```
UUU         UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP        SSSSSSSSSSSS  YYY         YYY
UUU         UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP        SSSSSSSSSSSS  YYY         YYY
UUU         UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP        SSSSSSSSSSSS  YYY         YYY
UUU         UUU  EEE                    TTT        PPP         PPP  SSS              YYY         YYY
UUU         UUU  EEE                    TTT        PPP         PPP  SSS              YYY         YYY
UUU         UUU  EEE                    TTT        PPP         PPP  SSS              YYY         YYY
UUU         UUU  EEE                    TTT        PPP         PPP  SSS                 YYY   YYY
UUU         UUU  EEE                    TTT        PPP         PPP  SSS                 YYY   YYY
UUU         UUU  EEE                    TTT        PPP         PPP  SSS                 YYY   YYY
UUU         UUU  EEEEEEEEEEEE           TTT        PPPPPPPPPPPP        SSSSSSSSS           YYY
UUU         UUU  EEEEEEEEEEEE           TTT        PPPPPPPPPPPP        SSSSSSSSS           YYY
UUU         UUU  EEEEEEEEEEEE           TTT        PPPPPPPPPPPP        SSSSSSSSS           YYY
UUU         UUU  EEE                    TTT        PPP                          SSS        YYY
UUU         UUU  EEE                    TTT        PPP                          SSS        YYY
UUU         UUU  EEE                    TTT        PPP                          SSS        YYY
UUU         UUU  EEE                    TTT        PPP                          SSS        YYY
UUU         UUU  EEE                    TTT        PPP                          SSS        YYY
UUU         UUU  EEE                    TTT        PPP                          SSS        YYY
UUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE        TTT        PPP              SSSSSSSSSSSS           YYY
UUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE        TTT        PPP              SSSSSSSSSSSS           YYY
UUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE        TTT        PPP              SSSSSSSSSSSS           YYY
```

```
**FILE**ID**SATSSF15                                  C 14
```

```
  SSSSSSSS    AAAAAA    TTTTTTTTTT    SSSSSSSS    SSSSSSSS  FFFFFFFFFF      11      5555555555
  SSSSSSSS    AAAAAA    TTTTTTTTTT    SSSSSSSS    SSSSSSSS  FFFFFFFFFF      11      5555555555
SS          AA      AA      TT      SS          SS          FF            1111      55
SS          AA      AA      TT      SS          SS          FF            1111      55
SS          AA      AA      TT      SS          SS          FF              11      555555
SS          AA      AA      TT      SS          SS          FF              11      555555
  SSSSSS    AA      AA      TT        SSSSSS      SSSSSS    FFFFFFFF        11            55
  SSSSSS    AA      AA      TT        SSSSSS      SSSSSS    FFFFFFFF        11            55
        SS  AAAAAAAAAA      TT              SS          SS  FF              11              55
        SS  AAAAAAAAAA      TT              SS          SS  FF              11              55
        SS  AA      AA      TT              SS          SS  FF              11      55      55      ....
        SS  AA      AA      TT              SS          SS  FF              11      55      55      ....
SSSSSSSS    AA      AA      TT      SSSSSSSS    SSSSSSSS    FF            111111      555555        ....
SSSSSSSS    AA      AA      TT      SSSSSSSS    SSSSSSSS    FF            111111      555555        ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
0000            .TITLE  SATSSF15 - SATS SYSTEM SERVICE TESTS  (FAILING S.C.)
0000            .IDENT  'V04-000'
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000
0000    ;++
0000    ; FACILITY:     SATS SYSTEM SERVICE TESTS
0000    ;
0000    ; ABSTRACT:     THE SATSSF15 MODULE TESTS THE EXECUTION OF CERTAIN
0000    ; VMS SYSTEM SERVICES, INVOKED IN SUCH A WAY AS TO EXPECT FAILING
0000    ; STATUS CODES. THE SYSTEM SERVICES TESTED AND THE STATUS CODES
0000    ; EXPECTED ARE SUMMARIZED AS ARGUMENTS TO THE TESTSERV MACROS
0000    ; WHICH APPEAR NEAR THE END OF THIS LISTING. SUCCESSFUL STATUS
0000    ; CODES ARE TESTED IN OTHER MODULES.
0000    ;
0000    ;
0000    ; ENVIRONMENT:  USER MODE IMAGE; NEEDS CMKRNL PRIVILEGE,
0000    ;               DYNAMICALLY ACQUIRES OTHER PRIVILEGES, AS NEEDED.
0000    ;
0000    ; AUTHOR: THOMAS L. CAFARELLA,         CREATION DATE: MMM, 1978
0000    ;         PAUL D. FAY (DISPSERV & TESTSERV MACROS)
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       , : VERSION
0000    ; 01    -
0000    ;--
```

```
                  0000              .SBTTL  DECLARATIONS
                  0000      ;
                  0000      ; INCLUDE FILES:
                  0000      ;
                  0000              $PHDDEF                                  ; PROCESS HEADER OFFSET SYMBOLS
                  0000              $PCBDEF                                  ; PROCESS CONTROL BLOCK OFFSET SYMBS
                  0000              $STSDEF                                  ; STATUS MESSAGE SYMBOLS
                  0000              $PRVDEF                                  ; SYMBOL DEFS FOR PRIVILEGES
                  0000              $UETPDEF                                 ; UETP MSG CODE DEFINITIONS
                  0000              $SHR_MESSAGES UETP,116,<<TEXT,INFO>>
                  0000                                                       ; DEFINE UETP$ TEXT
                  0000                                                       ; GET RID OF MACRO DEFINITIONS
                  0000              $PSLDEF                                  ; ACCESS MODE SYMBOLS
                  0000      ;
                  0000      ; MACROS:
                  0000      ;
                  0000      ;
                  0000      ; EQUATED SYMBOLS:
                  0000      ;
        00000000  0000      WARNING         = 0                              ; WARNING SEVERITY VALUE FOR MSGS
        00000001  0000      SUCCESS         = 1                              ; SUCCESS SEVERITY VALUE FOR MSGS
        00000002  0000      ERROR           = 2                              ; ERROR SEVERITY VALUE FOR MSGS
        00000003  0000      INFO            = 3                              ; INFORMATIONAL SEV VALUE FOR MSGS
        00000004  0000      SEVERE          = 4                              ; SEVERE (FATAL) SEV VALUE FOR MSGS
        00000000  0000      TCG_NO          =  0                             ; INITIALIZE TEST CASE GROUP NUMBER
        00000000  0000      GRP_TOTAL       =  0                             ; INITIALIZE TEST CASE GROUP TOTAL
        00007FFF  0000      R0_THRU_SP      = ^M<R0,R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,AP,FP,SP>
        00000001  0000      PRVHND_DCH20    = 1                              ; PRVHND ARG FOR DCLCMH (LOCATION 1)
                  0000      ;
                  0000      ; OWN STORAGE:
                  0000      ;
```

```
                 00000000           .PSECT  RODATA,RD,NOWRT,NOEXE,LONG
             BFFC  0000  REG_COMP_MASK:  .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,AP,FP> ! ^X8000 -
                   0002                                                  ; REG COMPARE MASK (HIGH-ORDER ...
                   0002                                                  ; ... BIT MUST BE ON
                   0002  ERR_MSG_FAOCTL: STRING  I,<!/!AC!1ZB!1ZB: REGISTER !2UW CONTENTS ALTERED>, -
                   0002                  <;  BEFORE SERVICE CALL: !8XL   AFTER SERVICE CALL: !8XL>
                   006E  TEST_MOD_NAME:  STRING  C,<SATSSF15>      ; TEST MODULE NAME
                   0077  TEST_MOD_BEG:   STRING  C,<begun>         ; DISPOSITION FIELD OF TEST MOD MSG
                   007D  TEST_MOD_SUCC:  STRING  C,<successful>    ; DISPOSITION FIELD OF TEST MOD MSG
                   0088  TEST_MOD_FAIL:  STRING  C,<failed>        ; DISPOSITION FIELD OF TEST MOD MSG
                   008F  TEST_MOD_NAME_D: STRING I,<SATSSF15>      ; TEST MODULE NAME DESCRIPTOR
                   009F  TTNAME:         STRING  I,<TT>            ; TERMINAL LOGICAL NAME
00000000'00000000' 00A9  INADR:          .LONG   NOACCESS,NOACCESS ; PAGE ADDRESS OF NOACCESS PSECT
         00000000' 00B1  PROT:           .LONG   PRT$C_NA          ; PROTECTION CODE FOR NOACCESS PSECT
FFFFFFFF FFFFFFFF  00B5  ONES:           .LONG   -1,-1             ; A QUADWORD OF 1-BITS
                   00BD  ADDRES_DCH:                               ; ADDRES ARGUMENT FOR DCLCMH
             0000  00BD                  .WORD   0                 ; ENTRY MASK FOR DCLCMH SERVICE
               02  00BF                  REI                       ; RETURN INSTRUCTION FOR DCLCMH SERV
         000000C4  00C0  PRVHND_DCH21:   .BLKL   1                 ; PRVHND ARGUMENT FOR DCLCMH
         00000000  00C4  TYPE_DCH:       .LONG   0                 ; TYPE ARGUMENT FOR DCLCMH SERVI(E
         00000003  00C8  ACMODE_ADS:     .LONG   PSL$C_USER        ; ACMODE ARGUMENT FOR ADJSTK
         00000002  00CC  ACMODE_ADS10:   .LONG   PSL$C_SUPER       ; ACMODE ARGUMENT FOR ADJSTK
         00000001  00D0  ACMODE_ADS11:   .LONG   PSL$C_EXEC        ; ACMODE ARGUMENT FOR ADJSTK
         00000000  00D4  ACMODE_ADS12:   .LONG   PSL$C_KERNEL      ; ACMODE ARGUMENT FOR ADJSTK
         00000000  00D8  ADJUST_ADS:     .LONG   0                 ; ADJUST ARGUMENT FOR ADJSTK
```

```
  00000000                 .PSECT  RWDATA,RD,WRT,NOEXE
00000004  0000  TPID:              .BLKL    1                ; PROCESS ID FOR THIS PROCESS
00000008  0004  CURRENT_TC:        .BLKL    1                ; PTR TO CURRENT TEST CASE
00000044  0008  REG_SAVE_AREA:     .BLKL    15               ; SAVE AREA FOR ALL REGS (SANS PC)
007480D9  0044  MOD_MSG_CODE:      .LONG    UETP$_SATSMS     ; TEST MODULE MSG CODE FOR PUTMSG
0000004C  0048  CLOB_REG_NO:       .BLKL    1                ; CLOBBERED REG NO (FOR FAO ERR MSG)
00000050  004C  REG_BEFORE_SS:     .BLKL    1                ; REG CONTENTS BEFORE S.S.
          0050                                               ; ... (FOR FAO ERROR MSG)
00000054  0050  REG_AFTER_SS:      .BLKL    1                ; REG CONTENTS AFTER S.S.
          0054                                               ; ... (FOR FAO ERROR MSG)
          0054  $$TSTN$$:          STRING  C,<  SF    >      ; ASCII PORTION OF TEST CASE NAME
0000006E' 005C  TMN_ADDR:          .ADDRESS TEST_MOD_NAME    ; ADDR OF TEST MOD NAME FOR FAO
00000077' 0060  TMD_ADDR:          .ADDRESS TEST_MOD_BEG     ; ADDR OF T.M. DISP FIELD FOR FAO
00000068  0064  TS_EP:             .BLKL    1                ; ENTRY PNT FOR CURR TESTSERV MACRO
00000070  0068  RETADR:            .BLKL    2                ; RETURN LONGWORDS FOR SETPRT
00000071  0070  PRVPRT:            .BLKB    1                ; PROT RETURN BYTE FOR SETPRT
00000079  0071  PRIVMASK:          .BLKQ    1                ; ADDR OF PRIVILEGE MASK (IN PHD)
0000007D  0079  CHM_CONT:          .BLKL    1                ; CHANGE MODE CONTINUE ADDRESS
00000091  007D  REGS:              .BLKL    5                ; AREA FOR COND INDEX REGS (R2-R6)
00000095  0091  PRVHND_DCH:        .BLKL    1                ; PRVHND ARGUMENT FOR DCLCMH SERVICE
          0095  DESBLK_CEH:                                  ; DESBLK ARGUMENT FOR CANEXH
          0095  DESBLK_CEH10:                                ; DESBLK ARGUMENT FOR CANEXH
00000099  0095                     .BLKL    1                ; EXIT CONTROL BLOCK (LINK PTR)
000000A9' 0099                     .ADDRESS 20$              ; ADDRESS OF ROUTINE ENTRY MASK
00000001  009D                     .LONG    1                ; ARGUMENT COUNT
000000A5' 00A1                     .ADDRESS 10$              ; ADDR OF REASON CODE FIELD
000000A9  00A5  10$:               .BLKL    1                ; REASON CODE FIELD
    0000  00A9  20$:               .WORD    0                ; EXIT HANDLER ENTRY MASK
      04  00AB                     RET                       ; EXIT HANDLER RETURN INSTR
00000000  00AC  NEWADR_ADS:        .LONG    0                ; NEWADR ARGUMENT FOR ADJSTK SERVICE
```

```
    00000000          .PSECT  SATS_ACCVIO_1,RD,WRT,NOEXE,PAGE
00000200  0000  EMPTY:          .BLKB   512     ; RESERVE A PAGE OF SPACE
          0200  ;
          0200  ; +
          0200  ; *****************************************************************
          0200  ; *                                                               *
          0200  ; *      THE ORDER OF STATEMENTS IN THIS PSECT IS CRITICAL.       *
          0200  ; *      DO NOT RE-ARRANGE THE VARIABLES. CONSULT SATS            *
          0200  ; *      FUNCTIONAL SPECIFICATION FOR A DESCRIPTION OF THE USE    *
          0200  ; *      OF THE EMPTY PSECT (AND ITS COMPANION PSECT, NOACCESS).  *
          0200  ; *                                                               *
          0200  ; *****************************************************************
          0200  ; -
          0200  ;
          0200  ; TYPE AAAAA_SSSX1 (TYPE AAAAA_SSSX2 IF NOT DESC) GO HERE:
000001FF  0200  PRVHND_DCH22    = . - 1         ; PRVHND ARG FOR DCLCMH (LAST BYTE IN PAGE
000001F3  0200                  . = . - 13      ; ALLOW ROOM FOR STRING DESCRIPTOR
          01F3  ; TYPE AAAAA_SSSX5 GO HERE:
00000006  01F3                  .LONG   6       ; STRING LENGTH (WILL CROSS PSECT BOUNDARY)
000001FB' 01F7                  .ADDRESS .+4    ; STRING ADDRESS
          01FB  ; TYPE AAAAA_SSSX3 GO HERE:
000001FC  01FB                  .BLKB   1       ; LOW-ORDER BYTE OF STRING LENGTH
          01FC  ; TYPE AAAAA_SSSX2 GO HERE:
00000200  01FC                  .BLKL   1       ; STRING LENGTH
          0200  ;
          0200  ;
          0200  ;
          0200  ;
    00000000          .PSECT  SATS_ACCVIO_2,RD,WRT,NOEXE,PAGE
00000200  0000  NOACCESS:       .BLKB   512     ; RESERVE A PAGE OF SPACE
00000000  0200                  . = . - 512     ; RETURN LOC CTR TO BEGINNING OF PSECT
00000000' 0000                  .ADDRESS EMPTY  ; ADDRESS OF ACCESSIBLE STRING
00000000' 0004                  .ADDRESS EMPTY/^X100 ; ADDRESS OF ACCESSIBLE STRING
          0008  ;+
          0008  ; *** NOTE -- DO NOT CHANGE LOCATION OR SEQUENCE OF ABOVE STATEMENTS!
          0008  ; ***         THIS PSECT (NOACCESS) MUST APPEAR IN MEMORY IMMEDIATELY
          0008  ; ***         FOLLOWING THE EMPTY PSECT. PSECT NAMES AND OPTIONS WILL BE
          0008  ; ***         CHOSEN TO FORCE THE DESIRED PSECT ORDERING.
          0008  ;-
          0008  ;
          0008  ;
          0008  ;
          0008  ;
    00000000          .PSECT  SATSSF15,RD,WRT,EXE,LONG
```

```
0000           .SBTTL  SATSSF15
0000     ;++
0000     ; FUNCTIONAL DESCRIPTION:
0000     ;
0000     ;               AFTER PERFORMING SOME INITIAL HOUSEKEEPING, SUCH AS
0000     ; PRINTING THE MODULE BEGIN MESSAGE AND ACQUIRING ALL PRIVILEGES,
0000     ; THE SATSSF15 ROUTINE EXECUTES THE TEST_SERV_EXEC MACRO TO RUN
0000     ; ALL TEST CASES. WHEN THE MACRO COMPLETES ITS EXECUTION, SATSSF15
0000     ; PRINTS A TEST MODULE SUCCESS OR FAIL MESSAGE AND EXITS TO THE
0000     ; OPERATING SYSTEM. TEST_SERV_EXEC CALLS THE TC_CONTROL/TESTSERV
0000     ; CO-ROUTINE PAIR ONCE PER TEST CASE GROUP TO EXECUTE ALL TEST
0000     ; CASES IN THAT GROUP. EACH TEST CASE GROUP IS DEFINED BY BOUNDING
0000     ; ITS TEST CASES WITH A TC_GROUP MACRO BEFORE THE FIRST TEST CASE
0000     ; AND A TCEND MACRO AFTER THE LAST ONE. THE TEST CASES THEMSELVES
0000     ; ARE DEFINED WITHIN THESE BOUNDS BY PRECEDING EACH WITH A
0000     ; NEXT_TEST_CASE MACRO. TC_CONTROL/TESTSERV EXECUTES THE CODE
0000     ; FOLLOWING EACH NEXT_TEST_CASE MACRO IMMEDIATELY BEFORE ISSUING
0000     ; THE SYSTEM SERVICE AS REQUESTED IN THE TESTSERV MACRO. TC_CONTROL/
0000     ; TESTSERV ALSO CHECKS THE RESULTS OF THE SERVICE WITH RESPECT
0000     ; TO ITS EXPECTED STATUS CODE AND PRINTS ANY REQUIRED FAILURE
0000     ; MESSAGES FOR THE TEST CASE. THE CODE APPEARING AFTER EACH
0000     ; NEXT_TEST_CASE MACRO IS MERELY TO SET UP CONDITIONS REQUIRED
0000     ; FOR THE SYSTEM SERVICE AND TO CLEAN UP ANY RESOURCES ACQUIRED
0000     ; BY THE PREVIOUS TEST CASE.
0000     ;
0000     ; CALLING SEQUENCE:
0000     ;
0000     ;       $ RUN SATSSF15  ...  (DCL COMMAND)
0000     ;
0000     ; INPUT PARAMETERS:
0000     ;
0000     ;       NONE
0000     ;
0000     ; IMPLICIT INPUTS:
0000     ;
0000     ;       NONE
0000     ;
0000     ; OUTPUT PARAMETERS:
0000     ;
0000     ;       NONE
0000     ;
0000     ; IMPLICIT OUTPUTS:
0000     ;
0000     ;       MESSAGES TO SYS$OUTPUT ARE THE ONLY OUTPUT FROM SATSSF15.
0000     ;       THEY ARE OF THE FORM:
0000     ;
0000     ;               %UETP-S-SATSMS, TEST MODULE SATSSF15 BEGUN ... (BEGIN MSG)
0000     ;               %UETP-S-SATSMS, TEST MODULE SATSSF15 SUCCESSFUL ... (END MSG)
0000     ;               %UETP-E-SATSMS, TEST MODULE SATSSF15 FAILED ... (END MSG)
0000     ;               %UETP-I-TEXT, ... (VARIABLE INFORMATION ABOUT A TEST MODULE FAILURE)
0000     ;
0000     ; COMPLETION CODES:
0000     ;
0000     ;       THE SATSSF15 ROUTINE TERMINATES WITH A $EXIT TO THE
0000     ;       OPERATING SYSTEM WITH A STATUS CODE DEFINED BY UETP$_SATSMS.
0000     ;
0000     ; SIDE EFFECTS:
```

```
                                 0000  ;
                                 0000  ;       NONE
                                 0000  ;
                                 0000  ;--
                                 0000
                                 0000
                                 0000
                                 0000  SATSSF15:
                           0FFC  0000          .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                                 0002                                                  ; ENTRY MASK
                                 0002          $WAKE_S  TPID                           ; GET PID OF THIS PROCESS
                                 0011          $HIBER_S                                ; UNDO WAKE
                                 0018          $SETPRN_S TEST_MOD_NAME_D               ; SET PROCESS NAME
                   0914    30    0025          BSBW    MOD_MSG_PRINT                   ; PRINT TEST MODULE BEGIN MSG
   00000060'EF  0000007D'EF  DE  0028          MOVAL   TEST_MOD_SUCC,TMD_ADDR          ; ASSUME END MSG WILL SHOW SUCCESS
  00000044'EF   03   00   01 F0  0033          INSV    #SUCCESS,#0,#3,MOD_MSG_CODE ; ADJUST STATUS CODE FOR SUCCESS
                                 003C          MODE    TO,10$,KRNL,NOREGS              ; KERNEL MODE TO ACCESS PHD
           59   00000000'9F  D0  0059          MOVL    @#CTL$GL_PHD,R9                 ; GET PROCESS HEADER ADDRESS
        00000071'EF    69    DE  0060          MOVAL   PHD$Q_PRIVMSK(R9),PRIVMASK ; GET PRIV MASK ADDRESS
                                 0067          MODE    FROM,10$                        ; GET BACK TO USER MODE
                                 0068          PRIV    ADD,ALL                         ; GET ALL PRIVILEGES
                                 0088          DISPSERV                                ; SET UP DISPLAY INFO FOR TESTSERV
                                 021D          $SETPRT_S INADR=INADR, RETADR=RETADR, -
                                 021D                  PROT=PROT, PRVPRT=PRVPRT
                                 023E                                                  ; SET NOACCESS PSECT ...
                                 023E                                                  ; ... FOR NO USER ACCESS
                   05A2    31    023E          BRW     EXECUTE                         ; GO EXECUTE ALL TEST CASES
                                 0241  ;
                                 0241          TC_GROUP        DCH,1,TS1
                                 0268  ;
                                 0268          NEXT_TEST_CASE  SFDCH20
```

```
0268  ;
0268  ;++
0268  ; ********************************************************
0268  ; *
0268  ; *  TEST CASE NAME:           SFDCH20
0268  ; *
0268  ; *  SYSTEM SERVICE:           DCLCMH
0268  ; *
0268  ; *  ARGUMENT UNDER TEST:      PRVHND_DCH20
0268  ; *
0268  ; *  INPUT CONDITIONS:
0268  ; *     PREVIOUS HANDLER ADDRESS BUFFER AT LOCATION 1.
0268  ; *
0268  ; *  EXPECTED RESULTS:
0268  ; *     1) SYSTEM STATUS CODE:  ACCVIO
0268  ; *     2) REGISTERS R2 THROUGH FP UNCHANGED
0268  ; *
0268  ; ********************************************************
0268  ;--
0268  ;
0268  ;
0268          NEXT_TEST_CASE  SFDCH21
```

```
0274    ;
0274    ;++
0274    ; ****************************************************
0274    ; *
0274    ; *  TEST CASE NAME:           SFDCH21
0274    ; *
0274    ; *  SYSTEM SERVICE:           DCLCMH
0274    ; *
0274    ; *  ARGUMENT UNDER TEST:      PRVHND_DCH21
0274    ; *
0274    ; *  INPUT CONDITIONS:
0274    ; *     PREVIOUS HANDLER ADDRESS BUFFER IN READ-ONLY PSECT.
0274    ; *
0274    ; *  EXPECTED RESULTS:
0274    ; *     1) SYSTEM STATUS CODE:  ACCVIO
0274    ; *     2) REGISTERS R2 THROUGH FP UNCHANGED
0274    ; *
0274    ; ****************************************************
0274    ;--
0274    ;
0274    ;
0274            NEXT_TEST_CASE  SFDCH22
```

```
0280    ;
0280    ;++
0280    ; ******************************************************
0280    ; *
0280    ; *  TEST CASE NAME:           SFDCH22
0280    ; *
0280    ; *  SYSTEM SERVICE:           DCLCMH
0280    ; *
0280    ; *  ARGUMENT UNDER TEST:      PRVHND_DCH22
0280    ; *
0280    ; *  INPUT CONDITIONS:
0280    ; *     PREVIOUS HANDLER ADDRESS BUFFER BEGINS IN ACCESSIBLE
0280    ; *       PSECT, ENDS IN NON-ACCESSIBLE PSECT.
0280    ; *
0280    ; *  EXPECTED RESULTS:
0280    ; *     1) SYSTEM STATUS CODE:  ACCVIO
0280    ; *     2) REGISTERS R2 THROUGH FP UNCHANGED
0280    ; *
0280    ; ******************************************************
0280    ;--
0280    ;
0280    ;
0280            TCEND
```

```
      0281    341 ;
      0281    342          TC_GROUP        CEH,1,TS2
      02A8    343 ;
      02A8    344          NEXT_TEST_CASE  SFCEH10
```

```
02A8  ;
02A8  ;++
02A8  ; ******************************************************
02A8  ; *
02A8  ; *  TEST CASE NAME:              SFCEH10
02A8  ; *
02A8  ; *  SYSTEM SERVICE:              CANEXH
02A8  ; *
02A8  ; *  ARGUMENT UNDER TEST:         DESBLK_CEH10
02A8  ; *
02A8  ; *  INPUT CONDITIONS:
02A8  ; *     SPECIFIED EXIT HANDLER NEVER DECLARED
02A8  ; *       WITH A $DCLEXH SYSTEM SERVICE.
02A8  ; *
02A8  ; *  EXPECTED RESULTS:
02A8  ; *     1) SYSTEM STATUS CODE:  NOHANDLER
02A8  ; *     2) REGISTERS R2 THROUGH FP UNCHANGED
02A8  ; *
02A8  ; ******************************************************
02A8  ;--
02A8  ;
02A8  ;
02A8          TCEND
```

```
02A9   368 ;
02A9   369        TC_GROUP        ADS,1,TS3
02D0   370 ;
02D0   371        NEXT_TEST_CASE  SFADS10
```

```
02D0  ;
02D0  ;++
02D0  ; ****************************************************
02D0  ; *
02D0  ; *  TEST CASE NAME:           SFADS10
02D0  ; *
02D0  ; *  SYSTEM SERVICE:           ADJSTK
02D0  ; *
02D0  ; *  ARGUMENT UNDER TEST:      ACMODE_ADS10
02D0  ; *
02D0  ; *  INPUT CONDITIONS:
02D0  ; *     ATTEMPT TO ADJUST SUPERVISOR STACK
02D0  ; *
02D0  ; *  EXPECTED RESULTS:
02D0  ; *     1) SYSTEM STATUS CODE:  NOPRIV
02D0  ; *     2) REGISTERS R2 THROUGH FP UNCHANGED
02D0  ; *
02D0  ; ****************************************************
02D0  ;--
02D0  ;
02D0  ;
02D0          NEXT_TEST_CASE  SFADS11
```

```
02DC    ;
02DC    ;++
02DC    ; ******************************************************
02DC    ; *
02DC    ; *  TEST CASE NAME:             SFADS11
02DC    ; *
02DC    ; *  SYSTEM SERVICE:             ADJSTK
02DC    ; *
02DC    ; *  ARGUMENT UNDER TEST:        ACMODE_ADS11
02DC    ; *
02DC    ; *  INPUT CONDITIONS:
02DC    ; *     ATTEMPT TO ADJUST EXECUTIVE STACK.
02DC    ; *
02DC    ; *  EXPECTED RESULTS:
02DC    ; *     1) SYSTEM STATUS CODE:  NOPRIV
02DC    ; *     2) REGISTERS R2 THROUGH FP UNCHANGED
02DC    ; *
02DC    ; ******************************************************
02DC    ;--
02DC    ;
02DC    ;
02DC            NEXT_TEST_CASE  SFADS12
```

```
02E8     ;
02E8     ;++
02E8     ; ******************************************************
02E8     ; *
02E8     ; *  TEST CASE NAME:            SFADS12
02E8     ; *
02E8     ; *  SYSTEM SERVICE:            ADJSTK
02E8     ; *
02E8     ; *  ARGUMENT UNDER TEST:       ACMODE_ADS12
02E8     ; *
02E8     ; *  INPUT CONDITIONS:
02E8     ; *     ATTEMPT TO ADJUST KERNEL STACK.
02E8     ; *
02E8     ; *  EXPECTED RESULTS:
02E8     ; *     1) SYSTEM STATUS CODE:  NOPRIV
02E8     ; *     2) REGISTERS R2 THROUGH FP UNCHANGED
02E8     ; *
02E8     ; ******************************************************
02E8     ;--
02E8     ;
02E8     ;
02E8             TCEND
```

```
02E9   438 TS1:
02E9   439         TESTSERV       DCLCMH,ERR,SATS,                                    -
02E9   440                                                                            -
02E9   441            <1,ADDRES_DCH,                                                  -
02E9   442                                                      >,                    -
02E9   443                                                                            -
02E9   444            <1,PRVHND_DCH,                                                  -
02E9   445                        PRVHND_DCH20,ACCVIO,       -  ; SFDCH20
02E9   446                        PRVHND_DCH21,ACCVIO,       -  ; SFDCH21
02E9   447                        PRVHND_DCH22,ACCVIO,       -  ; SFDCH22
02E9   448                                                      >,                    -
02E9   449                                                                            -
02E9   450            <1,TYPE_DCH,                                            -
02E9   451                                                      >,                    -
02E9   452
04D3   453         TS_CLEANUP              ; CLEAN UP & RETURN TO TEST_SERV_EXEC
```

```
04F3   454 TS2:
04F3   455         TESTSERV        CANEXH,ERR,SATS,                              -
04F3   456                                                                       -
04F3   457            <1,DESBLK_CEH,                                             -
04F3   458                         DESBLK_CEH10,NOHANDLER, -  ; SFCEH10
04F3   459                                                 >,                    -
04F3   460
05B6   461         TS_CLEANUP              ; CLEAN UP & RETURN TO TEST_SERV_EXEC
```

```
05D6      TS3:
05D6              TESTSERV        ADJSTK,ERR,SATS,                                -
05D6                                                                              -
05D6                 <1,ACMODE_ADS,                                               -
05D6                              ACMODE_ADS10,NOPRIV,      -  ; SFADS10
05D6                              ACMODE_ADS11,NOPRIV,      -  ; SFADS11
05D6                              ACMODE_ADS12,NOPRIV,      -  ; SFADS12
05D6                                                        >,                    -
05D6                                                                              -
05D6                 <1,ADJUST_ADS,                                               -
05D6                                                        >,                    -
05D6                                                                              -
05D6                 <1,NEWADR_ADS,                                               -
05D6                                                        >,                    -
05D6
07C3              TS_CLEANUP                 ; CLEAN UP & RETURN TO TEST_SERV_EXEC
```

```
                                   07E3      .SBTTL  EXECUTE & CLEANUP
                                   07E3  EXECUTE:
                                   07E3      TEST_SERV_EXEC                  ; EXECUTE ALL T. CASES IN ALL GROUPS
                                   0801  CLEANUP:
                         0138  30  0801      BSBW    MOD_MSG_PRINT           ; PRINT TEST MODULE END MSG
   00000044'EF  01  1C     01  F0  0804      INSV    #1,#STS$V_INHIB_MSG,#1,MOD_MSG_CODE
                                   080D                                      ; INHIBIT PRINTING
                                   080D      $EXIT_S MOD_MSG_CODE            ; EXIT TO OP SYS WITH MSG CODE
```

```
081A            .SBTTL  TC_CONTROL
0081A   ;++
081A    ; FUNCTIONAL DESCRIPTION:
081A    ;
081A    ;               THE TC_CONTROL SUBROUTINE IS CALLED BY THE TEST_SERV_EXEC
081A    ; MACRO TO EXECUTE A GROUP OF TEST CASES. A GROUP IS DEFINED BY A TC_GROUP
081A    ; MACRO. FOR EACH TC_GROUP MACRO, THERE IS A CORRESPONDING TESTSERV MACRO.
081A    ; TESTSERV CONTAINS CODE TO EXECUTE SYSTEM SERVICES AND CHECK THE RETURNED
081A    ; STATUS CODE VALUES. TESTSERV ARGUMENTS ARE CODED TO SPECIFY ALL THE SYSTEM
081A    ; SERVICE ARGUMENT VALUES AND THE EXPECTED STATUS CODE FOR EACH TEST CASE
081A    ; DEFINED BY A NEXT_TEST_CASE MACRO WITHIN THE GROUP. TC_CONTROL USES A
081A    ; CO-ROUTINE INTERFACE TO ENTER THE CODE OF THE APPROPRIATE TESTSERV MACRO
081A    ; IN VARIOUS PLACES. THE FIRST ENTRY OCCURS ONCE PER GROUP TO ALLOW TESTSERV
081A    ; TO DO SOME INITIALIZATION. THEN TWO ENTRIES ARE MADE FOR EACH TEST CASE IN
081A    ; THE GROUP. THE FIRST ALLOWS TESTSERV TO ISSUE THE SUBJECT SYSTEM SERVICE.
081A    ; THE SECOND ENTRY FOR THE TEST CASE CAUSES TESTSERV TO CHECK THE RETURNED
081A    ; STATUS CODE, PRINTING A FAILURE MESSAGE IF IT IS NOT THE EXPECTED CODE.
081A    ; IF THERE ARE NO MORE TEST CASES IN THE CURRENT GROUP, TESTSERV (NOT TC_CONTROL)
081A    ; RETURNS DIRECTLY TO TEST_SERV_EXEC (RSB ACTUALLY ISSUED IN TS_CLEANUP MACRO)
081A    ; FROM THIS SECOND ENTRY; OTHERWISE, CONTROL RETURNS TO TC_CONTROL WHICH
081A    ; IN TURN ENTERS TESTSERV AGAIN FOR THE NEXT TEST CASE. THE FAILURE OF A
081A    ; TEST CASE DOES NOT CAUSE TERMINATION OF THE TEST MODULE.
081A    ;
081A    ; CALLING SEQUENCE:
081A    ;
081A    ;       BSBW TC_CONTROL (ISSUED WITHIN THE TEST_SERV_EXEC MACRO)
081A    ;                       (RSB IS ISSUED WITHIN THE TS_CLEANUP MACRO)
081A    ;
081A    ; INPUT PARAMETERS:
081A    ;
081A    ;       NONE
081A    ;
081A    ; IMPLICIT INPUTS:
081A    ;
081A    ;       ARGUMENTS SPECIFIED ON EACH TESTSERV MACRO MAY BE VIEWED AS
081A    ;       INPUTS, SINCE TC_CONTROL AND TESTSERV ACT AS CO-ROUTINES.
081A    ;
081A    ; OUTPUT PARAMETERS:
081A    ;
081A    ;       SEVERITY CODE FIELD OF MOD_MSG_CODE (BITS 0,1,2) IS SET TO ERROR
081A    ;       IF ANY TEST CASE IN THE CURRENT GROUP FAILS; OTHERWISE IT REMAINS
081A    ;       SET TO SUCCESSFUL.
081A    ;
081A    ; IMPLICIT OUTPUTS:
081A    ;
081A    ;       %UETP-I-TEXT,   ERROR MESSAGES ARE WRITTEN TO SYS$OUTPUT BY
081A    ;                       THE TESTSERV MACRO (CO-ROUTINE WITH TC_CONTROL)
081A    ;
081A    ; COMPLETION CODES:
081A    ;
081A    ;       NONE
081A    ;
081A    ; SIDE EFFECTS:
081A    ;
081A    ;       NONE
081A    ;
081A    ;--
```

```
                               081A
                               081A
                               081A
                               081A  TC_CONTROL:
          00000064'EF     DD   081A          PUSHL   TS_EP                     ; PUSH TESTSERV ENTRY POINT
                   9E     16   0820          JSB     @(SP)+                    ; ENTER TESTSERV INITIALIZATION
                               0822  10$:                                      ; PROCESS NEXT TEST CASE
    00000056'EF    20     90   0822          MOVB    #^A/ /,$$TSTN$$+2         ; MAKE SURE T.C. NAME HAS A BLANK
                 002F     30   0829          BSBW    REG_SAVE                  ; SAVE REGISTERS
          00000004'FF     16   082C          JSB     @CURRENT_TC               ; JUMP TO CURRENT TEST CASE
                 0037     30   0832          BSBW    REG_REST                  ; RESTORE REGS FOR TESTSERV
                   9E     16   0835          JSB     @(SP)+                    ; LET TESTSERV ISSUE SYSTEM SERVICE
                 0042     30   0837          BSBW    REG_COMP                  ; COMPARE REGS TO SEE IF ...
                               083A                                            ; ... SYSTEM SERVICE CHANGED ANY
                   9E     16   083A          JSB     @(SP)+                    ; LET TESTSERV CHEK S.S. STATUS CODE
    00000056'EF    2A     91   083C          CMPB    #^A/*/,$$TSTN$$+2         ; HAS TESTSERV INDICATED FAILURE ?
                   DD     12   0843          BNEQU   10$                       ; NO -- PROCESS NEXT TEST CASE
00000060'EF   00000088'EF DE   0845          MOVAL   TEST_MOD_FAIL,TMD_ADDR    ; YES -- INDICATE FAILED IN END MSG
00000044'EF   03   00     02 F0 0850         INSV    #ERROR,#0,#3,MOD_MSG_CODE ; ADJUST STATUS CODE FOR ERROR
                   C7     11   0859          BRB     10$                       ; LOOP BAK TO PROCESS NEXT TEST CASE
                               085B  ;
                               085B  ;    TC_CONTROL RETURNS TO TEST_SERV_EXEC VIA TESTSERV (IN TS_CLEANUP MACRO)
                               085B  ;
```

```
                                  085B        .SBTTL  SUBROUTINES
                                  085B   REG_SAVE:
                                  085B   ;
                                  085B   ;    ************************************************************
                                  085B   ;    *                                                          *
                                  085B   ;    *  SAVES RO THRU SP IN REG_SAVE_AREA                       *
                                  085B   ;    *                                                          *
                                  085B   ;    ************************************************************
                                  085B   ;
                   7FFF 8F    BB  085B              PUSHR   #RO_THRU_SP              ; SAVE ALL REGS ON STACK
      00000008'EF  6E    3C   28  085F              MOVC3   #60,(SP),REG_SAVE_AREA   ; SAVE REGS (BEFORE S.S.)
                   7FFF 8F    BA  0867              POPR    #RO_THRU_SP              ; CLEAN UP STACK
                              05  086B              RSB                              ; .... AND RETURN
                                  086C   ;
                                  086C   ;
                                  086C   ;
                                  086C   ;
                                  086C   REG_REST:
                                  086C   ;
                                  086C   ;
                                  086C   ;    ************************************************************
                                  086C   ;    *                                                          *
                                  086C   ;    *  RESTORES RO THRU SP FROM REG_SAVE_AREA                  *
                                  086C   ;    *                                                          *
                                  086C   ;    ************************************************************
                                  086C   ;
                   5E    3C   C2  086C              SUBL2   #60,SP                   ; MOVE SP TO MAKE ROOM FOR REGS
      6E   00000008'EF   3C   28  086F              MOVC3   #60,REG_SAVE_AREA,(SP)   ; MOVE REGS ONTO STACK FOR POP
                   7FFF 8F    BA  0877              POPR    #RO_THRU_SP              ; RESTORE ALL REGS FOR TESTSERV
                              05  087B              RSB                              ; ... AND RETURN
                              087C      REG_COMP:
                              087C      ;
                              087C      ;     ****************************************************************
                              087C      ;     *                                                              *
                              087C      ;     *  1) PUSHES ALL REGS ONTO STACK                               *
                              087C      ;     *  2) COMPARES REGISTER IMAGES FROM STACK WITH CORRESPONDING   *
                              087C      ;     *       IMAGES FROM REG_SAVE_AREA FOR ALL REGISTERS SPECIFIED  *
                              087C      ;     *       IN REG_COMP_MASK.                                      *
                              087C      ;     *  3) FOR EACH UNEQUAL COMPARE, AN ERROR MESSAGE IS PRINTED    *
                              087C      ;     *       (USING $FAO AND $OUTPUT SYSTEM SERVICES).              *
                              087C      ;     *  4) POPS ALL REGS OFF OF STACK                               *
                              087C      ;     *                                                              *
                              087C      ;     ****************************************************************
                              087C      ;
              7FFF 8F     BB  087C              PUSHR   #R0_THRU_SP             ; SAVE ALL REGISTERS ON STACK
      56   00000008'EF    DE  0880              MOVAL   REG_SAVE_AREA,R6        ; POINT R6 TO BEG OF ...
                              0887                                              ; ... REGS (BEFORE S.S.)
              54    5E    D0  0887              MOVL    SP,R4                   ; POINT R4 TO BEG OF
                              088A                                              ; ... REGS (AFTER S.S.)
         53     FF 8F     98  088A              CVTBL   #-1,R3                  ; INITIALIZE REG_COMP_MASK INDEX
                              088E      REG_COMP_NEXT:
                    53    D6  088E              INCL    R3                      ; POINT TO NEXT BIT IN MASK
              53    0F    91  0890              CMPB    #15,R3                  ; END OF THE MASK ?
                    03    1A  0893              BGTRU   REG_COMP_CONT           ; NO -- CONTINUE
                  009F    31  0895              BRW     REG_COMP_RSB            ; YES -- GO TO COMMON RETURN
                              0898      REG_COMP_CONT:
              84    86    D1  0898              CMPL    (R6)+,(R4)+             ; REG BEFORE = REG AFTER ?
                    F1    13  089B              BEQLU   REG_COMP_NEXT           ; YES -- LOOK FOR NEXT REG
   E9 00000000'EF   53    E1  089D              BBC     R3,REG_COMP_MASK,REG_COMP_NEXT
                              08A5                                              ; NO -- GET NEXT IF BIT NOT SET
      00000048'EF   53    D0  08A5              MOVL    R3,CLOB_REG_NO          ; NO -- GIVE REG NUMBER TO FAO
   0000004C'EF   FC A6    D0  08AC              MOVL    -4(R6),REG_BEFORE_SS    ; GIVE "BEFORE" CONTENTS TO FAO
   00000050'EF   FC A4    D0  08B4              MOVL    -4(R4),REG_AFTER_SS     ; GIVE "AFTER" CONTENTS TO FAO
      00000056'EF   2A    90  08BC              MOVB    #^A/*/,$$TSTN$$+2       ; GIVE FAILURE INDIC'N IN ERROR MSG
                              08C3      ;
                              08C3              $FAO_S  ERR_MSG_FAOCTL,OUTL,OUTD,$$SNAD$$,  -
                              08C3                      $$ASEQ$$,$$PSEQ$$,CLOB_REG_NO,REG_BEFORE_SS,REG_AFTER_SS
                              08F6      ;
       F817 CF    F7E1 CF     B0  08F6              MOVW    OUTL,OUTD               ; ACTUAL OUTPUT LEN IN STRING DESC'R
                              08FD              PUTMSG  <#UETP$_TEXT,#1,#OUTD>  ; PRINT THE MSG
       F7FB CF    0084 8F     B0  0912              MOVW    #OUTE-OUTB,OUTD         ; GET MAX LEN BACK INTO DESCRIPTOR
      00000056'EF   20    90  0919              MOVB    #^A/ /,$$TSTN$$+2       ; REMOVE FAIL INDIC'N FOR NEXT MSG
 00000060'EF   00000088'EF    DE  0920              MOVAL   TEST_MOD_FAIL,TMD_ADDR  ; INDICATE FAILED IN END MSG
00000044'EF    03    00   02  F0  092B              INSV    #ERROR,#0,#3,MOD_MSG_CODE ; ADJUST STATUS CODE FOR ERROR
                  FF57    31  0934              BRW     REG_COMP_NEXT           ; GO LOOK FOR NEXT REG TO COMPARE
                              0937      REG_COMP_RSB:
              7FFF 8F     BA  0937              POPR    #R0_THRU_SP             ; CLEAN UP STACK
                          05  093B              RSB                             ; RETURN TO CALLER
```

```
093C        MOD_MSG_PRINT:
093C        ;
093C        ;    *****************************************************************
093C        ;    *                                                               *
093C        ;    *  PRINTS THE TEST MODULE BEGUN/SUCCESSFUL/FAILED MESSAGES      *
093C        ;    *       (USING THE PUTMSG MACRO).                               *
093C        ;    *                                                               *
093C        ;    *****************************************************************
```

```
                       093C   657 ;
                       093C   658         PUTMSG  <MOD_MSG_CODE,#2,TMN_ADDR,TMD_ADDR> ; PRINT MSG
                    05 0957   659         RSB                                 ; ... AND RETURN TO CALLER
                       0958   660 ;
                       0958   661 CHMRTN:
                       0958   662 ; ************************************************************
                       0958   663 ; *
                       0958   664 ; *       CHANGE MODE ROUTINE. THIS ROUTINE GETS CONTROL WHENEVER
                       0958   665 ; *       A CMKRNL, CMEXEC, OR CMSUP SYSTEM SERVICE IS ISSUED
                       0958   666 ; *       BY THE MODE MACRO ('TO' OPTION).  IT MERELY DOES
                       0958   667 ; *       A JUMP INDIRECT ON A FIELD SET UP BY MODE. IT HAS
                       0958   668 ; *       THE EFFECT OF RETURNING TO THE END OF THE MODE
                       0958   669 ; *       MACRO EXPANSION.
                       0958   670 ; *
                       0958   671 ; **************************************************************
                       0958   672 ;
                  0000 0958   673         .WORD   0                           ; ENTRY MASK
     00000079'FF    17 095A   674         JMP     @CHM_CONT                   ; RETURN TO MODE MACRO IN NEW MODE
                       0960   675 ;
                       0960   676 ; *       RET INSTR WILL BE ISSUED IN EXPANSION OF 'MODE FROM, ....' MACRO
                       0960   677 ;
                       0960   678         .END    SATSSF15
```

```
$$$CHARS                   = 00000048                PRVHND_DCH20               = 00000001
$$$FIRSTTC$$$              = 00000000                PRVHND_DCH21                 000000C0 R       02
$$$STRINGS                 = 00000000                PRVHND_DCH22               = 000001FF R       04
$$ACT$$                      000000F3 R       06     PRVPRT                       00000070 R       03
$$ARG$$                      000000FB R       06     PSL$C_EXEC                 = 00000001
$$ASEQ$$                     000000EB R       06     PSL$C_KERNEL               = 00000000
$$CALL$$                     000000DF R       06     PSL$C_SUPER                = 00000002
$$DISP$$                     000001E6 R       06     PSL$C_USER                 = 00000003
$$ERR$$                      000001A0 R       06     R0_THRU_SP                 = 00007FFF
$$EXP$$                      000000F7 R       06     REGS                         0000007D R       03
$$INIT$$                     000000E3 R       06     REG_AFTER_SS                 00000050 R       03
$$MAXP$$                   = 00000005                REG_BEFORE_SS                0000004C R       03
$$PSEQ$$                     000000EF R       06     REG_COMP                     0000087C R       06
$$SNAD$$                     000000E7 R       06     REG_COMP_CONT                00000898 R       06
$$T1                       = 00000004                REG_COMP_MASK                00000000 R       02
$$T2                       = 00000009                REG_COMP_NEXT                0000088E R       06
$$TSTN$$                     00000054 R       03     REG_COMP_RSB                 00000937 R       06
ACMODE_ADS                   000000C8 R       02     REG_REST                     0000086C R       06
ACMODE_ADS10                 000000CC R       02     REG_SAVE                     0000085B R       06
ACMODE_ADS11                 000000D0 R       02     REG_SAVE_AREA                00000008 R       03
ACMODE_ADS12                 000000D4 R       02     RETADR                       00000068 R       03
ADDRES_DCH                   000000BD R       02     SATSSF15                     00000000 R       06
ADJUST_ADS                   000000D8 R       02     SEVERE                     = 00000004
CHMRTN                       00000958 R       06     SHR$K_SHRDEF               = 00000001
CHM_CONT                     00000079 R       03     SHR$_TEXT                  = 00001130
CLEANUP                      00000801 R       06     SS$_ACCVIO                   ********    X    06
CLOB_REG_NO                  00000048 R       03     SS$_NOHANDLER                ********    X    06
CTL$GL_PHD                   ********    X    06     SS$_NOPRIV                   ********    X    06
CURRENT_TC                   00000004 R       03     STS$V_INHIB_MSG            = 0000001C
DESBLK_CEH                   00000095 R       03     SUCCESS                    = 00000001
DESBLK_CEH10                 00000095 R       03     SYS$ADJSTK                   ********   GX    06
EMPTY                        00000000 R       04     SYS$CANEXH                   ********   GX    06
ERROR                      = 00000002                SYS$CMKRNL                   ********   GX    06
ERR_MSG_FAOCTL               00000002 R       02     SYS$DCLCMH                   ********   GX    06
EXECUTE                      000007E3 R       06     SYS$EXIT                     ********   GX    06
GRP_TOTAL                  = 00000003                SYS$FAO                      ********    X    06
INADR                        000000A9 R       02     SYS$FAOL                     ********   GX    06
INFO                       = 00000003                SYS$HIBER                    ********   GX    06
LIB$SIGNAL                   ********    X    06     SYS$SETPRN                   ********   GX    06
MEXIT                      = 00000000                SYS$SETPRT                   ********   GX    06
MOD_MSG_CODE                 00000044 R       03     SYS$SETPRV                   ********   GX    06
MOD_MSG_PRINT                0000093C R       06     SYS$WAKE                     ********   GX    06
NARGS                      = 0000000E                TC1                          00000241 R       06
NEWADR_ADS                   000000AC R       03     TC2                          00000281 R       06
NOACCESS                     00000000 R       05     TC3                          000002A9 R       06
NSSARGS                    = 00000003                TCG_NO                     = 00000003
ONES                         000000B5 R       02     TC_CONTROL                   0000081A R       06
OUTB                         0000011C R       06     TEST_MOD_BEG                 00000077 R       02
OUTD                         00000114 R       06     TEST_MOD_FAIL                00000088 R       02
OUTE                         000001A0 R       06     TEST_MOD_NAME                0000006E R       02
OUTL                         000000DB R       06     TEST_MOD_NAME_D              0000008F R       02
PHD$Q_PRIVMSK              = 00000000                TEST_MOD_SUCC                0000007D R       02
PRIVMASK                     00000071 R       03     TMD_ADDR                     00000060 R       03
PRIV_ARGS                  = 00000002                TMN_ADDR                     0000005C R       03
PROT                         000000B1 R       02     TPID                         00000000 R       03
PRT$C_NA                     ********    X    02     TS1                          000002E9 R       06
PRVHND_DCH                   00000091 R       03     TS2                          000004F3 R       06
```

```
TS3                               000005D6 R     06
TS_EP                             00000064 R     03
TTNAME                            0000009F R     02
TYPE_DCH                          000000C4 R     02
UETP$_SATSMS                    = 007480D9
UETP$_TEXT                      = 00741133
WARNING                         = 00000000

                                                 +----------------+
                                                 ! Psect synopsis !
                                                 +----------------+

PSECT name                       Allocation          PSECT No.  Attributes
----------                       ----------          ---------  ----------
.  ABS  .                        00000000  (    0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
$ABS$                            00000000  (    0.)  01 (  1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
RODATA                           000000DC  (  220.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR NOEXE   RD  NOWRT NOVEC LONG
RWDATA                           000000B0  (  176.)  03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR NOEXE   RD    WRT NOVEC BYTE
SATS_ACCVIO_1                    00000200  (  512.)  04 (  4.)  NOPIC   USR   CON   REL   LCL NOSHR NOEXE   RD    WRT NOVEC PAGE
SATS_ACCVIO_2                    00000200  (  512.)  05 (  5.)  NOPIC   USR   CON   REL   LCL NOSHR NOEXE   RD    WRT NOVEC PAGE
SATSSF15                         00000960  ( 2400.)  06 (  6.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC LONG

                                              +------------------------+
                                              ! Performance indicators !
                                              +------------------------+

Phase                    Page faults     CPU Time       Elapsed Time
-----                    -----------     --------       ------------
Initialization                    31     00:00:00.07    00:00:00.29
Command processing               107     00:00:00.71    00:00:02.84
Pass 1                           358     00:00:13.33    00:00:22.59
Symbol table sort                  0     00:00:01.08    00:00:01.15
Pass 2                           142     00:00:03.02    00:00:03.87
Symbol table output               16     00:00:00.12    00:00:00.12
Psect synopsis output              2     00:00:00.03    00:00:00.03
Cross-reference output             0     00:00:00.00    00:00:00.00
Assembler run totals             658     00:00:18.37    00:00:30.92

The working set limit was 1650 pages.
68551 bytes (134 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 637 non-local and 88 local symbols.
678 source lines were read in Pass 1, producing 27 object records in Pass 2.
64 pages of virtual memory were used to define 48 macros.

                                             +--------------------------+
                                             ! Macro library statistics !
                                             +--------------------------+

Macro library name                            Macros defined
------------------                            --------------
_$255$DUA28:[SHRLIB]UETP.MLB;1                            19
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                             2
_$255$DUA28:[SYSLIB]STARLET.MLB;2                         21
TOTALS (all libraries)                                    42

1273 GETS were required to define 42 macros.
```

```
There were no errors, warnings or information messages.

MACRO/LIS=LIS$:SATSSF15/OBJ=OBJ$:SATSSF15 MSRC$:SATSSF15/UPDATE=(ENH$:SATSSF15)+EXECML$/LIB+SHRLIB$:UETP/LIB
```

0420 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY