```
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  EEEEEEEEEEE            TTT        PPPPPPPPPPPP      SSSSSSSSS             YYY
UUU         UUU  EEEEEEEEEEE            TTT        PPPPPPPPPPPP      SSSSSSSSS             YYY
UUU         UUU  EEEEEEEEEEE            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
UUUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE      TTT        PPP                 SSSSSSSSSSS          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        11
  SSSSSSSS    AAAAAA    TTTTTTTTTT    SSSSSSSS    SSSSSSSS  FFFFFFFFFF    000000        11
SS           AA    AA       TT      SS          SS          FF          00    00    1111
SS           AA    AA       TT      SS          SS          FF          00    00    1111
SS           AA    AA       TT      SS          SS          FF          00  0000      11
SS           AA    AA       TT      SS          SS          FF          00  0000      11
  SSSSSS     AA    AA       TT        SSSSSS      SSSSSS    FFFFFFFF    00  00  00    11
  SSSSSS     AA    AA       TT        SSSSSS      SSSSSS    FFFFFFFF    00  00  00    11
        SS   AAAAAAAAAA       TT              SS          SS  FF        0000    00    11
        SS   AAAAAAAAAA       TT              SS          SS  FF        0000    00    11
        SS   AA      AA       TT              SS          SS  FF        00      00    11          ....
        SS   AA      AA       TT              SS          SS  FF        00      00    11          ....
SSSSSSSS     AA      AA       TT      SSSSSSSS    SSSSSSSS    FF          000000    111111        ....
SSSSSSSS     AA      AA       TT      SSSSSSSS    SSSSSSSS    FF          000000    111111        ....
```

```
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     1          .TITLE  SATSSF01 - 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 SATSSF01 module tests the execution of the following
0000    33 ; VMS system services, invoked in such a way as to expect failing
0000    34 ; status codes:
0000    35 ;                   $DACEFC
0000    36 ;                   $DLCEFC
0000    37 ;                   $ASCEFC
0000    38 ;                   $SETEXV
0000    39 ;
0000    40 ;
0000    41 ; ENVIRONMENT:  User mode image; needs CMKRNL privilege,
0000    42 ;                   dynamically acquires other privileges, as needed.
0000    43 ;
0000    44 ; AUTHOR: THOMAS L. CAFARELLA,          CREATION DATE: AUG, 1978
0000    45 ;         PAUL D. FAY (DISPSERV & TESTSERV MACROS)
0000    46 ;
0000    47 ; MODIFIED BY:
0000    48 ;
0000    49 ;       V03-001 LDJ0001         Larry D. Jones,         17-Sep-1980
0000    50 ;                   Modified to conform to new build command procedures.
0000    51 ;**
0000    52 ;--
```

```
                                                L 15
                     0000     54 .SBTTL  DECLARATIONS
                     0000     55 ;
                     0000     56 ; MACRO LIBRARY CALLS
                     0000     57 ;
                     0000     58         $PRVDEF                                 ; privilege definitions
                     0000     59         $UETPDEF                                ; UETP message definitions
                     0000     60         $SHR_MESSAGES UETP,116,<<TEXT,INFO>> ; UETP$_TEXT definition
                     0000     61         $PHDDEF                                 ; process header definitions
                     0000     62         $PCBDEF                                 ; PCB definitions
                     0000     63         $SSDEF                                  ; SS definitions
                     0000     64         $STSDEF                                 ; STS definitions
                     0000     65 ;
                     0000     66 ; Equated symbols
                     0000     67 ;
00000000             0000     68 WARNING          = 0                             ; warning severity value for msgs
00000001             0000     69 SUCCESS          = 1                             ; success       "       "   "   "
00000002             0000     70 ERROR            = 2                             ; error         "       "   "   "
00000003             0000     71 INFO             = 3                             ; information   "       "   "   "
00000004             0000     72 SEVERE           = 4                             ; fatal         "       "   "   "
00000001             0000     73 PRVHND_SXV40     = 1                             ; page 0 address for SETEXV
                     0000     74
```

```
                                0000     76          .SBTTL  OWN STORAGE
                           00000000     77          .PSECT  RODATA,RD,NOWRT,NOEXE,LONG
                                0000     78 ;
                                0000     79 TEST_MOD_NAME:
        31 30 46 53 53 54 41 53 00'     0000     80          .ASCIC  /SATSSF01/              ; needed for SATSMS message
                                08       0000
                                0009     81 TEST_MOD_NAME_D:
46 53 53 54 41 53 00000011'010E0000'    0009     82          .ASCID  /SATSSF01/              ; module name
                             31 30      0017
                                0019     83 TEST_MOD_BEGIN:
                  6E 75 67 65 62 00'    0019     84          .ASCIC  /begun/
                                05       0019
                                001F     85 TEST_MOD_SUCC:
   6C 75 66 73 73 65 63 63 75 73 00'    001F     86          .ASCIC  /successful/
                                0A       001F
                                002A     87 TEST_MOD_FAIL:
               64 65 6C 69 61 66 00'    002A     88          .ASCIC  /failed/
                                06       002A
                                0031     89 DACEFC:
               43 46 45 43 41 44 00'    0031     90          .ASCIC  /DACEFC/
                                06       0031
                                0038     91 DLCEFC:
               43 46 45 43 4C 44 00'    0038     92          .ASCIC  /DLCEFC/
                                06       0038
                                003F     93 ASCEFC:
               43 46 45 43 53 41 00'    003F     94          .ASCIC  /ASCEFC/
                                06       003F
                                0046     95 SETEXV:
               56 58 45 54 45 53 00'    0046     96          .ASCIC  /SETEXV/
                                06       0046
                                004D     97 INADR:
                  00000000'00000000'    004D     98          .LONG   NOACCESS,NOACCESS       ; page address of noaccess psect
                                0055     99 PROT:
                           00000000'    0055    100          .LONG   PRT$C_NA                ; protection code for no access psect
                                0059    101 NAME_DLC:
   43 4C 44 46 53 00000061'010E0000'    0059    102          .ASCID  /SFDLC/                 ; legal name string
                                0066    103 NAME_DLC0:
                  0000006E'010E0000'    0066    104          .ASCID  //                      ; zero length string
                                006E    105 NAME_DLC15:
54 20 45 52 4F 4D 00000076'010E0000'    006E    106          .ASCID  /MORE THAN 15 CHARACTERS/ ; illegal string length test string
41 52 41 48 43 20 35 31 20 4E 41 48     007C
                     53 52 45 54 43     0088
                                008D    107 VECTOR_SXV:
                           00000000     008D    108          .LONG   0                       ; vector parameter for SETEXV
                                0091    109 ACMODE_SXV:
                           00000001     0091    110          .LONG   1                       ; access mode param. for SETEXV
                                0095    111 PRVHND_SXV41:
                           00000000     0095    112          .LONG   0                       ; readonly access for SETEXV
                                0099    113 CS1:
21 20 74 73 65 54 000000A1'010E0000'    0099    114          .ASCID  \Test !AC service name !AC step !UL failed.\
6E 20 65 63 69 76 72 65 73 20 43 41     00A7
70 65 74 73 20 43 41 21 20 65 6D 61     00B3
2E 64 65 6C 69 61 66 20 4C 55 21 20     00BF
                                00CB    115 CS2:
74 63 65 70 78 45 000000D3'010E0000'    00CB    116          .ASCID  \Expected !AS = !XL received !AS = !XL\
4C 58 21 20 3D 20 53 41 21 20 64 65     00D9
41 21 20 64 65 76 69 65 63 65 72 20     00E5
```

```
                4C 58 21 20 3D 20 53  00F1
                                      00F8    117 CS3:
74 63 65 70 78 45 00000100'010E0000'  00F8    118         .ASCID  \Expected !AS!UB = !XL received !AS!UB = !XL\
20 3D 20 42 55 21 53 41 21 20 64 65   0106
64 65 76 69 65 63 65 72 20 4C 58 21   0112
58 21 20 3D 20 42 55 21 53 41 21 20   011E
                                  4C  012A
                                      012B    119 EXP:
73 75 74 61 74 73 00000133'010E0000'  012B    120         .ASCID  \status\
```

```
                     0139   122 ;
                     0139   123         .SBTTL  R/W PSECT
                 00000000   124         .PSECT  RWDATA,RD,WRT,NOEXE,LONG
                     0000   125 ;
                     0000   126 TPID:
                 00000000 0000   127         .LONG   0                       ; PID for this process
                     0004   128 CURRENT_TC:
                 00000000 0004   129         .LONG   0                       ; ptr to current test case
                     0008   130         .ALIGN LONG
                     0008   131 REG_SAVE_AREA:
                 00000044 0008   132         .BLKL   15                      ; register save area
                     0044   133 MOD_MSG_CODE:
                 007480D9 0044   134         .LONG   UETP$_SATSMS            ; test module message code for putmsg
                     0048   135 TMN_ADDR:
                 00000000' 0048   136         .ADDRESS TEST_MOD_NAME
                     004C   137 TMD_ADDR:
                 00000019' 004C   138         .ADDRESS TEST_MOD_BEGIN
                     0050   139 PRVPRT:
                       00 0050   140         .BYTE   0                       ; protection return byte for SETPRT
                     0051   141 PRIVMASK:
            00000000 00000000 0051   142         .QUAD   0                       ; priv. mask
                     0059   143 CHM_CONT:
                 00000000 0059   144         .LONG   0                       ; change mode continue address
                     005D   145 RETADR:
                 00000065 005D   146         .BLKL   2                       ; returned address's from SETPRT
                     0065   147 DAC:
                     0065   148         $DACEFC 0                       ; DACEFC parameter list
                     006D   149 DLC:
                     006D   150         $DLCEFC NAME_DLC0                       ; DLCEFC parameter list
                     0075   151 ASC:
                     0075   152         $ASCEFC 0,0,0,1                 ; ASCEFC parameter list
                     0089   153 SET:
                     0089   154         $SETEXV VECTOR_SXV,0,ACMODE_SXV,PRVHND_SXV40 ; SETEXV parameter list
                     009D   155 REG:
74 73 69 67 65 72 00000A5'010E0000' 009D   156         .ASCID  \register R\
                    52 20 72 65 00AB
                     00AF   157 REGNUM:
                 00000000 00AF   158         .LONG   0                       ; register number
                     00B3   159 MSGL:
                 00000050 00B3   160         .LONG   80                      ; buffer desc.
                 000000BB' 00B7   161         .ADDRESS BUF
                     00BB   162 BUF:
                 0000010B 00BB   163         .BLKB   80
                     010B   164 MESSAGEL:
                 00000000 010B   165         .LONG   0                       ; message desc.
                 000000BB' 010F   166         .ADDRESS        BUF
                     0113   167 SERV_NAME:
                 00000000 0113   168         .LONG   0                       ; service name pointer
```

```
                     00000000   170                .PSECT  SATS_ACCVIO_1,RD,WRT,NOEXE,PAGE
00000200             0000   171 EMPTY:              .BLKB   512             ; reserve a page of space
                     0200   172 ;
                     0200   173 ; +
                     0200   174 ; ************************************************************************
                     0200   175 ; *                                                                      *
                     0200   176 ; *     THE ORDER OF STATEMENTS IN THIS PSECT IS CRITICAL.                *
                     0200   177 ; *     DO NOT RE-ARRANGE THE VARIABLES. CONSULT SATS                     *
                     0200   178 ; *     FUNCTIONAL SPECIFICATION FOR A DESCRIPTION OF THE USE            *
                     0200   179 ; *     OF THE EMPTY PSECT (AND ITS COMPANION PSECT, NOACCESS).           *
                     0200   180 ; *                                                                      *
                     0200   181 ; ************************************************************************
                     0200   182 ; -
                     0200   183 ;
000001FF             0200   184 PRVHND_SXV42    = . - 1                 ; prvhnd arg for SETEXV (last byte in the page)
000001F3             0200   185                 . = . - 13              ; allow room for string descriptor
                     01F3   186 ; type AAAAA_SSSX5 go here:
00000006             01F3   187                 .LONG   6               ; string length (will cross psect boundary)
000001FB'            01F7   188                 .ADDRESS .+4            ; string address
                     01FB   189 ; type AAAAA_SSSX3 go here:
000001FC             01FB   190                 .BLKB   1               ; low-order byte of string length
                     01FC   191 ; type AAAAA_SSSX2 go here:
00000200             01FC   192                 .BLKL   1               ; string length
                     0200   193 ;
                     0200   194 ;
                     0200   195 ;
                     0200   196 ;
                     00000000   197                .PSECT  SATS_ACCVIO_2,RD,WRT,NOEXE,PAGE
00000200             0000   198 NOACCESS:        .BLKB   512             ; reserve a page of space
00000000             0200   199                 . = . - 512             ; return loc ctr to beginning of psect
00000000'            0000   200                 .ADDRESS EMPTY          ; address of accessible string
00000000'            0004   201                 .ADDRESS EMPTY/^X100 ; address of accessible string
                     0008   202 ;+
                     0008   203 ; *** NOTE -- DO NOT CHANGE LOCATION OR SEQUENCE OF ABOVE STATEMENTS!
                     0008   204 ; ***          THIS PSECT (NOACCESS) MUST APPEAR IN MEMORY IMMEDIATELY
                     0008   205 ; ***          FOLLOWING THE EMPTY PSECT. PSECT NAMES AND OPTIONS WILL BE
                     0008   206 ; ***          CHOSEN TO FORCE THE DESIRED PSECT ORDERING.
                     0008   207 ;-
                     0008   208 ;
                     0008   209 ;
                     0008   210 ;
                     0008   211 ;
```

```
00000000   213          .PSECT  SATSSF01,RD,WRT,EXE,LONG
     0000  214          .SBTTL  SATSSF01
     0000  215 ;++
     0000  216 ; FUNCTIONAL DESCRIPTION:
     0000  217 ;
     0000  218 ;       After performing some initial housekeeping, such as
     0000  219 ; printing the module begin message and acquiring needed privileges,
     0000  220 ; the system services are tested in each of their failure conditions.
     0000  221 ; Detected failures are identified and  an error message is printed
     0000  222 ; on the terminal.  Upon completion of the test a success or fail
     0000  223 ; message is printed on the terminal.
     0000  224 ;
     0000  225 ; CALLING SEQUENCE:
     0000  226 ;
     0000  227 ;       $ RUN SATSSF01  ...  (DCL COMMAND)
     0000  228 ;
     0000  229 ; INPUT PARAMETERS:
     0000  230 ;
     0000  231 ;       none
     0000  232 ;
     0000  233 ; IMPLICIT INPUTS:
     0000  234 ;
     0000  235 ;       none
     0000  236 ;
     0000  237 ; OUTPUT PARAMETERS:
     0000  238 ;
     0000  239 ;       none
     0000  240 ;
     0000  241 ; IMPLICIT OUTPUTS:
     0000  242 ;
     0000  243 ;       Messages to SYS$OUTPUT are the only output from SATSSF01.
     0000  244 ;       They are of the form:
     0000  245 ;
     0000  246 ;               %UETP-S-SATSMS, TEST MODULE SATSSF01 BEGUN ... (BEGIN MSG)
     0000  247 ;               %UETP-S-SATSMS, TEST MODULE SATSSF01 SUCCESSFUL ... (END MSG)
     0000  248 ;               %UETP-E-SATSMS, TEST MODULE SATSSF01 FAILED ... (END MSG)
     0000  249 ;               %UETP-I-TEXT, ... (VARIABLE INFORMATION ABOUT A TEST MODULE FAILURE)
     0000  250 ;
     0000  251 ; COMPLETION CODES:
     0000  252 ;
     0000  253 ;       The SATSSF01 routine terminates with a $EXIT to the
     0000  254 ;       operating system with a status code defined by UETP$_SATSMS.
     0000  255 ;
     0000  256 ; SIDE EFFECTS:
     0000  257 ;
     0000  258 ;       none
     0000  259 ;
     0000  260 ;--
     0000  261
     0000  262
     0000  263
     0000  264          TEST_START SATSSF01              ; let the test begin
```

```
                                                E 16
                                0000
                          0000  0000                    .ENTRY SATSSF01,0
                0004'CF     D4  0002                    CLRL    W^CURRENT_TC
                      00    DD  0006                    PUSHL   #0
                0000'CF     DF  0008                    PUSHAL  W^TPID
      00000000'GF   02    FB  000C                    CALLS   #2,G^SYS$WAKE
      00000000'GF   00    FB  0013                    CALLS   #0,G^SYS$HIBER
                0009'CF     7F  001A                    PUSHAQ  W^TEST_MOD_NAME_D
      00000000'GF   01    FB  001E                    CALLS   #1,G^SYS$SETPRN
                  05AD      30  0025                    BSBW    W^MOD_MSG_PRINT
      004C'CF   001F'CF     DE  0028                    MOVAL   W^TEST_MOD_SUCC,W^TMD_ADDR
0044'CF   03    00    01    F0  002F                    INSV    #SUCCESS,#0,#3,W^MOD_MSG_CODE
                      00    DD  0036                    PUSHL   #0
            04DC'CF     01    FB  0038                    CALLS #1,W^REG_SAVE
                                003D            STP0:
                                003D  265               $SETPRT_S INADR=W^INADR, RETADR=W^RETADR, -
                                003D  266                         PROT=W^PROT, PRVPRT=W^PRVPRT ; set noaccess psect
                                0056  267                                                      ; ... for no user access
                                0056  268 .SBTTL DACEFC TESTS
                                0056  269 ;+
                                0056  270 ;
                                0056  271 ; $DACEFC tests
                                0056  272 ; test for an EFN of 0
                                0056  273 ;
                                0056  274 ;-
      0113'CF   0031'CF     DE  0056  275               MOVAL   W^DACEFC,W^SERV_NAME    ; set service name
                                005D  276               $DACEFC_G W^DAC
                                0066  277               FAIL_CHECK SS$_ILLEFC           ; check for correct failure
       000000EC 8F    DD  0066                            PUSHL   #SS$_ILLEFC
      04E6'CF   01    FB  006C                            CALLS   #1,W^REG_CHECK
                                0071  278               $DACEFC_S #0                    ; check _S form
                                007A  279               FAIL_CHECK SS$_ILLEFC           ; check for correct failure
       000000EC 8F    DD  007A                            PUSHL   #SS$_ILLEFC
      04E6'CF   01    FB  0080                            CALLS   #1,W^REG_CHECK
                                0085  280 ;+
                                0085  281 ;
                                0085  282 ; test for a non-zero but less than 64 EFN
                                0085  283 ;
                                0085  284 ;-
                                0085  285               NEXT_TEST
                                0085
                                0085            STP1:
      0004'CF   01    D0  0085                            MOVL    #1,W^CURRENT_TC
                      00    DD  008A                            PUSHL   #0
      04DC'CF   01    FB  008C                            CALLS   #1,W^REG_SAVE
      0069'CF   3F    D0  0091  286               MOVL    #63,W^DAC+DACEFC$_EFN   ; set EFN
                                0096  287               $DACEFC_G W^DAC
                                009F  288               FAIL_CHECK SS$_ILLEFC           ; check for correct failure
       000000EC 8F    DD  009F                            PUSHL   #SS$_ILLEFC
      04E6'CF   01    FB  00A5                            CALLS   #1,W^REG_CHECK
                                00AA  289               $DACEFC_S #63                   ; check the _S form
                                00B3  290               FAIL_CHECK SS$_ILLEFC           ; check for correct failure
       000000EC 8F    DD  00B3                            PUSHL   #SS$_ILLEFC
      04E6'CF   01    FB  00B9                            CALLS   #1,W^REG_CHECK
                                00BE  291 ;+
                                00BE  292 ;
                                00BE  293 ; test for a non-zero but greater than 127
```

```
                                00BE   294 ;
                                00BE   295 ;-
                                00BE   296          NEXT_TEST
                                00BE
                                00BE               STP2:
                0004'CF    02   D0   00BE                   MOVL    #2,W^CURRENT_TC
                           00   DD   00C3                   PUSHL   #0
                04DC'CF    01   FB   00C5                   CALLS   #1,W^REG_SAVE
        0069'CF  00000080  8F   D0   00CA   297          MOVL    #128,W^DAC+DACEFC$_EFN  ; set EFN
                                00D3   298          $DACEFC_G W^DAC
                                00DC   299          FAIL_CHECK SS$_ILLEFC           ; check for correct failure
                 000000EC  8F   DD   00DC                   PUSHL   #SS$_ILLEFC
                04E6'CF    01   FB   00E2                   CALLS   #1,W^REG_CHECK
                                00E7   300          $DACEFC_S #128                  ; check _S form
                                00F4   301          FAIL_CHECK SS$_ILLEFC           ; check for correct failure
                 000000EC  8F   DD   00F4                   PUSHL   #SS$_ILLEFC
                04E6'CF    01   FB   00FA                   CALLS   #1,W^REG_CHECK
```

```
                                                G 16
                     00FF    303 .SBTTL DLCEFC TESTS
                     00FF    304 ;+
                     00FF    305 ;
                     00FF    306 ; $DLCEFC tests
                     00FF    307 ; test for a zero length cluster name
                     00FF    308 ;
                     00FF    309 ;-
                     00FF    310         NEXT_TEST
                     00FF
                     00FF        STP3:
      0004'CF    03   D0  00FF                MOVL    #3,W^CURRENT_TC
                 00   DD  0104                PUSHL   #0
      04DC'CF    01   FB  0106                CALLS   #1,W^REG_SAVE
0113'CF  0038'CF    DE  010B    311          MOVAL   W^DLCEFC,W^SERV_NAME    ; set service name
                     0112    312          $DLCEFC_G W^DLC
                     011B    313          FAIL_CHECK SS$_IVLOGNAM         ; check for correct failure
       00000154 8F   DD  011B                PUSHL   #SS$_IVLOGNAM
      04E6'CF    01   FB  0121                CALLS   #1,W^REG_CHECK
                     0126    314          $DLCEFC_S W^NAME_DLC0           ; check the _S form
                     0131    315          FAIL_CHECK SS$_IVLOGNAM         ; check for correct failure
       00000154 8F   DD  0131                PUSHL   #SS$_IVLOGNAM
      04E6'CF    01   FB  0137                CALLS   #1,W^REG_CHECK
                     013C    316 ;+
                     013C    317 ;
                     013C    318 ; test for a non-zero but greater than 15 length cluster name
                     013C    319 ;
                     013C    320 ;-
                     013C    321         NEXT_TEST
                     013C
                     013C        STP4:
      0004'CF    04   D0  013C                MOVL    #4,W^CURRENT_TC
                 00   DD  0141                PUSHL   #0
      04DC'CF    01   FB  0143                CALLS   #1,W^REG_SAVE
0071'CF  006E'CF    DE  0148    322          MOVAL   W^NAME_DLC15,W^DLC+DLCEFC$_NAME ; set name address parameter
                     014F    323          $DLCEFC_G W^DLC
                     0158    324          FAIL_CHECK SS$_IVLOGNAM         ; check for correct failure
       00000154 8F   DD  0158                PUSHL   #SS$_IVLOGNAM
      04E6'CF    01   FB  015E                CALLS   #1,W^REG_CHECK
                     0163    325          $DLCEFC_S W^NAME_DLC15          ; check the _S form
                     016E    326          FAIL_CHECK SS$_IVLOGNAM         ; check for correct failure
       00000154 8F   DD  016E                PUSHL   #SS$_IVLOGNAM
      04E6'CF    01   FB  0174                CALLS   #1,W^REG_CHECK
                     0179    327 ;+
                     0179    328 ;
                     0179    329 ; a test for the requirement of PRMCEB privilege is not needed
                     0179    330 ; because a process, with the same UIC as the owner UIC of a
                     0179    331 ; created common EFC, can delete it without having the PRMCEB
                     0179    332 ; privilege.
                     0179    333 ;
                     0179    334 ;-
```

```
                                                H 16
                     0179   336 .SBTTL ASCEFC TESTS
                     0179   337 ;+
                     0179   338 ;
                     0179   339 ; $ASCEFC tests
                     0179   340 ; test for zero EFN
                     0179   341 ;
                     0179   342 ;-
                     0179   343         NEXT_TEST
                     0179
                     0179       STP5:
     0004'CF    05   D0   0179                 MOVL    #5,W^CURRENT_TC
                00   DD   017E                 PUSHL   #0
     04DC'CF    01   FB   0180                 CALLS   #1,W^REG_SAVE
0113'CF   003F'CF   DE   0185   344         MOVAL   W^ASCEFC,W^SERV_NAME    ; set service name
                     018C   345         $ASCEFC_G W^ASC
                     0195   346         FAIL_CHECK SS$_ILLEFC           ; check for correct failure
      000000EC 8F   DD   0195                 PUSHL   #SS$_ILLEFC
     04E6'CF    01   FB   019B                 CALLS   #1,W^REG_CHECK
                     01A0   347         $ASCEFC_S #0,W^NAME_DLC         ; check _S form
                     01B1   348         FAIL_CHECK SS$_ILLEFC           ; check for correct failure
      000000EC 8F   DD   01B1                 PUSHL   #SS$_ILLEFC
     04E6'CF    01   FB   01B7                 CALLS   #1,W^REG_CHECK
                     01BC   349 ;+
                     01BC   350 ;
                     01BC   351 ; test for non-zero but less than 64 EFN
                     01BC   352 ;
                     01BC   353 ;-
                     01BC   354         NEXT_TEST
                     01BC
                     01BC       STP6:
     0004'CF    06   D0   01BC                 MOVL    #6,W^CURRENT_TC
                00   DD   01C1                 PUSHL   #0
     04DC'CF    01   FB   01C3                 CALLS   #1,W^REG_SAVE
     0079'CF    3F   D0   01C8   355         MOVL    #63,W^ASC+ASCEFC$_EFN   ; set the EFN to 63
                     01CD   356         $ASCEFC_G W^ASC
                     01D6   357         FAIL_CHECK SS$_ILLEFC           ; check for correct failure
      000000EC 8F   DD   01D6                 PUSHL   #SS$_ILLEFC
     04E6'CF    01   FB   01DC                 CALLS   #1,W^REG_CHECK
                     01E1   358         $ASCEFC_S #63,W^NAME_DLC        ; check _S form
                     01F2   359         FAIL_CHECK SS$_ILLEFC           ; check for correct failure
      000000EC 8F   DD   01F2                 PUSHL   #SS$_ILLEFC
     04E6'CF    01   FB   01F8                 CALLS   #1,W^REG_CHECK
                     01FD   360 ;+
                     01FD   361 ;
                     01FD   362 ; test for a non-zero but greater than 127 EFN
                     01FD   363 ;
                     01FD   364 ;-
                     01FD   365         NEXT_TEST
                     01FD
                     01FD       STP7:
     0004'CF    07   D0   01FD                 MOVL    #7,W^CURRENT_TC
                00   DD   0202                 PUSHL   #0
     04DC'CF    01   FB   0204                 CALLS   #1,W^REG_SAVE
0079'CF   00000080 8F   D0   0209   366         MOVL    #128,W^ASC+ASCEFC$_EFN  ; set the EFN to 128
                     0212   367         $ASCEFC_G W^ASC
                     021B   368         FAIL_CHECK SS$_ILLEFC           ; check for the correct failure
      000000EC 8F   DD   021B                 PUSHL   #SS$_ILLEFC
```

```
                04E6'CF   01   FB  0221                        CALLS   #1,W^REG_CHECK
                                  0226   369          $ASCEFC_S #128,W^NAME_DLC        ; check _S form
                                  023B   370          FAIL_CHECK SS$_ILLEFC            ; check for correct failure
                 000000EC 8F   DD  023B                        PUSHL   #SS$_ILLEFC
                04E6'CF   01   FB  0241                        CALLS   #1,W^REG_CHECK
                                  0246   371 ;+
                                  0246   372 ;
                                  0246   373 ; test for a legal EFN but not addressable name string
                                  0246   374 ;
                                  0246   375 ;-
                                  0246   376          NEXT_TEST
                                  0246
                                  0246       STP8:
                0004'CF   08   D0  0246                        MOVL    #8,W^CURRENT_TC
                          00   DD  024B                        PUSHL   #0
                04DC'CF   01   FB  024D                        CALLS   #1,W^REG_SAVE
        0079'CF    00000040 8F   D0  0252   377          MOVL    #64,W^ASC+ASCEFC$_EFN    ; legalize the EFN
            007D'CF    0000'CF   DE  025B   378          MOVAL   W^NOACCESS,W^ASC+ASCEFC$_NAME ; set illegal address for name
                                  0262   379          $ASCEFC_G W^ASC
                                  026B   380          FAIL_CHECK SS$_ACCVIO            ; check for correct failure
                          0C   DD  026B                        PUSHL   #SS$_ACCVIO
                04E6'CF   01   FB  026D                        CALLS   #1,W^REG_CHECK
                                  0272   381          $ASCEFC_S #64,W^NOACCESS         ; check _S form
                                  0287   382          FAIL_CHECK SS$_ACCVIO            ; check for correct failure
                          0C   DD  0287                        PUSHL   #SS$_ACCVIO
                04E6'CF   01   FB  0289                        CALLS   #1,W^REG_CHECK
                                  028E   383 ;+
                                  028E   384 ;
                                  028E   385 ; test for 0 length cluster name
                                  028E   386 ;
                                  028E   387 ;-
                                  028E   388          NEXT_TEST
                                  028E
                                  028E       STP9:
                0004'CF   09   D0  028E                        MOVL    #9,W^CURRENT_TC
                          00   DD  0293                        PUSHL   #0
                04DC'CF   01   FB  0295                        CALLS   #1,W^REG_SAVE
        007D'CF    0066'CF   DE  029A   389          MOVAL   W^NAME_DLC0,W^ASC+ASCEFC$_NAME ; set 0 length name
                                  02A1   390          $ASCEFC_G W^ASC
                                  02AA   391          FAIL_CHECK SS$_IVLOGNAM          ; check for correct failure
                 00000154 8F   DD  02AA                        PUSHL   #SS$_IVLOGNAM
                04E6'CF   01   FB  02B0                        CALLS   #1,W^REG_CHECK
                                  02B5   392          $ASCEFC_S #64,W^NAME_DLC0        ; check _S form
                                  02CA   393          FAIL_CHECK SS$_IVLOGNAM          ; check for correct failure
                 00000154 8F   DD  02CA                        PUSHL   #SS$_IVLOGNAM
                04E6'CF   01   FB  02D0                        CALLS   #1,W^REG_CHECK
                                  02D5   394 ;+
                                  02D5   395 ;
                                  02D5   396 ; test for greater than 15 length cluster name
                                  02D5   397 ;
                                  02D5   398 ;-
                                  02D5   399          NEXT_TEST
                                  02D5
                                  02D5       STP10:
                0004'CF   0A   D0  02D5                        MOVL    #10,W^CURRENT_TC
                          00   DD  02DA                        PUSHL   #0
                04DC'CF   01   FB  02DC                        CALLS   #1,W^REG_SAVE
```

```
                                                J 16
007D'CF   006E'CF   DE  02E1   400          MOVAL   W^NAME_DLC15,W^ASC+ASCEFC$_NAME ; set 15 length name
                    02E8   401          $ASCEFC_G W^ASC
                    02F1   402          FAIL_CHECK SS$_IVLOGNAM          ; check for correct failure
                                                PUSHL   #SS$_IVLOGNAM
                                                CALLS   #1,W^REG_CHECK
                    02FC   403          $ASCEFC_S #64,W^NAME_DLC15       ; check _S form
                    0311   404          FAIL_CHECK SS$_IVLOGNAM          ; check for correct failure
                                                PUSHL   #SS$_IVLOGNAM
                                                CALLS   #1,W^REG_CHECK
                    031C   405 ;+
                    031C   406 ;
                    031C   407 ; test for need to have PRMCEB privilege
                    031C   408 ;
                    031C   409 ;-
                    031C   410          NEXT_TEST
                    031C
                    031C        STP11:
                                                MOVL    #11,W^CURRENT_TC
                                                PUSHL   #0
                                                CALLS   #1,W^REG_SAVE
                    0328   411          MODE    TO,10$,KRNL,NOREGS       ; kernal mode to access PHD
59                  0345   412          MOVL    @#CTL$GL_PHD,R9          ; get process header address
                    034C   413          MOVAL   PHD$Q_PRIVMSK(R9),W^PRIVMASK ; get priv mask address
                    0351   414          MODE    FROM,TO$                 ; get back to user mode
                    0352   415          PRIV    REM,PRMCEB               ; remove PRMCEB priv.
007D'CF   0059'CF   DE  0372   416          MOVAL   W^NAME_DLC,W^ASC+ASCEFC$_NAME ; set a legal name
                    00  DD  0379   417          PUSHL   #0                       ; push a dummy parameter
                    01  FB  037B   418          CALLS   #1,W^REG_SAVE            ; save the registers
                        0380   419          $ASCEFC_G W^ASC
                        0389   420          FAIL_CHECK SS$_NOPRIV            ; check for correct failure
                    24  DD  0389                        PUSHL   #SS$_NOPRIV
                    01  FB  038B                        CALLS   #1,W^REG_CHECK
                        0390   421          $ASCEFC_S #64,NAME_DLC,,#1       ; check _S form
                        03A7   422          FAIL_CHECK SS$_NOPRIV            ; check for correct failure
                    24  DD  03A7                        PUSHL   #SS$_NOPRIV
                    01  FB  03A9                        CALLS   #1,W^REG_CHECK
                        03AE   423          MODE    TO,20$,KRNL,NOREGS       ; kernal mode to access PHD
59                  03CB   424          MOVL    @#CTL$GL_PHD,R9          ; get process header address
                    03D2   425          MOVAL   PHD$Q_PRIVMSK(R9),W^PRIVMASK ; get priv mask address
                    03D7   426          MODE    FROM,20$                 ; get back to user mode
                    03D8   427          PRIV    ADD,PRMCEB               ; return PRMCEB priv.
```

```
                          03F8    429 .SBTTL SETEXV TESTS
                          03F8    430 ;+
                          03F8    431 ;
                          03F8    432 ; $SETEXV TESTS
                          03F8    433 ; test for page 0 access
                          03F8    434 ;
                          03F8    435 ;-
                          03F8    436         NEXT_TEST
                          03F8
                          03F8        STP12:
     0004'CF    0C    D0  03F8                MOVL    #12,W^CURRENT_TC
                0C    DD  03FD                PUSHL   #0
     04DC'CF    01    FB  03FF                CALLS   #1,W^REG_SAVE
0113'CF   0046'CF    DE  0404    437         MOVAL   W^SETEXV,W^SERV_NAME    ; set service name
                          040B    438         $SETEXV_G W^SET
                          0414    439         FAIL_CHECK SS$_ACCVIO           ; check for correct failure
                0C    DD  0414                PUSHL   #SS$_ACCVIO
     04E6'CF    01    FB  0416                CALLS   #1,W^REG_CHECK
                          041B    440         $SETEXV_S W^VECTOR_SXV,0,-
                          041B    441                 W^ACMODE_SXV,W^PRVHND_SXV40 ; check _S form
                          0430    442         FAIL_CHECK SS$_ACCVIO           ; check for correct failure
                0C    DD  0430                PUSHL   #SS$_ACCVIO
     04E6'CF    01    FB  0432                CALLS   #1,W^REG_CHECK
                          0437    443 ;+
                          0437    444 ;
                          0437    445 ; test for read-only psect access
                          0437    446 ;
                          0437    447 ;-
                          0437    448         NEXT_TEST
                          0437
                          0437        STP13:
     0004'CF    0D    D0  0437                MOVL    #13,W^CURRENT_TC
                0C    DD  043C                PUSHL   #0
     04DC'CF    01    FB  043E                CALLS   #1,W^REG_SAVE
0099'CF   0095'CF    DE  0443    449         MOVAL   W^PRVHND_SXV41,W^SET+SETEXV$_PRVHND
                          044A    450         $SETEXV_G W^SET
                          0453    451         FAIL_CHECK SS$_ACCVIO           ; check for correct failure
                0C    DD  0453                PUSHL   #SS$_ACCVIO
     04E6'CF    01    FB  0455                CALLS   #1,W^REG_CHECK
                          045A    452         $SETEXV_S W^VECTOR_SXV,0,-
                          045A    453                 W^ACMODE_SXV,W^PRVHND_SXV41 ; check _S form
                          046F    454         FAIL_CHECK SS$_ACCVIO           ; check for correct failure
                0C    DD  046F                PUSHL   #SS$_ACCVIO
     04E6'CF    01    FB  0471                CALLS   #1,W^REG_CHECK
                          0476    455 ;+
                          0476    456 ;
                          0476    457 ; test for noaccess psect protection
                          0476    458 ;
                          0476    459 ;-
                          0476    460         NEXT_TEST
                          0476
                          0476        STP14:
     0004'CF    0E    D0  0476                MOVL    #14,W^CURRENT_TC
                0C    DD  047B                PUSHL   #0
     04DC'CF    01    FB  047D                CALLS   #1,W^REG_SAVE
0099'CF   01FF'CF    DE  0482    461         MOVAL   W^PRVHND_SXV42,W^SET+SETEXV$_PRVHND
                          0489    462         $SETEXV_G W^SET
```

```
                                 0492   463        FAIL_CHECK SS$_ACCVIO                   ; check for correct failure
                          0C DD  0492                       PUSHL   #SS$_ACCVIO
                 04E6'CF  01 FB  0494                       CALLS   #1,W^REG_CHECK
                                 0499   464        $SETEXV_S W^VECTOR_SXV,0,-
                                 0499   465                  W^ACMODE_SXV,W^PRVHND_SXV42 ; check _S form
                                 04AE   466        FAIL_CHECK SS$_ACCVIO                   ; check for correct failure
                          0C DD  04AE                       PUSHL   #SS$_ACCVIO
                 04E6'CF  01 FB  04B0                       CALLS   #1,W^REG_CHECK
                                 04B5   467        TEST_END                                ; end the test
                     004C'CF DD  04B5                       PUSHL   W^TMD_ADDR
                     0048'CF DD  04B9                       PUSHL   W^TMN_ADDR
                          02 DD  04BD                       PUSHL   #2
                     0044'CF DD  04BF                       PUSHL   W^MOD_MSG_CODE
           00000000'GF   04 FB  04C3                       CALLS   #$$T1,G^LIB$SIGNAL
0044'CF    01    1C   01 F0  04CA                       INSV    #1,#STS$V_INHIB_MSG,#1,W^MOD_MSG_CODE
                     0044'CF DD  04D1                       PUSHL   W^MOD_MSG_CODE
           00000000'GF   01 FB  04D5                       CALLS   #1,G^SYS$EXIT
```

```
                                                M 16
                                04DC    469 .SBTTL REG_SAVE
                                04DC    470 ;++
                                04DC    471 ; FUNCTIONAL DESCRIPTION:
                                04DC    472 ;       Subroutine to save R2-R11 in the register save location.
                                04DC    473 ;
                                04DC    474 ; CALLING SEQUENCE:
                                04DC    475 ;       PUSHL   #0                      ; save a dummy parameter
                                04DC    476 ;       CALLS   #1,W^REG_SAVE   ; save R2-R11
                                04DC    477 ;
                                04DC    478 ; INPUT PARAMETERS:
                                04DC    479 ;       NONE
                                04DC    480 ;
                                04DC    481 ; OUTPUT PARAMETERS:
                                04DC    482 ;       NONE
                                04DC    483 ;
                                04DC    484 ;--
                                04DC    485
                                04DC    486 REG_SAVE:
                           0FFC 04DC    487         .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
        0008'CF   14 AD   28    28 04DE    488         MOVC3   #4*10,^X14(FP),W^REG_SAVE_AREA  ; save the registers in the program
                                04E5    489         RET
                                04E6    490         .SBTTL  REG_CHECK
                                04E6    491 ;++
                                04E6    492 ; FUNCTIONAL DESCRIPTION:
                                04E6    493 ;       Subroutine to test R0 & R2-R11 for proper content after a service
                                04E6    494 ;       execution. A snapshot is taken by the REG_SAVE routine at the
                                04E6    495 ;       beginning of each step and this routine is executed after the
                                04E6    496 ;       services have been executed.
                                04E6    497 ;
                                04E6    498 ; CALLING SEQUENCE:
                                04E6    499 ;       PUSHL   #SS$_XXXXXX     ; push expected R0 contents
                                04E6    500 ;       CALLS   #1,W^REG_CHECK  ; execute this routine
                                04E6    501 ;
                                04E6    502 ; INPUT PARAMETERS:
                                04E6    503 ;       expected R0 contents on the stack
                                04E6    504 ;
                                04E6    505 ; OUTPUT PARAMETERS:
                                04E6    506 ;       possible error messages printed using $PUTMSG
                                04E6    507 ;
                                04E6    508 ;--
                                04E6    509
                                04E6    510 REG_CHECK:
                           0FFC 04E6    511         .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
             50    04 AC      D1 04E8    512         CMPL    4(AP),R0                        ; is this the right fail code?
                      0E      13 04EC    513         BEQL    10$                             ; br if yes
                      50      DD 04EE    514         PUSHL   R0                              ; push received data
                   04 AC      DD 04F0    515         PUSHL   4(AP)                           ; push expected data
                012B'CF       DF 04F3    516         PUSHAL  W^EXP                           ; push the string variable
           052E'CF    03      FB 04F7    517         CALLS   #3,W^PRINT_FAIL                 ; print the error message
                                04FC    518 10$:
 0008'CF   14 AD   28    29 04FC    519         CMPC3   #4*10,^X14(FP),W^REG_SAVE_AREA  ; check all but R0
                   28    13 0503    520         BEQL    20$                             ; br if O.K.
56   53   00000008'8F    C3 0505    521         SUBL3   #REG_SAVE_AREA,R3,R6            ; calculate the register number
             56    04    C6 050D    522         DIVL2   #4,R6
     00AF'CF   56    02    81 0510    523         ADDB3   #^X2,R6,W^REGNUM                ; put it in the string
             51    03    CA 0516    524         BICL2   #3,R1                           ; backup to register boundry
             53    03    CA 0519    525         BICL2   #3,R3
```

```
                                                B 1
                00AF'CF    DD  051C   526          PUSHL   W^REGNUM                        ; push register number
                      61   DD  0520   527          PUSHL   (R1)                           ; push received data
                      63   DD  0522   528          PUSHL   (R3)                           ; push expected data
                009D'CF    DF  0524   529          PUSHAL  W^REG                          ; set string pntr param.
           052E'CF    04   FB  0528   530          CALLS   #4,W^PRINT_FAIL                ; print the error message
                           052D   531 20$:
                           052D   532          RET
                           052E   533          .SBTTL  PRINT_FAIL
                           052E   534 ;++
                           052E   535 ; FUNCTIONAL DESCRIPTION:
                           052E   536 ;       Subroutine to report failures using $PUTMSG
                           052E   537 ;
                           052E   538 ; CALLING SEQUENCE:
                           052E   539 ; Mode  #1      PUSHL EXPECTED  Mode    #2      PUSHL REG_NUMBER
                           052E   540 ;               PUSHL RECEIVED                   PUSHL EXPECTED
                           052E   541 ;               PUSHAL STRING_VAR                PUSHL RECEIVED
                           052E   542 ;               CALLS #3,W^PRINT_FAIL            PUSHAL STRING_VAR
                           052E   543 ;                                                CALLS #4,W^PRINT_FAIL
                           052E   544 ;
                           052E   545 ; INPUT PARAMETERS:
                           052E   546 ;       listed above
                           052E   547 ;
                           052E   548 ; OUTPUT PARAMETERS:
                           052E   549 ;       an error message is printed using $PUTMSG
                           052E   550 ;
                           052E   551 ;--
                           052E   552
                           052E   553 PRINT_FAIL:
                     003C  052E   554          .WORD   ^M<R2,R3,R4,R5>
                           0530   555          $FAO_S  W^CS1,W^MESSAGEL,W^MSGL,#TEST_MOD_NAME,W^SERV_NAME,W^CURRENT_TC
                           0551   556          PUTMSG  <#UETP$_TEXT,#1,#MESSAGEL>     ; print the message
                04   6C    91   0566   557          CMPB    (AP),#4                        ; is this a register message?
                     21    13   0569   558          BEQL    10$                            ; br if yes
                           056B   559          $FAO_S  W^CS2,W^MESSAGEL,W^MSGL,4(AP),8(AP),4(AP),12(AP)
                           25    11   058A   560          BRB     20$                            ; goto output message
                           058C   561 10$:
                           058C   562          $FAO_S  W^CS3,W^MESSAGEL,W^MSGL,4(AP),16(AP),8(AP),4(AP),16(AP),12(AP)
                           05B1   563 20$:
                           05B1   564          PUTMSG  <#UETP$_TEXT,#1,#MESSAGEL>     ; print the message
           004C'CF    002A'CF   DE   05C6   565          MOVAL   W^TEST_MOD_FAIL,W^TMD_ADDR     ; set failure message address
0044'CF    03    00    02   F0   05CD   566          INSV    #ERROR,#0,#3,W^MOD_MSG_CODE    ; set severity code
                           04    05D4   567          RET
```

```
                                C 1
                                        .SBTTL  MOD_MSG_PRINT
                                MOD_MSG_PRINT:
                                ;
                                ;  ***************************************************************************
                                ;  *                                                                         *
                                ;  *  PRINTS THE TEST MODULE BEGUN/SUCCESSFUL/FAILED MESSAGES                *
                                ;  *       (USING THE PUTMSG MACRO).                                         *
                                ;  *                                                                         *
                                ;  ***************************************************************************
                                ;
                                        PUTMSG  <W^MOD_MSG_CODE,#2,W^TMN_ADDR,W^TMD_ADDR> ; PRINT MSG
                             05 05EA    RSB                                     ; ... AND RETURN TO CALLER
                                ;
                                        .SBTTL  CHMRTN
                                CHMRTN:
                                ;  ***************************************************************************
                                ;  *                                                                         *
                                ;  *     CHANGE MODE ROUTINE. THIS ROUTINE GETS CONTROL WHENEVER             *
                                ;  *     A CMKRNL, CMEXEC, OR CMSUP SYSTEM SERVICE IS ISSUED                 *
                                ;  *     BY THE MODE MACRO ('TO' OPTION).  IT MERELY DOES                    *
                                ;  *     A JUMP INDIRECT ON A FIELD SET UP BY MODE. IT HAS                   *
                                ;  *     THE EFFECT OF RETURNING TO THE END OF THE MODE                      *
                                ;  *     MACRO EXPANSION.                                                    *
                                ;  *                                                                         *
                                ;  ***************************************************************************
                                ;
                           0000 05EB    .WORD   0                               ; ENTRY MASK
                 00000059'FF   17 05ED  JMP     @CHM_CONT                       ; RETURN TO MODE MACRO IN NEW MODE
                                ;
                                ;  *     RET INSTR WILL BE ISSUED IN EXPANSION OF 'MODE FROM, ....' MACRO
                                ;
                                        .END    SATSSF01
```

```
$$ARGS                  =  00000004
$$T1                    =  00000004
$$T2                    =  00000009
ACMODE_SXV                 00000091  R      02
ASC                        00000075  R      03
ASCEFC                     0000003F  R      02
ASCEFC$_EFN             =  00000004
ASCEFC$_NAME            =  00000008
ASCEFC$_NARGS           =  00000004
ASCEFC$_PERM            =  00000010
ASCEFC$_PROT            =  0000000C
BUF                        000000BB  R      03
CHMRTN                     000005EB  R      06
CHM_CONT                   00000059  R      03
CS1                        00000099  R      02
CS2                        000000CB  R      02
CS3                        000000F8  R      02
CTL$GL_PHD                 ********    X    06
CURRENT_TC                 00000004  R      03
DAC                        00000065  R      03
DACEFC                     00000031  R      02
DACEFC$_EFN             =  00000004
DACEFC$_NARGS           =  00000001
DLC                        0000006D  R      03
DLCEFC                     00000038  R      02
DLCEFC$_NAME            =  00000004
DLCEFC$_NARGS           =  00000001
EMPTY                      00000000  R      04
ERROR                   =  00000002
EXP                        0000012B  R      02
INADR                      0000004D  R      02
INFO                    =  00000003
LIB$SIGNAL                 ********    X    06
MESSAGEL                   0000010B  R      03
MOD_MSG_CODE               00000044  R      03
MOD_MSG_PRINT              000005D5  R      06
MSGL                       000000B3  R      03
NAME_DLC                   00000059  R      02
NAME_DLC0                  00000066  R      02
NAME_DLC15                 0000006E  R      02
NOACCESS                   00000000  R      05
PHD$Q_PRIVMSK           =  00000000
PRINT_FAIL                 0000052E  R      06
PRIVMASK                   00000051  R      03
PRIV_ARGS               =  00000002
PROT                       00000055  R      02
PRT$C_NA                   ********    X    02
PRV$V_PRMCEB            =  0000000A
PRVHND_SXV40            =  00000001
PRVHND_SXV41               00000095  R      02
PRVHND_SXV42            =  000001FF  R      04
PRVPRT                     00000050  R      03
REG                        0000009D  R      03
REGNUM                     000000AF  R      03
REG_CHECK                  000004E6  R      06
REG_SAVE                   000004DC  R      06
REG_SAVE_AREA              00000008  R      03
RETADR                     0000005D  R      03
SATSSF01                   00000000  RG     06
SERV_NAME                  00000113  R      03
SET                        00000089  R      03
SETEXV                     00000046  R      02
SETEXV$_ACMODE          =  0000000C
SETEXV$_ADDRES          =  00000008
SETEXV$_NARGS           =  00000004
SETEXV$_PRVHND          =  00000010
SETEXV$_VECTOR          =  00000004
SEVERE                  =  00000004
SHR$K_SHRDEF            =  00000001
SHR$_TEXT               =  00001130
SS$_ACCVIO              =  0000000C
SS$_ILLEFC              =  000000EC
SS$_IVLOGNAM            =  00000154
SS$_NOPRIV              =  00000024
STEP                    =  0000000E
STP0                       0000003D  R      06
STP1                       00000085  R      06
STP10                      000002D5  R      06
STP11                      0000031C  R      06
STP12                      000003F8  R      06
STP13                      00000437  R      06
STP14                      00000476  R      06
STP2                       000000BE  R      06
STP3                       000000FF  R      06
STP4                       0000013C  R      06
STP5                       00000179  R      06
STP6                       000001BC  R      06
STP7                       000001FD  R      06
STP8                       00000246  R      06
STP9                       0000028E  R      06
STS$V_INHIB_MSG         =  0000001C
SUCCESS                 =  00000001
SYS$ASCEFC                 ********    GX   06
SYS$CMKRNL                 ********    GX   06
SYS$DACEFC                 ********    GX   06
SYS$DLCEFC                 ********    GX   06
SYS$EXIT                   ********    GX   06
SYS$FAO                    ********     X   06
SYS$HIBER                  ********    GX   06
SYS$SETEXV                 ********    GX   06
SYS$SETPRN                 ********    GX   06
SYS$SETPRT                 ********    GX   06
SYS$SETPRV                 ********    GX   06
SYS$WAKE                   ********    GX   06
TEST_MOD_BEGIN             00000019  R      02
TEST_MOD_FAIL              0000002A  R      02
TEST_MOD_NAME              00000000  R      02
TEST_MOD_NAME_D            00000009  R      02
TEST_MOD_SUCC              0000001F  R      02
TMD_ADDR                   0000004C  R      03
TMN_ADDR                   00000048  R      03
TPID                       00000000  R      03
UETP$_SATSMS            =  007480D9
UETP$_TEXT              =  00741133
```

```
VECTOR_SXV                      0000008D R      02
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	00000139 (313.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC LONG
RWDATA	00000117 (279.)	03 (3.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
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
SATSSF01	000005F3 (1523.)	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	00:00:00.80
Command processing	113	00:00:00.67	00:00:02.19
Pass 1	392	00:00:13.04	00:00:26.28
Symbol table sort	0	00:00:01.56	00:00:02.84
Pass 2	144	00:00:02.92	00:00:05.54
Symbol table output	17	00:00:00.11	00:00:00.11
Psect synopsis output	5	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	703	00:00:18.39	00:00:37.79

The working set limit was 1350 pages.
79290 bytes (155 pages) of virtual memory were used to buffer the intermediate code.
There were 60 pages of symbol table space allocated to hold 999 non-local and 8 local symbols.
600 source lines were read in Pass 1, producing 30 object records in Pass 2.
58 pages of virtual memory were used to define 53 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SHRLIB]UETP.MLB;1	12
_255DUA28:[SYS.OBJ]LIB.MLB;1	2
_255DUA28:[SYSLIB]STARLET.MLB;2	36
TOTALS (all libraries)	50

1328 GETS were required to define 50 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:SATSSF01/OBJ=OBJ$:SATSSF01 MSRC$:SATSSF01/UPDATE=(ENH$:SATSSF01)+EXECML$/LIB+SHRLIB$:UETP/LIB
```

SATSUT10
MAP

SATSUT12
MAP

UETINIT00
MAP

SATSUT14
MAP

UETMA7800
MAP

SATSUT01
MAP

SATSUT05
MAP

SATSUT07
MAP

UETDR7800
MAP

UETSSMM00
MAP

SATSUT09
MAP

UETLOCK00
MAP

SATSUT11
MAP

SATSSF01
LIS

SATSUT13
MAP

UETCLIG00
MAP

SATSUT04
MAP

UETINIT01
MAP

SATSUT06
MAP

UETMEMY01
MAP

SATSUT08
MAP

CLSIODB
LIS