```
UUU          UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP        SSSSSSSSSSS  YYY          YYY
UUU          UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP        SSSSSSSSSSS  YYY          YYY
UUU          UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP        SSSSSSSSSSS  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
UUUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE        TTT        PPP                SSSSSSSSSSS              YYY
UUUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE        TTT        PPP                SSSSSSSSSSS              YYY
UUUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE        TTT        PPP                SSSSSSSSSSS              YYY
```

```
  SSSSSSSS    AAAAAA    TTTTTTTTTT    SSSSSSSS    SSSSSSSS  FFFFFFFFFF    000000      222222
  SSSSSSSS    AAAAAA    TTTTTTTTTT    SSSSSSSS    SSSSSSSS  FFFFFFFFFF    000000      222222
SS           AA    AA       TT      SS          SS          FF           00      00  22      22
SS           AA    AA       TT      SS          SS          FF           00      00  22      22
SS           AA    AA       TT      SS          SS          FF           00    0000          22
SS           AA    AA       TT      SS          SS          FF           00    0000          22
  SSSSSS     AA    AA       TT        SSSSSS      SSSSSS    FFFFFFFF     00  00  00        22
  SSSSSS     AA    AA       TT        SSSSSS      SSSSSS    FFFFFFFF     00  00  00        22
        SS   AAAAAAAAAA       TT              SS          SS  FF           0000    00      22
        SS   AAAAAAAAAA       TT              SS          SS  FF           0000    00      22
        SS   AA      AA       TT              SS          SS  FF           00      00    22
        SS   AA      AA       TT              SS          SS  FF           00      00    22          ....
SSSSSSSS     AA      AA       TT      SSSSSSSS    SSSSSSSS    FF             000000    2222222222  ....
SSSSSSSS     AA      AA       TT      SSSSSSSS    SSSSSSSS    FF             000000    2222222222  ....
```

```
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

```
                                    H 1
0000     1          .TITLE  SATSSF02 - SATS SYSTEM SERVICE TESTS  (FAILING S.C.)
0000     2          .IDENT  'V04-000'
0000     3
0000     4 ;
0000     5 ;*****************************************************************************
0000     6 ;*                                                                           *
0000     7 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9 ;*  ALL RIGHTS RESERVED.                                                     *
0000    10 ;*                                                                           *
0000    11 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    12 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    13 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    14 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    15 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    16 ;*  TRANSFERRED.                                                             *
0000    17 ;*                                                                           *
0000    18 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    19 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    20 ;*  CORPORATION.                                                             *
0000    21 ;*                                                                           *
0000    22 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    23 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24 ;*                                                                           *
0000    25 ;*                                                                           *
0000    26 ;*****************************************************************************
0000    27 ;
0000    28
0000    29 ;++
0000    30 ; FACILITY:     SATS SYSTEM SERVICE TESTS
0000    31 ;
0000    32 ; ABSTRACT:     THE SATSSF02 MODULE TESTS THE EXECUTION OF CERTAIN
0000    33 ; VMS SYSTEM SERVICES, INVOKED IN SUCH A WAY AS TO EXPECT FAILING
0000    34 ; STATUS CODES. THE SYSTEM SERVICES TESTED AND THE STATUS CODES
0000    35 ; EXPECTED ARE SUMMARIZED AS ARGUMENTS TO THE TESTSERV MACROS
0000    36 ; WHICH APPEAR NEAR THE END OF THIS LISTING. SUCCESSFUL STATUS
0000    37 ; CODES ARE TESTED IN OTHER MODULES.
0000    38 ;
0000    39 ;
0000    40 ; ENVIRONMENT:  USER MODE IMAGE; NEEDS CMKRNL PRIVILEGE,
0000    41 ;               DYNAMICALLY ACQUIRES OTHER PRIVILEGES, AS NEEDED.
0000    42 ;
0000    43 ; AUTHOR: THOMAS L. CAFARELLA,          CREATION DATE: MMM, 1978
0000    44 ;         PAUL D. FAY (DISPSERV & TESTSERV MACROS)
0000    45 ;
0000    46 ; MODIFIED BY:
0000    47 ;
0000    48 ;          : VERSION
0000    49 ; 01     -
0000    50 ;--
```

```
                                                .SBTTL  DECLARATIONS
                                        ;
                                        ; INCLUDE FILES:
                                        ;
                                                $PRVDEF                                 ; SYMBOL DEFS FOR PRIVILEGES
                                                $UETPDEF                                ; UETP MSG CODE DEFINITIONS
                                                $SHR_MESSAGES UETP,116,<<TEXT,INFO>>
                                                                                        ; DEFINE UETP$_TEXT
                                                                                        ; GET RID OF MACRO DEFINITIONS
                                        ;
                                        ; MACROS:
                                        ;
                                        ;
                                        ; EQUATED SYMBOLS:
                                        ;
00000000                                WARNING         = 0                             ; WARNING SEVERITY VALUE FOR MSGS
00000001                                SUCCESS         = 1                             ; SUCCESS SEVERITY VALUE FOR MSGS
00000002                                ERROR           = 2                             ; ERROR SEVERITY VALUE FOR MSGS
00000003                                INFO            = 3                             ; INFORMATIONAL SEV VALUE FOR MSGS
00000004                                SEVERE          = 4                             ; SEVERE (FATAL) SEV VALUE FOR MSGS
00000000                                TCG_NO          =  0                            ; INITIALIZE TEST CASE GROUP NUMBER
00000000                                GRP_TOTAL       =  0                            ; INITIALIZE TEST CASE GROUP TOTAL
00007FFF                                R0_THRU_SP      = ^M<R0,R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,AP,FP,SP>
00000001                                STATE_REF20     = 1                             ; STATE ARGUMENT FOR READEF (LOC 1)
                                        ;
                                        ; ****** THE FOLLOWING ASSIGNMENTS (IN PHD, PCB, STS) ARE BEING MADE
                                        ; ****** WITHOUT REFERENCE TO $PHDDEF, $PCBDEF, $STSDEF BECAUSE OF
                                        ; ****** SYMBOL TABLE OVERFLOW. FIX THIS WHEN MORE TABLE SPACE AVAILABLE.
                                        ;
00000000                                PHD$Q_PRIVMSK   = 0                             ; PRIV MASK OFFSET INTO PHD
00000020                                PCB$L_UIC       = ^X20                          ; UIC OFFSET INTO PCB
0000001C                                STS$V_INHIB_MSG = ^X1C                          ; INHIBIT_MSG BIT NUMBER IN MSG CODE
                                        ;
                                        ; OWN STORAGE:
                                        ;
```

```
                   00000000   88                .PSECT  RODATA,RD,NOWRT,NOEXE,LONG
               BFFC  0000   89 REG_COMP_MASK:  .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,AP,FP> ! ^X8000 -
                     0002   90                                          ; REG COMPARE MASK (HIGH-ORDER ...
                     0002   91                                          ; ... BIT MUST BE ON
                     0002   92 ERR_MSG_FAOCTL: STRING  I,<!/!AC!1ZB!1ZB: REGISTER !2UW CONTENTS ALTERED>, -
                     0002   93                 <;  BEFORE SERVICE CALL: !8XL   AFTER SERVICE CALL: !8XL>
                     006E   94 TEST_MOD_NAME:  STRING  C,<SATSSF02>     ; TEST MODULE NAME
                     0077   95 TEST_MOD_BEG:   STRING  C,<begun>        ; DISPOSITION FIELD OF TEST MOD MSG
                     007D   96 TEST_MOD_SUCC:  STRING  C,<successful>   ; DISPOSITION FIELD OF TEST MOD MSG
                     0088   97 TEST_MOD_FAIL:  STRING  C,<failed>       ; DISPOSITION FIELD OF TEST MOD MSG
                     008F   98 TEST_MOD_NAME_D: STRING I,<SATSSF02>     ; TEST MODULE NAME DESCRIPTOR
                     009F   99 TTNAME:         STRING  I,<TT>           ; TERMINAL LOGICAL NAME
00000000'00000000'   00A9  100 INADR:          .LONG   NOACCESS,NOACCESS ; PAGE ADDRESS OF NOACCESS PSECT
          00000000'  00B1  101 PROT:           .LONG   PRT$C_NA         ; PROTECTION CODE FOR NOACCESS PSECT
 FFFFFFFF FFFFFFFF   00B5  102 ONES:           .LONG   -1,-1            ; A QUADWORD OF 1-BITS
          00000016   00BD  103 EFN_SEF:        .LONG   22               ; EFN ARGUMENT FOR SETEF
          00000016   00C1  104 EFN_CEF:        .LONG   22               ; EFN ARGUMENT FOR CLREF
          00000000   00C5  105 EFN_REF:        .LONG   0                ; EFN ARGUMENT FOR READEF
          000000CD   00C9  106 STATE_REF21:    .BLKL   1                ; STATE ARGUMENT FOR READEF
          00000005   00CD  107 EFN_WFR:        .LONG   5                ; EFN ARGUMENT FOR WAITFR
          00000014   00D1  108 EFN_WFO:        .LONG   20               ; EFN ARGUMENT FOR WFLOR
          00000001   00D5  109 MASK_WFO:       .LONG   ^X1              ; MASK ARGUMENT FOR WFLOR
          00000002   00D9  110 EFN_WFA:        .LONG   2                ; EFN ARGUMENT FOR WFLAND
          FFFFFFFF   00DD  111 MASK_WFA:       .LONG   ^XFFFFFFFF       ; MASK ARGUMENT FOR WFLAND
```

```
                                    K 1
00000000    113          .PSECT  RWDATA,RD,WRT,NOEXE
00000004 0000    114 TPID:           .BLKL   1               ; PROCESS ID FOR THIS PROCESS
00000008 0004    115 CURRENT_TC:     .BLKL   1               ; PTR TO CURRENT TEST CASE
00000044 0008    116 REG_SAVE_AREA:  .BLKL   15              ; SAVE AREA FOR ALL REGS (SANS PC)
00748009 0044    117 MOD_MSG_CODE:   .LONG   UETP$_SATSMS    ; TEST MODULE MSG CODE FOR PUTMSG
0000004C 0048    118 CLOB_REG_NO:    .BLKL   1               ; CLOBBERED REG NO (FOR FAO ERR MSG)
00000050 004C    119 REG_BEFORE_SS:  .BLKL   1               ; REG CONTENTS BEFORE S.S.
         0050    120                                         ;  ... (FOR FAO ERROR MSG)
00000054 0050    121 REG_AFTER_SS:   .BLKL   1               ; REG CONTENTS AFTER S.S.
         0054    122                                         ;  ... (FOR FAO ERROR MSG)
         0054    123 $$TSTN$$:       STRING  C,<  SF   >     ; ASCII PORTION OF TEST CASE NAME
0000006E' 005C   124 TMN_ADDR:       .ADDRESS TEST_MOD_NAME  ; ADDR OF TEST MOD NAME FOR FAO
00000077' 0060   125 TMD_ADDR:       .ADDRESS TEST_MOD_BEG   ; ADDR OF T.M. DISP FIELD FOR FAO
00000068 0064    126 TS_EP:          .BLKL   1               ; ENTRY PNT FOR CURR TESTSERV MACRO
00000070 0068    127 RETADR:         .BLKL   2               ; RETURN LONGWORDS FOR SETPRT
00000071 0070    128 PRVPRT:         .BLKB   1               ; PROT RETURN BYTE FOR SETPRT
00000079 0071    129 PRIVMASK:       .BLKQ   1               ; ADDR OF PRIVILEGE MASK (IN PHD)
0000007D 0079    130 CHM_CONT:       .BLKL   1               ; CHANGE MODE CONTINUE ADDRESS
00000091 007D    131 REGS:           .BLKL   5               ; AREA FOR COND INDEX REGS (R2-R6)
00000095 0091    132 EFN_SEF10:      .BLKL   1               ; EFN ARGUMENT FOR SETEF
00000099 C095    133 EFN_SEF11:      .BLKL   1               ; EFN ARGUMENT FOR SETEF
0000009D 0099    134 EFN_SEF12:      .BLKL   1               ; EFN ARGUMENT FOR SETEF
000000A1 009D    135 EFN_SEF13:      .BLKL   1               ; EFN ARGUMENT FOR SETEF
000000A5 00A1    136 EFN_CEF10:      .BLKL   1               ; EFN ARGUMENT FOR CLREF
000000A9 00A5    137 EFN_CEF11:      .BLKL   1               ; EFN ARGUMENT FOR CLREF
000000AD 00A9    138 EFN_CEF12:      .BLKL   1               ; EFN ARGUMENT FOR CLREF
000000B1 00AD    139 EFN_CEF13:      .BLKL   1               ; EFN ARGUMENT FOR CLREF
000000B5 00B1    140 EFN_REF10:      .BLKL   1               ; EFN ARGUMENT FOR READEF
000000B9 00B5    141 EFN_REF11:      .BLKL   1               ; EFN ARGUMENT FOR READEF
000000BD 00B9    142 EFN_REF12:      .BLKL   1               ; EFN ARGUMENT FOR READEF
000000C1 00BD    143 EFN_REF13:      .BLKL   1               ; EFN ARGUMENT FOR READEF
000000C5 00C1    144 STATE_REF:      .BLKL   1               ; STATE ARGUMENT FOR READEF
000000C9 00C5    145 EFN_WFR10:      .BLKL   1               ; EFN ARGUMENT FOR WAITFR
000000CD 00C9    146 EFN_WFR11:      .BLKL   1               ; EFN ARGUMENT FOR WAITFR
000000D1 00CD    147 EFN_WFR12:      .BLKL   1               ; EFN ARGUMENT FOR WAITFR
000000D5 00D1    148 EFN_WFR13:      .BLKL   1               ; EFN ARGUMENT FOR WAITFR
000000D9 00D5    149 EFN_WFO10:      .BLKL   1               ; EFN ARGUMENT FOR WFLOR
000000DD 00D9    150 EFN_WFO11:      .BLKL   1               ; EFN ARGUMENT FOR WFLOR
000000E1 00DD    151 EFN_WFO12:      .BLKL   1               ; EFN ARGUMENT FOR WFLOR
000000E5 00E1    152 EFN_WFO13:      .BLKL   1               ; EFN ARGUMENT FOR WFLOR
000000E9 00E5    153 EFN_WFA10:      .BLKL   1               ; EFN ARGUMENT FOR WFLAND
000000ED 00E9    154 EFN_WFA11:      .BLKL   1               ; EFN ARGUMENT FOR WFLAND
000000F1 00ED    155 EFN_WFA12:      .BLKL   1               ; EFN ARGUMENT FOR WFLAND
000000F5 00F1    156 EFN_WFA13:      .BLKL   1               ; EFN ARGUMENT FOR WFLAND
```

```
                                        .PSECT  SATS_ACCVIO_1,RD,WRT,NOEXE,PAGE
00000200  0000    159 EMPTY:              .BLKB   512     ; RESERVE A PAGE OF SPACE
          0200    160 ;
          0200    161 ; +
          0200    162 ; ************************************************************************
          0200    163 ; *                                                                      *
          0200    164 ; *     THE ORDER OF STATEMENTS IN THIS PSECT IS CRITICAL.               *
          0200    165 ; *     DO NOT RE-ARRANGE THE VARIABLES. CONSULT SATS                    *
          0200    166 ; *     FUNCTIONAL SPECIFICATION FOR A DESCRIPTION OF THE USE            *
          0200    167 ; *     OF THE EMPTY PSECT (AND ITS COMPANION PSECT, NOACCESS).          *
          0200    168 ; *                                                                      *
          0200    169 ; ************************************************************************
          0200    170 ; -
          0200    171 ;
000001FF  0200    172 STATE_REF22         = . - 1         ; STATE ARG FOR READEF (LAST BYTE IN THE PAGE)
000001F3  0200    173                     . = . - 13      ; ALLOW ROOM FOR STRING DESCRIPTOR
          01F3    174 ; TYPE AAAAA_SSSX5 GO HERE:
00000006  01F3    175                     .LONG   6       ; STRING LENGTH (WILL CROSS PSECT BOUNDARY)
000001FB' 01F7    176                     .ADDRESS .+4    ; STRING ADDRESS
          01FB    177 ; TYPE AAAAA_SSSX3 GO HERE:
000001FC  01FB    178                     .BLKB   1       ; LOW-ORDER BYTE OF STRING LENGTH
          01FC    179 ; TYPE AAAAA_SSSX2 GO HERE:
00000200  01FC    180                     .BLKL   1       ; STRING LENGTH
          0200    181 ;
          0200    182 ;
          0200    183 ;
          0200    184 ;
                                        .PSECT  SATS_ACCVIO_2,RD,WRT,NOEXE,PAGE
00000200  0000    186 NOACCESS:           .BLKB   512     ; RESERVE A PAGE OF SPACE
00000000  0200    187                     . = . - 512     ; RETURN LOC CTR TO BEGINNING OF PSECT
00000000' 0000    188                     .ADDRESS EMPTY  ; ADDRESS OF ACCESSIBLE STRING
00000000' 0004    189                     .ADDRESS EMPTY/^X100 ; ADDRESS OF ACCESSIBLE STRING
          0008    190 ;+
          0008    191 ; *** NOTE -- DO NOT CHANGE LOCATION OR SEQUENCE OF ABOVE STATEMENTS!
          0008    192 ; ***         THIS PSECT (NOACCESS) MUST APPEAR IN MEMORY IMMEDIATELY
          0008    193 ; ***         FOLLOWING THE EMPTY PSECT. PSECT NAMES AND OPTIONS WILL BE
          0008    194 ; ***         CHOSEN TO FORCE THE DESIRED PSECT ORDERING.
          0008    195 ;-
          0008    196 ;
          0008    197 ;
          0008    198 ;
          0008    199 ;
                  200             .PSECT  SATSSF02,RD,WRT,EXE,LONG
```

```
                                    M 1
0000    202          .SBTTL  SATSSF02
0000    203 ;++
0000    204 ; FUNCTIONAL DESCRIPTION:
0000    205 ;
0000    206 ;                   AFTER PERFORMING SOME INITIAL HOUSEKEEPING, SUCH AS
0000    207 ; PRINTING THE MODULE BEGIN MESSAGE AND ACQUIRING ALL PRIVILEGES,
0000    208 ; THE SATSSF02 ROUTINE EXECUTES THE TEST_SERV_EXEC MACRO TO RUN
0000    209 ; ALL TEST CASES. WHEN THE MACRO COMPLETES ITS EXECUTION, SATSSF02
0000    210 ; PRINTS A TEST MODULE SUCCESS OR FAIL MESSAGE AND EXITS TO THE
0000    211 ; OPERATING SYSTEM. TEST_SERV_EXEC CALLS THE TC_CONTROL/TESTSERV
0000    212 ; CO-ROUTINE PAIR ONCE PER TEST CASE GROUP TO EXECUTE ALL TEST
0000    213 ; CASES IN THAT GROUP. EACH TEST CASE GROUP IS DEFINED BY BOUNDING
0000    214 ; ITS TEST CASES WITH A TC_GROUP MACRO BEFORE THE FIRST TEST CASE
0000    215 ; AND A TCEND MACRO AFTER THE LAST ONE. THE TEST CASES THEMSELVES
0000    216 ; ARE DEFINED WITHIN THESE BOUNDS BY PRECEDING EACH WITH A
0000    217 ; NEXT_TEST_CASE MACRO. TC_CONTROL/TESTSERV EXECUTES THE CODE
0000    218 ; FOLLOWING EACH NEXT_TEST_CASE MACRO IMMEDIATELY BEFORE ISSUING
0000    219 ; THE SYSTEM SERVICE AS REQUESTED IN THE TESTSERV MACRO. TC_CONTROL/
0000    220 ; TESTSERV ALSO CHECKS THE RESULTS OF THE SERVICE WITH RESPECT
0000    221 ; TO ITS EXPECTED STATUS CODE AND PRINTS ANY REQUIRED FAILURE
0000    222 ; MESSAGES FOR THE TEST CASE. THE CODE APPEARING AFTER EACH
0000    223 ; NEXT_TEST_CASE MACRO IS MERELY TO SET UP CONDITIONS REQUIRED
0000    224 ; FOR THE SYSTEM SERVICE AND TO CLEAN UP ANY RESOURCES ACQUIRED
0000    225 ; BY THE PREVIOUS TEST CASE.
0000    226 ;
0000    227 ; CALLING SEQUENCE:
0000    228 ;
0000    229 ;       $ RUN SATSSF02  ...  (DCL COMMAND)
0000    230 ;
0000    231 ; INPUT PARAMETERS:
0000    232 ;
0000    233 ;       NONE
0000    234 ;
0000    235 ; IMPLICIT INPUTS:
0000    236 ;
0000    237 ;       NONE
0000    238 ;
0000    239 ; OUTPUT PARAMETERS:
0000    240 ;
0000    241 ;       NONE
0000    242 ;
0000    243 ; IMPLICIT OUTPUTS:
0000    244 ;
0000    245 ;       MESSAGES TO SYS$OUTPUT ARE THE ONLY OUTPUT FROM SATSSF02.
0000    246 ;       THEY ARE OF THE FORM:
0000    247 ;
0000    248 ;               %UETP-S-SATSMS, TEST MODULE SATSSF02 BEGUN ... (BEGIN MSG)
0000    249 ;               %UETP-S-SATSMS, TEST MODULE SATSSF02 SUCCESSFUL ... (END MSG)
0000    250 ;               %UETP-E-SATSMS, TEST MODULE SATSSF02 FAILED ... (END MSG)
0000    251 ;               %UETP-I-TEXT, ... (VARIABLE INFORMATION ABOUT A TEST MODULE FAILURE)
0000    252 ;
0000    253 ; COMPLETION CODES:
0000    254 ;
0000    255 ;       THE SATSSF02 ROUTINE TERMINATES WITH A $EXIT TO THE
0000    256 ;       OPERATING SYSTEM WITH A STATUS CODE DEFINED BY UETP$_SATSMS.
0000    257 ;
0000    258 ; SIDE EFFECTS:
```

```
                                                N 1
                                0000   259 ;
                                0000   260 ;       NONE
                                0000   261 ;
                                0000   262 ;--
                                0000   263
                                0000   264
                                0000   265
                                0000   266 SATSSF02:
                           0FFC 0000   267         .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                                0002   268                                         ; ENTRY MASK
                                0002   269         $WAKE_S  TPID                   ; GET PID OF THIS PROCESS
                                0011   270         $HIBER_S                        ; UNDO WAKE
                                0018   271         $SETPRN_S TEST_MOD_NAME_D       ; SET PROCESS NAME
                         0DD2   30   0025   272         BSBW    MOD_MSG_PRINT           ; PRINT TEST MODULE BEGIN MSG
 00000060'EF   0000007D'EF   DE   0028   273         MOVAL   TEST_MOD_SUCC,TMD_ADDR  ; ASSUME END MSG WILL SHOW SUCCESS
00000044'EF   03   00   01   F0   0033   274         INSV    #SUCCESS,#0,#3,MOD_MSG_CODE ; ADJUST STATUS CODE FOR SUCCESS
                                003C   275         MODE    TO,10$,KRNL,NOREGS      ; KERNEL MODE TO ACCESS PHD
          59   00000000'9F   D0   0059   276         MOVL    @#CTL$GL_PHD,R9         ; GET PROCESS HEADER ADDRESS
          00000071'EF   69   DE   0060   277         MOVAL   PHD$Q_PRIVMSK(R9),PRIVMASK ; GET PRIV MASK ADDRESS
                                0067   278         MODE    FROM,TO$                ; GET BACK TO USER MODE
                                0068   279         PRIV    ADD,ALL                 ; GET ALL PRIVILEGES
                                0088   280         DISPSERV                        ; SET UP DISPLAY INFO FOR TESTSERV
                                021D   281         $SETPRT_S INADR=INADR, RETADR=RETADR, -
                                021D   282                   PROT=PROT, PRVPRT=PRVPRT
                                023E   283                                         ; SET NOACCESS PSECT ...
                                023E   284                                         ; ... FOR NO USER ACCESS
                         0A42   31   023E   285         BRW     EXECUTE                 ; GO EXECUTE ALL TEST CASES
                                0241   286         TC_GROUP        SEF,1,TS1
                                0268   287         NEXT_TEST_CASE  SFSEF10
```

```
00000091'EF    FF 8F    98  0268   288          CVTBL   #-1,EFN_SEF10   ; ILLEGAL EVENT FLAG NUMBER
                         0270   289          NEXT_TEST_CASE  SFSEF11
```

```
00000095'EF   80 8F   9A  027C   290          MOVZBL  #128,EFN_SEF11  ; ILLEGAL EVENT FLAG NUMBER
                      0284   291          NEXT_TEST_CASE  SFSEF12
```

```
                                                D 2
00000099'EF   01F4 8F   3C  0290   292          MOVZWL  #500,EFN_SEF12  ; ILLEGAL EVENT FLAG NUMBER
                        0299   293          NEXT_TEST_CASE  SFSEF13
```

```
                                                E 2
0000009D'EF   7F 8F   9A  02A5   294          MOVZBL  #127,EFN_SEF13  ; E.F. IN AN UNASSG'D COMM CLUSTER
                      02AD   295          TCEND
```

```
                02AE    296         TC_GROUP        CEF,1,TS2
                02D5    297         NEXT_TEST_CASE  SFCEF10
```

```
                                    G 2
000000A1'EF   FF 8F   98  02D5   298          CVTBL   #-1,EFN_CEF10   ; ILLEGAL EVENT FLAG NUMBER
                          02DD   299          NEXT_TEST_CASE  SFCEF11
```

```
                                                H 2
000000A5'EF    80 8F    9A  02E9    300         MOVZBL  #128,EFN_CEF11  ; ILLEGAL EVENT FLAG NUMBER
                                    02F1    301         NEXT_TEST_CASE  SFCEF12
```

```
                                                I 2
000000A9'EF   01F4 8F   3C  02FD   302          MOVZWL  #500,EFN_CEF12  ; ILLEGAL EVENT FLAG NUMBER
                        0306   303              NEXT_TEST_CASE  SFCEF13
```

```
                000000AD'EF   7F 8F    9A  0312    304         MOVZBL  #127,EFN_CEF13  ; E.F. IN AN UNASSG'D COMM CLUSTER
                                        031A    305         TCEND
```

```
                031B    306         TC_GROUP        REF,1,TS3
                0342    307         NEXT_TEST_CASE  SFREF10
```

```
                                                L 2
000000B1'EF   FF 8F   98  0342   308          CVTBL   #-1,EFN_REF10   ; ILLEGAL EVENT FLAG NUMBER
                          034A   309          NEXT_TEST_CASE  SFREF11
```

```
                                                M 2
000000B5'EF    80 8F    9A  0356    310         MOVZBL  #128,EFN_REF11  ; ILLEGAL EVENT FLAG NUMBER
                             035E    311         NEXT_TEST_CASE  SFREF12
```

```
000000B9'EF   01F4 8F   3C  036A   312          MOVZWL  #500,EFN_REF12  ; ILLEGAL EVENT FLAG NUMBER
                        0373   313          NEXT_TEST_CASE  SFREF13
```

```
000000BD'EF   7F 8F   9A  037F   314          MOVZBL  #127,EFN_REF13  ; E.F. IN AN UNASSG'D COMM CLUSTER
                      0387   315          NEXT_TEST_CASE  SFREF20
```

```
                0393    316         NEXT_TEST_CASE  SFREF21
```

```
     039F    317          NEXT_TEST_CASE  SFREF22
```

```
                03AB   318          TCEND
```

```
03AC   319          TC_GROUP        WFR,1,TS4
03D3   320          NEXT_TEST_CASE  SFWFR10
```

```
                                                G 3
000000C5'EF   FF 8F   98  03D3   321          CVTBL   #-1,EFN_WFR10   ; ILLEGAL EVENT FLAG NUMBER
                          03DB   322          NEXT_TEST_CASE  SFWFR11
```

```
                                                    H 3
000000C9'EF    80 8F    9A  03E7    323          MOVZBL  #128,EFN_WFR11  ; ILLEGAL EVENT FLAG NUMBER
                         03EF    324          NEXT_TEST_CASE  SFWFR12
```

```
                                                I 3
000000CD'EF   01F4 8F   3C  03FB   325          MOVZWL  #500,EFN_WFR12  ; ILLEGAL EVENT FLAG NUMBER
                        0404   326          NEXT_TEST_CASE  SFWFR13
```

```
000000D1'EF   7F 8F   9A  0410   327          MOVZBL  #127,EFN_WFR13  ; E.F. IN AN UNASSG'D COMM CLUSTER
                      0418   328          TCEND
```

```
                0419    329         TC_GROUP        WFO,1,TS5
                0440    330         NEXT_TEST_CASE  SFWF010
```

```
000000D5'EF   FF 8F   98  0440   331          CVTBL   #-1,EFN_WF010   ; ILLEGAL EVENT FLAG NUMBER
                      0448   332          NEXT_TEST_CASE  SFWF011
```

```
000000D9'EF    80 8F    9A  0454    333          MOVZBL  #128,EFN_WF011  ; ILLEGAL EVENT FLAG NUMBER
                         045C    334          NEXT_TEST_CASE  SFWF012
```

```
000000DD'EF   01F4 8F   3C  0468   335          MOVZWL  #500,EFN_WF012  ; ILLEGAL EVENT FLAG NUMBER
                        0471   336          NEXT_TEST_CASE  SFWF013
```

```
000000E1'EF   7F 8F   9A  047D   337          MOVZBL  #127,EFN_WF013  ; E.F. IN AN UNASSG'D COMM CLUSTER
                      0485   338          TCEND
```

```
                0486    339         TC_GROUP        WFA,1,TS6
                04AD    340         NEXT_TEST_CASE  SFWFA10
```

```
000000E5'EF    FF 8F    98  04AD    341          CVTBL   #-1,EFN_WFA10   ; ILLEGAL EVENT FLAG NUMBER
                         04B5    342          NEXT_TEST_CASE  SFWFA11
```

```
000000E9'EF   80 8F   9A  04C1    343          MOVZBL  #128,EFN_WFA11  ; ILLEGAL EVENT FLAG NUMBER
                      04C9    344          NEXT_TEST_CASE  SFWFA12
```

```
                                F 4
000000ED'EF   01F4 8F   3C  04D5   345          MOVZWL  #500,EFN_WFA12  ; ILLEGAL EVENT FLAG NUMBER
                        04DE   346          NEXT_TEST_CASE  SFWFA13
```

```
                                                G 4

000000F1'EF   7F 8F   9A  04EA    347         MOVZBL  #127,EFN_WFA13  ; E.F. IN AN UNASSG'D COMM CLUSTER
                      04F2    348         TCEND
```

```
04F3    349 TS1:
04F3    350         TESTSERV        SETEF,ERR,SATS,                             -
04F3    351                                                                             -
04F3    352            <1,EFN_SEF,                                              -
04F3    353                         EFN_SEF10,ILLEF.,       -  ; SFSEF10
04F3    354                         EFN_SEF11,ILLEFC,       -  ; SFSEF11
04F3    355                         EFN_SEF12,ILLEFC,       -  ; SFSEF12
04F3    356                         EFN_SEF13,UNASEFC,      -  ; SFSEF13
04F3    357                                                   >,                    -
04F3    358
05CD    359         TS_CLEANUP              ; CLEAN UP & RETURN TO TEST_SERV_EXEC
```

```
                                I 4
05ED    360 TS2:
05ED    361             TESTSERV        CLREF,ERR,SATS,                         -
05ED    362                                                                             -
05ED    363                <1,EFN_CEF,                                          -
05ED    364                                EFN_CEF10,ILLEFC,       -  ; SFCEF10
05ED    365                                EFN_CEF11,ILLEFC,       -  ; SFCEF11
05ED    366                                EFN_CEF12,ILLEFC,       -  ; SFCEF12
05ED    367                                EFN_CEF13,UNASEFC,      -  ; SFCEF13
05ED    368                                                          >,             -
05ED    369
06C7    370             TS_CLEANUP              ; CLEAN UP & RETURN TO TEST_SERV_EXEC
```

```
06E7   371 TS3:
06E7   372          TESTSERV        READEF,ERR,SATS,                                -
06E7   373                                                                          -
06E7   374             <1,EFN_REF,                                    -
06E7   375                          EFN_REF10,ILLEFC,       -  ; SFREF10
06E7   376                          EFN_REF11,ILLEFC,       -  ; SFREF11
06E7   377                          EFN_REF12,ILLEFC,       -  ; SFREF12
06E7   378                          EFN_REF13,UNASEFC,      -  ; SFREF13
06E7   379                                                    >,                    -
06E7   380                                                                          -
06E7   381             <1,STATE_REF,                                               -
06E7   382                          STATE_REF20,ACCVIO,     -  ; SFREF20
06E7   383                          STATE_REF21,ACCVIO,     -  ; SFREF21
06E7   384                          STATE_REF22,ACCVIO,     -  ; SFREF22
06E7   385                                                    >,                    -
06E7   386
0863   387          TS_CLEANUP                 ; CLEAN UP & RETURN TO TEST_SERV_EXEC
```

```
0883   388 TS4:
0883   389         TESTSERV        WAITFR,ERR,SATS,                                -
0883   390                                                                         -
0883   391            <1,EFN_WFR,                                                  -
0883   392                         EFN_WFR10,ILLEFC,       -  ; SFWFR10
0883   393                         EFN_WFR11,ILLEFC,       -  ; SFWFR11
0883   394                         EFN_WFR12,ILLEFC,       -  ; SFWFR12
0883   395                         EFN_WFR13,UNASEFC,      -  ; SFWFR13
0883   396                                                   >,                    -
0883   397
095E   398         TS_CLEANUP              ; CLEAN UP & RETURN TO TEST_SERV_EXEC
```

```
097E    399 TS5:
097E    400         TESTSERV        WFLOR,ERR,SATS,                                 -
097E    401                                                                                 -
097E    402            <1,EFN_WFO,                                                  -
097E    403                         EFN_WF010,ILLEFC,       -  ; SFWF010
097E    404                         EFN_WF011,ILLEFC,       -  ; SFWF011
097E    405                         EFN_WF012,ILLEFC,       -  ; SFWF012
097E    406                         EFN_WF013,UNASEFC,      -  ; SFWF013
097E    407                                                   >,                        -
097E    408                                                                             -
097E    409            <1,MASK_WFO,                                                 -
097E    410                                                   >,                        -
097E    411
0AE0    412         TS_CLEANUP              ; CLEAN UP & RETURN TO TEST_SERV_EXEC
```

```
0B00    413 TS6:
0B00    414         TESTSERV        WFLAND,ERR,SATS,                                -
0B00    415                                                                         -
0B00    416            <1,EFN_WFA,                                          -
0B00    417                         EFN_WFA10,ILLEFC,       -  ; SFWFA10
0B00    418                         EFN_WFA11,ILLEFC,       -  ; SFWFA11
0B00    419                         EFN_WFA12,ILLEFC,       -  ; SFWFA12
0B00    420                         EFN_WFA13,UNASEFC,      -  ; SFWFA13
0B00    421                                                   >,                    -
0B00    422                                                                         -
0B00    423            <1,MASK_WFA,                                         -
0B00    424                                                   >,                    -
0B00    425
0C63    426         TS_CLEANUP              ; CLEAN UP & RETURN TO TEST_SERV_EXEC
```

```
                                0C83   427          .SBTTL  EXECUTE & CLEANUP
                                0C83   428 EXECUTE:
                                0C83   429          TEST_SERV_EXEC                  ; EXECUTE ALL T. CASES IN ALL GROUPS
                                0CBF   430 CLEANUP:
                          0138   30    0CBF   431          BSBW    MOD_MSG_PRINT           ; PRINT TEST MODULE END MSG
00000044'EF   01    1C    01    F0    0CC2   432          INSV    #1,#STS$V_INHIB_MSG,#1,MOD_MSG_CODE
                                0CCB   433                                          ; INHIBIT PRINTING
                                0CCB   434          $EXIT_S MOD_MSG_CODE            ; EXIT TO OP SYS WITH MSG CODE
```

```
0CD8   436          .SBTTL  TC_CONTROL
0CD8   437 ;++
0CD8   438 ; FUNCTIONAL DESCRIPTION:
0CD8   439 ;
0CD8   440 ;                    THE TC_CONTROL SUBROUTINE IS CALLED BY THE TEST_SERV_EXEC
0CD8   441 ; MACRO TO EXECUTE A GROUP OF TEST CASES. A GROUP IS DEFINED BY A TC_GROUP
0CD8   442 ; MACRO. FOR EACH TC_GROUP MACRO, THERE IS A CORRESPONDING TESTSERV MACRO.
0CD8   443 ; TESTSERV CONTAINS CODE TO EXECUTE SYSTEM SERVICES AND CHECK THE RETURNED
0CD8   444 ; STATUS CODE VALUES. TESTSERV ARGUMENTS ARE CODED TO SPECIFY ALL THE SYSTEM
0CD8   445 ; SERVICE ARGUMENT VALUES AND THE EXPECTED STATUS CODE FOR EACH TEST CASE
0CD8   446 ; DEFINED BY A NEXT TEST CASE MACRO WITHIN THE GROUP. TC_CONTROL USES A
0CD8   447 ; CO-ROUTINE INTERFACE TO ENTER THE CODE OF THE APPROPRIATE TESTSERV MACRO
0CD8   448 ; IN VARIOUS PLACES. THE FIRST ENTRY OCCURS ONCE PER GROUP TO ALLOW TESTSERV
0CD8   449 ; TO DO SOME INITIALIZATION. THEN TWO ENTRIES ARE MADE FOR EACH TEST CASE IN
0CD8   450 ; THE GROUP. THE FIRST ALLOWS TESTSERV TO ISSUE THE SUBJECT SYSTEM SERVICE.
0CD8   451 ; THE SECOND ENTRY FOR THE TEST CASE CAUSES TESTSERV TO CHECK THE RETURNED
0CD8   452 ; STATUS CODE, PRINTING A FAILURE MESSAGE IF IT IS NOT THE EXPECTED CODE.
0CD8   453 ; IF THERE ARE NO MORE TEST CASES IN THE CURRENT GROUP, TESTSERV (NOT TC_CONTROL)
0CD8   454 ; RETURNS DIRECTLY TO TEST_SERV_EXEC (RSB ACTUALLY ISSUED IN TS_CLEANUP MACRO)
0CD8   455 ; FROM THIS SECOND ENTRY; OTHERWISE, CONTROL RETURNS TO TC_CONTROL WHICH
0CD8   456 ; IN TURN ENTERS TESTSERV AGAIN FOR THE NEXT TEST CASE. THE FAILURE OF A
0CD8   457 ; TEST CASE DOES NOT CAUSE TERMINATION OF THE TEST MODULE.
0CD8   458 ;
0CD8   459 ; CALLING SEQUENCE:
0CD8   460 ;
0CD8   461 ;       BSBW TC_CONTROL (ISSUED WITHIN THE TEST_SERV_EXEC MACRO)
0CD8   462 ;                       (RSB IS ISSUED WITHIN THE TS_CLEANUP MACRO)
0CD8   463 ;
0CD8   464 ; INPUT PARAMETERS:
0CD8   465 ;
0CD8   466 ;       NONE
0CD8   467 ;
0CD8   468 ; IMPLICIT INPUTS:
0CD8   469 ;
0CD8   470 ;       ARGUMENTS SPECIFIED ON EACH TESTSERV MACRO MAY BE VIEWED AS
0CD8   471 ;       INPUTS, SINCE TC_CONTROL AND TESTSERV ACT AS CO-ROUTINES.
0CD8   472 ;
0CD8   473 ; OUTPUT PARAMETERS:
0CD8   474 ;
0CD8   475 ;       SEVERITY CODE FIELD OF MOD_MSG_CODE (BITS 0,1,2) IS SET TO ERROR
0CD8   476 ;       IF ANY TEST CASE IN THE CURRENT GROUP FAILS; OTHERWISE IT REMAINS
0CD8   477 ;       SET TO SUCCESSFUL.
0CD8   478 ;
0CD8   479 ; IMPLICIT OUTPUTS:
0CD8   480 ;
0CD8   481 ;       %UETP-I-TEXT,   ERROR MESSAGES ARE WRITTEN TO SYS$OUTPUT BY
0CD8   482 ;                       THE TESTSERV MACRO (CO-ROUTINE WITH TC_CONTROL)
0CD8   483 ;
0CD8   484 ; COMPLETION CODES:
0CD8   485 ;
0CD8   486 ;       NONE
0CD8   487 ;
0CD8   488 ; SIDE EFFECTS:
0CD8   489 ;
0CD8   490 ;       NONE
0CD8   491 ;
0CD8   492 ;--
```

```
                          0CD8   493
                          0CD8   494
                          0CD8   495
                          0CD8   496 TC_CONTROL:
          00000064'EF   DD 0CD8   497          PUSHL   TS_EP                   ; PUSH TESTSERV ENTRY POINT
                 9E   16  0CDE   498          JSB     @(SP)+                  ; ENTER TESTSERV INITIALIZATION
                          0CE0   499 10$:                                     ; PROCESS NEXT TEST CASE
      00000056'EF   20 90 0CE0   500          MOVB    #^A/ /,$$TSTN$$+2       ; MAKE SURE T.C. NAME HAS A BLANK
               002F   30  0CE7   501          BSBW    REG_SAVE                ; SAVE REGISTERS
      00000004'FF   16 0CEA   502          JSB     @CURRENT_TC             ; JUMP TO CURRENT TEST CASE
               0037   30  0CF0   503          BSBW    REG_REST                ; RESTORE REGS FOR TESTSERV
                 9E   16  0CF3   504          JSB     @(SP)+                  ; LET TESTSERV ISSUE SYSTEM SERVICE
               0042   30  0CF5   505          BSBW    REG_COMP                ; COMPARE REGS TO SEE IF ...
                          0CF8   506                                          ; ... SYSTEM SERVICE CHANGED ANY
                 9E   16  0CF8   507          JSB     @(SP)+                  ; LET TESTSERV CHEK S.S. STATUS CODE
      00000056'EF   2A 91 0CFA   508          CMPB    #^A/*/,$$TSTN$$+2       ; HAS TESTSERV INDICATED FAILURE ?
                 DD   12  0D01   509          BNEQU   10$                     ; NO -- PROCESS NEXT TEST CASE
 00000060'EF   00000088'EF   DE  0D03   510          MOVAL   TEST_MOD_FAIL,TMD_ADDR  ; YES -- INDICATE FAILED IN END MSG
00000044'EF   03   00   02   F0  0D0E   511          INSV    #ERROR,#0,#3,MOD_MSG_CODE ; ADJUST STATUS CODE FOR ERROR
                 C7   11  0D17   512          BRB     10$                     ; LOOP BAK TO PROCESS NEXT TEST CASE
                          0D19   513 ;
                          0D19   514 ;    TC_CONTROL RETURNS TO TEST_SERV_EXEC VIA TESTSERV (IN TS_CLEANUP MACRO)
                          0D19   515 ;
```

```
                                0D19   517          .SBTTL  SUBROUTINES
                                0D19   518 REG_SAVE:
                                0D19   519 ;
                                0D19   520 ;  ************************************************************************
                                0D19   521 ;  *                                                                      *
                                0D19   522 ;  *  SAVES R0 THRU SP IN REG_SAVE_AREA                                    *
                                0D19   523 ;  *                                                                      *
                                0D19   524 ;  ************************************************************************
                                0D19   525 ;
                    7FFF 8F     BB 0D19   526          PUSHR   #R0_THRU_SP             ; SAVE ALL REGS ON STACK
        00000008'EF 6E    3C    28 0D1D   527          MOVC3   #60,(SP),REG_SAVE_AREA  ; SAVE REGS (BEFORE S.S.)
                    7FFF 8F     BA 0D25   528          POPR    #R0_THRU_SP             ; CLEAN UP STACK
                                05 0D29   529          RSB                             ; .... AND RETURN
                                0D2A   530 ;
                                0D2A   531 ;
                                0D2A   532 ;
                                0D2A   533 ;
                                0D2A   534 REG_REST:
                                0D2A   535 ;
                                0D2A   536 ;
                                0D2A   537 ;  ************************************************************************
                                0D2A   538 ;  *                                                                      *
                                0D2A   539 ;  *  RESTORES R0 THRU SP FROM REG_SAVE_AREA                               *
                                0D2A   540 ;  *                                                                      *
                                0D2A   541 ;  ************************************************************************
                                0D2A   542 ;
                    5E    3C    C2 0D2A   543          SUBL2   #60,SP                  ; MOVE SP TO MAKE ROOM FOR REGS
    6E  00000008'EF   3C    28 0D2D   544          MOVC3   #60,REG_SAVE_AREA,(SP)  ; MOVE REGS ONTO STACK FOR POP
                    7FFF 8F     BA 0D35   545          POPR    #R0_THR0_SP             ; RESTORE ALL REGS FOR TESTSERV
                                05 0D39   546          RSB                             ; ... AND RETURN
```

```
                                0D3A   548 REG_COMP:
                                0D3A   549 ;
                                0D3A   550 ;  ************************************************************************
                                0D3A   551 ;  *                                                                      *
                                0D3A   552 ;  *  1) PUSHES ALL REGS ONTO STACK                                        *
                                0D3A   553 ;  *  2) COMPARES REGISTER IMAGES FROM STACK WITH CORRESPONDING             *
                                0D3A   554 ;  *        IMAGES FROM REG_SAVE_AREA FOR ALL REGISTERS SPECIFIED         *
                                0D3A   555 ;  *        IN REG_COMP_MASK.                                             *
                                0D3A   556 ;  *  3) FOR EACH UNEQUAL COMPARE, AN ERROR MESSAGE IS PRINTED             *
                                0D3A   557 ;  *        (USING $FAO AND $OUTPUT SYSTEM SERVICES).                     *
                                0D3A   558 ;  *  4) POPS ALL REGS OFF OF STACK                                        *
                                0D3A   559 ;  *                                                                      *
                                0D3A   560 ;  ************************************************************************
                                0D3A   561 ;
                    7FFF 8F     BB 0D3A   562         PUSHR   #RO_THRU_SP             ; SAVE ALL REGISTERS ON STACK
            56  00000008'EF     DE 0D3E   563         MOVAL   REG_SAVE_AREA,R6        ; POINT R6 TO BEG OF ...
                                0D45   564                                         ; ... REGS (BEFORE S.S.)
                      54    5E  D0 0D45   565         MOVL    SP,R4                   ; POINT R4 TO BEG OF
                                0D48   566                                         ; ... REGS (AFTER S.S.)
                     53   FF 8F  98 0D48   567         CVTBL   #-1,R3                  ; INITIALIZE REG_COMP_MASK INDEX
                                0D4C   568 REG_COMP_NEXT:
                          53    D6 0D4C   569         INCL    R3                      ; POINT TO NEXT BIT IN MASK
                      53    0F  91 0D4E   570         CMPB    #15,R3                  ; END OF THE MASK ?
                            03  1A 0D51   571         BGTRU   REG_COMP_CONT           ; NO -- CONTINUE
                         009F   31 0D53   572         BRW     REG_COMP_RSB            ; YES -- GO TO COMMON RETURN
                                0D56   573 REG_COMP_CONT:
                      84    86  D1 0D56   574         CMPL    (R6)+,(R4)+             ; REG BEFORE = REG AFTER ?
                            F1  13 0D59   575         BEQLU   REG_COMP_NEXT           ; YES -- LOOK FOR NEXT REG
        E9 00000000'EF      53  E1 0D5B   576         BBC     R3,REG_COMP_MASK,REG_COMP_NEXT
                                0D63   577                                         ; NO -- GET NEXT IF BIT NOT SET
           00000048'EF      53  D0 0D63   578         MOVL    R3,CLOB_REG_NO          ; NO -- GIVE REG NUMBER TO FAO
         0000004C'EF        FC  A6 0D6A   579         MOVL    -4(R6),REG_BEFORE_SS    ; GIVE "BEFORE" CONTENTS TO FAO
         00000050'EF        FC  A4 0D72   580         MOVL    -4(R4),REG_AFTER_SS     ; GIVE "AFTER" CONTENTS TO FAO
            00000056'EF     2A  90 0D7A   581         MOVB    #^A/*/,$$TSTN$$+2       ; GIVE FAILURE INDIC'N IN ERROR MSG
                                0D81   582 ;
                                0D81   583         $FAO_S  ERR_MSG_FAOCTL,OUTL,OUTD,$$SNAD$$,  -
                                0D81   584                 $$ASEQ$$,$$PSEQ$$,CLOB_REG_NO,REG_BEFORE_SS,REG_AFTER_SS
                                0DB4   585 ;
              F359 CF    F323 CF  B0 0DB4   586         MOVW    OUTL,OUTD               ; ACTUAL OUTPUT LEN IN STRING DESC'R
                                0DBB   587         PUTMSG  <#UETP$_TEXT,#1,#OUTD>  ; PRINT THE MSG
              F33D CF    0084 8F  B0 0DD0   588         MOVW    #OUTE-OUTB,OUTD         ; GET MAX LEN BACK INTO DESCRIPTOR
               00000056'EF    20  90 0DD7   589         MOVB    #^A/ /,$$TSTN$$+2       ; REMOVE FAIL INDIC'N FOR NEXT MSG
 00000060'EF    00000088'EF     DE 0DDE   590         MOVAL   TEST_MOD_FAIL,TMD_ADDR  ; INDICATE FAILED IN END MSG
00000044'EF   03    00    02   F0 0DE9   591         INSV    #ERROR,#0,#3,MOD_MSG_CODE ; ADJUST STATUS CODE FOR ERROR
                       FF57   31 0DF2   592         BRW     REG_COMP_NEXT           ; GO LOOK FOR NEXT REG TO COMPARE
                                0DF5   593 REG_COMP_RSB:
                    7FFF 8F     BA 0DF5   594         POPR    #RO_THRU_SP             ; CLEAN UP STACK
                                05 0DF9   595         RSB                             ; RETURN TO CALLER
```

```
                                                F 5
                     0DFA   597 MOD_MSG_PRINT:
                     0DFA   598 ;
                     0DFA   599 ;  ************************************************************************
                     0DFA   600 ;  *                                                                      *
                     0DFA   601 ;  *  PRINTS THE TEST MODULE BEGUN/SUCCESSFUL/FAILED MESSAGES              *
                     0DFA   602 ;  *       (USING THE PUTMSG MACRO).                                      *
                     0DFA   603 ;  *                                                                      *
                     0DFA   604 ;  ************************************************************************
                     0DFA   605 ;
                     0DFA   606          PUTMSG  <MOD_MSG_CODE,#2,TMN_ADDR,TMD_ADDR> ; PRINT MSG
                  05 0E15   607          RSB                                         ; ... AND RETURN TO CALLER
                     0E16   608 ;
                     0E16   609 CHMRTN:
                     0E16   610 ; ************************************************************
                     0E16   611 ; *
                     0E16   612 ; *     CHANGE MODE ROUTINE. THIS ROUTINE GETS CONTROL WHENEVER
                     0E16   613 ; *     A CMKRNL, CMEXEC, OR CMSUP SYSTEM SERVICE IS ISSUED
                     0E16   614 ; *     BY THE MODE MACRO ('TO' OPTION).  IT MERELY DOES
                     0E16   615 ; *     A JUMP INDIRECT ON A FIELD SET UP BY MODE. IT HAS
                     0E16   616 ; *     THE EFFECT OF RETURNING TO THE END OF THE MODE
                     0E16   617 ; *     MACRO EXPANSION.
                     0E16   618 ; *
                     0E16   619 ; ************************************************************
                     0E16   620 ;
                0000 0E16   621          .WORD   0                                   ; ENTRY MASK
00000079'FF      17 0E18   622          JMP     @CHM_CONT                           ; RETURN TO MODE MACRO IN NEW MODE
                     0E1E   623 ;
                     0E1E   624 ; *     RET INSTR WILL BE ISSUED IN EXPANSION OF 'MODE FROM, ....' MACRO
                     0E1E   625 ;
                     0E1E   626          .END    SATSSF02
```

```
$$$CHARS                = 00000048
$$$FIRSTTC$$$           = 00000000
$$$STRINGS              = 00000000
$$ACT$$                   000000F3 R      06
$$ARG$$                   000000FB R      06
$$ASEQ$$                  000000EB R      06
$$CALL$$                  000000DF R      06
$$DISP$$                  000001E6 R      06
$$ERR$$                   000001A0 R      06
$$EXP$$                   000000F7 R      06
$$INIT$$                  000000E3 R      06
$$MAXP$$                = 00000005
$$PSEQ$$                  000000EF R      06
$$SNAD$$                  000000E7 R      06
$$T1                    = 00000004
$$T2                    = 00000009
$$TSTN$$                  00000054 R      03
CHMRTN                    00000E16 R      06
CHM_CONT                  00000079 R      03
CLEANUP                   00000CBF R      06
CLOB_REG_NO               00000048 R      03
CTL$GL_PRD                ******** X      06
CURRENT_TC                00000004 R      03
EFN_CEF                   000000C1 R      02
EFN_CEF10                 000000A1 R      03
EFN_CEF11                 000000A5 R      03
EFN_CEF12                 000000A9 R      03
EFN_CEF13                 000000AD R      03
EFN_REF                   000000C5 R      02
EFN_REF10                 000000B1 R      03
EFN_REF11                 000000B5 R      03
EFN_REF12                 000000B9 R      03
EFN_REF13                 000000BD R      03
EFN_SEF                   000000BD R      02
EFN_SEF10                 00000091 R      03
EFN_SEF11                 00000095 R      03
EFN_SEF12                 00000099 R      03
EFN_SEF13                 0000009D R      03
EFN_WFA                   000000D9 R      02
EFN_WFA10                 000000E5 R      03
EFN_WFA11                 000000E9 R      03
EFN_WFA12                 000000ED R      03
EFN_WFA13                 000000F1 R      03
EFN_WFO                   000000D1 R      02
EFN_WFO10                 000000D5 R      03
EFN_WFO11                 00000CD9 R      03
EFN_WFO12                 000000DD R      03
EFN_WFO13                 000000E1 R      03
EFN_WFR                   000000CD R      02
EFN_WFR10                 000000C5 R      03
EFN_WFR11                 000000C9 R      03
EFN_WFR12                 000000CD R      03
EFN_WFR13                 000000D1 R      03
EMPTY                     00000000 R      04
ERROR                   = 00000002
ERR_MSG_FAOCTL            00000002 R      02
EXECUTE                   00000C83 R      06
GRP_TOTAL               = 00000006
INADR                     000000A9 R      02
INFO                    = 00000003
LIB$SIGNAL                ******** X      06
MASK_WFA                  000000DD R      02
MASK_WFO                  000000D5 R      02
MEXIT                   = 00000000
MOD_MSG_CODE              00000044 R      03
MOD_MSG_PRINT             00000DFA R      06
NARGS                   = 0000000E
NOACCESS                  00000000 R      05
NSSARGS                 = 00000002
ONES                      000000B5 R      02
OUTB                      0000011C R      06
OUTD                      00000114 R      06
OUTE                      000001A0 R      06
OUTL                      000000DB R      06
PCB$L_UIC               = 00000020
PHD$Q_PRIVMSK           = 00000000
PRIVMASK                  00000071 R      03
PRIV_ARGS               = 00000002
PROT                      000000B1 R      02
PRT$C_NA                  ******** X      02
PRVPRT                    00000070 R      03
RO_THRU_SP              = 00007FFF
REGS                      0000007D R      03
REG_AFTER_SS              00000050 R      03
REG_BEFORE_SS             0000004C R      03
REG_COMP                  00000D3A R      06
REG_COMP_CONT             00000D56 R      06
REG_COMP_MASK             00000000 R      02
REG_COMP_NEXT             00000D4C R      06
REG_COMP_RSB              00000DF5 R      06
REG_REST                  00000D2A R      06
REG_SAVE                  00000D19 R      06
REG_SAVE_AREA             00000008 R      03
RETADR                    00000068 R      03
SATSSF02                  00000000 R      06
SEVERE                  = 00000004
SHR$K_SHRDEF            = 00000001
SHR$_TEXT               = 00001130
SS$_ACCVIO                ********   X    06
SS$_ILLEFC                ********   X    06
SS$_UNASEFC               ********   X    06
STATE_REF                 000000C1 R      03
STATE_REF20             = 00000001
STATE_REF21               000000C9 R      02
STATE_REF22             = 000001FF R      04
STS$V_INHIB_MSG         = 0000001C
SUCCESS                 = 00000001
SYS$CLREF                 ********  GX    06
SYS$CMKRNL                ********  GX    06
SYS$EXIT                  ********  GX    06
SYS$FAO                   ********    X   06
SYS$FAOL                  ********  GX    06
SYS$HIBER                 ********  GX    06
SYS$READEF                ********  GX    06
```

```
                                                H 5
SYS$SETEF                       ********  GX    06
SYS$SETPRN                      ********  GX    06
SYS$SETPRT                      ********  GX    06
SYS$SETPRV                      ********  GX    06
SYS$WAITFR                      ********  GX    06
SYS$WAKE                        ********  GX    06
SYS$WFLAND                      ********  GX    06
SYS$WFLOR                       ********  GX    06
TC1                             00000241  R     06
TC2                             000002AE  R     06
TC3                             000003B  R     06
TC4                             000003AC  R     06
TC5                             00000419  R     06
TC6                             00000486  R     06
TCG_NO                        = 00000006
TC_CONTROL                      00000CD8  R     06
TEST_MOD_BEG                    00000077  R     02
TEST_MOD_FAIL                   00000088  R     02
TEST_MOD_NAME                   0000006E  R     02
TEST_MOD_NAME_D                 0000008F  R     02
TEST_MOD_SUCC                   0000007D  R     02
TMD_ADDR                        00000060  R     03
TMN_ADDR                        0000005C  R     03
TPID                            00000000  R     03
TS1                             000004F3  R     06
TS2                             000005ED  R     06
TS3                             000006E7  R     06
TS4                             00000883  R     06
TS5                             0000097E  R     06
TS6                             00000B00  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	000000E1	(225.)	02	(2.)	NOPIC	USR	CON	REL	LCL	NOSHR	NOEXE	RD	NOWRT NOVEC LONG
RWDATA	000000F5	(245.)	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
SATSSF02	00000E1E	(3614.)	06	(6.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	34	00:00:00.09	00:00:00.33
Command processing	115	00:00:00.72	00:00:02.16
Pass 1	341	00:00:12.69	00:00:24.53
Symbol table sort	0	00:00:00.63	00:00:00.70
Pass 2	154	00:00:02.88	00:00:06.29
Symbol table output	18	00:00:00.13	00:00:00.13
Psect synopsis output	3	00:00:00.03	00:00:00.05
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	667	00:00:17.17	00:00:34.19

The working set limit was 1650 pages.
65959 bytes (129 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 349 non-local and 137 local symbols.
626 source lines were read in Pass 1, producing 30 object records in Pass 2.
63 pages of virtual memory were used to define 47 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SHRLIB]UETP.MLB;1	19
_255DUA28:[SYS.OBJ]LIB.MLB;1	0
_255DUA28:[SYSLIB]STARLET.MLB;2	22
TOTALS (all libraries)	41

947 GETS were required to define 41 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:SATSSF02/OBJ=OBJ$:SATSSF02 MSRC$:SATSSF02/UPDATE=(ENH$:SATSSF02)+EXECML$/LIB+SHRLIB$:UETP/LIB