```
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
```

FILEID**SATSSF16


```
  SSSSSSSS    AAAAAA    TTTTTTTTTT    SSSSSSSS    SSSSSSSS  FFFFFFFFFF      11        666666
  SSSSSSSS    AAAAAA    TTTTTTTTTT    SSSSSSSS    SSSSSSSS  FFFFFFFFFF      11        666666
SS          AA      AA      TT      SS          SS          FF            1111      66
SS          AA      AA      TT      SS          SS          FF            1111      66
SS          AA      AA      TT      SS          SS          FF              11      66
SS          AA      AA      TT      SS          SS          FF              11      66
  SSSSSS    AA      AA      TT        SSSSSS      SSSSSS    FFFFFFFF        11      66666666
  SSSSSS    AA      AA      TT        SSSSSS      SSSSSS    FFFFFFFF        11      66666666
        SS  AAAAAAAAAA      TT              SS          SS  FF              11      66      66
        SS  AAAAAAAAAA      TT              SS          SS  FF              11      66      66
        SS  AA      AA      TT              SS          SS  FF              11      66      66      ....
        SS  AA      AA      TT              SS          SS  FF              11      66      66      ....
SSSSSSSS    AA      AA      TT      SSSSSSSS    SSSSSSSS    FF            111111      666666        ....
SSSSSSSS    AA      AA      TT      SSSSSSSS    SSSSSSSS    FF            111111      666666        ....

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  SATSSF16 - 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 SATSSF16 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     ;
                  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>
                  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,<SATSSF16>      ; 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,<SATSSF16>      ; 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$C_NA          ; PROTECTION CODE FOR NOACCESS PSECT
 FFFFFFFF FFFFFFFF  00B5  ONES:           .LONG   -1,-1             ; A QUADWORD OF 1-BITS
                    00BD  CTRSTR_FAO:                               ; CTRSTR ARGUMENT FOR FAO
                    00BD  CTRSTR_FAL:                               ; CTRSTR ARGUMENT FOR FAOL
                    00BD                  STRING  I,<TEST: !AC>
                    00CE  CTRSTR_FAO10:                             ; CTRSTR ARGUMENT FOR FAO
                    00CE  CTRSTR_FAL10:                             ; CTRSTR ARGUMENT FOR FAOL
                    00CE                  STRING  I,<TEST: !BC>
         000000E3'  00DF  P1_FAO:         .ADDRESS COUNTED_STR      ; P1 ARGUMENT FOR FAO
                    00E3  COUNTED_STR:    STRING  C,<A VALID COUNTED STRING>
          00000000  00FA  P2_FAO:         .LONG   0                 ; P2 ARGUMENT FOR FAO
         000000DF'  00FE  PRMLST_FAL:     .ADDRESS P1_FAO           ; PRMLST ARGUMENT FOR FAOL
                    0102  MSGBUF_SNE:
                    0102  MSGBUF_SNE10:   STRING  I,<SATSSF16 ERROR LOG BUFFER> ; MSGBUF ARG FOR SNDERR
                    0123  MY_DISK:        STRING  I,<SYS$DISK>      ; LOGICAL NAME FOR USER DISK
```

```
        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)
          0091  OUTLEN_FAO:                                 ; OUTLEN ARGUMENT FOR FAO
          0091  OUTLEN_FAL:                                 ; OUTLEN ARGUMENT FOR FAOL
00000093  0091                  .BLKW   1
          0093  OUTBUF_FAO:                                 ; OUTBUF ARGUMENT FOR FAO
          0093  OUTBUF_FAL:                                 ; OUTBUF ARGUMENT FOR FAOL
          0093                  STRING  0,50
          00CD  OUTBUF_FAO30:                               ; OUTBUF ARGUMENT FOR FAO
          00CD  OUTBUF_FAL30:                               ; OUTBUF ARGUMENT FOR FAOL
          00CD                  STRING  0,1
```

```
        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:
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  MSGBUF_SNE11:   STRING  I,<SFSNE11>    ; MSGBUF ARGUMENT FOR SNDERR
          0017  ;
          0017  ;
          0017  ;
        00000000          .PSECT  SATSSF16,RD,WRT,EXE,LONG
```

```
0000          .SBTTL  SATSSF16
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 SATSSF16 ROUTINE EXECUTES THE TEST_SERV_EXEC MACRO TO RUN
0000  ; ALL TEST CASES. WHEN THE MACRO COMPLETES ITS EXECUTION, SATSSF16
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 SATSSF16  ...  (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 SATSSF16.
0000  ;       THEY ARE OF THE FORM:
0000  ;
0000  ;               %UETP-S-SATSMS, TEST MODULE SATSSF16 BEGUN ... (BEGIN MSG)
0000  ;               %UETP-S-SATSMS, TEST MODULE SATSSF16 SUCCESSFUL ... (END MSG)
0000  ;               %UETP-E-SATSMS, TEST MODULE SATSSF16 FAILED ... (END MSG)
0000  ;               %UETP-I-TEXT, ... (VARIABLE INFORMATION ABOUT A TEST MODULE FAILURE)
0000  ;
0000  ; COMPLETION CODES:
0000  ;
0000  ;       THE SATSSF16 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  SATSSF16:
                                 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          $SETPRN_S TEST_MOD_NAME_D         ; SET PROCESS NAME
                        0AFA      30   0025          BSBW    MOD_MSG_PRINT             ; PRINT TEST MODULE BEGIN MSG
   00000060'EF     0000007D'EF    DE   0028          MOVAL   TEST_MOD_SUCC,TMD_ADDR    ; ASSUME END MSG WILL SHOW SUCCESS
  00000044'EF    03    00    01   F0   0033          INSV    #SUCCESS,#0,#3,MOD_MSG_CODE ; ADJUST STATUS CODE FOR SUCCESS
                                       003C          MODE    TO,10$,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
                        0788      31   023E          BRW     EXECUTE                   ; GO EXECUTE ALL TEST CASES
                                       0241  ;
                                       0241          TC_GROUP        FAO,1,TS1
                                       0268  ;
                                       0268          NEXT_TEST_CASE  SFFA010
```

```
0268    ;
0268    ;++
0268    ; ***********************************************************
0268    ; *
0268    ; *  TEST CASE NAME:            SFFA010
0268    ; *
0268    ; *  SYSTEM SERVICE:            FAO
0268    ; *
0268    ; *  ARGUMENT UNDER TEST:       CTRSTR_FA010
0268    ; *
0268    ; *  INPUT CONDITIONS:
0268    ; *     INVALID DIRECTIVE IN FAO CONTROL STRING
0268    ; *
0268    ; *  EXPECTED RESULTS:
0268    ; *     1) SYSTEM STATUS CODE:  BADPARAM
0268    ; *     2) REGISTERS R2 THROUGH FP UNCHANGED
0268    ; *
0268    ; ***********************************************************
0268    ;--
0268    ;
0268    ;
0268            NEXT_TEST_CASE  SFFA030
```

```
0274     ;
0274     ;++
0274     ; ***********************************************************
0274     ; *
0274     ; *  TEST CASE NAME:               SFFA030
0274     ; *
0274     ; *  SYSTEM SERVICE:               FAO
0274     ; *
0274     ; *  ARGUMENT UNDER TEST:          OUTBUF_FA030
0274     ; *
0274     ; *  INPUT CONDITIONS:
0274     ; *     OUTPUT BUFFER (LENGTH 1) TOO SHORT FOR
0274     ; *       INCOMING DATA.
0274     ; *
0274     ; *  EXPECTED RESULTS:
0274     ; *     1) SYSTEM STATUS CODE:  BUFFEROVF
0274     ; *     2) REGISTERS R2 THROUGH FP UNCHANGED
0274     ; *
0274     ; ***********************************************************
0274     ;--
0274     ;
0274     ;
0274           TCEND
```

G 1

```
0275        ;
0275                TC_GROUP        FAL,1,TS2
029C        ;
029C                NEXT_TEST_CASE  SFFAL10
```

```
029C ;
029C ;++
029C ; ***********************************************************
029C ; *
029C ; *  TEST CASE NAME:            SFFAL10
029C ; *
029C ; *  SYSTEM SERVICE:            FAOL
029C ; *
029C ; *  ARGUMENT UNDER TEST:       CTRSTR_FAL10
029C ; *
029C ; *  INPUT CONDITIONS:
029C ; *     INVALID DIRECTIVE IN FAO CONTROL STRING
029C ; *
029C ; *  EXPECTED RESULTS:
029C ; *     1) SYSTEM STATUS CODE:  BADPARAM
029C ; *     2) REGISTERS R2 THROUGH FP UNCHANGED
029C ; *
029C ; ***********************************************************
029C ;--
029C ;
029C ;
029C        NEXT_TEST_CASE  SFFAL30
```

```
02A8   ;
02A8   ;++
02A8   ; ***********************************************************
02A8   ; *
02A8   ; *  TEST CASE NAME:              SFFAL30
02A8   ; *
02A8   ; *  SYSTEM SERVICE:              FAOL
02A8   ; *
02A8   ; *  ARGUMENT UNDER TEST:         OUTBUF_FAL30
02A8   ; *
02A8   ; *  INPUT CONDITIONS:
02A8   ; *     OUTPUT BUFFER (LENGTH 1) TOO SHORT FOR
02A8   ; *        INCOMING DATA.
02A8   ; *
02A8   ; *  EXPECTED RESULTS:
02A8   ; *     1) SYSTEM STATUS CODE:  BUFFEROVF
02A8   ; *     2) REGISTERS R2 THROUGH FP UNCHANGED
02A8   ; *
02A8   ; ***********************************************************
02A8   ;--
02A8   ;
02A8   ;
02A8           TCEND
```

```
02A9        ;
02A9                TC_GROUP        SNE,1,TS3
02D0        ;
02D0                NEXT_TEST_CASE  SFSNE10
```

```
02D0  ;
02D0  ;++
02D0  ; ***********************************************************
02D0  ; *
02D0  ; *  TEST CASE NAME:              SFSNE10
02D0  ; *
02D0  ; *  SYSTEM SERVICE:              SNDERR
02D0  ; *
02D0  ; *  ARGUMENT UNDER TEST:         MSGBUF_SNE10
02D0  ; *
02D0  ; *  INPUT CONDITIONS:
02D0  ; *     USER DOES NOT HAVE THE PRIVILEGE TO SEND A
02D0  ; *       MESSAGE TO THE ERROR LOGGER.
02D0  ; *
02D0  ; *  EXPECTED RESULTS:
02D0  ; *     1) SYSTEM STATUS CODE:  NOPRIV
02D0  ; *     2) REGISTERS R2 THROUGH FP UNCHANGED
02D0  ; *
02D0  ; ***********************************************************
02D0  ;--
02D0  ;
02D0          PRIV    REM,BUGCHK              ; REMOVE BUGCHK PRIVILEGE
02F0  ;
02F0          NEXT_TEST_CASE  SFSNE11
```

```
02FC    ;
02FC    ;++
02FC    ; ******************************************************
02FC    ; *
02FC    ; *  TEST CASE NAME:              SFSNE11
02FC    ; *
02FC    ; *  SYSTEM SERVICE:              SNDERR
02FC    ; *
02FC    ; *  ARGUMENT UNDER TEST:         MSGBUF_SNE11
02FC    ; *
02FC    ; *  INPUT CONDITIONS:
02FC    ; *     MESSAGE BUFFER DESCRIPTOR IN NON-ACCESSIBLE PSECT.
02FC    ; *
02FC    ; *  EXPECTED RESULTS:
02FC    ; *     1) SYSTEM STATUS CODE:  ACCVIO
02FC    ; *     2) REGISTERS R2 THROUGH FP UNCHANGED
02FC    ; *
02FC    ; ******************************************************
02FC    ;--
02FC    ;
02FC            PRIV    ADD,BUGCHK              ; RE-ACQUIRE PRIVILEGE REMOVED BY SFSNE10
031C    ;
031C            TCEND
```

```
031D      TS1:
031D              TESTSERV        FAO,ERR,SATS,                             -
031D                                                                          -
031D                  <1,CTRSTR_FAO,                                          -
031D                                  CTRSTR_FAO10,BADPARAM,  -  ; SFFAO10
031D                                                          >,              -
031D                                                                          -
031D                  <1,OUTLEN_FAO,                                          -
031D                                                          >,              -
031D                                                                          -
031D                  <1,OUTBUF_FAO,                                          -
031D                                  OUTBUF_FAO30,BUFFEROVF, -  ; SFFAO30
031D                                                          >,              -
031D                                                                          -
031D                  <1,P1_FAO,                                        -
031D                                                          >,              -
031D                                                                          -
031D                  <1,P2_FAO,                                        -
031D                                                          >,              -
031D
062A              TS_CLEANUP                  ; CLEAN UP & RETURN TO TEST_SERV_EXEC
```

```
064A      TS2:
064A              TESTSERV      FAOL,ERR,SATS,                                  -
064A                                                                            -
064A                 <1,CTRSTR_FAL,                                             -
064A                                CTRSTR_FAL10,BADPARAM,  -  ; SFFAL10
064A                                                      >,                    -
064A                                                                            -
064A                 <1,OUTLEN_FAL,                                             -
064A                                                      >,                    -
064A                                                                            -
064A                 <1,OUTBUF_FAL,                                             -
064A                                OUTBUF_FAL30,BUFFEROVF, -  ; SFFAL30
064A                                                      >,                    -
064A                                                                            -
064A                 <1,PRMLST_FAL,                                             -
064A                                                      >,                    -
064A
08BE              TS_CLEANUP                   ; CLEAN UP & RETURN TO TEST_SERV_EXEC
```

```
08DE  TS3:
08DE          TESTSERV        SNDERR,ERR,SATS,                                  -
08DE                                                                            -
08DE             <1,MSGBUF_SNE,                                                 -
08DE                          MSGBUF_SNE10,NOPRIV,     -  ; SFSNE10
08DE                          MSGBUF_SNE11,ACCVIO,     -  ; SFSNE11
08DE                                                  >,                        -
08DE
09A9          TS_CLEANUP                  ; CLEAN UP & RETURN TO TEST_SERV_EXEC
```

```
                                      09C9   466         .SBTTL  EXECUTE & CLEANUP
                                      09C9   467 EXECUTE:
                                      09C9   468         TEST_SERV_EXEC                  ; EXECUTE ALL T. CASES IN ALL GROUPS
                                      09E7   469 CLEANUP:
                          0138    30  09E7   470         BSBW    MOD_MSG_PRINT           ; PRINT TEST MODULE END MSG
   00000044'EF    01   1C   01    F0  09EA   471         INSV    #1,#STS$V_INHIB_MSG,#1,MOD_MSG_CODE
                                      09F3   472                                         ; INHIBIT PRINTING
                                      09F3   473         $EXIT_S MOD_MSG_CODE            ; EXIT TO OP SYS WITH MSG CODE
```

```
0A00          .SBTTL  TC_CONTROL
0A00  ;++
0A00  ; FUNCTIONAL DESCRIPTION:
0A00  ;
0A00  ;               THE TC_CONTROL SUBROUTINE IS CALLED BY THE TEST_SERV_EXEC
0A00  ; MACRO TO EXECUTE A GROUP OF TEST CASES. A GROUP IS DEFINED BY A TC_GROUP
0A00  ; MACRO. FOR EACH TC_GROUP MACRO, THERE IS A CORRESPONDING TESTSERV MACRO.
0A00  ; TESTSERV CONTAINS CODE TO EXECUTE SYSTEM SERVICES AND CHECK THE RETURNED
0A00  ; STATUS CODE VALUES. TESTSERV ARGUMENTS ARE CODED TO SPECIFY ALL THE SYSTEM
0A00  ; SERVICE ARGUMENT VALUES AND THE EXPECTED STATUS CODE FOR EACH TEST CASE
0A00  ; DEFINED BY A NEXT_TEST_CASE MACRO WITHIN THE GROUP. TC_CONTROL USES A
0A00  ; CO-ROUTINE INTERFACE TO ENTER THE CODE OF THE APPROPRIATE TESTSERV MACRO
0A00  ; IN VARIOUS PLACES. THE FIRST ENTRY OCCURS ONCE PER GROUP TO ALLOW TESTSERV
0A00  ; TO DO SOME INITIALIZATION. THEN TWO ENTRIES ARE MADE FOR EACH TEST CASE IN
0A00  ; THE GROUP. THE FIRST ALLOWS TESTSERV TO ISSUE THE SUBJECT SYSTEM SERVICE.
0A00  ; THE SECOND ENTRY FOR THE TEST CASE CAUSES TESTSERV TO CHECK THE RETURNED
0A00  ; STATUS CODE, PRINTING A FAILURE MESSAGE IF IT IS NOT THE EXPECTED CODE.
0A00  ; IF THERE ARE NO MORE TEST CASES IN THE CURRENT GROUP, TESTSERV (NOT TC_CONTROL)
0A00  ; RETURNS DIRECTLY TO TEST_SERV_EXEC (RSB ACTUALLY ISSUED IN TS_CLEANUP MACRO)
0A00  ; FROM THIS SECOND ENTRY; OTHERWISE, CONTROL RETURNS TO TC_CONTROL WHICH
0A00  ; IN TURN ENTERS TESTSERV AGAIN FOR THE NEXT TEST CASE. THE FAILURE OF A
0A00  ; TEST CASE DOES NOT CAUSE TERMINATION OF THE TEST MODULE.
0A00  ;
0A00  ; CALLING SEQUENCE:
0A00  ;
0A00  ;       BSBW TC_CONTROL (ISSUED WITHIN THE TEST_SERV_EXEC MACRO)
0A00  ;                       (RSB IS ISSUED WITHIN THE TS_CLEANUP MACRO)
0A00  ;
0A00  ; INPUT PARAMETERS:
0A00  ;
0A00  ;       NONE
0A00  ;
0A00  ; IMPLICIT INPUTS:
0A00  ;
0A00  ;       ARGUMENTS SPECIFIED ON EACH TESTSERV MACRO MAY BE VIEWED AS
0A00  ;       INPUTS, SINCE TC_CONTROL AND TESTSERV ACT AS CO-ROUTINES.
0A00  ;
0A00  ; OUTPUT PARAMETERS:
0A00  ;
0A00  ;       SEVERITY CODE FIELD OF MOD_MSG_CODE (BITS 0,1,2) IS SET TO ERROR
0A00  ;       IF ANY TEST CASE IN THE CURRENT GROUP FAILS; OTHERWISE IT REMAINS
0A00  ;       SET TO SUCCESSFUL.
0A00  ;
0A00  ; IMPLICIT OUTPUTS:
0A00  ;
0A00  ;       %UETP-I-TEXT,   ERROR MESSAGES ARE WRITTEN TO SYS$OUTPUT BY
0A00  ;                       THE TESTSERV MACRO (CO-ROUTINE WITH TC_CONTROL)
0A00  ;
0A00  ; COMPLETION CODES:
0A00  ;
0A00  ;       NONE
0A00  ;
0A00  ; SIDE EFFECTS:
0A00  ;
0A00  ;       NONE
0A00  ;
0A00  ;--
```

```
                                 0A00
                                 0A00
                                 0A00
                                 0A00   TC_CONTROL:
         00000064'EF     DD      0A00          PUSHL   TS_EP                     ; PUSH TESTSERV ENTRY POINT
                  9E     16      0A06          JSB     @(SP)+                    ; ENTER TESTSERV INITIALIZATION
                                 0A08   10$:                                     ; PROCESS NEXT TEST CASE
      00000056'EF    20  90      0A08          MOVB    #^A/ /,$$TSTN$$+2         ; MAKE SURE T.C. NAME HAS A BLANK
                   002F  30      0A0F          BSBW    REG_SAVE                  ; SAVE REGISTERS
         00000004'FF     16      0A12          JSB     @CURRENT_TC               ; JUMP TO CURRENT TEST CASE
                   0037  30      0A18          BSBW    REG_REST                  ; RESTORE REGS FOR TESTSERV
                     9E  16      0A1B          JSB     @(SP)+                    ; LET TESTSERV ISSUE SYSTEM SERVICE
                   0042  30      0A1D          BSBW    REG_COMP                  ; COMPARE REGS TO SEE IF ...
                                 0A20                                            ; ... SYSTEM SERVICE CHANGED ANY
                     9E  16      0A20          JSB     @(SP)+                    ; LET TESTSERV CHEK S.S. STATUS CODE
      00000056'EF    2A  91      0A22          CMPB    #^A/*/,$$TSTN$$+2         ; HAS TESTSERV INDICATED FAILURE ?
                     DD  12      0A29          BNEQU   10$                       ; NO -- PROCESS NEXT TEST CASE
 00000060'EF   00000088'EF  DE   0A2B          MOVAL   TEST_MOD_FAIL,TMD_ADDR    ; YES -- INDICATE FAILED IN END MSG
00000044'EF  03   00   02  F0    0A36          INSV    #ERROR,#0,#3,MOD_MSG_CODE ; ADJUST STATUS CODE FOR ERROR
                     C7  11      0A3F          BRB     10$                       ; LOOP BAK TO PROCESS NEXT TEST CASE
                                 0A41   ;
                                 0A41   ;      TC_CONTROL RETURNS TO TEST_SERV_EXEC VIA TESTSERV (IN TS_CLEANUP MACRO)
                                 0A41   ;
```

```
                             0A41      .SBTTL  SUBROUTINES
                             0A41 REG_SAVE:
                             0A41 ;
                             0A41 ;   ***********************************************************************
                             0A41 ;   *                                                                     *
                             0A41 ;   *  SAVES RO THRU SP IN REG_SAVE_AREA                                  *
                             0A41 ;   *                                                                     *
                             0A41 ;   ***********************************************************************
                             0A41 ;
              7FFF 8F    BB  0A41          PUSHR   #RO_THRU_SP                  ; SAVE ALL REGS ON STACK
  00000008'EF   6E  3C   28  0A45          MOVC3   #60,(SP),REG_SAVE_AREA       ; SAVE REGS (BEFORE S.S.)
              7FFF 8F    BA  0A4D          POPR    #RO_THRU_SP                  ; CLEAN UP STACK
                         05  0A51          RSB                                  ; .... AND RETURN
                             0A52 ;
                             0A52 ;
                             0A52 ;
                             0A52 ;
                             0A52 REG_REST:
                             0A52 ;
                             0A52 ;
                             0A52 ;   ***********************************************************************
                             0A52 ;   *                                                                     *
                             0A52 ;   *  RESTORES RO THRU SP FROM REG_SAVE_AREA                             *
                             0A52 ;   *                                                                     *
                             0A52 ;   ***********************************************************************
                             0A52 ;
                5E  3C   C2  0A52          SUBL2   #60,SP                       ; MOVE SP TO MAKE ROOM FOR REGS
 6E  00000008'EF    3C   28  0A55          MOVC3   #60,REG_SAVE_AREA,(SP)       ; MOVE REGS ONTO STACK FOR POP
              7FFF 8F    BA  0A5D          POPR    #RO_THRU_SP                  ; RESTORE ALL REGS FOR TESTSERV
                         05  0A61          RSB                                  ; ... AND RETURN
```

```
                                 0A62  REG_COMP:
                                 0A62  ;
                                 0A62  ;   ******************************************************************
                                 0A62  ;   *                                                                *
                                 0A62  ;   *   1) PUSHES ALL REGS ONTO STACK                                *
                                 0A62  ;   *   2) COMPARES REGISTER IMAGES FROM STACK WITH CORRESPONDING    *
                                 0A62  ;   *         IMAGES FROM REG_SAVE_AREA FOR ALL REGISTERS SPECIFIED  *
                                 0A62  ;   *         IN REG_COMP_MASK.                                      *
                                 0A62  ;   *   3) FOR EACH UNEQUAL COMPARE, AN ERROR MESSAGE IS PRINTED     *
                                 0A62  ;   *         (USING $FAO AND $OUTPUT SYSTEM SERVICES).              *
                                 0A62  ;   *   4) POPS ALL REGS OFF OF STACK                                *
                                 0A62  ;   *                                                                *
                                 0A62  ;   ******************************************************************
                                 0A62  ;
               7FFF 8F     BB    0A62          PUSHR   #R0_THRU_SP                     ; SAVE ALL REGISTERS ON STACK
      56   00000008'EF     DE    0A66          MOVAL   REG_SAVE_AREA,R6                ; POINT R6 TO BEG OF ...
                                 0A6D                                                  ; ... REGS (BEFORE S.S.)
              54    5E     D0    0A6D          MOVL    SP,R4                           ; POINT R4 TO BEG OF
                                 0A70                                                  ; ... REGS (AFTER S.S.)
           53    FF 8F     98    0A70          CVTBL   #-1,R3                          ; INITIALIZE REG_COMP_MASK INDEX
                                 0A74  REG_COMP_NEXT:
                    53     D6    0A74          INCL    R3                              ; POINT TO NEXT BIT IN MASK
              53    0F     91    0A76          CMPB    #15,R3                          ; END OF THE MASK ?
                    03     1A    0A79          BGTRU   REG_COMP_CONT                   ; NO -- CONTINUE
                  009F     31    0A7B          BRW     REG_COMP_RSB                    ; YES -- GO TO COMMON RETURN
                                 0A7E  REG_COMP_CONT:
              84    86     D1    0A7E          CMPL    (R6)+,(R4)+                     ; REG BEFORE = REG AFTER ?
                    F1     13    0A81          BEQLU   REG_COMP_NEXT                   ; YES -- LOOK FOR NEXT REG
   E9 00000000'EF   53     E1    0A83          BBC     R3,REG_COMP_MASK,REG_COMP_NEXT
                                 0A8B                                                  ; NO -- GET NEXT IF BIT NOT SET
      00000048'EF   53     D0    0A8B          MOVL    R3,CLOB_REG_NO                  ; NO -- GIVE REG NUMBER TO FAO
   0000004C'EF    FC A6    D0    0A92          MOVL    -4(R6),REG_BEFORE_SS            ; GIVE "BEFORE" CONTENTS TO FAO
   00000050'EF    FC A4    D0    0A9A          MOVL    -4(R4),REG_AFTER_SS             ; GIVE "AFTER" CONTENTS TO FAO
      00000056'EF   2A     90    0AA2          MOVB    #^A/*/,$$TSTN$$+2               ; GIVE FAILURE INDIC'N IN ERROR MSG
                                 0AA9  ;
                                 0AA9          $FAO_S  ERR_MSG_FAOCTL,OUTL,OUTD,$$SNAD$$,     -
                                 0AA9                  $$ASEQ$$,$$PSEQ$$,CLOB_REG_NO,REG_BEFORE_SS,REG_AFTER_SS
                                 0ADC  ;
   F631 CF    F5FB CF      B0    0ADC          MOVW    OUTL,OUTD                       ; ACTUAL OUTPUT LEN IN STRING DESC'R
                                 0AE3          PUTMSG  <#UETP$_TEXT,#1,#OUTD>          ; PRINT THE MSG
   F615 CF    0084 8F      B0    0AF8          MOVW    #OUTE-OUTB,OUTD                 ; GET MAX LEN BACK INTO DESCRIPTOR
      00000056'EF   20     90    0AFF          MOVB    #^A/ /,$$TSTN$$+2               ; REMOVE FAIL INDIC'N FOR NEXT MSG
00000060'EF   00000088'EF  DE    0B06          MOVAL   TEST_MOD_FAIL,TMD_ADDR          ; INDICATE FAILED IN END MSG
00000044'EF   03   00  02  F0    0B11          INSV    #ERROR,#0,#3,MOD_MSG_CODE       ; ADJUST STATUS CODE FOR ERROR
                  FF57     31    0B1A          BRW     REG_COMP_NEXT                   ; GO LOOK FOR NEXT REG TO COMPARE
                                 0B1D  REG_COMP_RSB:
               7FFF 8F     BA    0B1D          POPR    #R0_THRU_SP                     ; CLEAN UP STACK
                           05    0B21          RSB                                     ; RETURN TO CALLER
```

```
        0B22  MOD_MSG_PRINT:
        0B22  ;
        0B22  ;  ****************************************************************
        0B22  ;  *                                                              *
        0B22  ;  *   PRINTS THE TEST MODULE BEGUN/SUCCESSFUL/FAILED MESSAGES   *
        0B22  ;  *        (USING THE PUTMSG MACRO).                             *
        0B22  ;  *                                                              *
        0B22  ;  ****************************************************************
        0B22  ;
        0B22          PUTMSG  <MOD_MSG_CODE,#2,TMN_ADDR,TMD_ADDR> ; PRINT MSG
    05  0B3D          RSB                              ; ... AND RETURN TO CALLER
        0B3E  ;
        0B3E  CHMRTN:
        0B3E  ;  *********************************************************
        0B3E  ;  *
        0B3E  ;  *       CHANGE MODE ROUTINE. THIS ROUTINE GETS CONTROL WHENEVER
        0B3E  ;  *       A CMKRNL, CMEXEC, OR CMSUP SYSTEM SERVICE IS ISSUED
        0B3E  ;  *       BY THE MODE MACRO ('TO' OPTION).  IT MERELY DOES
        0B3E  ;  *       A JUMP INDIRECT ON A FIELD SET UP BY MODE. IT HAS
        0B3E  ;  *       THE EFFECT OF RETURNING TO THE END OF THE MODE
```

```
                              0B3E    ; *      MACRO EXPANSION.
                              0B3E    ; *
                              0B3E    ; *****************************************************************
                              0B3E    ;
                       0000   0B3E             .WORD   0                      ; ENTRY MASK
         00000079'FF    17    0B40             JMP     @CHM_CONT              ; RETURN TO MODE MACRO IN NEW MODE
                              0B46    ;
                              0B46    ; *      RET INSTR WILL BE ISSUED IN EXPANSION OF 'MODE FROM, ....' MACRO
                              0B46    ;
                              0B46             .END    SATSSF16
```

```
$$$CHARS            = 00000048                 P1_FAO                 000000DF R     02
$$$FIRSTTC$$$       = 00000000                 P2_FAO                 000000FA R     02
$$$STRINGS          = 00000000                 PHD$Q_PRIVMSK        = 00000000
$$ACT$$               000000F3 R      06       PRIVMASK               00000071 R     03
$$ARG$$               000000FB R      06       PRIV_ARGS            = 00000002
$$ASEQ$$              000000EB R      06       PRMLST_FAL             000000FE R     02
$$CALL$$              000000DF R      06       PROT                   000000B1 R     02
$$DISP$$              000001E6 R      06       PRT$C_NA               ********    X  02
$$ERR$$               000001A0 R      06       PRV$V_BUGCHK         = 00000017
$$EXP$$               000000F7 R      06       PRVPRT                 00000070 R     03
$$INIT$$              000000E3 R      06       R0_THRU_SP           = 00007FFF
$$MAXP$$            = 00000005                 REGS                   0000007D R     03
$$PSEQ$$              000000EF R      06       REG_AFTER_SS           00000050 R     03
$$SNAD$$              000000E7 R      06       REG_BEFORE_SS          0000004C R     03
$$T1                = 00000004                 REG_COMP               00000A62 R     06
$$T2                = 00000009                 REG_COMP_CONT          00000A7E R     06
$$TSTN$$              00000054 R      03       REG_COMP_MASK          00000000 R     02
CHMRTN                00000B3E R      06       REG_COMP_NEXT          00000A74 R     06
CHM_CONT              00000079 R      03       REG_COMP_RSB           00000B1D R     06
CLEANUP               000009E7 R      06       REG_REST               00000A52 R     06
CLOB_REG_NO           00000048 R      03       REG_SAVE               00000A41 R     06
COUNTED_STR           000000E3 R      02       REG_SAVE_AREA          00000008 R     03
CTL$GL_PHD            ********     X  06       RETADR                 00000068 R     03
CTRSTR_FAL            000000BD R      02       SATSSF16               00000000 R     06
CTRSTR_FAL10          000000CE R      02       SEVERE               = 00000004
CTRSTR_FAO            000000BD R      02       SHR$K_SHRDEF         = 00000001
CTRSTR_FAO10          000000CE R      02       SHR$_TEXT            = 00001130
CURRENT_TC            00000004 R      03       SS$_ACCVIO             ********    X  06
EMPTY                 00000000 R      04       SS$_BADPARAM           ********    X  06
ERROR               = 00000002                 SS$_BUFFEROVF          ********    X  06
ERR_MSG_FAOCTL        00000002 R      02       SS$_NOPRIV             ********    X  06
EXECUTE               000009C9 R      06       STS$V_INHIB_MSG      = 0000001C
GRP_TOTAL           = 00000003                 SUCCESS              = 00000001
INADR                 000000A9 R      02       SYS$CMKRNL             ********   GX  06
INFO                = 00000003                 SYS$EXIT               ********   GX  06
LIB$SIGNAL            ********     X  06       SYS$FAO                ********    X  06
MEXIT               = 00000000                 SYS$FAOL               ********   GX  06
MOD_MSG_CODE          00000044 R      03       SYS$HIBER              ********   GX  06
MOD_MSG_PRINT         00000B22 R      06       SYS$SETPRN             ********   GX  06
MSGBUF_SNE            00000102 R      02       SYS$SETPRT             ********   GX  06
MSGBUF_SNE10          00000102 R      02       SYS$SETPRV             ********   GX  06
MSGBUF_SNE11          00000008 R      05       SYS$SNDERR             ********   GX  06
MY_DISK               00000123 R      02       SYS$WAKE               ********   GX  06
NARGS               = 00000012                 TC1                    00000241 R     06
NOACCESS              00000000 R      05       TC2                    00000275 R     06
NSSARGS             = 00000001                 TC3                    000002A9 R     06
ONES                  000000B5 R      02       TCG_NO               = 00000003
OUTB                  0000011C R      06       TC_CONTROL             00000A00 R     06
OUTBUF_FAL            00000093 R      03       TEST_MOD_BEG           00000077 R     02
OUTBUF_FAL30          000000CD R      03       TEST_MOD_FAIL          00000088 R     02
OUTBUF_FAO            00000093 R      03       TEST_MOD_NAME          0000006E R     02
OUTBUF_FAO30          000000CD R      03       TEST_MOD_NAME_D        0000008F R     02
OUTD                  00000114 R      06       TEST_MOD_SUCC          0000007D R     02
OUTE                  000001A0 R      06       TMD_ADDR               00000060 R     03
OUTL                  000000DB R      06       TMN_ADDR               0000005C R     03
OUTLEN_FAL            00000091 R      03       TPID                   00000000 R     03
OUTLEN_FAO            00000091 R      03       TS1                    0000031D R     06
```

```
TS2                              0000064A R      06
TS3                              000008DE R      06
TS_EP                            00000064 R      03
TTNAME                           0000009F 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                          00000133  (   307.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR NOEXE   RD  NOWRT NOVEC LONG
RWDATA                          000000D6  (   214.)  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
SATSSF16                        00000B46  (  2886.)  06 (  6.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC LONG

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

Phase                    Page faults    CPU Time       Elapsed Time
-----                    -----------    --------       ------------
Initialization                   30    00:00:00.06    00:00:00.30
Command processing              112    00:00:00.65    00:00:01.84
Pass 1                          395    00:00:15.21    00:00:23.43
Symbol table sort                 0    00:00:01.06    00:00:01.09
Pass 2                          141    00:00:03.45    00:00:04.32
Symbol table output              16    00:00:00.09    00:00:00.11
Psect synopsis output             3    00:00:00.03    00:00:00.04
Cross-reference output            0    00:00:00.00    00:00:00.00
Assembler run totals            699    00:00:20.55    00:00:31.13

The working set limit was 1500 pages.
80713 bytes (158 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 601 non-local and 109 local symbols.
666 source lines were read in Pass 1, producing 28 object records in Pass 2.
61 pages of virtual memory were used to define 45 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                  18
TOTALS (all libraries)                             39

1216 GETS were required to define 39 macros.
```

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

MACRO/LIS=LIS$:SATSSF16/OBJ=OBJ$:SATSSF16 MSRC$:SATSSF16/UPDATE=(ENH$:SATSSF16)+EXECML$/LIB+SHRLIB$:UETP/LIB
```

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SATSSF15
LIS

SATSSF16
LIS

SATSSF14
LIS

SATSSF12
LIS

SATSSF13
LIS

0421 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SATSSS08
LIS

SATSSS05
LIS

SATSSS22
LIS

SATSSS07
LIS

SATSSS01
LIS