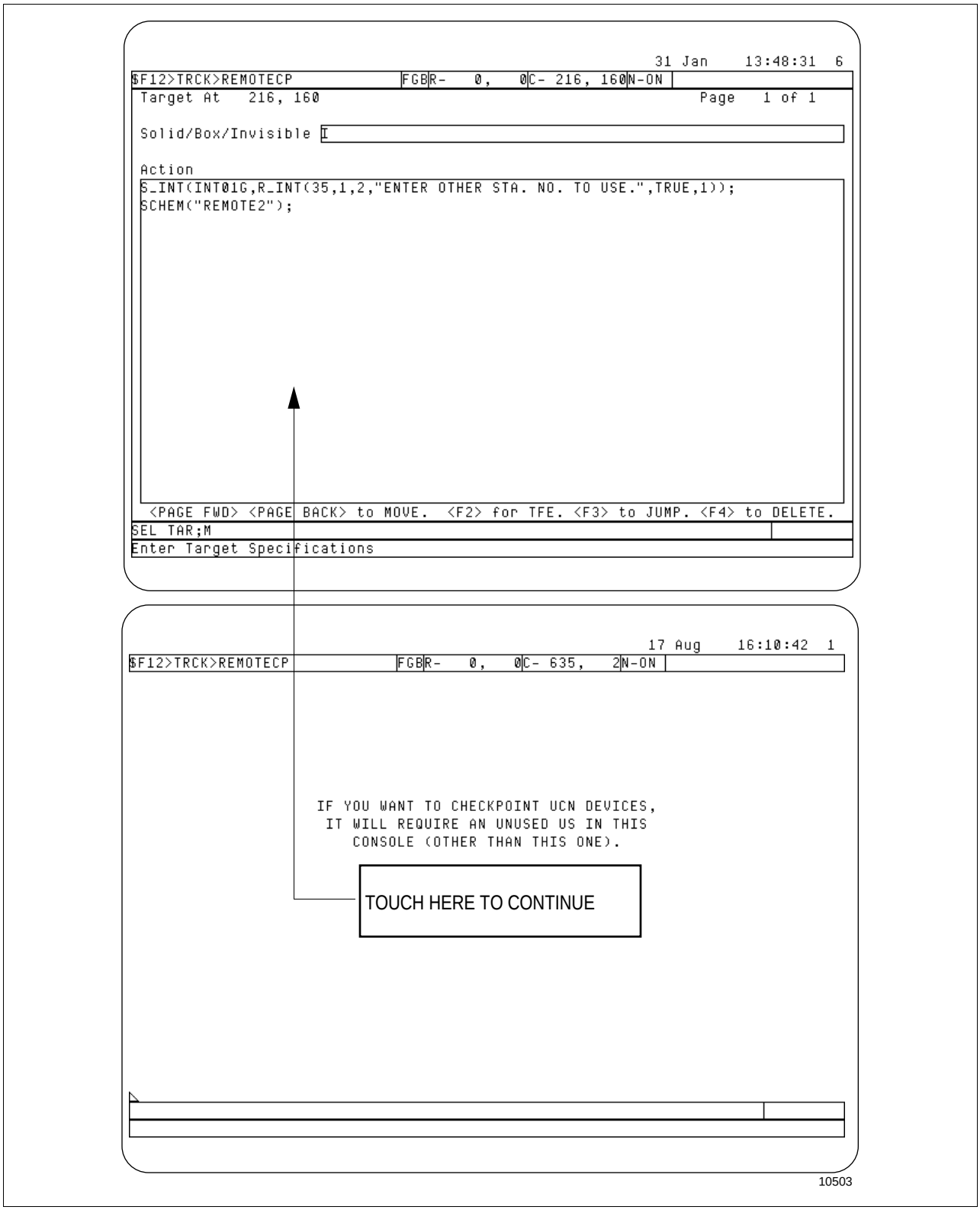
Figures 13 through 18 illustrate the *targets* of the display REMOTE2 when it is configured for remote keystroking:

- Figure 13—Initial Action pgs. 1 & 2
- Figure 14—Initial Action pgs. 1 & 2
- Figure 15—Target to Checkpoint All Devices
- Figure 16—Target to Checkpoint PM 3
- Figure 17—Target to Checkpoint APM 13
- Figure 18—Target to DO IT, pgs. 1 & 2
- Figure 19—Target to DO IT, pgs. 3 & 4
- Figure 20—Target to DO IT, pg. 5

Continued on next page

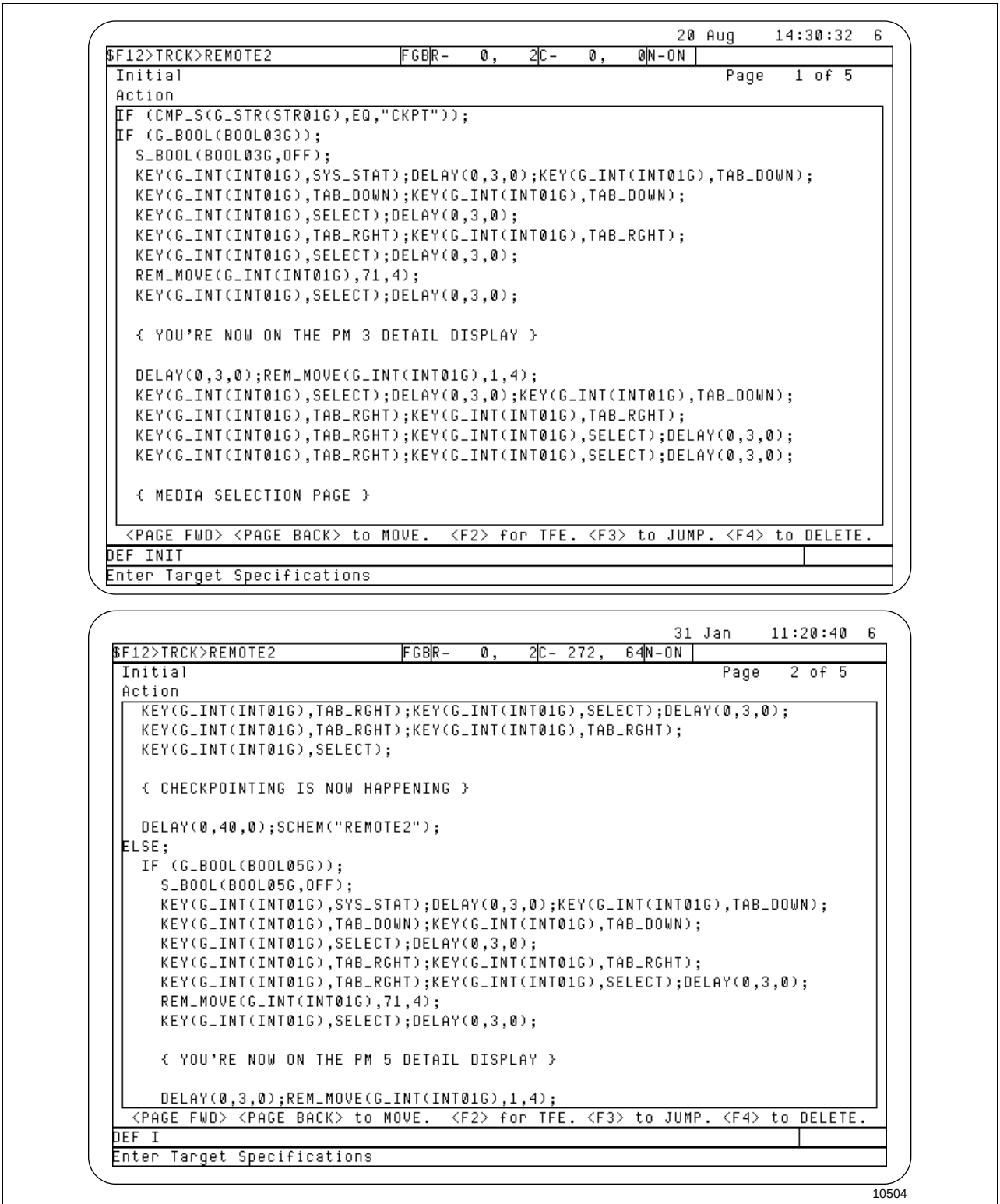
Remoting Keystroke Macros, Continued

Figure 12 Remote Keystroke Macros—Target to Specify “Robot” Station



Continued on next page

Figure 13 Remote Keystroke Macros—Initial Action pg. 1 & 2



```
                                                          20 Aug    14:30:32  6
$F12>TRCK>REMOTE2              FGBR-   0,   2C-   0,   0N-ON
Initial                                                        Page   1 of 5
Action
IF (CMP_S(G_STR(STR01G),EQ,"CKPT"));
IF (G_BOOL(BOOL03G));
  S_BOOL(BOOL03G,OFF);
  KEY(G_INT(INT01G),SYS_STAT);DELAY(0,3,0);KEY(G_INT(INT01G),TAB_DOWN);
  KEY(G_INT(INT01G),TAB_DOWN);KEY(G_INT(INT01G),TAB_DOWN);
  KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);
  KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),TAB_RGHT);
  KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);
  REM_MOVE(G_INT(INT01G),71,4);
  KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);

  { YOU'RE NOW ON THE PM 3 DETAIL DISPLAY }

  DELAY(0,3,0);REM_MOVE(G_INT(INT01G),1,4);
  KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);KEY(G_INT(INT01G),TAB_DOWN);
  KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),TAB_RGHT);
  KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);
  KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);

  { MEDIA SELECTION PAGE }

 <PAGE FWD> <PAGE BACK> to MOVE.  <F2> for TFE. <F3> to JUMP. <F4> to DELETE.
DEF INIT
Enter Target Specifications
```

```
                                                          31 Jan    11:20:40  6
$F12>TRCK>REMOTE2              FGBR-   0,   2C- 272,  64N-ON
Initial                                                        Page   2 of 5
Action
  KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);
  KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),TAB_RGHT);
  KEY(G_INT(INT01G),SELECT);

  { CHECKPOINTING IS NOW HAPPENING }

  DELAY(0,40,0);SCHEM("REMOTE2");
ELSE;
  IF (G_BOOL(BOOL05G));
    S_BOOL(BOOL05G,OFF);
    KEY(G_INT(INT01G),SYS_STAT);DELAY(0,3,0);KEY(G_INT(INT01G),TAB_DOWN);
    KEY(G_INT(INT01G),TAB_DOWN);KEY(G_INT(INT01G),TAB_DOWN);
    KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);
    KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),TAB_RGHT);
    KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);
    REM_MOVE(G_INT(INT01G),71,4);
    KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);

    { YOU'RE NOW ON THE PM 5 DETAIL DISPLAY }

    DELAY(0,3,0);REM_MOVE(G_INT(INT01G),1,4);
 <PAGE FWD> <PAGE BACK> to MOVE.  <F2> for TFE. <F3> to JUMP. <F4> to DELETE.
DEF I
Enter Target Specifications
```

10504

Continued on next page

Figure 14 Remote Keystroke Macros—Initial Action pgs. 3 & 4

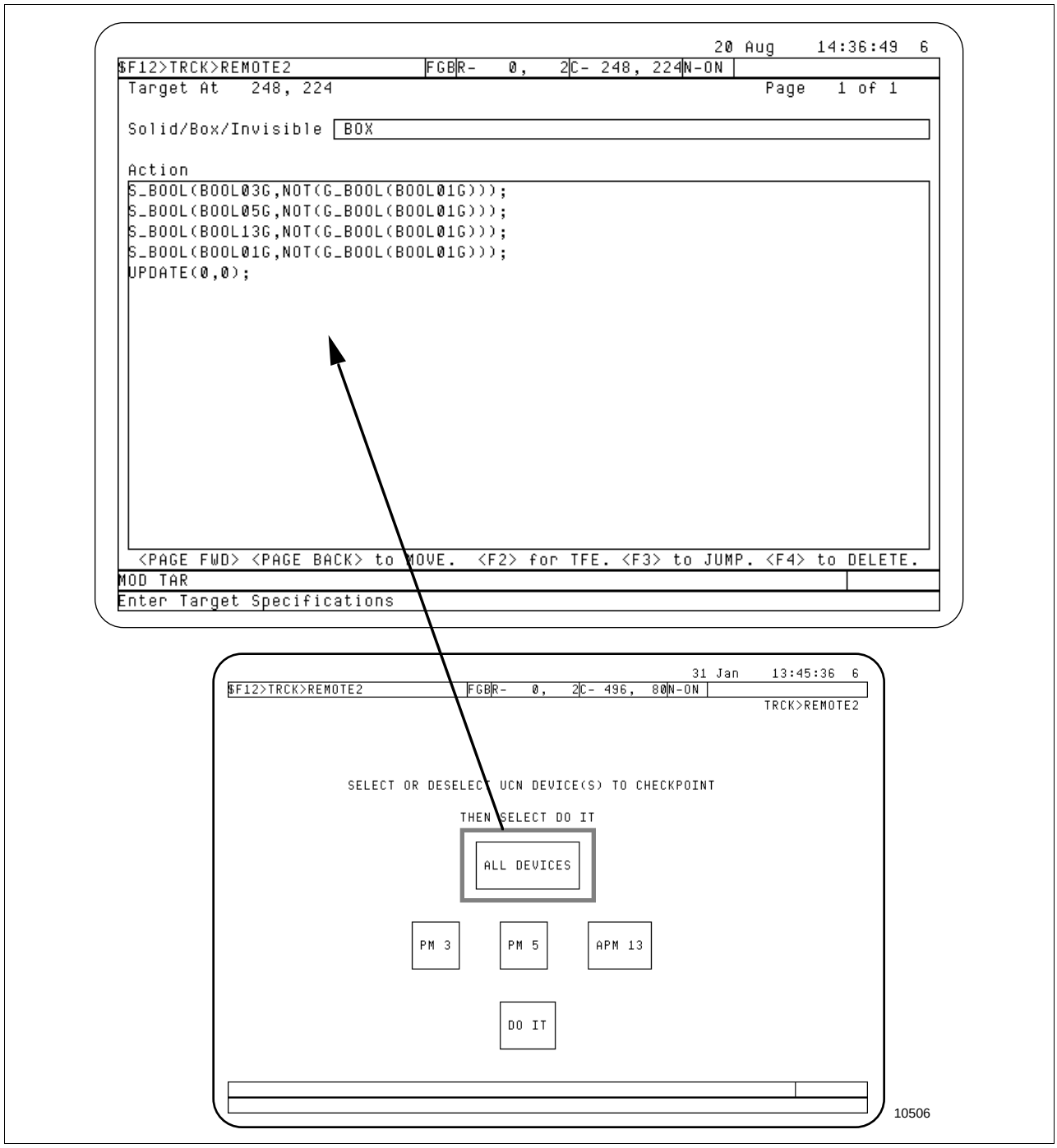
```
                                                          31 Jan    11:22:49  6
$F12>TRCK>REMOTE2                 FGBR-   0,   2C- 272,  64N-ON
 Initial                                                        Page   3 of 5
 Action
    KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);KEY(G_INT(INT01G),TAB_DOWN);
    KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),TAB_RGHT);
    KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);
    KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);

    { MEDIA SELECTION PAGE }

    KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);
    KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),TAB_RGHT);
    KEY(G_INT(INT01G),SELECT);

    { CHECKPOINTING IS NOW HAPPENING }

    DELAY(0,40,0);SCHEM("REMOTE2");
  ELSE;
    IF (G_BOOL(BOOL13G));
      S_BOOL(BOOL13G,OFF);
      KEY(G_INT(INT01G),SYS_STAT);DELAY(0,3,0);KEY(G_INT(INT01G),TAB_DOWN);
      KEY(G_INT(INT01G),TAB_DOWN);KEY(G_INT(INT01G),TAB_DOWN);
      KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);
      KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),TAB_RGHT);
 <PAGE FWD> <PAGE BACK> to MOVE.  <F2> for TFE. <F3> to JUMP. <F4> to DELETE.
DEF I
Enter Target Specifications
```

```
                                                          20 Aug    14:34:35  6
$F12>TRCK>REMOTE2                 FGBR-   0,   2C-   0,   0N-ON
 Initial                                                        Page   4 of 5
 Action
      KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),TAB_RGHT);
      KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);
      REM_MOVE(G_INT(INT01G),71,4);
      KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);

      { YOU'RE NOW ON THE PM 13 DETAIL DISPLAY }

      DELAY(0,3,0);REM_MOVE(G_INT(INT01G),1,4);
      KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);KEY(G_INT(INT01G),TAB_DOWN);
      KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),TAB_RGHT);
      KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);
      KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);

      { MEDIA SELECTION PAGE }

      KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);
      KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),TAB_RGHT);
      KEY(G_INT(INT01G),SELECT);

      { CHECKPOINTING IS NOW HAPPENING }

 <PAGE FWD> <PAGE BACK> to MOVE.  <F2> for TFE. <F3> to JUMP. <F4> to DELETE.
DEF INIT
Enter Target Specifications
```

10505

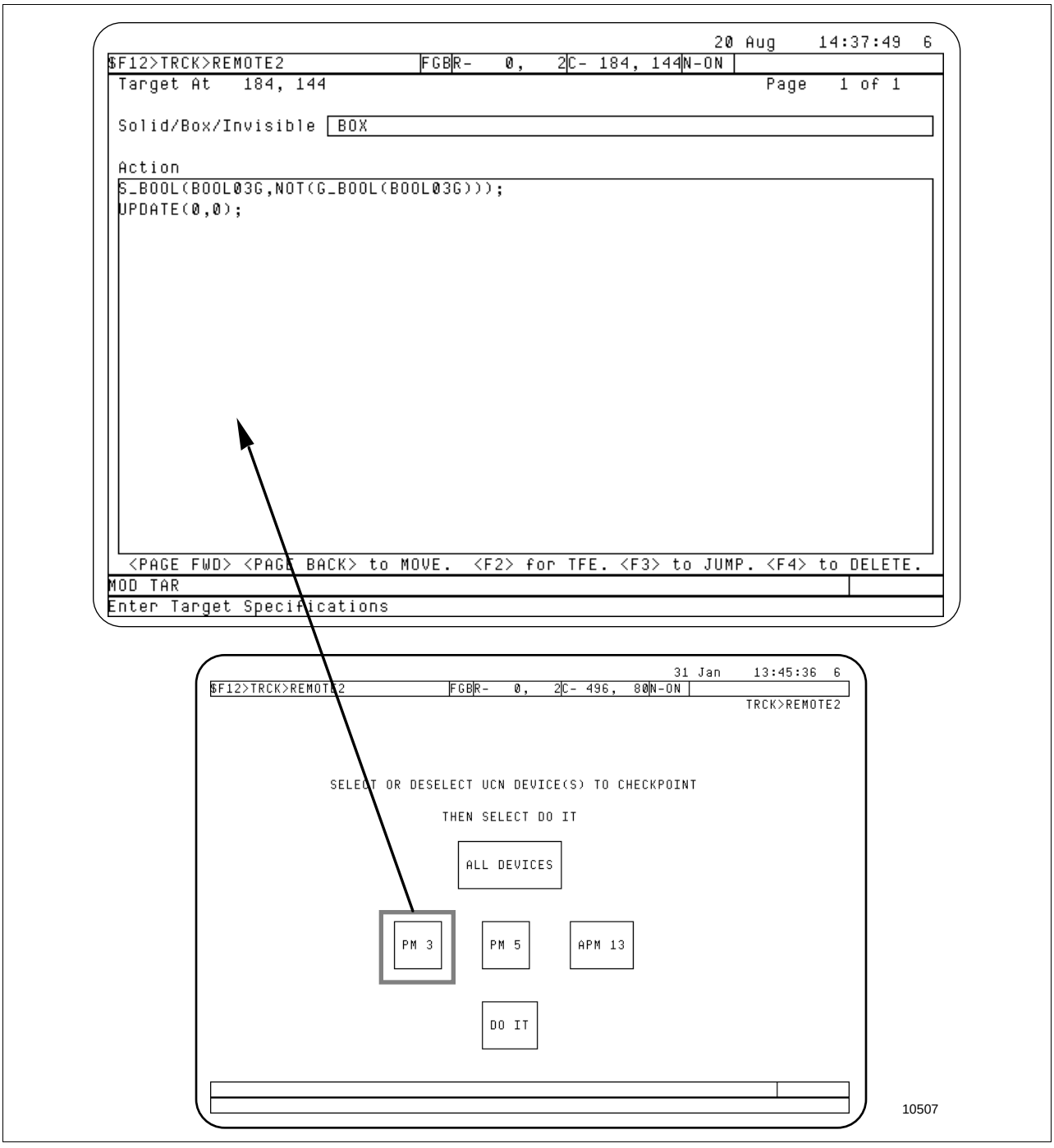
Continued on next page

Figure 15 Remote Keystroke Macros—All Devices Target



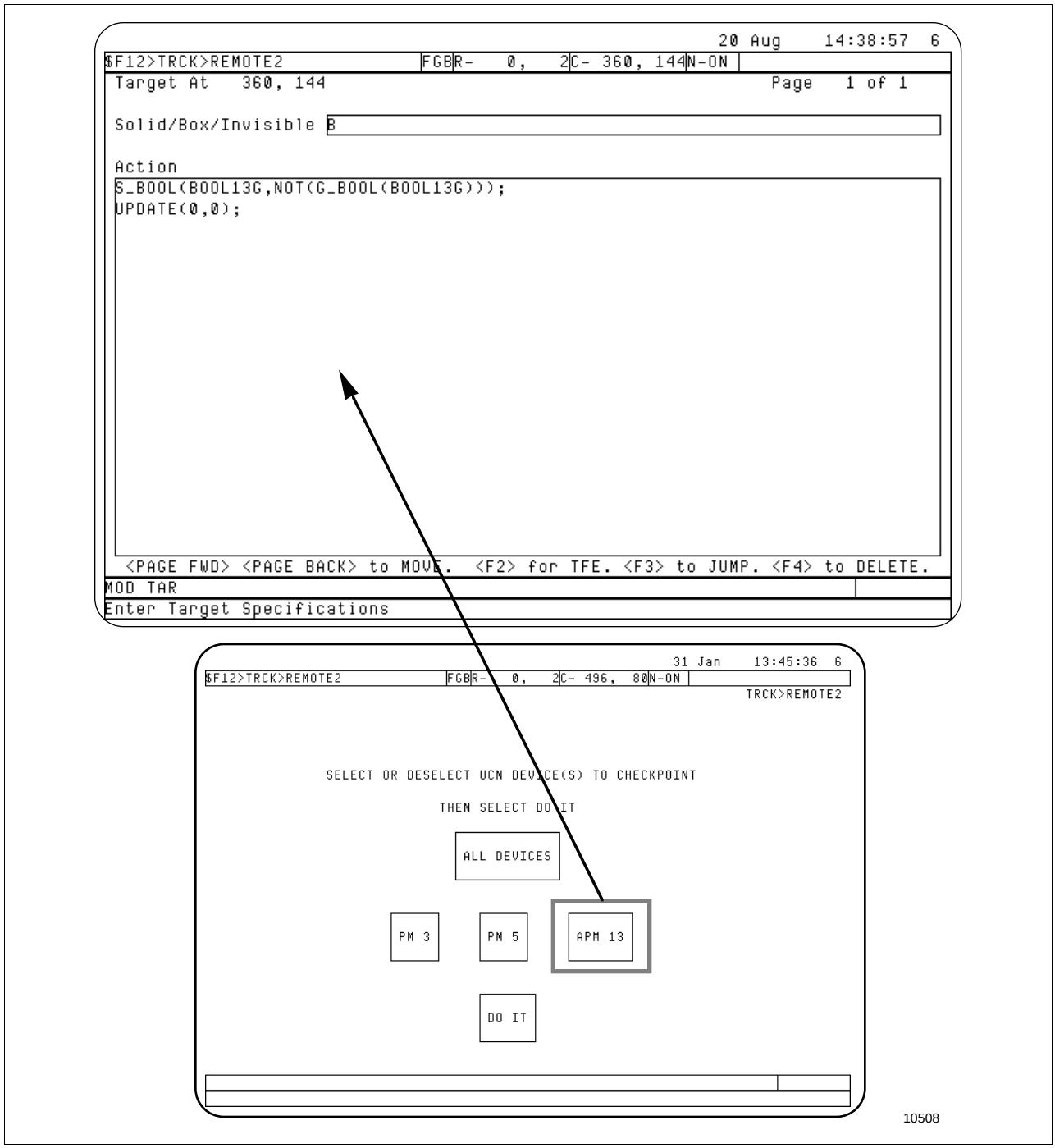
Continued on next page

Figure 16 Remote Keystroke Macros—PM 3 Target



Continued on next page

Figure 17 Remote Keystroke Macros—APM 13 Target



Continued on next page

Figure 18 Remote Keystroke Macros—DO IT Target, pgs. 1 & 2

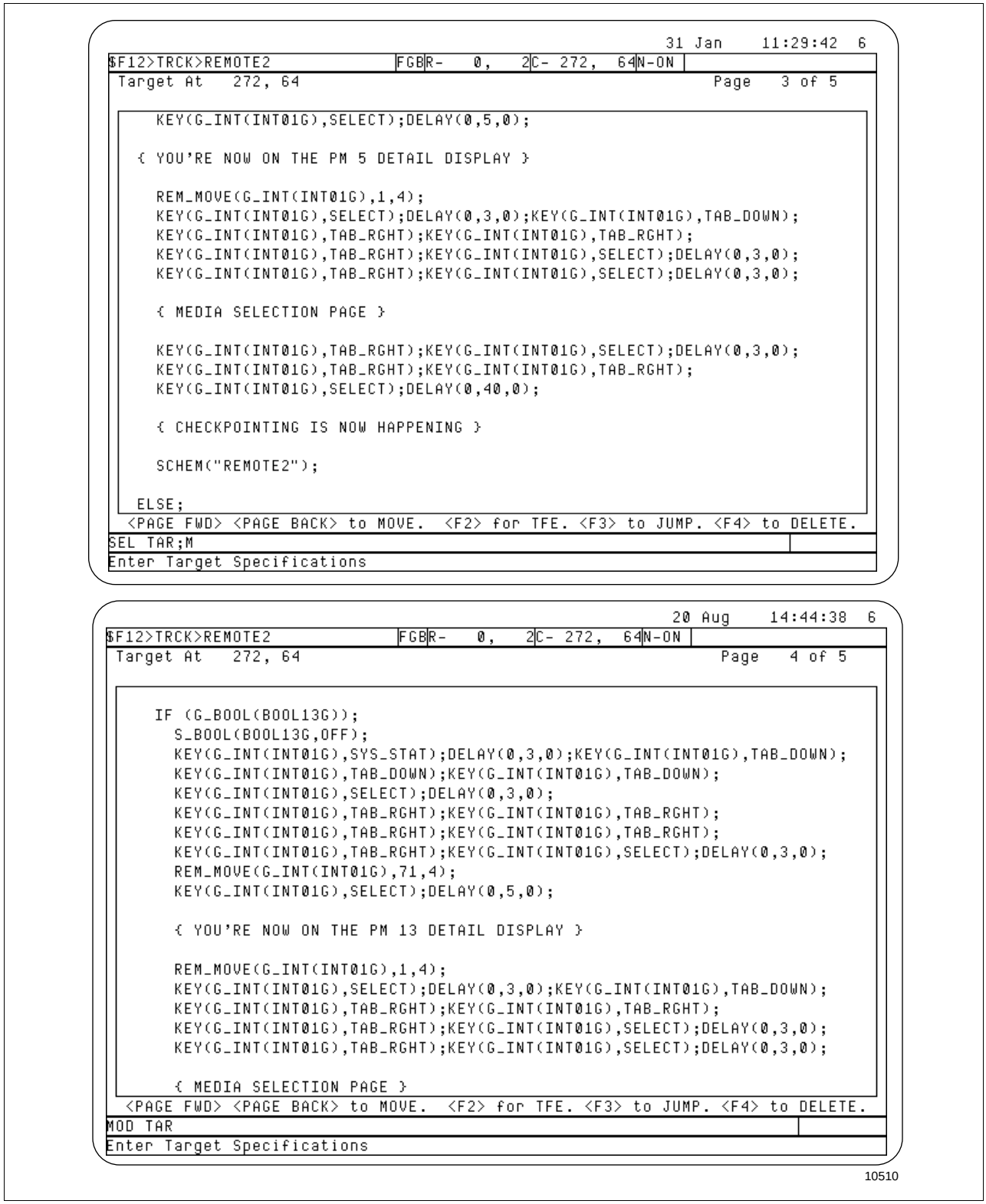
```
                                                          20 Aug    14:40:10  6
$F12>TRCK>REMOTE2                 FGB R-   0,   2 C- 272,  64 N-ON
 Target At   272, 64                                          Page   1 of 5

 Solid/Box/Invisible  B

 Action
 S_STR(STR01G,"CKPT");
 IF (G_BOOL(BOOL03G));
   S_BOOL(BOOL03G,OFF);
   KEY(G_INT(INT01G),SYS_STAT);DELAY(0,3,0);
   KEY(G_INT(INT01G),TAB_DOWN);KEY(G_INT(INT01G),TAB_DOWN);
   KEY(G_INT(INT01G),TAB_DOWN);KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);
   KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),TAB_RGHT);
   KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);
   REM_MOVE(G_INT(INT01G),71,4);
   KEY(G_INT(INT01G),SELECT);DELAY(0,5,0);

   { YOU'RE NOW ON THE PM 3 DETAIL DISPLAY }

   REM_MOVE(G_INT(INT01G),1,4);
   KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);KEY(G_INT(INT01G),TAB_DOWN);
   KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),TAB_RGHT);
   KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);
   KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);
  <PAGE FWD> <PAGE BACK> to MOVE.  <F2> for TFE. <F3> to JUMP. <F4> to DELETE.
MOD TAR
Enter Target Specifications
```

```
                                                          31 Jan    11:28:04  6
$F12>TRCK>REMOTE2                 FGB R-   0,   2 C- 272,  64 N-ON
 Target At   272, 64                                          Page   2 of 5

   { MEDIA SELECTION PAGE }

   KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);
   KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),TAB_RGHT);
   KEY(G_INT(INT01G),SELECT);DELAY(0,40,0);

   { CHECKPOINTING IS NOW HAPPENING }

   SCHEM("REMOTE2");

 ELSE;

   IF (G_BOOL(BOOL05G));
     S_BOOL(BOOL05G,OFF);
     KEY(G_INT(INT01G),SYS_STAT);DELAY(0,3,0);
     KEY(G_INT(INT01G),TAB_DOWN);
     KEY(G_INT(INT01G),TAB_DOWN);KEY(G_INT(INT01G),TAB_DOWN);
     KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);
     KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),TAB_RGHT);
     KEY(G_INT(INT01G),TAB_RGHT);KEY(G_INT(INT01G),SELECT);DELAY(0,3,0);
     REM_MOVE(G_INT(INT01G),71,4);
  <PAGE FWD> <PAGE BACK> to MOVE.  <F2> for TFE. <F3> to JUMP. <F4> to DELETE.
SEL TAR;M
Enter Target Specifications
```

10509

Continued on next page

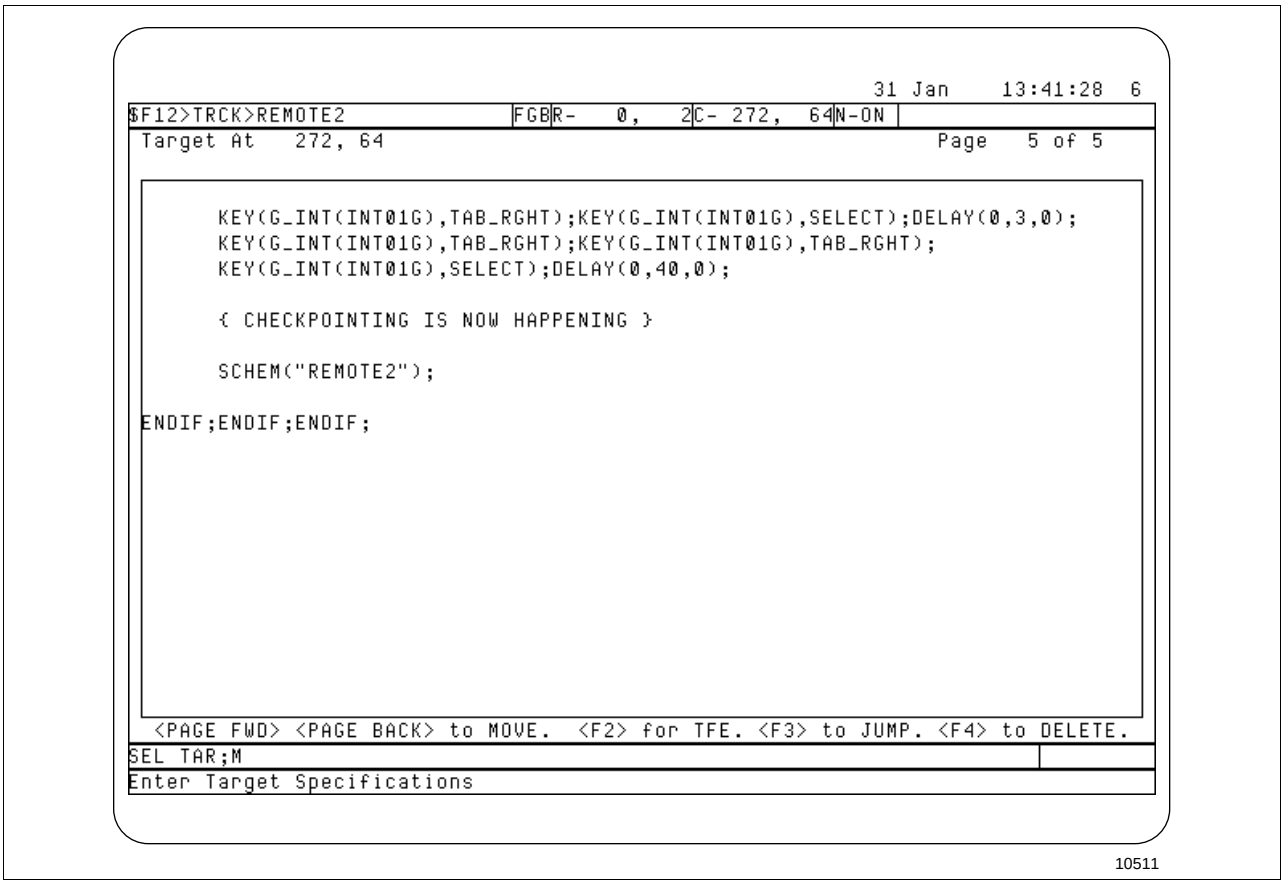
Figure 19 Remote Keystroke Macros—DO IT Target, pgs. 3 & 4



Continued on next page

Remoting Keystroke Macros, Continued

Figure 20 Remote Keystroke Macros—DO IT Target, pg. 5



Beep

Description

Sometimes it would be nice to be able to cause the station to produce an audible beep.

The slick trick described here lets you produce a beep in any target, including those invoked by the DEFINE command.

This is a nice feature you can use to signal an invalid operator input or get the attention of the operator when a message is displayed.

Slick trick

Use the actor QUE_KEY (ESCAPE).

This requests the ESCAPE key, which is invalid in the Operator Personality, so the station beeps as it always does when an invalid key is pressed.

The station will not print any error messages and it will not stop execution of the rest of the Target.

How to Create Self-Formatting Values

Description

When you build a point, such as a Regulatory Control point, one of the configuration parameters is the PV and SP display format (PVFORMAT). This format is used for displaying the decimal place on standard displays, such as Group and Detail. What about Custom Displays? Will they use PVFORMAT?

Problem

PVFORMAT is not used by Custom Displays. Instead, each real Value added to the display must have its format defined at display buildtime.

The problem is, unless you look up each format when adding a value to the display (possibly a lot of duplicate work), the value may end up with the wrong fractional precision.

Partial solution

A partial solution to this problem is to build a Subpicture library containing a Subpicture for each value format you intend to use.

With the exception that you can build such Subpictures with complex conditional behavior, this approach "buys" you little. You still must know the correct format for any particular point when you add a value to the display.

Scenario

Consider this scenario:

After adding a particular point's PV and SP to several custom displays, you decide to change the PVFORMAT of the point. If you want the PV and SP values to be consistently displayed on the system, you should go back and fix each display that contains the values.

You now are forced to use FIND NAMES to locate each occurrence of the point, then edit and recompile each display. Well, you get the picture (pun intended)!

It would be nice if the display always used the point's *current* PVFORMAT.

Continued on next page

How to Create Self-Formatting Values, Continued

Preferred solution

To cause a custom display to always use a point's current PVFORMAT, provide a single Subpicture to be used for all real PVs and SPs.

Build a 3-level nested subpicture, which is a subpicture used by a second Subpicture, that is used by a third, which is the end product.

Slick trick

The following steps outline how to cause a custom display to use a point's current PVFORMAT:

- **STEP 1**

  Build a generic Subpicture for each PVFORMAT (D0, D1, D2, D3).

- **STEP 2**

  Build a second Subpicture with a single Variant that conditionally selects one of the previous subpictures, based on its PVFORMAT.

  The trick here is to keep the point name generic for later use. Use something like &TAG.PVFORMAT.

- **STEP 3**

  At this point you need to know the data type of the parameter PVFORMAT.

  Take the time now to look it up in the *AM Parameter Reference Dictionary*, because the points you are using in this course are AM points.

  For future reference, write the data type in the space provided below:

  _____

- **STEP 4**

  Because you would like this subpicture to be usable for both PVs and SPs, keep the parameter generic. Use something like &TAG.&PARM to answer each value prompt.

If you stop here, the subpicture is completely usable as it stands, but carry it one step further to make it even simpler to use.

- **STEP 5**

  Create separate subpictures for PVs and SPs.

  Keep the point name generic by answering &TAG to the point name prompt and PV to the parameter name prompt.

  Give the subpicture a simple name, such as REAL_PV, because this subpicture won't work for PVs of other data types. Now you can simply ADD SUB REAL_PV and supply a point name.

  Because the runtime operation is dynamic, the display won't require modification if the PVFORMAT of the point is changed.

Stuff Text into Prompt Port

Description

Occasionally, when building a Custom Display, you might want to solicit operator input through a prompt port. If a value or text is required, but you do not want the operator to have to type it in, you can cause the port to open with a value or text already typed-in.

When the port opens, the cursor moves to the first position in the port. Using the QUE_KEY actor, you can specify each keyboard character to place in the port.

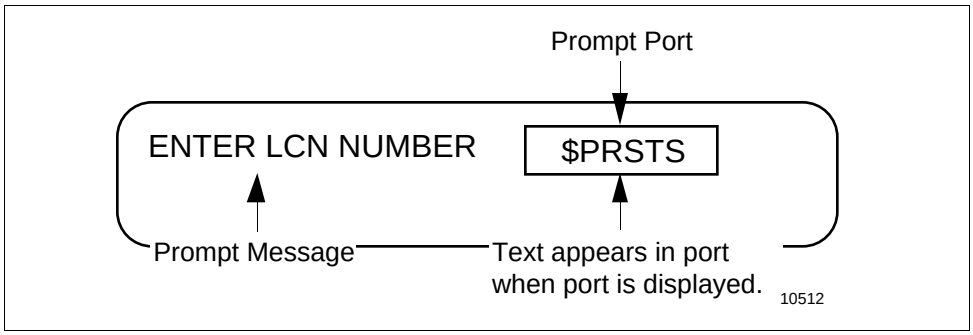
The operator presses the [ENTER] key to choose the default, or presses the [CLR ENT] key and types-in the desired data.

Example 1

The following target action puts text into the port

```
QUE_KEY(DOLLAR); QUE_KEY(P); QUE_KEY(R);
QUE_KEY (S); QUE_KEY (T);QUE_KEY (S);
S_ENT(ENT01,R_ENT(22,1,8,"ENTER LCN
NUMBER",TRUE,1); DMD_UPD (3)
```

Figure 21 Stuffing Text Into A Port

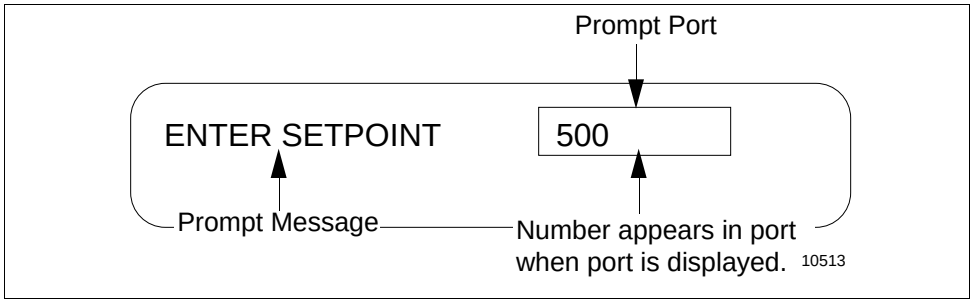


Example 2

The following sequence of actors allows the operator to accept the default of 500, appearing next to the prompt in the port for TIC21001's setpoint:

```
QUE_KEY (FIVE); QUE_KEY (ZERO); QUE_KEY (ZERO);
RS_SYS (TIC21001.SP,24,1,6,"ENTER SETPOINT",ON,1);
DMD_UPD (0)
```

Figure 22 Stuffing A Value Into A Port



Continued on next page

Stuff Text into Prompt Port, Continued

Slick trick

There is a trick associated with the sequence of QUE_KEY actor execution.

You might think that you should first open the prompt port, then queue up the characters to place in the port. This is logical but it does not work. Once a port opens, the display immediately goes into a "wait for input" mode. Only the characters typed-in by the operator are read when the [ENTER] key is pressed.

The trick is to put the desired characters into the keyboard queue by using the QUE_KEY actor before opening the port. If the keyboard queue has characters in it when the port is opened, they are placed in the port!

Application Subpictures

RING

The RING subpicture can be used to indicate a status condition (see Figure 23).

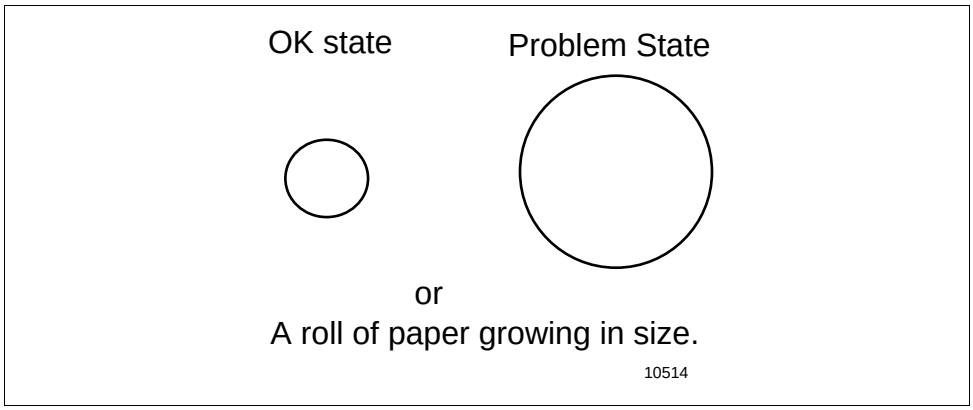
Example 1:

Small ring = OK
Large ring = problem

Example 2:

Could represent a roll of paper growing in size.

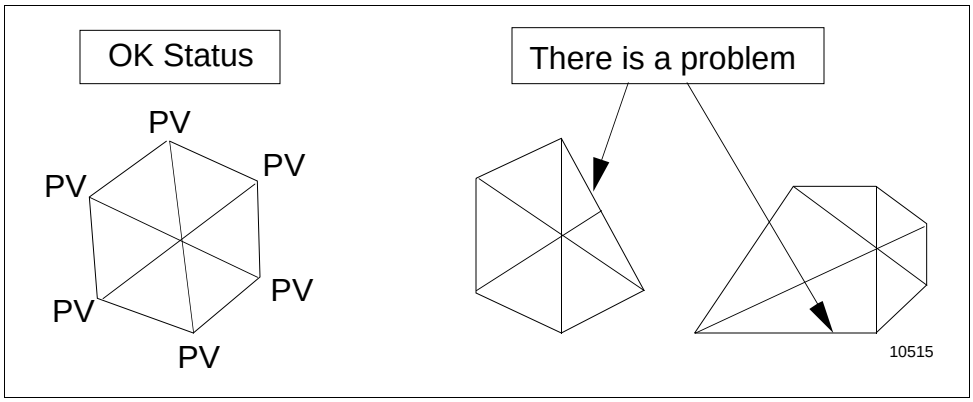
Figure 23 RING Application Subpicture



RADAR

RADAR displays are for pattern recognition, often used for operator who is walking around (see Figure 24).

Figure 24 RADAR Application Subpicture



Lab Exercise

Introduction

Overview

In this lab exercise, you have the opportunity to practice all of the slick tricks described in this course module:

- text editing keys,
- Combine Date/Time actor,
- “tick mark,”
- test for a bad value,
- keystroke macros, and
- default value in TIP.

The lab exercise is approximately 2 hours in length.

Lab Instructions

Text editing keys

STEP 1

Read the display TRCK>REMOTE2 into the Picture Editor, then select its Initial Action for modification.

STEP 2

Practice using the text editing keys described in Table 1 of this course module until you feel that you understand their use.

Combine Date/Time actor

STEP 1

Call up MOOSE, then select ACTORS

From the Actors Menu, select COMBINE DATE AND TIME ACTOR

STEP 2

Review the instruction provided for the actor.

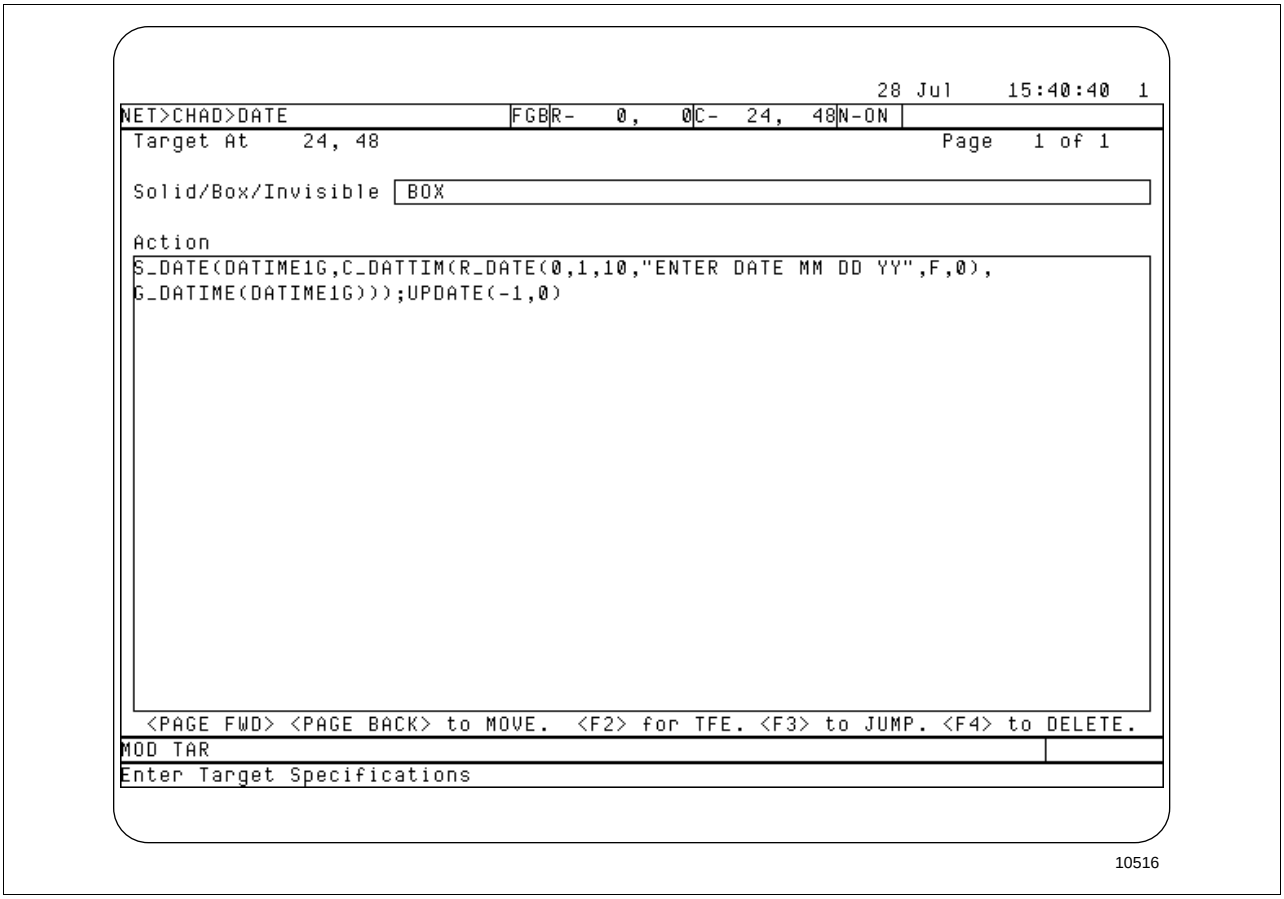
STEP 3

Create a target that reads a date from a Text Input Port and stores it to the standard global DDB parameter DATIME1G, without affecting the time portion of the parameter.

(Figure 25 shows an example target. The example display named DATE is on your cartridge disk in directory TRCK)

Continued on next page

Figure 25 Combine Date/Time Actor



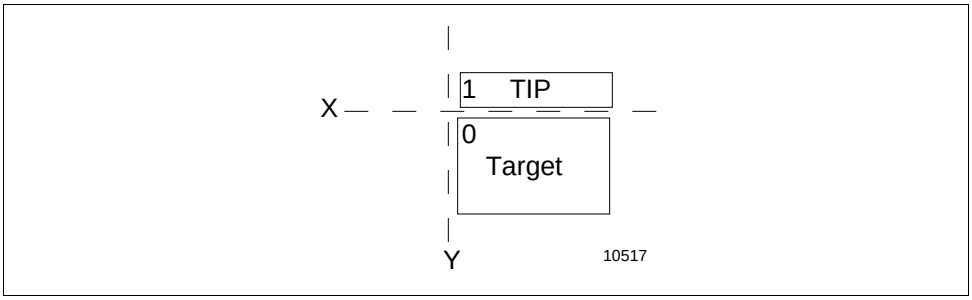
TIP coordinates

Look at the R_DATE actor in Figure 25.
`R_DATE(0,1...` defines the X,Y coordinates for the TIP.

To open the TIP *above* the target, specify `1` as the `Y` coordinate and specify `F` (FALSE) to position the TIP relative to the target's upper-left corner (see Figure 26).

The ACTORS section of the MOOSE menu discusses TIP coordinates.

Figure 26 TIP X/Y Coordinates



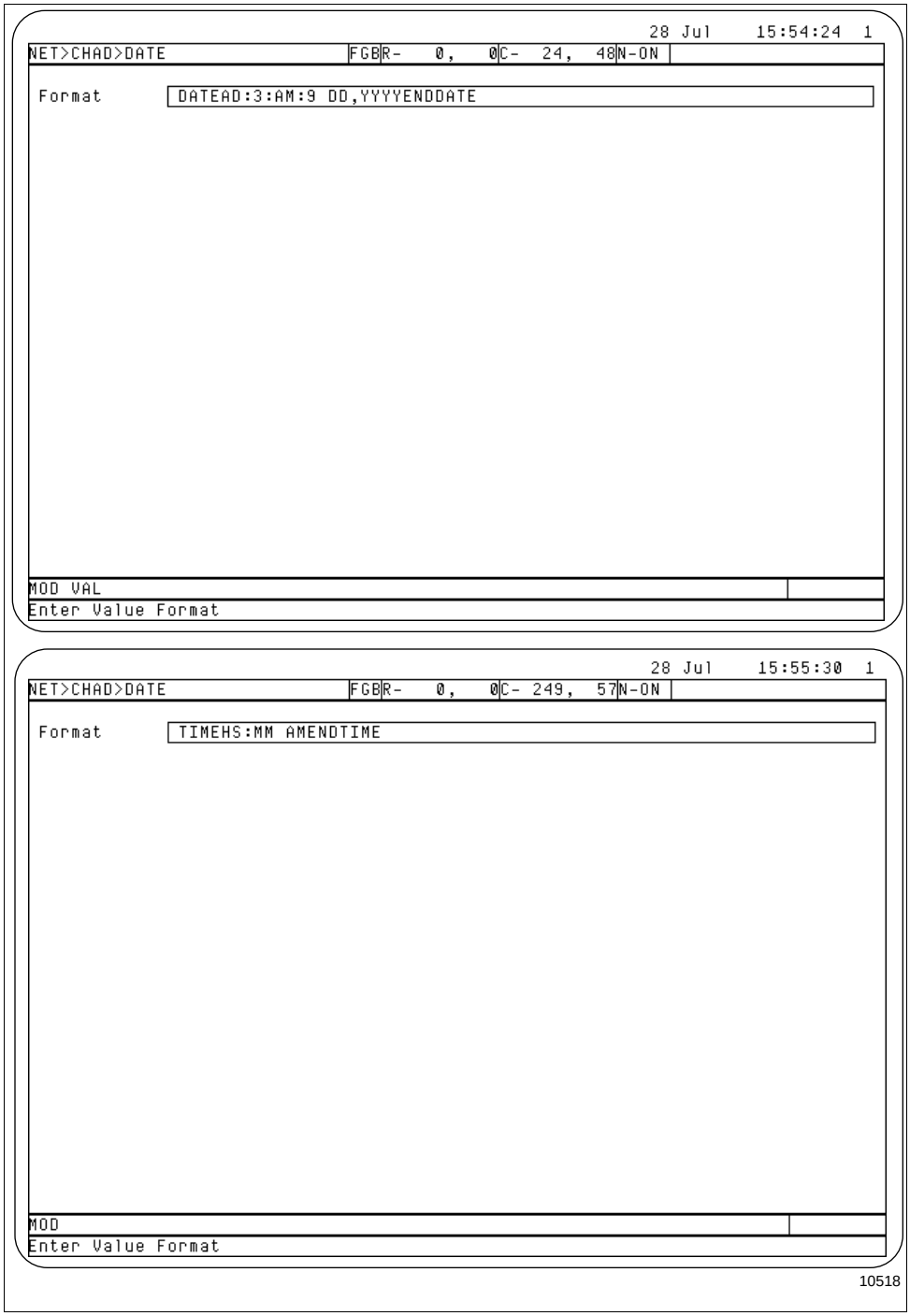
Continued on next page

Date/Time values

STEP 4

Add date and time values to your display to see that changing the date does not affect the time. Figures 27 shows the default value formats. Appendix A of the *Picture Editor Reference Manual* describes all date/time formats.

Figure 27 Date/Time Value Format Examples



Tick mark

If you want to experiment with the "tick mark" slick trick, do the following:

STEP 1

Add to a custom display a trend bar representing the PV of RAMP### (where ### is your assigned number).

STEP 2

Next to the PV bar, add a tick mark representing the high alarm limit value. Make the color of the tick mark low intensity red.

STEP 3

Add text that indicates the maximum and minimum values of the PV bar.

STEP 4

After compiling your display, change the alarm limit value to see the change reflected by the tick mark.

Test for NaN

If you want to experiment with the "test for NaN" slick trick, do the following:

STEP 1

Choose one of the points on your partition sheet to use for this exercise.

STEP 2
In the INITIAL action of a display, test the point's PV for NaN.

If the PV is NaN, store a string to a local string variable. Add the the string variable to the display.

STEP 2

After compiling your display, activate and deactivate your point to observe that the string reflects the PV change from good to NaN.

Continued on next page

Keystroke macros

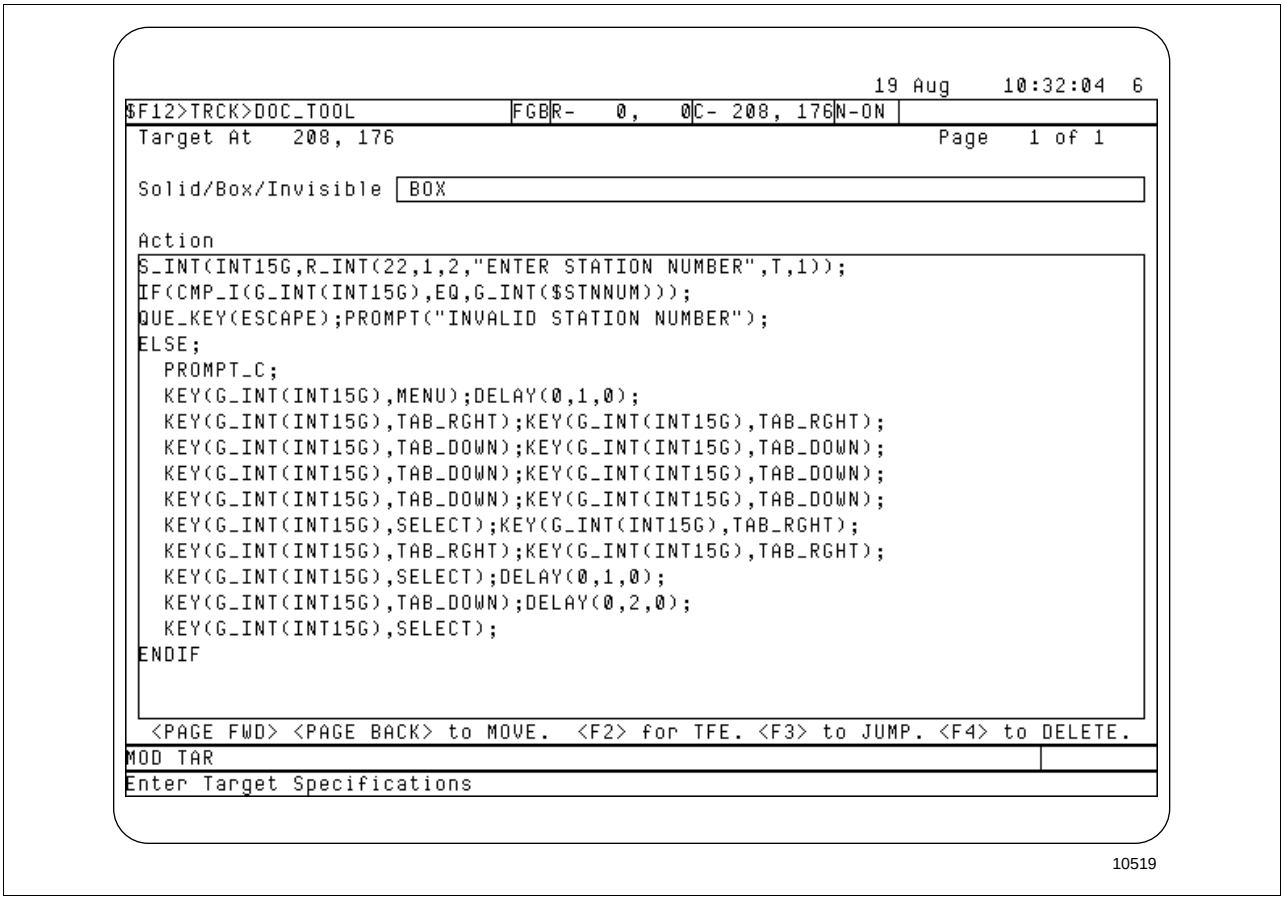
For more information on the new R400 actors pertaining to keystroke macros (KEY, REM_MOVE, and REM_STR), select the ACTORS target on the MOOSE Menu, then select the R400 Actors target.

If you want to experiment with keystroke macros, do the following:

STEP 1

Create a target that provides the operator with **one step access to prebuilt queries in the R400 Documentation Tool** (see Figure 28).

Figure 28 Display Doc Tool On Any Station With Universal Personality



Continued on next page

Return from Doc Tool

Without the engineering keyboard, an operator has no easy way to get out of the Documentation Tool, so you must configure a button that calls up the Engineering Main Menu, then calls up a nonengineering display from the Engineering Main Menu, such as the Console Status display.

This button must be pressed from another station in the console, then you enter the number of the station that has Doc Tool active. The example in Figure 29 calls up the Engineering Menu, then the System Status display.

STEP 1

Modify the Button Configuration file for your assigned Area database (&Dnn>BUTTON.KS on your cartridge disk).

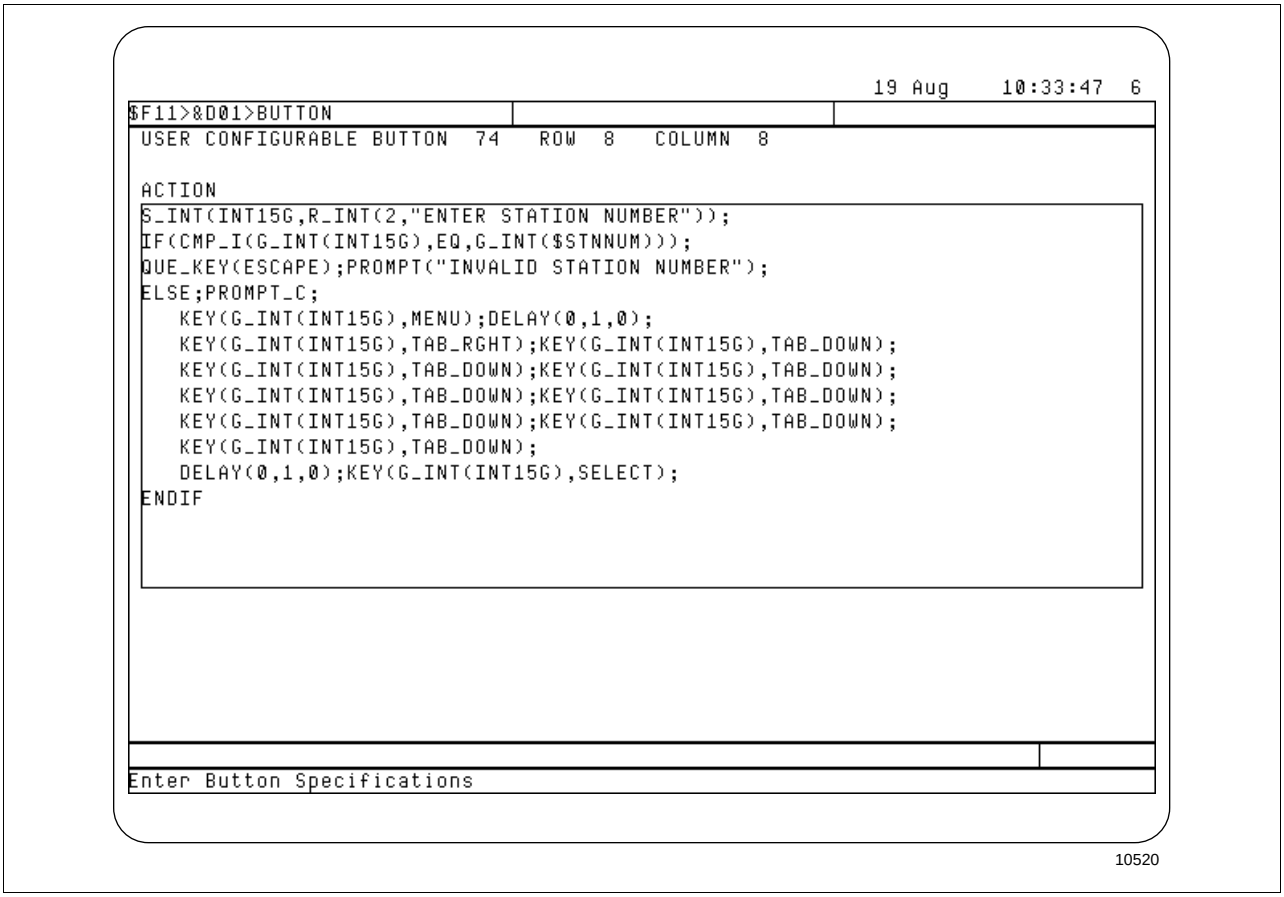
STEP 2

After compiling your Button Configuration, remember to Change Areas.

STEP 3

Find another station in your console that you can use to test your keystroke macros.

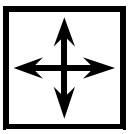
Figure 29 Don't Forget the Button to Remove Doc Tool



Continued on next page

Other keystroke macros	If you want to try out the prebuilt display named CHKPNT that uses checkpointing macros, in the Phoenix training center, use a station on the far west side of the lab, or ask your course manager for assistance.
Stuff text or value into prompt port	To practice stuffing text or values into prompt ports, follow the instructions provided earlier in the course module. If you want to look at or modify a prebuilt example, use the display TRCK>DATE.

Directions



DIRECTIONS—This is the end of the study material for this module. Discuss questions concerning the study material or the lab activities with a colleague or a course manager

If you are satisfied that you have achieved the objectives of this module, continue with the next section, the Student Proficiency Evaluation.

Student Proficiency Evaluation

Criterion Test

Instructions

1. Build a custom display that demonstrates *at least three* slick tricks:
   a. target that uses the Combine Date/Time actor to store either a date or time,
   b. “tick mark” by using Add Bar command (The tick mark should represent one value in relation to another.),
   c. target, variant, or condition that tests for NaN,
   d. target that performs a keystroke macro,
   e. subpicture that displays a PV value in its currently configured PVFORMAT, and
   f. target that puts text or values into an input port displayed to the operator

   When you have completed your display, demonstrate to your course manager that it successfully performs as defined.

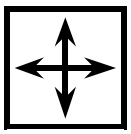
2. Demonstrate to your course manager how to use the text editing keys to cut and paste blocks of text into the entry port of a target, variant, or condition.

3. Demonstrate to your course manager how to write the entry port of a target, variant, or condition to a text file, then read the file into the entry port of a different target, variant, or condition.

Self-Evaluation

1.a. Combine Data/Time actor	*Demonstrate that you can change either the date or the time without affecting the other.*
1.b. Tick mark	*Demonstrate that the tick mark changes in response to a parameter change.*
1.c. Test for NaN	*Demonstrate that the custom display reflects when the value becomes NaN.*
1.d. Keystroke macros	*Demonstrate that the keystrokes execute correctly when the target is selected.*
1.e. PV format	*Change the PVFORMAT of a point and demonstrate that the custom display's subpicture reflects the format change.*
1.f. Stuff a port	*Demonstrate that text or values appear in the port when a target is selected.*
2.	*Follow the procedure described in Table 1.*
3.	*Follow the procedures described in Tables 2 and 3.*

Directions



DIRECTIONS—This is the end of this module.

Use your course map to

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

If you have a question

- Ask your course manager.

LAST PAGE

Industrial Automation and Control
Honeywell Inc.
2820 West Kelton Lane
Phoenix, Arizona 85023

Helping You Control Your World