To: Distribution

From: John Rhodes

Date: February 3, 1975

Subject: Online T & D and Operator Interface

Introduction

This document describes in detail how to use the Peripheral Online Test Subsystem (POLTS) in MULTICS.

POLTS provides the online capability of functional testing and online trouble shooting of malfunctioned equipment. This is accomplished without unduly interfering with the overall system capability to continue to service other users while testing is being conducted.

A secondary function is that during slack periods of user operation, or peripheral availability, equipment testing can be accomplished. This allows the normally scheduled P.M. time to be used more effectively in the actual maintenance of equipment rather than the running of tests.

FUNCTIONAL CAPABILITIES

The following services are accomplished by POLTS:

* Resource Allocation
* Test Page Selection
* Test Page Dispatching
* Test Sequencing
* Option Processing
* Test and Diagnostic Language (TDL) Interpretation
* I/O Set Up and Issue

- Error Checking and Error Message Formatting
- Test Page Termination
- Resource De-Allocation

Hardware Tested

The hardware tested includes:

- ASA Seven-track Magnetic Tape (AT7C)
- ASA Nine-Track Magnetic Tape (AT9A,AT9B)
- MTS500 Seven-Track Magnetic Tape (T57A)
- MTS500 Nine-Track Magnetic Tape (T59A, T59B, T59C)
- CR20 Card Reader (R20C)
- CR21 Card Reader (R21C)
- CRZ301 Card Reader (R34C)
- CPZ201 Card Punch (P23C)
- CPZ300 Card Punch (P32C)
- PRT201 Printer (PRTC)
- PRT300/301 Printers (AP3C, BP3C)
- PRT303 Printer (AP4C, BP4C)
- PRT401/402 Printer (136 line) (P40A,P40B)
- PRT401/402 Printer (160 line) (P40C,P40D)
- System Console (CONC)
- System Control Center (SCCC)

OPERATOR COMMAND STRUCTURE

Control of the POLTS test system is achieved by typing in a command as below after a "???" typeout when the POLTS system is originally called into execution or subsequently resulting from depressing of the "interrupt" or "break" button on the TTY during test execution.

The generalized test system command is entered as:

```
TEST xxxxxxxxxxxx
```

Where "TEST" is entered as it is and
"xxx---" make up any of the requests or
options listed below.

NOTE: Multics POLTS input is in lower case. The upper case TEST is for GCOS; use lower case ???test for Multics.

Test PCD Request

```
???test pcd
```

causes the current peripheral configuration to be printed as follows:

```
Configuration:

     opc  01600 emc
     prtb 01200  301   3
     tape 018xx  500 1 7tks starting at 8 find 7 9tks starting at 1
     prtc 01200  301   2
     rdrr 01001  301
     punb 01500  201
     rdrb 01400  201
     punr 01101  301
     prtd 00801 1600 600 136
     prtr 00901 1200 600 136
```

where:

prtr = Multics device name
00901 = iom/channel/device in decimal, exactly the way the test piccdd request expects input.
1200 = device typ
600 = train
136 = numbar of prt columns
tape 018xx = iom0 channel 18 device xx (handler 01 or 01 etc.)

Test List Request

```
???TEST PLSTAL
    or
???TEST LSTAL
```

List all of the active test pages in POLTS

Sample output:

```
POLTS LSTAL:

**0(11900C) IN EXECUTION
**1(01201C) WAITING ALLOCATION
```

The "WAITING ALLOCATION" message indicates that the device cannot be assigned to POLTS becuase it is in use by some process already (possibly by POLTS itself).

Test Wrapup Request

```
???TEST PW
    or
???TEST W
```

Wrap up all POLTS operation on the system

New Test Request

```
???TEST Piccddooooo
```

Where:

```
iccdd  = iom/chan/dev

ooooo = Option characters or a control mnemonic
```

Test Communications Request

```
???TEST POiccddoooo
```

Where:

```
iccdd  = iom/chan/dev

oooo   = The new options or control mnemonic (use the
         option O if you want to interrupt the test page)
```

Test End Request

```
???TEST PEiccdd
```

Where:

```
iccdd  = iom/chan/dev
```

Configuration Request

???TEST PCD

This request will print the system configuration in the format:

```
configuration:

rdra 01400   200
rdrb 01801   301
prtc 11201  1600    4 136
opc  01600  ibm
prta 01200   201    2
prtb 01200   301    3
tape 118xx   500 4  7tks starting at 1 and 2 9tks starting at 7
```

Where:

The first field, such as "rdra", is the configured device name.

The second field, such as "01400", is the iom/chan/dev just as it will be used by the "test p---" request.

For tapes, the dev. field is xx as they are multi device channels.

The third field, such as "200" is the unit "type" (model).

The fourth field, for printers, is the print train image for those printers that have replaceable trains.

The fifth field, for the printers with different line lengths, 71), is the # of columns in a print line.

For tapes, starting at the fourth field, information is given as to the first 7tk tape dev. # , the first 9tk dev. #. and the number of consecutive devices of that type starting with the first # for that type.

STANDARD TEST PAGE OPTIONS

The following error and control options can be entered in response to an Enter Options message or designated in the option string of a new Test request.

Option Characters

B - Bypass error message output. Bypass overrides a Pass or Cycle message unless Halt is set. Halt forces these messages out over Bypass.

E - Output Transient Error message. If a numeric digit (or two digits) follows, the value is used to override the test page standard value for the number of retries to be made. If a "-1" is input, then the test standard is restored. This control of retry count is independent of the turn ON or OFF capabilities of E or NE.

H - Halt for input of options following error messages, Test End messages, Pass End message, and Cycle End messages.

I - Inform operator of each normal Test End. If H is also set then an Enter Options is appended to the End Test message. If the next next segment is being called or if an End Pass message is being output, or if a cycle ends, the End Test message is overridden.

L - Loop on current test (cannot loop on test 0)

N - Negate the following option character (valid preceeding A-B-E-H-I-L-P-R-T-X-Z options only).

O - Go to ENTER OPTIONS following processing of the complete option string containing the O.

P - Issue an End Pass message any time a back jump is detected by the next test sequencing. If H is also set then an Enter Options is appended to the End Pass message. The error tallies for the current pass are reset when an End Pass is reported or when P is being turned ON.

R - Issue an End Cycle message any time a normal test page termination would occur and cycle back to the first test in the current sequence again. If H is also set then an Enter Options is appended to the End Cycle message. The error tallies for both the current pass and current cycle are reset when End Cycle is reported. The error tallies for the cycle are reset whenever R is being turned ON.

S - Unconditionally skip to the next test.

T - If turn ON (no preceding N) then unconditionally jump to the first occurence of the test in the current sequence. The test number must follow and must be nonzero, and can consist of one, two or three digits. For segmented test pages, a value outside of the current segment causes a jump to the corresponding segment after processing of the current option string. If turn OFF (NTxx) then the test number must be in the current segment and sequence and not the forced term test number.

X - Enables the extended status portion of the standard error message for MPC test pages. This option is initially forced ON automatically. The user should utilize the NX option characters if no extended status message is desired. The message could subsequently be enabled by utilizing the X option character.

Z - Trace i/o setup A message is output for each test io issued in the form:

```
**p(iccddC) 02/05A I/O TRACE
PCWA 33 01 00 02 01 00
FDCWP 313|3
DCWS 000045 0 1 0001
     000001 0 0 0001
```

Where:

p = Polts page #

iccdd = iom/chan/dev

02/05A = test #,line #, per.op. on line

next is IOM PCWA, segment pointer to first dcw, then dcws

. - A control mnemonic follows (see "Control Mnemonics" listed below) and is only valid if it is the first character in the string.

Control Mnemonics

The following control mnemonics (.OPTIONS) are processed only if found at the beginning of the option string:

.GO - Return to the test page where interrupted. If S or Txx has been specified the next test sequencing is done.

.OPT - An Enter Options message is output.

.TAL - A message with a tally of errors is output. P or R must be set and the message includes the information for the current pass or cycle or both depending on the state of P or R or both. The error tallies are reset for the pass or cycle or both when reported.

Sample output:

```
**(01900C) R,P,I
FOR PASS 3:
3 STATUS AND 0 DATA ERRORS
TRANSIENT ERRORS: 1 READ AND 2 WRITE
AND CYCLE 4:
3 STATUS AND 0 DATA ERRORS
TRANSIENT ERRORS: 1 READ AND 2 WRITE
ENTER OPTIONS:
```

.TEST E - The test page is forced terminated.

.TEST W - POLTS is wrapped up

.WAIT - The test page is put in a Wait condition

POLTS MESSAGES

Informative Messages

POLTS LOG ON MESSAGE

This message is issued when POLTS is called initially at command level.

```
***POLTS EXECUTIVE VERSION xxxxxx ON yyyyyy AT zz.zzz
```

Where:

xxxxxx = The POLTS version date in yymmdd

yyyyyy = The current data in yymmdd

zz.zzz = The time in hours and thousands

POLTS LOG OFF MESSAGE

This message is issued when POLTS comes to an orderly (not forced or error caused) conclusion of testing.

```
***POLTS EXECUTIVE VERSION xxxxxx OFF yyyyyy AT zz.zzz
```

Where:

xxxxxx = The POLTS version date in yymmdd

yyyyyy = The current data in yymmdd

zz.zzz = The time in hours and thousands

POLTS FORCED TERM MESSAGE

This message is issued when POLTS terminates due to a "TEST W" or "TEST PW" request, a ".TEST W" option, or "ABT" TDL instruction.

```
***POLTS EXECUTIVE VERSION xxxxxx FORCED TERM yyyyyy AT zz.zzz
rrrrrrrrrrr--
```

Where:

xxxxxx = The POLTS version date in yymmdd

yyyyyy = The current data in yymmdd

zz.zzz = The time in hours and thousands

rrrrrrrrrrr-- is the termination reason:

```
TEST W REQUEST RECEIVED (routine test_request)
.TEST W REQUEST RECEIVED (routine options)
```

POLTS ABORT MESSAGE

This message is issued whenever an internal error is detected by POLTS or POLTS is given a fatal error code from some MULTICS routine call.

```
***POLTS EXECUTIVE VERSION xxxxxx ABORT yyyyyy AT zz.zzz
aaaaaaaaaaaa--
```

Where:

xxxxxx = The POLTS version date in yymmdd

yyyyyy = The current data in yymmdd

zz.zzz = The time in hours and thousands

aaaaaaaaaaaa-- is a formatted abort reason:

NOTE: "system-error-message" below is a formatted message interpreting the returned system error code where the error involved one. Also the (routine xxxx) information indicates the source for each message.

ALLOC--CALLING NON EXISTANT PAGE (routine alloc)

ARG_PTR ERROR (routine test)
system-error-message

CANNOT CHANGE WAIT CHANNEL TO CALL CHANNEL FOR PAGE CLOCK (routine alloc)
system-error-message

CANNOT CHANGE WAIT CHANNEL TO CALL CHANNEL FOR TEST IO (routine alloc)
system-error-message

CANNOT CREATE EVENT CHANNEL FOR PAGE CLOCK (routine alloc)
system-error-message

CANNOT GET CONFIG DECK (routine alloc)
system-error-mesage

CANNOT GET DESCRIPTOR SEG FOR CONFIG DECK (routine alloc)
system-error-message

CANNOT SET CALL PRIORITY OVER WAIT FOR TEST IO (routine alloc)
system-error-message

DEALC--ERROR UNASSIGNING DEVICE (routine dealc_page)
system-error-message

DEALC_PAGE--ERROR DELETING PAGE CLOCK CHANNEL (routine dealc_page)
system-error-message

DEALC_PAGE--ERROR DELETING TEST IO COMPLETION CHANNEL
(routine dealc_page)
system-error-message

ERROR CALLING tttttt (routine test)
NO SUCH TEST PAGE

Where:

tttttt = page called

ERROR CREATING TEST IO EVENT CHANNEL (routine alloc)
system-error-message

ERROR DELETING CLOCK_EVENT (routine alloc)
system-error-message

ERROR DELETING STATUS_EVENT (routine alloc)
system-error-message

ERROR ON PAGE CALL tttttt (routine tpinit)
system-error-message

Where tttttt = page called

ERROR ON TEST I/O STATUS FETCH (routine test_io_ccc)
system-error-message

INIT--CANT WAKEUP REQUEST EVENT (routine test)
system-error-message

IOI_ASSIGN ERROR--COULDNT FIND (routine assign)
system-error-message

NO VACANT PAGE ENTRY FOUND WITH PAGE COUNT <8 (routine alloc)

TDL IMPLEMENTATION ERROR-NON TDL LINE AT NXLIN (routine test)

TDL.TDTYP > 10 IN SET_UP_IO (routine test)

TDL.TDTYP xx ILLEGAL IN DTYPST (routine test)

Where xx = data type

UNASSIGN ERROR (routine assign)
system-error-message

WORKSPACE ASSIGN ERROR (routine assign)
system-error-message

Start Test Execution Message

This message is output whenever a page starts into execution.

```
**p(iccddC) START cccccc nnnn TTLDAT yymmdd
```

Where:

p = Polts page #

iccdd = iom/chan/dev

cccccc = the page "call" name (such as "TD12CA")

nnnn = the POLTS test name (such as AT7A)

yymmdd = The version date in yymmdd

Test Termination Message

This message is output when a test page terminates from execution.

```
**p(iccddC) rrrrrr TERM ccc: sss STATUS AND ddd DATA ERRORS
TRANSIENT ERRORS: rrr READ AND www WRITE
```

Where:

p = Polts page #

iccdd = iom/chan/dev

rrrrrr = term reason (NORMAL,FORCED)

ccc = # number of complete cycles

sss = number of status errors detected during execution

ddd = number of data errors detected during execution

The "TRANSIENT ERRORS" line will appear only if there have been transient errors since the start of the test.

rrr and www = total number of transient read and write errors.

End Cycle Message

```
**p(iccddC) END CYCLE ccc: sss STATUS AND ddd DATA ERRORS
TRANSIENT ERRORS: rrr READ AND www WRITE
```

Where:

p = Polts page #

iccdd = iom/chan/dev

ccc = the number of complete cycles executed

sss = the number of status errors for the last cycle

ddd = the number of data errors for the last cycle

The "TRANSIENT ERRORS" line will appear only if there are accumulated transient errors for the cycle that have not yet been reported.

rrr and www = number of transient read and write errors for the current cycle.

End Pass Message

```
**p(iccddC) END PASS ppp: sss STATUS AND ddd DATA ERRORS
TRANSIENT ERRORS: rrr READ AND www WRITE
```

Where:

p = Polts page #

iccdd = iom/chan/dev

ppp = the number of complete passes executed

sss = the number of status errors for the last pass

ddd = the number of data errors for the last pass

The "TRANSIENT ERRORS" line will appear only if there are accumulated transient errors for the pass that have not yet been reported.

rrr and www = number of transient read and write errors for the current pass.

Inform Message

```
**p(iccddC) END Teee NEXT Tnnn
oooooo
```

Where:

p = Polts page #

iccdd = Iom/chan/dev

eee = the last test executed

nnn = the next test to be executed

oooooo = an appended (ENTER OPTIONS) message if the "H" (halt) option is on

POLTS STANDARD ERROR MESSAGE

The following is the POLTS standard error message which is output whenever an error is detected and where the "B" (bypass) option is not on and where the "NT" modifier is not used.

The following is an example of an error message. Each line usually has several alternatives for either fields or for the whole line, and some lines may not be present. Information of the pages following this one give a detailied breakdown of the error message, line by line.

```
**0(01206C) 42/75B 05-RTB   00/OK 00/OK 00 T/OK LN 040/OK  123456123456
SMB1 010203040506/010203040506 RRC 00/OK 166 DATA ERS D/543210 P/------
(002)012345012345 (003)012345012345 (004)012345012345 (005)012345012345
 S/B 010101010101  S/B 010101010101  S/B 010101010101  S/B 010101010101
IO#105
POSSIBLE TRANSIENT ERROR, RETRY WILL BE MADE
(a message supplied by a NSdd.dd instruction)
DEVICE EXTENDED STATUS IN HEX
(eep message line from TYSdd instruction)
c,c,c,c, ENTER OPTIONS:
```

Line 01 (always present):

```
**0(01206C) 42/75B 05-RTB   00/OK 00/OK 00 T/OK LN 040/OK   123456123456
                            00/03 00/01         LN 040/041 POS 004/OK
                                                LN 040/--- POS 004/---
                                                REC CNT XX POS 004/005
                                                (blanks)   (blanks)
```

```
Field  0             = POLTS test page number
Field  00106         = iom/chan/dev
Field  42            = Test number
Field  75            = Line Number
Field  B             = The number of the device i/o on the line
Field  05            = The octal Op-code
Field  RTB           = The Mnemomic for the op-code
Field  00/OK         = Major Status was/sb if ok
  or   00/05         = if bad
Field  00/OK         = Sub Status was/sb if and if not ignore substatus
   or  00/01         = if bad and if not ignore substatus
   or  00/--         = if ignore substatus
Field  00            = IOM/Channel status
Field  T/OK          = Interrupt returned was/sb if good
   or  T/I           = If bad
Field  LN 040/OK     = Last dcw word count if residual dcw good if
                       data transfer io and not "NR" and not init
                       interrupt
   or  LN 040/041    = if bad and if not "NR"
   or  LN 040/---    = "NR" and not init interrupt
   or  REC CNT 01    = If non-data command
   or  (blanks)      = If Init interrupt and data transfer io
Field  123456123456  = If First i/o of dual io was seek for mass
                       storage
   or  POS 004/ok    = If "POS" positioning in effect for mag tape
                       and positioning good.
   or  POS 004/---   = If "POS" but position cannot be checked due to
                       data error
   or  POS 004/005   = If "POS" and position error
   or  (blanks)      = If not seek or "POS" as above
```

Line 02 (present if not iom fault or timeout):

```
SMB1 012345671234/OK            RRC 00/01    POSITIONING ERROR      ¬¬¬
SMB1 010273040506/010203040506 RRC 00/OK 166 DATA ERS D/543210 P/------
SMB1 010203040506/------------ RRC 00/--     DATA CHKS OK 012345012345
INITIATION INTERRUPT                         DATA NOT CHECKED (INIT)
NON-DATA COMMAND                             DATA NOT CHECKED (NC)
                                             WRITE DATA 123456123456
                                             WRITE DATA XXXXXXXXXXXX
                                             WRITE DATA CHARACTER 00
POWER OFF, OPERATOR INTERVENTION REQUIRED
DEVICE ATTENTION, OPERATOR INTERVENTION REQUIRED
```

```
Field  SMB1 012345671234/OK              = Data xfer/not init interrupt
                                           not "NM" and smbl ok.
   or  SMB1 010203040506/------------ = Same but "NM" used
   or  SMB1 010203040506/010203040507 = Same not "NM" and checks bad
   or  INITIATION INTERRUPT              = If Init interrupt occurred for
                                           data transfer command
   or  NON-DATA COMMAND                  = If Non-data command
Field  RRC 00/01                         = Residual record count was/sb if
                                           error, "NR"and not init.
   or  RRC 00/OK                         = Same but checks good.
   or  RRC 00/--                         = If "NR" or init interrupt
Field  POSITIONING ERROR       ¬¬¬       = If read, data checks good, not
                                           "NC", not initiation interrupt
                                           "POS" with position error.
   or  166 DATA ERS D/543210 P/------ = If read, data bad, not "NC"
                                           not initiation interrupt
                   166 = # number of data words in error.
                   D/543210 P/------ bits dropped and picked in a char.
                   "-" if bit is ok, otherwise # is bit number.

   or  DATA CHKS OK 012345012345         = Read, data ok, not "NC" or
                                           init.  Reports first data word.
   or  DATA NOT CHECKED (INIT)           = If read, not "NC" and init
                                           interrupt.
   or  DATA NOT CHECKED (NC)             = If read and "NC"
   or  WRITE DATA 123456123456           = If write and first dcw not
                                           non-data-transfer.
   or  WRITE DATA XXXXXXXXXXXX           = If write and first dcw
                                           non-data-transfer.
   or  WRITE DATA CHARACTER 00           = If write and ioc cmd. "sing.
                                           char." used.
```

If power off, line 02 is:

```
    POWER OFF, OPERATOR INTERVENTION REQUIRED
```

If unexpected device or mpc attention, line 02 is:

```
DEVICE ATTENTION, OPERATOR INTERVENTION REQUIRED
```

Line 03:

Present only if a read data error is reported in line 02.

```
(002)012345012345 (003)012345012345 (004)012345012345 (005)012345012345
(PRE)012345012345 (FOL)012345012345 (002)012345012345 (003)012345012345
```

If present the (002) or (PRE) reports the location (offset) in the read area of the data word in error if numeric, or that the word preceeding the read transfer(PRE) or following the read transfer (FOL) was bad. The 012345012345 is the data "was".

Line 04:

Present only if a read data error reported in line 2
above and a line 03 "was" is output.

```
 S/B 010101010101  S/B 010101010101  S/B 010101010101  S/B 010101010101
```

If present the data indicates the "s/b" for the word immediately above.

Line 05:

Always present.

```
IO#105
```

This line indicates the absolute # of connects (io's) issued to the device since the test started.

Line 06:

```
UNRECOVERABLE ERROR AFTER 3 RETRIES
POSSIBLE TRANSIENT ERROR, RETRY WILL BE MADE
```

This line is output whenever there is a transient error recovery routine for the test page (mag tapes) and an error is detected for either a read or write. An ioc error, attention or power off, or init interrupt overrides the transient recovery routine and this line.

Line 07:

(a message supplied by a NSdd.dd instruction)

This line is present if a "NSdd.dd" instruction applies to this i/o. the data is the data pointed to by the "NSdd.dd".

Line 08 and 09:

```
DEVICE EXTENDED STATUS IN HEX
EXTENDED STATUS UNREADABLE
```

(eep message line from TYSdd instruction)

```
STATUS WAS 123456654321
```

These lines are present if the "X" eep option is on. If there was no error on fetching the extended status the first of the two alternatives for line 08 and 09 are output. If there was an eep error, then the second alternative for the two is output. For this case the status given is the iom status word #1 in octal format.

Line 10:

```
c,c,c,c, ENTER OPTIONS:
```

This line is output if the "H" (halt) option is on and "NS" or "NSdd.dd" is not used. The TTY pauses for input options.

Extended Status Output Done as Freestanding Test:

One of the following messages is output whenever the extended status test is run as a free standing test via a "Txx" request.

The following message occurs if there were no errors detected while fetching the extended status via a device i/o:

```
**p(iccddC) DEVICE EXTENDED STATUS IN HEX
eeeeee
```

Where:

p = Polts page #

iccdd = iom/chan/dev

eeeee is the formatted extended status line

The following message is output if a status error was detected while fetching extended status:

```
**p(iccddC) EXTENDED STATUS UNREADABLE
STATUS WAS ssssss
```

Where:

p = Polts page

iccdd = iom/chan/dev

sssss = the IOM status return #1 in octal

Input Error Messages

POLTS INVALID INPUT MESSAGE

This message is issued whenever an input following "TEST" contains an error:

```
***POLTS EXECUTIVE (TEST iiiiiiiiiiii) INVALID INPUT
rrrrrrrrrrr--
```

Where:

iiiiiiiiiiii is the input data that is erroneous

rrrrrrrrrrr-- is one of the following reasons:

NOTE: The (routine xxxxxx) information indicates source of the message.

CURRENT OPTIONS NOT PROCESSED YET (routine test_request)
The request which is to put new options in a test page cannot be honored because the pages current options have not yet been processed.

DEVICE NOT CONFIGURED (routine alloc)
An original test request specified a device that was not configured on the system.

INVALID ICCDD (routine test_request)
The request, which should contain an iccdd did not do so or the iccdd input could not be interpreted as one(i.e. not numerics)

INVALID SUB-EXEC CODE (routine tty_ccc)
This message indicates that the first character after "TEST " did not correspond to any legal defined character interpretable by POLTS

NO SUCH ACTIVE TEST PAGE (routine test_request)
The request, which contains an iccdd specifier does not match any current test's iccdd.

NO SUCH PRINTER TRAIN NUMBER (routine assign)
This message is an indication that the MULTICS configuration deck is not proper. The configured printer Train number found for an original test request did not correspond to any known printer train image.

POLTS BUSY--8 REQUESTS OR PAGES ACTIVE (routine alloc)
This message is issued when a new test request is entered and there are already 8 active test pages in POLTS.

USE "TEST XX--" (routine tty_ccc)
This message occurs when the input does not have a "TEST " as the first 5 characters.

UNKNOWN LINE LENGTH (routine alloc)
This message, which is really not an input error but rather a MULTICS configuration deck error is issued whenever an original test request is entered that specifies a type 401/402 printer, but the configuration deck does not have either a 136 or 160 for the printers line length.

UNKNOWN PERIPHERAL (routine alloc)
This message is issued when a new test request is entered but the peripheral "device" or "model" as determined by the MULTICS configuration deck is not known to POLTS.

UNKNOWN REQUEST (routine test_request)
The input is not interpretable

Options Error Messages

This message is typed when options input to the test are processed and some error is detected in those options. A request to enter new options is always appended to this error message.

An input option "Txx" to go to a specific test may be illegal but not detectable at the time that the options are processed. The illegal input will only be detected when the test sequencing code in POLTS attempts to go to that test. For this case, the error will be reported as an INVALID TEST SEQUENCING message as described after this Options Error Message.

```
**p(iccddC) ILLEGAL OPTION: ooooo
rrr--
c,c,c,c,c, ENTER OPTIONS:
```

Where:

p = Polts page #

iccdd = iom/chan/dev

ooooo = The illegal option

c,c,c - current options

rrr-- = any of the following reasons:

"." MUST BE THE FIRST OPTION CHARACTER
A "." was detected as other than the first option character.

CANT TURN OFF FORCED TERM TEST
An "NTxx" was input where "xx" was the number of the present segments forced termination test.

CANT TURN OFF INITIALIZATION TEST IN SEGMENT
An "NTxx" was input where "xx" was the pages "initialization" test (not in the test table--line 0).

CANT TURN OFF TEST IN ANOTHER SEGMENT
An "NTxx" was input where the test number "xx" does not refer to any test contained in the present test segment(assumed to exist in another segment although there may be no such other segment).

EEP ILLEGAL IF NOT MPC OR MANUAL
An "X" or "NX" option was entered for a test page which is not for an MPC type device or the current test page is a manual test page.

ILLEGAL CONTROL MNEMONIC (.OPTION) ENCOUNTERED
An option starting with "." was entered which does not correspond to any option known to POLTS.

NONNUMERICS INPUT FOR LINE NUMBER FOR CONTROL MNEMONIC
Either no character or some character other than 0-9 appears after the alpha portion of one of the control mnemonics ".SEQnn", .STRnn", -- ect.

ONLY A "1" IS ALLOWED FOLLOWING "E-"
The numeral "1" must follow "E-".

OPTION NOT IMPLEMENTED
Some option defined in the GCOS POLTS implementation has not been incorporated in the MULTICS POLTS implementation. Retained for compatability information.

OPTIONS ILLEGAL AFTER (.OPTION)
After validating that a .OPTION was input it was determined that more characters were input than are defined for that option. Only one control mnemonic (.OPTION) may be input for any single options input.

PASS OR RECYCLE MUST BE SAT TO OUTPUT ERROR TALLIES
A .TAL control mnemonic was entered and neither the "p" or "R" option had been set for the test page.

TEST NUMBER CANNOT BE "0"
A "Txx" or "NTxx" where "xx" was "0" or "00" was input.

TEST NUMBER MUST FOLLOW "T"
A "T" or "NT" was input where there was either no following character or that character was not 0-9.

UNKNOWN OPTION
Some string of characters was input other than a "." that cannot be interpreted as an option (not an invalid control mnemonic).

Invalid Test Sequencing Message

```
**p(iccddC) INVALID TEST SEQUENCING
 rrrr--
 c,c,c,c, ENTER OPTIONS:
```

Where:

p = Polts page #

iccdd = iom/chan/dev

c,c,c, = current options

rrrr-- is one of the following reasons:

TRYING TO JUMP TO A TEST NOT IN CURRENT SEQUENCE
During test sequencing, a "jump" condition was detected that specifies a test not defined in the test table for the current sequence.

NO EXECUTABLE TESTS IN THIS SEQUENCE
During test sequencing, no test could be found in the current test sequence that could be executed because all of them were turned off.

Invalid TDL Instruction

During sequential processing of the TDL instructions, an unknown TDL mnemonic or erroneous TDL instruction usage was encountered.

```
**p(iccddC) INVALID TDL INSTRUCTION, LINE pp, FIELD , aaannn
rrr--
c,c,c,c, ENTER OPTIONS:
```

Where:

p	=	Polts page #
iccdd	=	iom/chan/dev
xx	=	line number
yy	=	field number
aaa	=	isolated alpha part of instruction
nnn	=	isolated numeric part of instruction
c,c,c,	=	current options

rrr-- is one of the following reasons:

> 6 ALPHA OR >12 NUMBERS
More than 12 numbers (0-9) or more than 6 non-numerics other than a "," or space were encountered on a TDL instruction line with no intervening "," or space.

CANNOT SPECIFY AN IOC NON-DATA COMMAND
An "IC01" (iom channel command "02") was encountered.

CANNOT USE TDCW OR IDCW AS FIRST DCW
Either "CW0Bn" or ""CW0I" was encountered.

DLN DATA IS NOT ALL FROM TEST DATA LINE
Detected at data setup time after some prior "DLNxx" was used. With the given dcw length, the line "xx" or some following line, if the length requires the use of following lines, was found to be flagged as other than an octal data line.

EEP DATA LINE NOT ASCII DATA
The line number pointed to by "ITALxx" was not flagged as an ascii data line.

EEP INSTRUCTIONS ILLEGAL IN MANUAL MODE
Any of the instructions used for extended status processing such as "ITALxx" or "RETEP" was encountered in a manual test page during TDL instruction processing.

END OF LINE SEQUENCING WOULD PROCEED ON A NON-TDL LINE
The end of the current TDL instruction line was encountered and the last instruction processed required continuing to the next successive line and that latter line was found to be flagged as other than a TDL instruction line in the test page.

ILLEGAL USE OF TDCW (CWXBY), TWO TDCWS IN A ROW
During dcw tracing for io setup, two successive TDCWS were encountered.

INSUFFICIENT FIXED NUMERICS
A TDL instruction was isolated that had less than the required number of numerics that are defined for the instruction.

INVALID HEXIDECIMAL CHARACTER IN UHDLN OR PHDLN
Detected during data setup possibly after the "UHDLNxx" or "PHDLNxx" instruction. Some character other than 0-9 or a-f was detected in the lines covered by the range of the TDL instruction considering the current length.

LAST WORD OF DCW WOULD BE OUTSIDE OF WRITE AREA
An "Ldd" or control word instruction such as "CWdSddd.ddd" was encountered such that the resulting last word of the of the transfer area defined by the starting offset and length would be outside of the data transfer area (considered to be 320 maximum).

LAST WORD OF MESSAGE WOULD BE OUTSIDE OF MESSAGE AREA
An instruction to output a message line such as "TYdd.dddd" was made with a length of >72 characters.

LCW POINTS TO TDCW
An "LCWd" was encountered where dcw #d is a TDCW "CWdBd".

LOOPING TO NON_TDL_LINE
A loop type instruction was used that would cause a loop to a line that was not flagged as a TDL instruction line.

MORE NUMERICS THAN DEFINED FOR INSTRUCTION
A TDL instruction was encountered with more numbers than are defined for the instruction.

OBSOLETE INSTRUCTION, NOT IMPLEMENTED
A TDL instruction was encountered that was valid for the GCOS POLTS implementation that is not legal for the MULTICS POLTS implementation. Retained for compatability indications.

ONLY 3 NUMBERS, EACH 0 OR 1 PERMITTED IN "TIS"
A "TISooo" was encountered that had either more than 3 numbers or one of the numbers was other than a 0 or 1.

ONLY 36 FLAGS AVAILABLE, 0-35
A "SFLxx", "CKFxx" or RFLxx" instruction was encountered where "xx" was >36.

ONLY OCTAL NUMERICS ALLOWED
A TDL instruction was encountered that must have only octal numerics (0-7) and an 8 or 9 was found to be in the instruction.

OP-CODE <0 IN PERIF--SHOULD NOT HAPPEN
The peripheral op-code type was found to be <0 at such a place that it should not have occured--implementation error.

POSITIONING VALID ONLY FOR MAG TAPE
A "POS" instruction was found in a test page for other than a mag tape.

RECORD COUNT MUST BE 0-63
An "Rdd" instruction was encountered with dd>63.

RESIDUAL WORD COUNT CANNOT EXCEED 320
An "RWdd" was encountered with dd>320.

RETEP ILLEGAL IF NOT IN EEP
A "RETEP" instruction was encountered where POLTS was not in an "automatic" extended status fetch mode (not fetching extended status due to the X option after an error).

TDL LANGUAGE LOCKUP FAULT, NO IO FOR 275 MAJOR INSTRUCTIONS
Greater than 275 TDL instructions were isolated without ever causing a test io to be issued.

TRANSIENT ERROR RECOVERY SUBROUTINE IS A NON_TDL LINE
When an attempt was made to enter either a read or write transient error recovery subroutine, the line pointed to by the "TRRdd" or "TWRdd" instruction was found to not be a TDL instruction line (can happen if the "TRRdd" or "TWRdd" instructions were not used for the current page.

TRYING TO ADD OFFSET OR LENGTH TO A TDCW OR IDCW
A "CWdAddd.dddd" instruction was encountered that pointed to a TDCW (CWdBd") or IDCW (CWdI").

TRYING TO EXECUTE A RET WITHOUT A PREVIOUS SV
A "RETd" was encountered without a "SVd" having been used prior to the "RETd".

TRYING TO ISSUE ILLEGAL OP-CODE TO DEVICE
A peripheral op-code defined as illegal for the device under test was found resulting from a peripheral mnemonic such as "REW" or special i/o instruction such as "SWoo".

TRYING TO SET UP THE LAST DCW AS AN IONTP OR IOTP
A CWdPddd.dddd" or "CWdNddd.dddd" was encountered that would select dcw #9.

TRYING TO TYPE 4096 WORDS (0)
A TDL instruction to type message data such as "TYdd.dddd" was encountered that specified a word count of 0.

TRYING TO TYPE FROM NON_ASCII DATA LINE
A TDL instruction to type message data such as "TYdd.dddd" was encountered that subtends TDL lines that are not flagged as ascii data type lines.

TRYING TO USE A DCW WITH WORD COUNT OF 4096 (0)
A dcw setup such as via "CWdSddd.dddd" was attempted that had a word count of 0.

TRYING TO USE A SING. CHAR. IOC CMD WITH A READ OR NONDATA OPCODE
During i/o setup to issue a test i/o, it was found that an "IC04" instruction has been used but the peripheral command is a read or non-data type op-code.

UNKNOWN IOC COMMAND OR BOOTLOAD
An illegal IOM channel command or "BOOT" was specified in an "ICoo" instruction.

UNKNOWN MNEMONIC
The TDL instruction isolated was unknown to POLTS

USE OF TDCW (CWXBY) HAS CAUSAD DCW STRING LOOP WITHOUT IOTD (CWXS)
During the tracing of DCW setup before issuing a test i/o, more than 10 dcws were traced without encountering a stop dcw. This can only occur if the dcws are setup in such a form as to "loop".

WRONG IOC COMMAND USED WITH LOC DATA TYPE
During i/o setup to issue a test i/o a data type of "LOC" was encountered without an iom command "IC04".

POLTS EMERGENCY MESSAGES

In some cases, there may be certain types of errors detected by POLTS that there is no reliable way to guarantee that the message can be output by the normal TTY handling routines in POLTS. These errors are related to the execution of thosa TTY handling routines so that some other method of outputting these errors must be used.

The method used to output these error indications is to use the MULTICS routine "com_err_" which directly writes out an error message over the stream "error_output_". This message does not use the TTY completion wait logic of POLTS so that if it is interrupted by a "QUIT" condition, the message may be lost. The message typed out will be the last one so typed and POLTS will immediately return to command level without a "sign off" message. It is possible, however, for several of these "emergency" messages to appear one after the other if multiple problems are encountered.

The various emergency messages and the routine from which they came are listed below. Because these are system errors that must be analysed by a cognizant programmer their significance is not interpreted.

FROM TEST_INIT:

```
CANNOT CHANGE QUITH_EVENT TO A CALL CHANNEL
CANNOT CHANGE TTY_CCC_EVENT TO A CALL CHANNEL
CANNOT CHANGE TTY_ISSUE_EVENT TO A CALL CHANNEL
CANNOT GET TTY SPEED
CANNOT CREATE QUITH_EVENT
CANNOT CREATE TTY_CCC_EVENT
CANNOT CREATE TTY_ISSUE_EVENT
CANNOT SET CALL CHANNELS PRIORITY OVER WAIT
ERROR CREATING DUMMY_EVENT
ERROR DELETING QUITH_EVENT
ERROR DELETING TTY_CCC_EVENT
ERROR DELETING TTY_ISSUE_EVENT
ERROR DELETING WAIT_EVENT_ID
UNKNOWN TTY BAUD RATE 111
```

FROM TEST_TERM(part of test):

```
     ERROR DELETING QUITH_EVENT
     ERROR DELETING TTY_CCC_EVENT
     ERROR DELETING TTY_ISSUE_EVENT
     ERROR DELETING WAIT_EVENT_ID
```