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

FILEID**UETSSMM00   H 9

```
UU      UU  EEEEEEEEEE  TTTTTTTTTT    SSSSSSSS    SSSSSSSS  MM      MM  MM      MM    000000      000000
UU      UU  EEEEEEEEEE  TTTTTTTTTT    SSSSSSSS    SSSSSSSS  MM      MM  MM      MM    000000      000000
UU      UU  EE              TT      SS          SS          MMMM  MMMM  MMMM  MMMM  00      00  00      00
UU      UU  EE              TT      SS          SS          MMMM  MMMM  MMMM  MMMM  00      00  00      00
UU      UU  EE              TT      SS          SS          MM  MM  MM  MM  MM  MM  00    0000  00    0000
UU      UU  EE              TT      SS          SS          MM  MM  MM  MM  MM  MM  00    0000  00    0000
UU      UU  EEEEEEEE        TT        SSSSSS      SSSSSS    MM      MM  MM      MM  00  00  00  00  00  00
UU      UU  EEEEEEEE        TT        SSSSSS      SSSSSS    MM      MM  MM      MM  00  00  00  00  00  00
UU      UU  EE              TT              SS          SS  MM      MM  MM      MM  0000    00  0000    00
UU      UU  EE              TT              SS          SS  MM      MM  MM      MM  0000    00  0000    00
UU      UU  EE              TT              SS          SS  MM      MM  MM      MM  00      00  00      00    ....
UU      UU  EE              TT              SS          SS  MM      MM  MM      MM  00      00  00      00    ....
UUUUUUUUUU  EEEEEEEEEE      TT      SSSSSSSS    SSSSSSSS    MM      MM  MM      MM    000000      000000      ....
UUUUUUUUUU  EEEEEEEEEE      TT      SSSSSSSS    SSSSSSSS    MM      MM  MM      MM    000000      000000      ....

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

Table of contents

```
0000        .TITLE UETSSMM00 VAX/VMS UETP SYSTEM SERVICE TEST
0000        .IDENT  'V04-000'
0000        .DEFAULT DISPLACEMENT,WORD
0000        .ENABLE SUPPRESSION
0000  ;
0000  ;****************************************************************************
0000  ;*                                                                          *
0000  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000  ;*  ALL RIGHTS RESERVED.                                                    *
0000  ;*                                                                          *
0000  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000  ;*  TRANSFERRED.                                                            *
0000  ;*                                                                          *
0000  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000  ;*  CORPORATION.                                                            *
0000  ;*                                                                          *
0000  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000  ;*                                                                          *
0000  ;*                                                                          *
0000  ;****************************************************************************
0000  ;
0000
0000  ;++
0000  ; FACILITY:      SATS SYSTEM SERVICE TESTS
0000  ;
0000  ; ABSTRACT:      The UETSSMM00 module tests the execution of the following
0000  ;                VMS system services:
0000  ;
0000  ;                $CRMPSC
0000  ;                $DGBLSC
0000  ;                $MGBLSC
0000  ;                $UPDSEC
0000  ;
0000  ;
0000  ; ENVIRONMENT:   User, Supervisor and Executive mode image.
0000  ;                Needs SETPRV privilege and dynamically acquires other
0000  ;                privileges, as needed.  Privileges used are:
0000  ;
0000  ;                SYSGBL  SHMEM  SYSNAM PSWAPM
0000  ;                PRMGBL  PFNMAP CMEXEC
0000  ;
0000  ;--
0000  ;
0000  ; AUTHOR: Larry D. Jones,       CREATION DATE: August, 1981
0000  ;
0000  ; MODIFIED BY:
0000  ;
0000  ;       V03-004 LDJ0001         Larry D. Jones,         22-Feb-1984
0000  ;               Extended the GSDNAM length limit to 44 from 15 for the
0000  ;               the change put in $CRMPSC.
```

```
0000    58 ;
0000    59 ;        V03-003 KDM0002         Kathleen D. Morse       28-Jun-1982
0000    60 ;                Added $SSDEF and $VADEF.
0000    61 ;
0000    62 ;**
```

```
                   0000          .SBTTL  Declarations
                   0000     ;
                   0000     ; INCLUDE FILES:
                   0000     ;
                   0000     ;        SYS$LIBRARY:LIB.MLB       for general definitions
                   0000     ;        SHRLIB$:UETP.MLB          for UETP definitions
                   0000
                   0000     ;
                   0000     ; MACROS:
                   0000     ;
                   0000          $CHFDEF                                 ; Condition handler frame definitions
                   0000          $PHDDEF                                 ; Process header definitions
                   0000          $PSLDEF                                 ; PSL definitions
                   0000          $PRTDEF                                 ; Protection definitions
                   0000          $PRVDEF                                 ; Privilege bit definitions
                   0000          $PTEDEF                                 ; Page Table Entry definitions
                   0000          $SECDEF                                 ; Section definitions
                   0000          $SHRDEF                                 ; Shared messages
                   0000          $SSDEF                                  ; System status codes
                   0000          $STSDEF                                 ; Status return
                   0000          $UETPDEF                                ; UETP
                   0000          $VADEF                                  ; Virtual address fields
                   0000     ;
                   0000     ; Equated symbols
                   0000     ;
         00741130  0000          UETP$_TEXT = <<116@16>!SHR$_TEXT>
                   0000     ;
         00000002  0000          EF2         = 2
         00000100  0000          TEXT_BUFFER = 256
         00000001  0000          SUCCESS     = 1
         00000002  0000          ERROR       = 2
         00000001  0000          EFN1        = 1
                   0000     ;
                   0000     ; MACROS
                   0000     ;
                   0000
```

```
                                      0000  101           .SBTTL  RO PSECT
                                  00000000  102           .PSECT  RODATA,RD,NOWRT,NOEXE,LONG
                                      0000  103 ;
                                      0000  104
                                      0000  105 TEST_MOD_NAME:
     30 30 4D 4D 53 53 54 45 55 00'   0000  106           .ASCIC  /UETSSMM00/             ; Needed for SATSMS message
                                09    0000
                                      000A  107
                                      000A  108 TEST_MOD_NAME_D:
4D 53 53 54 45 55 00000012'010E0000'  000A  109           .ASCID /UETSSMM00/              ; Module name
                          30 30 4D    0018
                                      001B  110
                                      001B  111 TEST_MOD_BEGIN:
             6E 75 67 65 62 00'       001B  112           .ASCIC  /begun/                 ; Begin message
                                05    001B
                                      0021  113
                                      0021  114 TEST_MOD_SUCC:
   6C 75 66 73 73 65 63 63 75 73 00'  0021  115           .ASCIC  /successful/            ; Success message
                                0A    0021
                                      002C  116
                                      002C  117 TEST_MOD_FAIL:
             64 65 6C 69 61 66 00'    002C  118           .ASCIC  /failed/                ; Failure message
                                06    002C
                                      0033  119
                                      0033  120 CRMPSC:
             43 53 50 4D 52 43 00'    0033  121           .ASCIC  /CRMPSC/                ; System service names
                                06    0033
                                      003A  122
                                      003A  123 DGBLSC:
             43 53 4C 42 47 44 00'    003A  124           .ASCIC  /DGBLSC/
                                06    003A
                                      0041  125
                                      0041  126 MGBLSC:
             43 53 4C 42 47 4D 00'    0041  127           .ASCIC  /MGBLSC/
                                06    0041
                                      0048  128
                                      0048  129 UPDSEC:
             43 45 53 44 50 55 00'    0048  130           .ASCIC  /UPDSEC/
                                06    0048
                                      004F  131
                                      004F  132 RMS_ERR_MSG:                              ; Announces an RMS error
72 65 20 53 4D 52 00000057'010E0000'  004F  133           .ASCID  /RMS error in file !AD/
20 65 6C 69 66 20 6E 69 20 72 6F 72   005D
                          44 41 21    0069
                                      006C  134                                           ; Failure messages
                                      006C  135 UPDSEC_FAILED:
45 53 44 50 55 24 00000074'010E0000'  006C  136           .ASCID  /$UPDSEC failed to modify the section file./
20 6F 74 20 64 65 6C 69 61 66 20 43   007A
73 20 65 68 74 20 79 66 69 64 6F 6D   0086
2E 65 6C 69 66 20 6E 6F 69 74 63 65   0092
                                      009E  137
                                      009E  138 BAD_PFN:
46 50 20 64 61 42 000000A6'010E0000'  009E  139           .ASCID  /Bad PFN returned by the exec. No PFNMAP testing done./
62 20 64 65 6E 72 75 74 65 72 20 4E   00AC
20 2E 63 65 78 65 20 65 68 74 20 79   00B8
65 74 20 50 41 4D 4E 46 50 20 6F 4E   00C4
   2E 65 6E 6F 64 20 67 6E 69 74 73   00D0
```

```
                                       00DB
                                       00DB  CS1:
21 20 74 73 65 54 000000E3'010E0000'   00DB          .ASCID  \Test !AC service name !AC step !UL failed.\
6E 20 65 63 69 76 72 65 73 20 43 41    00E9
70 65 74 73 20 43 41 21 20 65 6D 61    00F5
2E 64 65 6C 69 61 66 20 4C 55 21 20    0101
                                       010D
                                       010D  CS2:
74 63 65 70 78 45 00000115'010E0000'   010D          .ASCID  \Expected !AS = !XL received !AS = !XL\
4C 58 21 20 3D 20 53 41 21 20 64 65    011B
41 21 20 64 65 76 69 65 63 65 72 20    0127
                  4C 58 21 20 3D 20 53 0133
                                       013A
                                       013A  CS3:
74 63 65 70 78 45 00000142'010E0000'   013A          .ASCID  \Expected !AS!UB = !XL received !AS!UB = !XL\
20 3D 20 42 55 21 53 41 21 20 64 65    0148
64 65 76 69 65 63 65 72 20 4C 58 21    0154
58 21 20 3D 20 42 55 21 53 41 21 20    0160
                                    4C 016C
                                       016D
                                       016D  CS4:
72 65 66 66 75 42 00000175'010E0000'   016D          .ASCID  \Buffer offset was !XL\
20 73 61 77 20 74 65 73 66 66 6F 20    017B
                              4C 58 21 0187
                                       018A
                                       018A  CS5:
77 20 65 64 6F 4D 00000192'010E0000'   018A          .ASCID  \Mode was !AS.\
            2E 53 41 21 20 73 61       0198
                                       019F
                                       019F  NSSSF:
75 73 2D 6E 6F 4E 000001A7'010E0000'   019F          .ASCID  \Non-subject system service failure of : !/!_       !AS\
6D 65 74 73 79 73 20 74 63 65 6A 62    01AD
69 61 66 20 65 63 69 76 72 65 73 20    01B9
2F 21 20 3A 20 66 6F 20 65 72 75 6C    01C5
   53 41 21 20 20 20 20 20 20 5F 21    01D1
                                       01DC
                                       01DC  UM:                                           ; CPU modes
      72 65 73 75 000001E4'010E0000'   01DC          .ASCID  \user\
                                       01E8
                                       01E8  SM:
   72 65 70 75 73 000001F0'010E0000'   01E8          .ASCID  \super\
                                       01F5
                                       01F5  EM:
74 75 63 65 78 65 000001FD'010E0000'   01F5          .ASCID  \executive\
                             65 76 69  0203
                                       0206
                                       0206  KM:
6C 61 6E 72 65 6B 0000020E'010E0000'   0206          .ASCID  \kernal\
                                       0214
                                       0214  EXP:
73 75 74 61 74 73 0000021C'010E0000'   0214          .ASCID  \status\
                                       0222
                                       0222  ZEROSTRING:                                   ; Illegal length strings
                  0000022A'010E0000'   0222          .ASCID  //
                                       022A
                                       022A  TOOLONGSTRING:
41 41 41 41 41 41 00000232'010E0000'   022A          .ASCID  /AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA/
```

```
41 41 41 41 41 41 41 41 41 41 41 41  0238
41 41 41 41 41 41 41 41 41 41 41 41  0244
41 41 41 41 41 41 41 41 41 41 41 41  0250
                               41 41  025C
                                      025E
                                      025E   TTNAM:
          54 54 00000266'010E0000'    025E           .ASCID  /TT/
                                      0268
                                      0268   REG:                                     ; Output message variables
74 73 69 67 65 72 00000270'010E0000'  0268           .ASCID  \register R\
                         52 20 72 65  0276
                                      027A
                                      027A   ADR:
73 65 72 64 64 61 00000282'010E0000'  027A           .ASCID  \address\
                                  73  0288
                                      0289
                                      0289   DATA:
      61 74 61 64 00000291'010E0000'  0289           .ASCID  \data\
                                      0295
                                      0295   STATUS:
73 75 74 61 74 73 0000029D'010E0000'  0295           .ASCID  \status\
                                      02A3
                                      02A3   MS:
      65 64 6F 6D 000002AB'010E0000'  02A3           .ASCID  \mode\
                                      02AF
                                      02AF   MSGVEC:                                  ; PUTMSG message vector
                            00000003  02AF           .LONG   3
                            00741130  02B3           .LONG   UETP$_TEXT
                            00000001  02B7           .LONG   1
                           00000122'  02BB           .ADDRESS MESSAGEL
                                      02BF
                                      02BF   AN_ILLEGAL_IDENT:                        ; An illegal IDENT
                   00000000 00000003  02BF           .QUAD   <SEC$K_MATLEQ+1>
```

```
              02C7  ;
              02C7          .SBTTL  R/W PSECT
          00000000          .PSECT  RWDATA,RD,WRT,NOEXE,LONG
              0000  ;
              0000
              0000  TPID:
00000000      0000          .LONG   0                       ; PID for this process
              0004
              0004  TTCHAN:
    0000      0004          .WORD   0
              0006
              0006  CURRENT_TC:
00000000      0006          .LONG   0                       ; Ptr to current test case
              000A
              000A  MODE:
00000000      000A          .LONG   0                       ; Current mode string pointer
              000E
              000E  REGNUM:                                 ; Register number
00000000      000E          .LONG   0
              0012
              0012  STATUS_CODE:                            ; Failure status code
00000000      0012          .LONG   0
              0016
              0016  ARG_COUNT:                              ; Storage for argument counts
00000000      0016          .LONG   0
              001A
              001A  MSGL:
00000100      001A          .LONG   TEXT_BUFFER             ; Buffer desc.
00000022'     001E          .ADDRESS BUF
              0022
              0022  BUF:
00000122      0022          .BLKB   TEXT_BUFFER
              0122
              0122  MESSAGEL:
00000000      0122          .LONG   0                       ; Message length
00000022'     0126          .ADDRESS BUF
              012A
              012A  MSG1L:
00000100      012A          .LONG   TEXT_BUFFER             ; Buffer #1 desc.
00000132'     012E          .ADDRESS BUF1
              0132
              0132  BUF1:
00000232      0132          .BLKB   TEXT_BUFFER
              0232
              0232  MESSAGE1L:
00000000      0232          .LONG   0                       ; Message length
00000132'     0236          .ADDRESS BUF1
              023A
              023A  SERV_NAME:
00000000      023A          .LONG   0                       ; Service name pointer
              023E
              023E  REG_SAVE_AREA:
0000027A      023E          .BLKL   15                      ; Register save area
              027A
              027A  MOD_MSG_CODE:
007480D9      027A          .LONG   UETP$_SATSMS            ; Test module message code for putmsg
              027E
```

```
                              027E  TMN_ADDR:
                   00000000'  027E          .ADDRESS TEST_MOD_NAME
                              0282
                              0282  TMD_ADDR:
                   0000001B'  0282          .ADDRESS TEST_MOD_BEGIN
                              0286
                              0286  START_ADDRESS:                          ; Storage for buffer start address
                   00000000   0286          .LONG   0
                              028A
                              028A  WORK:
                   00000000   028A          .LONG   0                       ; Scratch long word
                              028E
                              028E  MSGVEC1:                                ; PUTMSG message vector
                   00000003   028E          .LONG   3
                   00741130   0292          .LONG   UETP$_TEXT
                   00000001   0296          .LONG   1
                   00000000   029A          .LONG   0
                              029E
                              029E  EXIT_DESC:                              ; Exit handler descriptor
                   00000000   029E          .LONG   0
                   00001FBF'  02A2          .ADDRESS EXIT_HANDLER
                   00000001   02A6          .LONG   1
                   00000012'  02AA          .ADDRESS STATUS_CODE
                              02AE
                              02AE  UPD_IOSB:                               ; IOSB for the $UPDSEC
          00000000 00000000   02AE          .QUAD   0
                              02B6
                              02B6  CRMPSCG:                                ; Allocate space for _G parameters
                              02B6          $CRMPSC
                              02EA
                              02EA  MGBLSCG:
                              02EA          $MGBLSC
                              030A
                              030A  DGBLSCG:
                              030A          $DGBLSC
                              031A
                              031A  UPDSECG:
                              031A          $UPDSEC
                              033E
                              033E  AN_IDENT:                               ; A section identifier
          00000000 00000000   033E          .QUAD 0
                              0346
                              0346  GSDNAM:                                 ; Global section name
45 53 4D 4D 53 53 0000034E'010E0000'  0346  .ASCID  /SSMMSEC/
                         43   0354
                              0355
                              0355  PRIV_MASK:                              ; Privilege mask
          00000000 00000000   0355          .QUAD   0
                              035D
                              035D  VBN:                                    ; Storage for Virtual Block Number
                   00000000   035D          .LONG   0
                              0361
                              0361  OUTADDRESS:                             ; Address descriptors for sections
                   00000600'  0361          .ADDRESS READ_BUF
                   00000600'  0365          .ADDRESS READ_BUF
                              0369
                              0369  INADDRESS:
```

```
                00000600'  0369          .ADDRESS READ_BUF
                00000600'  036D          .ADDRESS READ_BUF
                           0371
                           0371  FAO_BUF:                                   ; FAO output string descriptor
                0000 0100  0371          .WORD    TEXT_BUFFER,0
                00000381'  0375          .ADDRESS BUFFER
                           0379
                           0379  BUFFER_PTR:                                ; Fake .ASCID buffer for misc. strings
                0000 0100  0379          .WORD    TEXT_BUFFER,0             ; A word for length, a word for desc.
                00000381'  037D          .ADDRESS BUFFER
                           0381
                           0381  BUFFER:                                    ; FAO output and other misc. buffer
                00000481   0381          .BLKB    TEXT_BUFFER
                           0481
                           0481  ; The following two user defined symbols must remain contiguous.
                           0481
                           0481  LOGICAL_NAME:
                00000009   0481          .LONG    9                         ; Logical name descriptors
                00000489'  0485          .ADDRESS .+4
  45 4D 41 4E 24 4C 42 47  0489          .ASCII   /GBL$NAME/
                           0491  NAME_INC1:
                       42  0491          .BYTE    ^A/B/
                           0492
                           0492  ; The following two user defined symbols must remain contiguous.
                           0492
                           0492  EQUIV_NAME:
                00000005   0492          .LONG    5
                0000049A'  0496          .ADDRESS .+4
              45 4D 41 4E  049A          .ASCII   /NAME/
                           049E  NAME_INC2:
                       41  049E          .BYTE    ^A/A/
                           049F
                           049F  GSD_LOGNAM:
4C 45 4D 41 4E 000004A7'010E0000'  049F  .ASCID   /NAMEL/
                           04AC
```

```
                   04AC          .SBTTL RMS PSECT
               00000000          .PSECT RMS,PAGE
                   0000
                   0000  WRITE_BUF:                          ; 3 Page buffer on page boundary
FFFFFFFF           0000          A = -1
                   0000          .REPT <512*3>/4
                   0000          .LONG A
                   0000          A = A-1
FFFFFFFF           0000          .ENDR
                   0600
                   0600          .ALIGN  PAGE
                   0600
                   0600  READ_BUF:
00000C00           0600          .BLKB   512*3
00000E00           0C00          .BLKB   512                 ; End buffer bumper
                   0E00
                   0E00          .ALIGN PAGE
                   0E00
                   0E00  READ_BUF1:
00001400           0E00          .BLKB   512*3
00001600           1400          .BLKB   512                 ; End buffer bumper
                   1600
                   1600          .ALIGN LONG
                   1600
                   1600  SECFAB:                             ; Section file FAB
                   1600          $FAB FNM=<UETPSECT.DAT>,-
                   1600               FAC=<GET>,-
                   1600               RAT=CR,-
                   1600               FOP=UFO
                   1650
                   1650  FILEFAB:                            ; Section file creation FAB
                   1650          $FAB FNM = <UETPSECT.DAT>,-
                   1650               FAC = <PUT,BIO>
                   16A0
                   16A0  FILERAB:                            ; Section file RAB
                   16A0          $RAB FAB = FILEFAB,-
                   16A0               UBF = READ_BUF,-
                   16A0               USZ = 512,-
                   16A0               RBF = WRITE_BUF,-
                   16A0               RSZ = 512
                   16E4
```

```
    16E4        .SBTTL  UETSSMM00
00000000        .PSECT  UETSSMM00,RD,WRT,EXE,LONG
    0000 ;++
    0000 ; FUNCTIONAL DESCRIPTION:
    0000 ;
    0000 ;       After performing some initial housekeeping, such as
    0000 ; printing the module begin message and acquiring needed privileges,
    0000 ; the system services are tested in each of their normal and failure
    0000 ; conditions. Detected failures are identified and  an error message
    0000 ; is printed on the terminal.  Upon completion of the test a success
    0000 ; or fail message is printed on the terminal.
    0000 ;
    0000 ; CALLING SEQUENCE:
    0000 ;
    0000 ;       $ RUN UETSSMM00  ...  (DCL COMMAND)
    0000 ;
    0000 ; INPUT PARAMETERS:
    0000 ;
    0000 ;       none
    0000 ;
    0000 ; IMPLICIT INPUTS:
    0000 ;
    0000 ;       none
    0000 ;
    0000 ; OUTPUT PARAMETERS:
    0000 ;
    0000 ;       none
    0000 ;
    0000 ; IMPLICIT OUTPUTS:
    0000 ;
    0000 ;       Messages to SYS$OUTPUT are the only output from UETSSMM00.
    0000 ;       They are of the form:
    0000 ;
    0000 ;               %UETP-S-SATSMS, TEST MODULE UETSSMM00 BEGUN ... (BEGIN MSG)
    0000 ;               %UETP-S-SATSMS, TEST MODULE UETSSMM00 SUCCESSFUL ... (END MSG)
    0000 ;               %UETP-E-SATSMS, TEST MODULE UETSSMM00 FAILED ... (END MSG)
    0000 ;               %UETP-I-TEXT, ... (VARIABLE INFORMATION ABOUT A TEST MODULE FAILURE)
    0000 ;
    0000 ; COMPLETION CODES:
    0000 ;
    0000 ;       The UETSSMM00 routine terminates with a $EXIT to the
    0000 ;       operating system with a status code defined by UETP$_SATSMS.
    0000 ;
    0000 ; SIDE EFFECTS:
    0000 ;
    0000 ;       none
    0000 ;
    0000 ;--
    0000
    0000         TEST_START UETSSMM00            ; let the test begin
```

```
                              0000
                      0000    0000                 .ENTRY UETSSMM00,0
             0006'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
             000A'CF    7F    001A                 PUSHAQ  W^TEST_MOD_NAME_D
      00000000'GF  01   FB    001E                 CALLS   #1,G^SYS$SETPRN
                1F81    30    0025                 BSBW    W^MOD_MSG_PRINT
        0282'CF 0021'CF DE    0028                 MOVAL   W^TEST_MOD_SUCC,W^TMD_ADDR
 027A'CF   03    00     01 F0 002F                 INSV    #SUCCESS,#0,#3,W^MOD_MSG_CODE
                  00    DD    0036                 PUSHL   #0
        1D2C'CF   01    FB    0038                 CALLS #1,W^REG_SAVE
                              003D        STP0:
                              003D
                              003D              .SBTTL  CRMPSC TESTS
                              003D        ;+
                              003D        ;
                              003D        ; $CRMPSC tests
                              003D        ;
                              003D        ; Test simplest case
                              003D        ;
                              003D        ;-
                              003D
                    00000000  003D              STEP=0
                              003D              NEXT_TEST
                              003D
                              003D        STP1:
        0006'CF   01    D0    003D                 MOVL    #1,W^CURRENT_TC
                  00    DD    0042                 PUSHL   #0
        1D2C'CF   01    FB    0044                 CALLS   #1,W^REG_SAVE
                              0049
                              0049              $DCLEXH_S DESBLK = EXIT_DESC        ; Declare an exit handler
 000A'CF   01DC'CF      DE    0054              MOVAL   UM,MODE                     ; Set the mode
 023A'CF   0033'CF      DE    005B              MOVAL   CRMPSC,SERV_NAME            ; Set the service name
                              0062              $CRMPSC_S                           ; No inadr on non global section
                              007C              FAIL_CHECK SS$_ACCVIO
                  0C    DD    007C                 PUSHL   #SS$_ACCVIO
        1D36'CF   01    FB    007E                 CALLS   #1,W^REG_CHECK
                              0083        ;+
                              0083        ;
                              0083        ; Test no inadr on non-permanent global section
                              0083        ;
                              0083        ;-
                              0083
                              0083              NEXT_TEST
                              0083
                              0083        STP2:
        0006'CF   02    D0    0083                 MOVL    #2,W^CURRENT_TC
                  00    DD    0088                 PUSHL   #0
        1D2C'CF   01    FB    008A                 CALLS   #1,W^REG_SAVE
                              008F
                              008F              $CRMPSC_S FLAGS = #SEC$M_GBL
                              00AB              FAIL_CHECK SS$_ACCVIO
                  0C    DD    00AB                 PUSHL   #SS$_ACCVIO
        1D36'CF   01    FB    00AD                 CALLS   #1,W^REG_CHECK
```

```
                              00B2  ;+
                              00B2  ;
                              00B2  ; Test No read access on inadr
                              00B2  ;
                              00B2  ;-
                              00B2
                              00B2          NEXT_TEST
                              00B2
                              00B2  STP3:
          0006'CF   03   D0   00B2                  MOVL    #3,W^CURRENT_TC
                    00   DD   00B7                  PUSHL   #0
          1D2C'CF   01   FB   00B9                  CALLS   #1,W^REG_SAVE
                              00BE
                              00BE          $CREATE FAB = FILEFAB,-
                              00BE                  ERR = RMS_ERROR                 ; Create the data file
                              00CD          $CONNECT RAB = FILERAB,-
                              00CD                  ERR = RMS_ERROR                 ; Connect to the RAB
                              00DC          $WRITE RAB = FILERAB,-
                              00DC                  ERR = RMS_ERROR                 ; Write 3 pages of long word
                              00EB                                                  ; unique data to the file
                              00EB          $CLOSE FAB = FILEFAB,-
                              00EB                  ERR = RMS_ERROR                 ; Close the file
                              00FA          $OPEN   FAB = SECFAB,-
                              00FA                  ERR = RMS_ERROR                 ; Open the file to get a channel
                              0109          $SETPRT_S INADR = INADDRESS,-           ; Set buffer protection
                              0109                  PROT   = #PRT$C_NA
               07 50     E8   011C          BLBS    R0,10$                          ; Skip error report if OK
                  50     DD   011F          PUSHL   R0                              ; Save error code
          1F15'CF   01   FB   0121          CALLS   #1,NONSUB_SSE                   ; Print the failure
                              0126  10$:
                              0126          $CRMPSC_S INADR = READ_BUF,-
                              0126                    CHAN  = SECFAB+FAB$L_STV      ; Try for accvio
                              0144          FAIL_CHECK SS$_ACCVIO
                    0C   DD   0144                  PUSHL   #SS$_ACCVIO
          1D36'CF   01   FB   0146                  CALLS   #1,W^REG_CHECK
                              014B          $SETPRT_S INADR = INADDRESS,-           ; Reset buffer protection
                              014B                  PROT   = #PRT$C_UW
               07 50     E8   015E          BLBS    R0,20$                          ; Skip error report if OK
                  50     DD   0161          PUSHL   R0                              ; Save error code
          1F15'CF   01   FB   0163          CALLS   #1,NONSUB_SSE                   ; Print the failure
                              0168  20$:
                              0168  ;+
                              0168  ;
                              0168  ; Test inadr of system space address
                              0168  ;
                              0168  ;-
                              0168
                              0168          NEXT_TEST
                              0168
                              0168  STP4:
          0006'CF   04   D0   0168                  MOVL    #4,W^CURRENT_TC
                    00   DD   016D                  PUSHL   #0
          1D2C'CF   01   FB   016F                  CALLS   #1,W^REG_SAVE
                              0174
 0369'CF  80000000'EF    DE   0174          MOVAL   ^X80000000,INADDRESS            ; Set a system space address
 036D'CF  80000000'EF    DE   017D          MOVAL   ^X80000000,INADDRESS+4
                              0186          $CRMPSC_S INADR = INADDRESS,-
```

```
                              0186              CHAN  = SECFAB+FAB$L_STV
                              01A4        FAIL_CHECK SS$_NOPRIV
                    24   DD   01A4              PUSHL   #SS$_NOPRIV
          1D36'CF   01   FB   01A6              CALLS   #1,W^REG_CHECK
                              01AB   ;+
                              01AB   ;
                              01AB   ; Test virtual address space full
                              01AB   ;
                              01AB   ;-
                              01AB
                              01AB        NEXT_TEST
                              01AB
                              01AB   STP5:
          0006'CF   05   D0   01AB              MOVL    #5,W^CURRENT_TC
                    00   DD   01B0              PUSHL   #0
          1D2C'CF   01   FB   01B2              CALLS   #1,W^REG_SAVE
                              01B7
 0369'CF   70000000'EF   DE   01B7        MOVAL   ^X70000000,INADDRESS           ; Set a P1 space address
 036D'CF   70000000'EF   DE   01C0        MOVAL   ^X70000000,INADDRESS+4
                              01C9        $CRMPSC_S INADR = INADDRESS,-
                              01C9              CHAN  = SECFAB+FAB$L_STV
                              01E7        FAIL_CHECK SS$_VASFULL
       00000244 8F       DD   01E7              PUSHL   #SS$_VASFULL
          1D36'CF   01   FB   01ED              CALLS   #1,W^REG_CHECK
   0369'CF   0600'CF     DE   01F2        MOVAL   READ_BUF,INADDRESS             ; Reset to P0 space address
   036D'CF   07FF'CF     DE   01F9        MOVAL   READ_BUF+^X1FF,INADDRESS+4
                              0200   ;+
                              0200   ;
                              0200   ; Test illegal address in retadr
                              0200   ;
                              0200   ;-
                              0200
                              0200        NEXT_TEST
                              0200
                              0200   STP6:
          0006'CF   06   D0   0200              MOVL    #6,W^CURRENT_TC
                    00   DD   0205              PUSHL   #0
          1D2C'CF   01   FB   0207              CALLS   #1,W^REG_SAVE
                              020C
                              020C        $CRMPSC_S INADR = INADDRESS,-
                              020C              CHAN  = SECFAB+FAB$L_STV,-
                              020C              RETADR= 1
                              022E        FAIL_CHECK SS$_ACCVIO
                    0C   DD   022E              PUSHL   #SS$_ACCVIO
          1D36'CF   01   FB   0230              CALLS   #1,W^REG_CHECK
                              0235   ;+
                              0235   ;
                              0235   ; Test no write access for retadr
                              0235   ;
                              0235   ;-
                              0235
                              0235        NEXT_TEST
                              0235
                              0235   STP7:
          0006'CF   07   D0   0235              MOVL    #7,W^CURRENT_TC
                    00   DD   023A              PUSHL   #0
          1D2C'CF   01   FB   023C              CALLS   #1,W^REG_SAVE
```

```
                          0241
                          0241          $SETPRT_S INADR = OUTADDRESS,-              ; Set retadr protection
                          0241                    PROT  = #PRT$C_NA
           07 50     E8   0254          BLBS      R0,10$                            ; Skip error report if OK
              50     DD   0257          PUSHL     R0                                ; Save error code
   1F15'CF    01     FB   0259          CALLS     #1,NONSUB_SSE                     ; Print the failure
                          025E  10$:
                          025E          $CRMPSC_S INADR = INADDRESS,-
                          025E                    CHAN  = SECFAB+FAB$L_STV,-
                          025E                    RETADR= READ_BUF
                          027E          FAIL_CHECK SS$_ACCVIO
              0C     DD   027E                    PUSHL   #SS$_ACCVIO
   1D36'CF    01     FB   0280                    CALLS   #1,W^REG_CHECK
                          0285          $SETPRT_S INADR = OUTADDRESS,-              ; Reset retadr protection
                          0285                    PROT  = #PRT$C_UW
           07 50     E8   0298          BLBS      R0,20$                            ; Skip error report if OK
              50     DD   029B          PUSHL     R0                                ; Save error code
   1F15'CF    01     FB   029D          CALLS     #1,NONSUB_SSE                     ; Print the failure
                          02A2  20$:
                          02A2  ;+
                          02A2  ;
                          02A2  ; Test no channel and no PFNMAP flag bit set
                          02A2  ;
                          02A2  ;-
                          02A2
                          02A2          NEXT_TEST
                          02A2
                          02A2  STP8:
   0006'CF    08     D0   02A2                    MOVL    #8,W^CURRENT_TC
              00     DD   02A7                    PUSHL   #0
   1D2C'CF    01     FB   02A9                    CALLS   #1,W^REG_SAVE
                          02AE
                          02AE          $CRMPSC_S INADR = INADDRESS
                          02CA          FAIL_CHECK SS$_IVSECFLG
  0000016C 8F        DD   02CA                    PUSHL   #SS$_IVSECFLG
   1D36'CF    01     FB   02D0                    CALLS   #1,W^REG_CHECK
                          02D5  ;+
                          02D5  ;
                          02D5  ; Test chan and PFNMAP
                          02D5  ;
                          02D5  ;-
                          02D5
                          02D5          NEXT_TEST
                          02D5
                          02D5  STP9:
   0006'CF    09     D0   02D5                    MOVL    #9,W^CURRENT_TC
              00     DD   02DA                    PUSHL   #0
   1D2C'CF    01     FB   02DC                    CALLS   #1,W^REG_SAVE
                          02E1
                          02E1          $CRMPSC_S INADR = INADDRESS,-
                          02E1                    FLAGS = #SEC$M_PFNMAP,-
                          02E1                    CHAN  = SECFAB+FAB$L_STV
                          0305          FAIL_CHECK SS$_IVSECFLG
  0000016C 8F        DD   0305                    PUSHL   #SS$_IVSECFLG
   1D36'CF    01     FB   030B                    CALLS   #1,W^REG_CHECK
                          0310  ;+
                          0310  ;
```

```
                                  0310   ; Test illegal channel
                                  0310   ;
                                  0310   ;-
                                  0310
                                  0310           NEXT_TEST
                                  0310
                                  0310   STP10:
              0006'CF    0A   D0  0310                   MOVL    #10,W^CURRENT_TC
                         00   DD  0315                   PUSHL   #0
              1D2C'CF    01   FB  0317                   CALLS   #1,W^REG_SAVE
                                  031C
                                  031C           $CRMPSC_S INADR = INADDRESS,-
                                  031C                   CHAN  = #1
                                  0338           FAIL_CHECK SS$_IVCHAN
             0000013C 8F      DD  0338                   PUSHL   #SS$_IVCHAN
              1D36'CF    01   FB  033E                   CALLS   #1,W^REG_CHECK
                                  0343   ;+
                                  0343   ;
                                  0343   ; Test non-file channel device
                                  0343   ;
                                  0343   ;-
                                  0343
                                  0343           NEXT_TEST
                                  0343
                                  0343   STP11:
              0006'CF    0B   D0  0343                   MOVL    #11,W^CURRENT_TC
                         00   DD  0348                   PUSHL   #0
              1D2C'CF    01   FB  034A                   CALLS   #1,W^REG_SAVE
                                  034F
                                  034F           $CREMBX_S CHAN  = TTCHAN                          ; Make a no RND, FOD, DIR channel
                      07 50   E8  0362           BLBS    R0,10$                                    ; Skip error report if OK
                         50   DD  0365           PUSHL   R0                                        ; Save error code
              1F15'CF    01   FB  0367           CALLS   #1,NONSUB_SSE                             ; Print the failure
                                  036C   10$:
                                  036C           $CRMPSC_S INADR = INADDRESS,-                     ; Try it
                                  036C                   CHAN  = TTCHAN
                                  038A           FAIL_CHECK SS$_NOTFILEDEV
             000001CC 8F      DD  038A                   PUSHL   #SS$_NOTFILEDEV
              1D36'CF    01   FB  0390                   CALLS   #1,W^REG_CHECK
                                  0395           $DASSGN_S CHAN  = TTCHAN                          ; Get rid of the mail box
                      07 50   E8  03A1           BLBS    R0,20$                                    ; Skip error report if OK
                         50   DD  03A4           PUSHL   R0                                        ; Save error code
              1F15'CF    01   FB  03A6           CALLS   #1,NONSUB_SSE                             ; Print the failure
                                  03AB   20$:
 0204 8F    00  0600'CF  00   2C  03AB           MOVC5   #0,READ_BUF,#0,#512+4,READ_BUF            ; Clear the buffer and it's bumper
                0600'CF           03B4
                                  03B7   ;+
                                  03B7   ;
                                  03B7   ; Create the simplest section. One page, user, temporary, P0, private, local memory,
                                  03B7   ; read only section.
                                  03B7   ;
                                  03B7   ;-
                                  03B7
                                  03B7           NEXT_TEST
                                  03B7
                                  03B7   STP12:
              0006'CF    0C   D0  03B7                   MOVL    #12,W^CURRENT_TC
```

```
                  00   DD 03BC                     PUSHL   #0
         1D2C'CF  01   FB 03BE                     CALLS   #1,W^REG_SAVE
                          03C3
                          03C3        $CRMPSC_S CHAN = SECFAB+FAB$L_STV,-      ; Create the temp P0 private read
                          03C3                  INADR = INADDRESS              ; local memory section
                          03E1        FAIL_CHECK SS$_CREATED
       00000619 8F        DD 03E1                  PUSHL   #SS$_CREATED
         1D36'CF  01   FB 03E7                     CALLS   #1,W^REG_CHECK
            0369'CF    DF 03EC        PUSHAL  INADDRESS                        ; Push address quad word
       FFFFFFFF 8F     DD 03F0        PUSHL   #-1                              ; Push start data value
         1E89'CF  02   FB 03F6        CALLS   #2,DATA_CHECK                    ; Check section data
                          03FB        $DELTVA_S INADR = INADDRESS              ; Clean up
               07 50   E8 040A        BLBS    R0,10$                           ; Skip error report if OK
                  50   DD 040D        PUSHL   R0                               ; Save error code
         1F15'CF  01   FB 040F        CALLS   #1,NONSUB_SSE                    ; Print the failure
                          0414   10$:
                          0414   ;+
                          0414   ;
                          0414   ; Create the second simplest section. One page, user, temporary, P1, private, local
                          0414   ; read only section. Introduce the use of RETADR and FLAGS.
                          0414   ;
                          0414   ;-
                          0414
                          0414        NEXT_TEST
                          0414
                          0414   STP13:
         0006'CF  0D   D0 0414                     MOVL    #13,W^CURRENT_TC
                  00   DD 0419                     PUSHL   #0
         1D2C'CF  01   FB 041B                     CALLS   #1,W^REG_SAVE
                          0420
 0369'CF   D8F0 CE     DE 0420        MOVAL   -10000(SP),INADDRESS             ; Put a P1 address in the inadr
 036D'CF   D6F1 CE     DE 0427        MOVAL   -10000-^X1FF(SP),INADDRESS+4
                          042E        $CRMPSC_S CHAN = SECFAB+FAB$L_STV,-
                          042E                  FLAGS = #SEC$M_EXPREG,-
                          042E                  INADR = INADDRESS,-
                          042E                  RETADR = OUTADDRESS
                          0454        FAIL_CHECK SS$_CREATED
       00000619 8F     DD 0454                     PUSHL   #SS$_CREATED
         1D36'CF  01   FB 045A                     CALLS   #1,W^REG_CHECK
            0361'CF    DF 045F        PUSHAL  OUTADDRESS                       ; Push address quad word
       FFFFFFFF 8F     DD 0463        PUSHL   #-1                              ; Push start data value
         1E89'CF  02   FB 0469        CALLS   #2,DATA_CHECK                    ; Check section data
                          046E        $DELTVA_S INADR = OUTADDRESS             ; Clean up
               07 50   E8 047D        BLBS    R0,10$                           ; Skip error report if OK
                  50   DD 0480        PUSHL   R0                               ; Save error code
         1F15'CF  01   FB 0482        CALLS   #1,NONSUB_SSE                    ; Print the failure
                          0487   10$:
                          0487   ;+
                          0487   ;
                          0487   ; Test read/write section creation on a read only file.
                          0487   ;
                          0487   ;-
                          0487
                          0487        NEXT_TEST
                          0487
                          0487   STP14:
         0006'CF  0E   D0 0487                     MOVL    #14,W^CURRENT_TC
```

```
              00   DD  048C                PUSHL   #0
     1D2C'CF  01   FB  048E                CALLS   #1,W^REG_SAVE
                       0493
                       0493        $CRMPSC_S CHAN = SECFAB+FAB$L_STV,-
                       0493                  FLAGS = #SEC$M_WRT,-
                       0493                  INADR = INADDRESS
                       04B3        FAIL_CHECK SS$_NOWRT
   000003FC 8F     DD  04B3                PUSHL   #SS$_NOWRT
     1D36'CF  01   FB  04B9                CALLS   #1,W^REG_CHECK
                       04BE        $DELTVA_S INADR = OUTADDRESS             ; Clean up
           07 50   E8  04CD        BLBS    R0,10$                           ; Skip error report if OK
              50   DD  04D0        PUSHL   R0                               ; Save error code
     1F15'CF  01   FB  04D2        CALLS   #1,NONSUB_SSE                    ; Print the failure
                       04D7   10$:
                       04D7        $DASSGN_S CHAN = SECFAB+FAB$L_STV        ; Deassign the read-only channel
           07 50   E8  04E3        BLBS    R0,20$                           ; Skip error report if OK
              50   DD  04E6        PUSHL   R0                               ; Save error code
     1F15'CF  01   FB  04E8        CALLS   #1,NONSUB_SSE                    ; Print the failure
                       04ED   20$:
     1616'CF  01   88  04ED        BISB2   #FAB$M_PUT,SECFAB+FAB$B_FAC      ; Make the file read/write
                       04F2        $OPEN   FAB = SECFAB,-
                       04F2                ERR = RMS_ERROR                  ; Open the read/write channel
                       0501   ;+
                       0501   ;
                       0501   ; Create a one page user temporary P1 private local memory read/write section.
                       0501   ;
                       0501   ;-
                       0501
                       0501        NEXT_TEST
                       0501
                       0501   STP15:
     0006'CF  0F   D0  0501                MOVL    #15,W^CURRENT_TC
              00   DD  0506                PUSHL   #0
     1D2C'CF  01   FB  0508                CALLS   #1,W^REG_SAVE
                       050D
                       050D        $CRMPSC_S CHAN = SECFAB+FAB$L_STV,-
                       050D                  FLAGS = #SEC$M_EXPREG!SEC$M_WRT,-
                       050D                  INADR = INADDRESS,-
                       050D                 RETADR = OUTADDRESS
                       0533        FAIL_CHECK SS$_CREATED
   00000619 8F     DD  0533                PUSHL   #SS$_CREATED
     1D36'CF  01   FB  0539                CALLS   #1,W^REG_CHECK
        0361'CF    DF  053E        PUSHAL  OUTADDRESS                       ; Push address quad word
   FFFFFFFF 8F     DD  0542        PUSHL   #-1                              ; Push start data value
     1E89'CF  02   FB  0548        CALLS   #2,DATA_CHECK                    ; Check section data
                       054D        $DELTVA_S INADR = OUTADDRESS             ; Clean up
           07 50   E8  055C        BLBS    R0,10$                           ; Skip error report if OK
              50   DD  055F        PUSHL   R0                               ; Save error code
     1F15'CF  01   FB  0561        CALLS   #1,NONSUB_SSE                    ; Print the failure
                       0566   10$:
                       0566   ;+
                       0566   ;
                       0566   ; Create a one page user temporary P0 private local memory read/write section.
                       0566   ;
                       0566   ;-
                       0566
                       0566        NEXT_TEST
```

```
                                    0566
                                    0566      STP16:
          0006'CF    10     D0      0566                         MOVL    #16,W^CURRENT_TC
                     00     DD      056B                         PUSHL   #0
          1D2C'CF    01     FB      056D                         CALLS   #1,W^REG_SAVE
                                    0572
 0369'CF    0600'CF         DE      0572            MOVAL   READ_BUF,INADDRESS               ; Reset to a P0 address
 036D'CF    07FF'CF         DE      0579            MOVAL   READ_BUF+^X1FF,INADDRESS+4
                                    0580            $CRMPSC_S CHAN = SECFAB+FAB$L_STV,-
                                    0580                      FLAGS = #SEC$M_EXPREG!SEC$M_WRT,-
                                    0580                      INADR = INADDRESS,-
                                    0580                      RETADR = OUTADDRESS
                                    05A6            FAIL_CHECK SS$_CREATED
       00000619 8F          DD      05A6                         PUSHL   #SS$_CREATED
        1D36'CF    01       FB      05AC                         CALLS   #1,W^REG_CHECK
            0361'CF         DF      05B1            PUSHAL  OUTADDRESS                       ; Push address quad word
       FFFFFFFF 8F          DD      05B5            PUSHL   #-1                              ; Push start data value
        1E89'CF    02       FB      05BB            CALLS   #2,DATA_CHECK                    ; Check section data
                                    05C0            $DELTVA_S INADR = OUTADDRESS             ; Clean up
             07 50          E8      05CF            BLBS    R0,10$                           ; Skip error report if OK
                50          DD      05D2            PUSHL   R0                               ; Save error code
        1F15'CF    01       FB      05D4            CALLS   #1,NONSUB_SSE                    ; Print the failure
                                    05D9      10$:
                                    05D9            $DASSGN_S CHAN  = SECFAB+FAB$L_STV       ; Deassign the channel to the old fi
             07 50          E8      05E5            BLBS    R0,20$                           ; Skip error report if OK
                50          DD      05E8            PUSHL   R0                               ; Save error code
        1F15'CF    01       FB      05EA            CALLS   #1,NONSUB_SSE                    ; Print the failure
                                    05EF      20$:
                                    05EF      ;+
                                    05EF      ;
                                    05EF      ; Create a three page user temporary P0 private local memory read/write section.
                                    05EF      ; INADR is specified in reverse order.
                                    05EF      ;
                                    05EF      ;-
                                    05EF
                                    05EF            NEXT_TEST
                                    05EF
                                    05EF      STP17:
          0006'CF    11     D0      05EF                         MOVL    #17,W^CURRENT_TC
                     00     DD      05F4                         PUSHL   #0
          1D2C'CF    01     FB      05F6                         CALLS   #1,W^REG_SAVE
                                    05FB
                                    05FB            $ERASE FAB = FILEFAB,-
                                    05FB                   ERR = RMS_ERROR                   ; Delete old one page file
 16C2'CF     0600 8F        B0      060A            MOVW    #<3*512>,FILERAB+RAB$W_RSZ       ; Set 3 page size
                                    0611            $CREATE FAB = FILEFAB,-
                                    0611                    ERR = RMS_ERROR                  ; Create a 3 page file
                                    0620            $CONNECT RAB = FILERAB,-
                                    0620                     ERR = RMS_ERROR                 ; Connect the RAB
                                    062F            $WRITE RAB = FILERAB,-
                                    062F                   ERR = RMS_ERROR                   ; Write 3 pages of incrementing data
                                    063E            $CLOSE FAB = FILEFAB,-
                                    063E                   ERR = RMS_ERROR                   ; Close the 3 page file
                                    064D            $OPEN FAB = SECFAB,-
                                    064D                  ERR = RMS_ERROR
 036D'CF   00000400 8F      C1      065C            ADDL3   #^X400,INADDRESS+4,INADDRESS     ; expand the input buffer size
                0369'CF             0665
```

```
        036D'CF    0600'CF   DE  0668        MOVAL   READ_BUF,INADDRESS+4
                                 066F        $CRMPSC_S CHAN = SECFAB+FAB$L_STV,-
                                 066F                FLAGS = #SEC$M_WRT,-
                                 066F                INADR = INADDRESS,-
                                 066F                RETADR = OUTADDRESS
                                 0691        FAIL_CHECK SS$_CREATED
               00000619 8F   DD  0691                PUSHL   #SS$_CREATED
               1D36'CF   01  FB  0697                CALLS   #1,W^REG_CHECK
                  0361'CF    DF  069C        PUSHAL  OUTADDRESS                  ; Push address quad word
               FFFFFFFF 8F   DD  06A0        PUSHL   #-1                         ; Push start data value
               1E89'CF   02  FB  06A6        CALLS   #2,DATA_CHECK               ; Check section data
                                 06AB        $DELTVA_S INADR = OUTADDRESS        ; Clean up
                      07 50  E8  06BA        BLBS    R0,10$                      ; Skip error report if OK
                         50  DD  06BD        PUSHL   R0                          ; Save error code
               1F15'CF   01  FB  06BF        CALLS   #1,NONSUB_SSE               ; Print the failure
                                 06C4   10$:
                                 06C4
                                 06C4   ;+
                                 06C4   ;
                                 06C4   ; Create a three page user temporary P1 private local memory read/write section.
                                 06C4   ;
                                 06C4   ;-
                                 06C4
                                 06C4        NEXT_TEST
                                 06C4
                                 06C4   STP18:
               0006'CF   12  D0  06C4                MOVL    #18,W^CURRENT_TC
                         00  DD  06C9                PUSHL   #0
               1D2C'CF   01  FB  06CB                CALLS   #1,W^REG_SAVE
                                 06D0
0369'CF    7F000000'EF       DE  06D0        MOVAL   ^X7F000000,INADDRESS        ; Set P1 address up
036D'CF    7F000000'EF       DE  06D9        MOVAL   ^X7F000000,INADDRESS+4
                                 06E2        $CRMPSC_S CHAN = SECFAB+FAB$L_STV,-
                                 06E2                FLAGS = #SEC$M_WRT!SEC$M_EXPREG,-
                                 06E2                INADR = INADDRESS,-
                                 06E2                RETADR = OUTADDRESS
                                 0708        FAIL_CHECK SS$_CREATED
               00000619 8F   DD  0708                PUSHL   #SS$_CREATED
               1D36'CF   01  FB  070E                CALLS   #1,W^REG_CHECK
                  0361'CF    DF  0713        PUSHAL  OUTADDRESS                  ; Push address quad word
               FFFFFFFF 8F   DD  0717        PUSHL   #-1                         ; Push start data value
               1E89'CF   02  FB  071D        CALLS   #2,DATA_CHECK               ; Check section data
                                 0722        $DELTVA_S INADR = OUTADDRESS        ; Clean up
                      07 50  E8  0731        BLBS    R0,10$                      ; Skip error report if OK
                         50  DD  0734        PUSHL   R0                          ; Save error code
               1F15'CF   01  FB  0736        CALLS   #1,NONSUB_SSE               ; Print the failure
                                 073B   10$:
                                 073B
                                 073B   ;+
                                 073B   ;
                                 073B   ; Create a two page user temporary P0 private local DZRO read/write section.
                                 073B   ; INADR is specified in reverse order.
                                 073B   ;
                                 073B   ;-
                                 073B
                                 073B        NEXT_TEST
                                 073B
```

```
                          073B      STP19:
      0006'CF    13   D0  073B                  MOVL    #19,W^CURRENT_TC
                 00   DD  0740                  PUSHL   #0
      1D2C'CF    01   FB  0742                  CALLS   #1,W^REG_SAVE
                          0747
   036D'CF   0600'CF  DE  0747          MOVAL   READ_BUF,INADDRESS+4
   0369'CF   09FF'CF  DE  074E          MOVAL   READ_BUF+<<512*2>-1>,INADDRESS
                          0755          $CRMPSC_S CHAN = SECFAB+FAB$L_STV,-
                          0755                  FLAGS = #SEC$M_WRT!SEC$M_DZRO,-
                          0755                  INADR = INADDRESS,-
                          0755                  PAGCNT = #2,-
                          0755                  RETADR = OUTADDRESS
                          0779          FAIL_CHECK SS$_CREATED
       00000619 8F    DD  0779                  PUSHL   #SS$_CREATED
      1D36'CF    01   FB  077F                  CALLS   #1,W^REG_CHECK
         0361'CF      DF  0784          PUSHAL  OUTADDRESS                      ; Push address quad word
                 00   DD  0788          PUSHL   #0                              ; Push start data value
      1E89'CF    02   FB  078A          CALLS   #2,DATA_CHECK                   ; Check section data
 0365'DF   0000'CF   0400 8F  28  078F  MOVC3   #<2*512>,WRITE_BUF,@<OUTADDRESS+4> ; Fix 2 DZRO pages
                          0799          $DELTVA_S INADR = OUTADDRESS            ; Clean up
              07 50   E8  07A8          BLBS    R0,10$                          ; Skip error report if OK
                 50   DD  07AB          PUSHL   R0                              ; Save error code
      1F15'CF    01   FB  07AD          CALLS   #1,NONSUB_SSE                   ; Print the failure
                          07B2      10$:
                          07B2
                          07B2      ;+
                          07B2      ;
                          07B2      ; Create a three page user temporary P1 private local DZRO read/write section.
                          07B2      ;
                          07B2      ;-
                          07B2
                          07B2          NEXT_TEST
                          07B2
                          07B2      STP20:
      0006'CF    14   D0  07B2                  MOVL    #20,W^CURRENT_TC
                 00   DD  07B7                  PUSHL   #0
      1D2C'CF    01   FB  07B9                  CALLS   #1,W^REG_SAVE
                          07BE
   0369'CF   70000000'EF  DE  07BE      MOVAL   ^X70000000,INADDRESS
   036D'CF   70000000'EF  DE  07C7      MOVAL   ^X70000000,INADDRESS+4
                          07D0          $CRMPSC_S CHAN = SECFAB+FAB$L_STV,-
                          07D0                  FLAGS = #SEC$M_WRT!SEC$M_DZRO!SEC$M_EXPREG,-
                          07D0                  INADR = INADDRESS,-
                          07D0                  RETADR = OUTADDRESS
                          07F6          FAIL_CHECK SS$_CREATED
       00000619 8F    DD  07F6                  PUSHL   #SS$_CREATED
      1D36'CF    01   FB  07FC                  CALLS   #1,W^REG_CHECK
         0361'CF      DF  0801          PUSHAL  OUTADDRESS                      ; Push address quad word
                 00   DD  0805          PUSHL   #0                              ; Push start data value
      1E89'CF    02   FB  0807          CALLS   #2,DATA_CHECK                   ; Check section data
 0361'DF   0000'CF   0600 8F  28  080C  MOVC3   #<3*512>,WRITE_BUF,@OUTADDRESS  ; Fix 3 DZRO pages
                          0816          $DELTVA_S INADR = OUTADDRESS            ; Clean up
              07 50   E8  0825          BLBS    R0,10$                          ; Skip error report if OK
                 50   DD  0828          PUSHL   R0                              ; Save error code
      1F15'CF    01   FB  082A          CALLS   #1,NONSUB_SSE                   ; Print the failure
                          082F      10$:
                          082F
```

```
                          082F   ;+
                          082F   ;
                          082F   ; Create a three page user temporary P1 private local CRF read/write section.
                          082F   ;
                          082F   ;-
                          082F
                          082F          NEXT_TEST
                          082F
                          082F   STP21:
     0006'CF    15   D0   082F                   MOVL    #21,W^CURRENT_TC
                00   DD   0834                   PUSHL   #0
     1D2C'CF    01   FB   0836                   CALLS   #1,W^REG_SAVE
                          083B
                          083B          $CRMPSC_S CHAN = SECFAB+FAB$L_STV,-
                          083B                   FLAGS = #SEC$M_WRT!SEC$M_CRF!SEC$M_EXPREG,-
                          083B                   INADR = INADDRESS,-
                          083B                  RETADR = OUTADDRESS
                          0861          FAIL_CHECK SS$_CREATED
    00000619 8F      DD   0861                   PUSHL   #SS$_CREATED
     1D36'CF    01   FB   0867                   CALLS   #1,W^REG_CHECK
         0361'CF     DF   086C          PUSHAL  OUTADDRESS                      ; Push address quad word
    FFFFFFFF 8F      DD   0870          PUSHL   #-1                             ; Push start data value
     1E89'CF    02   FB   0876          CALLS   #2,DATA_CHECK                   ; Check section data
                          087B          $DELTVA_S INADR = OUTADDRESS            ; Clean up
          07 50      E8   088A          BLBS    R0,10$                          ; Skip error report if OK
             50      DD   088D          PUSHL   R0                              ; Save error code
     1F15'CF    01   FB   088F          CALLS   #1,NONSUB_SSE                   ; Print the failure
                          0894   10$:
                          0894
                          0894   ;+
                          0894   ;
                          0894   ; Create a two page user temporary P0 private local CRF read/write section.
                          0894   ;
                          0894   ;-
                          0894
                          0894          NEXT_TEST
                          0894
                          0894   STP22:
     0006'CF    16   D0   0894                   MOVL    #22,W^CURRENT_TC
                00   DD   0899                   PUSHL   #0
     1D2C'CF    01   FB   089B                   CALLS   #1,W^REG_SAVE
                          08A0
0369'CF    0600'CF   DE   08A0          MOVAL   READ_BUF,INADDRESS
036D'CF    0A00'CF   DE   08A7          MOVAL   READ_BUF+<512*2>,INADDRESS+4
                          08AE          $CRMPSC_S CHAN = SECFAB+FAB$L_STV,-
                          08AE                   FLAGS = #SEC$M_WRT!SEC$M_CRF,-
                          08AE                   INADR = INADDRESS,-
                          08AE                  PAGCNT = #2,-
                          08AE                  RETADR = OUTADDRESS
                          08D2          FAIL_CHECK SS$_CREATED
    00000619 8F      DD   08D2                   PUSHL   #SS$_CREATED
     1D36'CF    01   FB   08D8                   CALLS   #1,W^REG_CHECK
         0361'CF     DF   08DD          PUSHAL  OUTADDRESS                      ; Push address quad word
    FFFFFFFF 8F      DD   08E1          PUSHL   #-1                             ; Push start data value
     1E89'CF    02   FB   08E7          CALLS   #2,DATA_CHECK                   ; Check section data
                          08EC          $DELTVA_S INADR = OUTADDRESS            ; Clean up
          07 50      E8   08FB          BLBS    R0,10$                          ; Skip error report if OK
```

```
                   50  DD 08FE        PUSHL   R0                            ; Save error code
           1F15'CF 01  FB 0900        CALLS   #1,NONSUB_SSE                 ; Print the failure
                          0905 10$:
                          0905
                          0905 ;+
                          0905 ;
                          0905 ; Create a one page (by PAGCNT) user temporary P0 private local CRF
                          0905 ; read/write section.
                          0905 ;
                          0905 ;-
                          0905
                          0905        NEXT_TEST
                          0905
                          0905 STP23:
           0006'CF 17  D0 0905                MOVL    #23,W^CURRENT_TC
                   00  DD 090A                PUSHL   #0
           1D2C'CF 01  FB 090C                CALLS   #1,W^REG_SAVE
                          0911
                          0911        $CRMPSC_S CHAN = SECFAB+FAB$L_STV,-
                          0911                FLAGS = #SEC$M_WRT!SEC$M_CRF,-
                          0911                 INADR = INADDRESS,-
                          0911                RETADR = OUTADDRESS,-
                          0911                PAGCNT = #1
                          0935        FAIL_CHECK SS$_CREATED
          00000619 8F  DD 0935                PUSHL   #SS$_CREATED
           1D36'CF 01  FB 093B                CALLS   #1,W^REG_CHECK
              0361'CF  DF 0940        PUSHAL  OUTADDRESS                    ; Push address quad word
          FFFFFFFF 8F  DD 0944        PUSHL   #-1                           ; Push start data value
           1E89'CF 02  FB 094A        CALLS   #2,DATA_CHECK                 ; Check section data
                          094F        $DELTVA_S INADR = OUTADDRESS          ; Clean up
                07 50  E8 095E        BLBS    R0,10$                        ; Skip error report if OK
                   50  DD 0961        PUSHL   R0                            ; Save error code
           1F15'CF 01  FB 0963        CALLS   #1,NONSUB_SSE                 ; Print the failure
                          0968 10$:
                          0968
                          0968 ;+
                          0968 ;
                          0968 ; Create a three page (by PAGCNT) user temporary P1 private local DZRO
                          0968 ; read/write section.
                          0968 ;
                          0968 ;-
                          0968
                          0968        NEXT_TEST
                          0968
                          0968 STP24:
           0006'CF 18  D0 0968                MOVL    #24,W^CURRENT_TC
                   00  DD 096D                PUSHL   #0
           1D2C'CF 01  FB 096F                CALLS   #1,W^REG_SAVE
                          0974
0369'CF  70000000'EF  DE 0974        MOVAL   ^X70000000,INADDRESS
036D'CF  70000000'EF  DE 097D        MOVAL   ^X70000000,INADDRESS+4
                          0986        $CRMPSC_S CHAN = SECFAB+FAB$L_STV,-
                          0986                FLAGS = #SEC$M_EXPREG!SEC$M_WRT!SEC$M_DZRO,-
                          0986                 INADR = INADDRESS,-
                          0986                RETADR = OUTADDRESS,-
                          0986                PAGCNT = #3
                          09AE        FAIL_CHECK SS$_CREATED
```

```
              00000619 8F   DD  09AE               PUSHL   #SS$_CREATED
               1D36'CF  01  FB  09B4               CALLS   #1,W^REG_CHECK
                  0361'CF   DF  09B9         PUSHAL  OUTADDRESS                     ; Push address quad word
                       00   DD  09BD         PUSHL   #0                             ; Push start data value
               1E89'CF  02  FB  09BF         CALLS   #2,DATA_CHECK                  ; Check section data
0361'DF   0000'CF  0600 8F  28  09C4         MOVC3   #<3*512>,WRITE_BUF,@OUTADDRESS ; Fix 3 DZRO pages
                                09CE         $DELTVA_S INADR = OUTADDRESS           ; Clean up
                    07 50   E8  09DD         BLBS    R0,10$                         ; Skip error report if OK
                       50   DD  09E0         PUSHL   R0                             ; Save error code
               1F15'CF  01  FB  09E2         CALLS   #1,NONSUB_SSE                  ; Print the failure
                                09E7   10$:
                                09E7
                                09E7   ;+
                                09E7   ;
                                09E7   ; Create a three page (PAGCNT=4) user temporary P0 private local read only section.
                                09E7   ;
                                09E7   ;-
                                09E7
                                09E7         NEXT_TEST
                                09E7
                                09E7   STP25:
               0006'CF  19  D0  09E7               MOVL    #25,W^CURRENT_TC
                        00  DD  09EC               PUSHL   #0
               1D2C'CF  01  FB  09EE               CALLS   #1,W^REG_SAVE
                                09F3
         0369'CF   0600'CF  DE  09F3         MOVAL   READ_BUF,INADDRESS
         036D'CF   0E00'CF  DE  09FA         MOVAL   READ_BUF+<512*4>,INADDRESS+4
                                0A01         $CRMPSC_S CHAN = SECFAB+FAB$L_STV,-
                                0A01                 FLAGS = #SEC$M_EXPREG,-
                                0A01                 INADR = INADDRESS,-
                                0A01                RETADR = OUTADDRESS,-
                                0A01                PAGCNT = #4
                                0A29         FAIL_CHECK SS$_CREATED
              00000619 8F   DD  0A29               PUSHL   #SS$_CREATED
               1D36'CF  01  FB  0A2F               CALLS   #1,W^REG_CHECK
                  0361'CF   DF  0A34         PUSHAL  OUTADDRESS                     ; Push address quad word
              FFFFFFFF 8F   DD  0A38         PUSHL   #-1                            ; Push start data value
               1E89'CF  02  FB  0A3E         CALLS   #2,DATA_CHECK                  ; Check section data
                                0A43         $DELTVA_S INADR = OUTADDRESS           ; Clean up
                    07 50   E8  0A52         BLBS    R0,10$                         ; Skip error report if OK
                       50   DD  0A55         PUSHL   R0                             ; Save error code
               1F15'CF  01  FB  0A57         CALLS   #1,NONSUB_SSE                  ; Print the failure
                                0A5C   10$:
                                0A5C
                                0A5C   ;+
                                0A5C   ;
                                0A5C   ; Create a one page VBN 1 user temporary P0 private local read only section.
                                0A5C   ;
                                0A5C   ;-
                                0A5C
                                0A5C         NEXT_TEST
                                0A5C
                                0A5C   STP26:
               0006'CF  1A  D0  0A5C               MOVL    #26,W^CURRENT_TC
                        00  DD  0A61               PUSHL   #0
               1D2C'CF  01  FB  0A63               CALLS   #1,W^REG_SAVE
                                0A68
```

```
                                 0A68       $CRMPSC_S CHAN = SECFAB+FAB$L_STV,-
                                 0A68                FLAGS = #SEC$M_EXPREG,-
                                 0A68                INADR = INADDRESS,-
                                 0A68               RETADR = OUTADDRESS,-
                                 0A68               PAGCNT = #1,-
                                 0A68                  VBN = #1
                                 0A90       FAIL_CHECK SS$_CREATED
           00000619 8F      DD   0A90                 PUSHL   #SS$_CREATED
            1D36'CF 01      FB   0A96                 CALLS   #1,W^REG_CHECK
               0361'CF      DF   0A9B       PUSHAL  OUTADDRESS                    ; Push address quad word
           FFFFFFFF 8F      DD   0A9F       PUSHL   #-1                           ; Push start data value
            1E89'CF 02      FB   0AA5       CALLS   #2,DATA_CHECK                 ; Check section data
                                 0AAA       $DELTVA_S INADR = OUTADDRESS          ; Clean up
                  07 50     E8   0AB9       BLBS    R0,10$                        ; Skip error report if OK
                     50     DD   0ABC       PUSHL   R0                            ; Save error code
            1F15'CF 01      FB   0ABE       CALLS   #1,NONSUB_SSE                 ; Print the failure
                                 0AC3  10$:
                                 0AC3
                                 0AC3  ;+
                                 0AC3  ;
                                 0AC3  ; Create a two page VBN 2 user temporary P1 private local read only section.
                                 0AC3  ;
                                 0AC3  ;-
                                 0AC3
                                 0AC3       NEXT_TEST
                                 0AC3
                                 0AC3  STP27:
            0006'CF 1B      D0   0AC3                 MOVL    #27,W^CURRENT_TC
                     00     DD   0AC8                 PUSHL   #0
            1D2C'CF 01      FB   0ACA                 CALLS   #1,W^REG_SAVE
                                 0ACF
   0369'CF  70000000'EF     DE   0ACF       MOVAL   ^X70000000,INADDRESS
   036D'CF  70000000'EF     DE   0AD8       MOVAL   ^X70000000,INADDRESS+4
                                 0AE1       $CRMPSC_S CHAN = SECFAB+FAB$L_STV,-
                                 0AE1                FLAGS = #SEC$M_EXPREG,-
                                 0AE1                INADR = INADDRESS,-
                                 0AE1               RETADR = OUTADDRESS,-
                                 0AE1               PAGCNT = #2,-
                                 0AE1                  VBN = #2
                                 0B09       FAIL_CHECK SS$_CREATED
           00000619 8F      DD   0B09                 PUSHL   #SS$_CREATED
            1D36'CF 01      FB   0B0F                 CALLS   #1,W^REG_CHECK
               0361'CF      DF   0B14       PUSHAL  OUTADDRESS                    ; Push address quad word
           FFFFFF7F 8F      DD   0B18       PUSHL   #<-1-128>                     ; Push start data value
            1E89'CF 02      FB   0B1E       CALLS   #2,DATA_CHECK                 ; Check section data
                                 0B23       $DELTVA_S INADR = OUTADDRESS          ; Clean up
                  07 50     E8   0B32       BLBS    R0,10$                        ; Skip error report if OK
                     50     DD   0B35       PUSHL   R0                            ; Save error code
            1F15'CF 01      FB   0B37       CALLS   #1,NONSUB_SSE                 ; Print the failure
                                 0B3C  10$:
                                 0B3C
                                 0B3C  ;+
                                 0B3C  ;
                                 0B3C  ; Create a three page VBN 3 user temporary P0 private local read only section.
                                 0B3C  ;
                                 0B3C  ;-
                                 0B3C
```

```
                         0B3C          NEXT_TEST
                         0B3C
                         0B3C  STP28:
      0006'CF    1C  D0  0B3C                  MOVL    #28,W^CURRENT_TC
                 00  DD  0B41                  PUSHL   #0
      1D2C'CF    01  FB  0B43                  CALLS   #1,W^REG_SAVE
                         0B48
 0369'CF   0600'CF   DE  0B48          MOVAL   READ_BUF,INADDRESS
 036D'CF   0C00'CF   DE  0B4F          MOVAL   READ_BUF+<512*3>,INADDRESS+4
                         0B56          $CRMPSC_S CHAN = SECFAB+FAB$L_STV,-
                         0B56                   FLAGS = #SEC$M_EXPREG,-
                         0B56                   INADR = INADDRESS,-
                         0B56                  RETADR = OUTADDRESS,-
                         0B56                  PAGCNT = #3,-
                         0B56                     VBN = #3
                         0B7E          FAIL_CHECK SS$_CREATED
    00000619 8F  DD  0B7E                  PUSHL   #SS$_CREATED
    1D36'CF  01  FB  0B84                  CALLS   #1,W^REG_CHECK
        0361'CF  DF  0B89          PUSHAL  OUTADDRESS                 ; Push address quad word
    FFFFFEFF 8F  DD  0B8D          PUSHL   #<-1-<2*128>>              ; Push start data value
    1E89'CF  02  FB  0B93          CALLS   #2,DATA_CHECK              ; Check section data
                     0B98          $DELTVA_S INADR = OUTADDRESS       ; Clean up
          07 50  E8  0BA7          BLBS    R0,10$                     ; Skip error report if OK
             50  DD  0BAA          PUSHL   R0                         ; Save error code
    1F15'CF  01  FB  0BAC          CALLS   #1,NONSUB_SSE              ; Print the failure
                     0BB1  10$:
                     0BB1
                     0BB1  ;+
                     0BB1  ;
                     0BB1  ; Create a one page VBN 4 user temporary P0 private local read only section.
                     0BB1  ;
                     0BB1  ;-
                     0BB1
                     0BB1          NEXT_TEST
                     0BB1
                     0BB1  STP29:
      0006'CF    1D  D0  0BB1                  MOVL    #29,W^CURRENT_TC
                 00  DD  0BB6                  PUSHL   #0
      1D2C'CF    01  FB  0BB8                  CALLS   #1,W^REG_SAVE
                         0BBD
                         0BBD          $CRMPSC_S CHAN = SECFAB+FAB$L_STV,-
                         0BBD                   FLAGS = #SEC$M_EXPREG,-
                         0BBD                   INADR = INADDRESS,-
                         0BBD                  RETADR = OUTADDRESS,-
                         0BBD                  PAGCNT = #1,-
                         0BBD                     VBN = #4
                         0BE5          FAIL_CHECK SS$_ENDOFFILE
    00000870 8F  DD  0BE5                  PUSHL   #SS$_ENDOFFILE
    1D36'CF  01  FB  0BEB                  CALLS   #1,W^REG_CHECK
                     0BF0
                     0BF0  ;+
                     0BF0  ;
                     0BF0  ; Create a Supervisor section
                     0BF0  ;
                     0BF0  ;-
                     0BF0
                     0BF0          NEXT_TEST
```

```
                          0BF0
                          0BF0        STP30:
     0006'CF   1E    D0   0BF0                   MOVL    #30,W^CURRENT_TC
               00    DD   0BF5                   PUSHL   #0
     1D2C'CF   01    FB   0BF7                   CALLS   #1,W^REG_SAVE
                          0BFC
000A'CF   01E8'CF    DE   0BFC             MOVAL   SM,MODE                      ; Set the mode ID address
                          0C03
                          0C03        ;+
                          0C03        ;
                          0C03        ; Create a Executive section
                          0C03        ;
                          0C03        ;-
                          0C03
                          0C03             NEXT_TEST
                          0C03
                          0C03        STP31:
     0006'CF   1F    D0   0C03                   MOVL    #31,W^CURRENT_TC
               00    DD   0C08                   PUSHL   #0
     1D2C'CF   01    FB   0C0A                   CALLS   #1,W^REG_SAVE
                          0C0F
000A'CF   01F5'CF    DE   0C0F             MOVAL   EM,MODE                      ; Set the mode ID address
                          0C16
                          0C16        ;+
                          0C16        ;
                          0C16        ; Create a Kernal section
                          0C16        ;
                          0C16        ;-
                          0C16
                          0C16             NEXT_TEST
                          0C16
                          0C16        STP32:
     0006'CF   20    D0   0C16                   MOVL    #32,W^CURRENT_TC
               00    DD   0C1B                   PUSHL   #0
     1D2C'CF   01    FB   0C1D                   CALLS   #1,W^REG_SAVE
                          0C22
000A'CF   0206'CF    DE   0C22             MOVAL   KM,MODE                      ; Set the mode ID address
000A'CF   01DC'CF    DE   0C29             MOVAL   UM,MODE                      ; Set the mode ID address
                          0C30
                          0C30        ;+
                          0C30        ; Global section tests
                          0C30        ;
                          0C30        ; 0 length GSDNAM
                          0C30        ;
                          0C30        ;-
                          0C30
                          0C30             NEXT_TEST
                          0C30
                          0C30        STP33:
     0006'CF   21    D0   0C30                   MOVL    #33,W^CURRENT_TC
               00    DD   0C35                   PUSHL   #0
     1D2C'CF   01    FB   0C37                   CALLS   #1,W^REG_SAVE
                          0C3C
                          0C3C             $CRMPSC_S GSDNAM = ZEROSTRING,-
                          0C3C                         INADR = INADDRESS,-
                          0C3C                          CHAN = SECFAB+FAB$L_STV,-
                          0C3C                         FLAGS = #SEC$M_GBL
```

```
                    0C5E        FAIL_CHECK SS$_IVLOGNAM
 00000154 8F   DD   0C5E                   PUSHL   #SS$_IVLOGNAM
 1D36'CF  01   FB   0C64                   CALLS   #1,W^REG_CHECK
                    0C69   ;+
                    0C69   ;
                    0C69   ; GSDNAM too long (16 characters)
                    0C69   ;
                    0C69   ;-
                    0C69
                    0C69        NEXT_TEST
                    0C69
                    0C69   STP34:
 0006'CF  22   D0   0C69                   MOVL    #34,W^CURRENT_TC
          00   DD   0C6E                   PUSHL   #0
 1D2C'CF  01   FB   0C70                   CALLS   #1,W^REG_SAVE
                    0C75
                    0C75        $CRMPSC_S GSDNAM = TOOLONGSTRING,-
                    0C75                   INADR = INADDRESS,-
                    0C75                    CHAN = SECFAB+FAB$L_STV,-
                    0C75                   FLAGS = #SEC$M_GBL
                    0C97        FAIL_CHECK SS$_IVLOGNAM
 00000154 8F   DD   0C97                   PUSHL   #SS$_IVLOGNAM
 1D36'CF  01   FB   0C9D                   CALLS   #1,W^REG_CHECK
                    0CA2   ;+
                    0CA2   ;
                    0CA2   ; GSDNAM address unaccessable
                    0CA2   ;
                    0CA2   ;-
                    0CA2
                    0CA2        NEXT_TEST
                    0CA2
                    0CA2   STP35:
 0006'CF  23   D0   0CA2                   MOVL    #35,W^CURRENT_TC
          00   DD   0CA7                   PUSHL   #0
 1D2C'CF  01   FB   0CA9                   CALLS   #1,W^REG_SAVE
                    0CAE
                    0CAE        $CRMPSC_S GSDNAM = 0,-
                    0CAE                   INADR = INADDRESS,-
                    0CAE                    CHAN = SECFAB+FAB$L_STV,-
                    0CAE                   FLAGS = #SEC$M_GBL
                    0CCE        FAIL_CHECK SS$_ACCVIO
          0C   DD   0CCE                   PUSHL   #SS$_ACCVIO
 1D36'CF  01   FB   0CD0                   CALLS   #1,W^REG_CHECK
                    0CD5   ;+
                    0CD5   ;
                    0CD5   ; Test no read access on GSDNAM
                    0CD5   ;
                    0CD5   ;-
                    0CD5
                    0CD5        NEXT_TEST
                    0CD5
                    0CD5   STP36:
 0006'CF  24   D0   0CD5                   MOVL    #36,W^CURRENT_TC
          00   DD   0CDA                   PUSHL   #0
 1D2C'CF  01   FB   0CDC                   CALLS   #1,W^REG_SAVE
                    0CE1
                    0CE1        $CRMPSC_S GSDNAM = READ_BUF,-
```

```
                       0CE1               INADR = INADDRESS,-
                       0CE1                CHAN = SECFAB+FAB$L_STV,-
                       0CE1               FLAGS = #SEC$M_GBL                ; Try for accvio
                       0D03        FAIL_CHECK SS$_ACCVIO
              0C   DD  0D03                PUSHL   #SS$_ACCVIO
   1D36'CF    01   FB  0D05                CALLS   #1,W^REG_CHECK
                       0D0A   ;+
                       0D0A   ;
                       0D0A   ; Test no read access on IDENT
                       0D0A   ;
                       0D0A   ;-
                       0D0A
                       0D0A        NEXT_TEST
                       0D0A
                       0D0A   STP37:
   0006'CF    25   D0  0D0A                MOVL    #37,W^CURRENT_TC
              00   DD  0D0F                PUSHL   #0
   1D2C'CF    01   FB  0D11                CALLS   #1,W^REG_SAVE
                       0D16
                       0D16        $CRMPSC_S INADR = OUTADDRESS,-
                       0D16                   CHAN  = SECFAB+FAB$L_STV,-
                       0D16                   FLAGS = #SEC$M_GBL,-
                       0D16                  GSDNAM = GSDNAM,-
                       0D16                   IDENT = READ_BUF
                       0D3A        FAIL_CHECK SS$_ACCVIO
              0C   DD  0D3A                PUSHL   #SS$_ACCVIO
   1D36'CF    01   FB  0D3C                CALLS   #1,W^REG_CHECK
                       0D41   ;+
                       0D41   ;
                       0D41   ; Test 11 levels of logical name translation error
                       0D41   ;
                       0D41   ;-
                       0D41
                       0D41        NEXT_TEST
                       0D41
                       0D41   STP38:
   0006'CF    26   D0  0D41                MOVL    #38,W^CURRENT_TC
              00   DD  0D46                PUSHL   #0
   1D2C'CF    01   FB  0D48                CALLS   #1,W^REG_SAVE
                       0D4D
              04   BB  0D4D        PUSHR   #^M<R2>                     ; Save it's contents
              52   D4  0D4F        CLRL    R2                          ; Set index variable
                       0D51   10$:
                       0D51        $CRELOG_S LOGNAM = LOGICAL_NAME,-
                       0D51                  TBLFLG = #2,-
                       0D51                  EQLNAM = EQUIV_NAME         ; Create a level of logical name
           07 50   E8  0D64        BLBS    R0,20$                      ; Skip error report if OK
              50   DD  0D67        PUSHL   R0                          ; Save error code
   1F15'CF    01   FB  0D69        CALLS   #1,NONSUB_SSE               ; Print the failure
                       0D6E   20$:
        0491'CF   96   0D6E        INCB    NAME_INC1                   ; Bump the logical name by one
        049E'CF   96   0D72        INCB    NAME_INC2                   ; Bump the equiv name by one
      D7 52   0B   F2  0D76        AOBLSS  #11,R2,10$                  ; Make 11 levels
              04   BA  0D7A        POPR    #^M<R2>                     ; Restore it
                       0D7C        $CRMPSC_S FLAGS = #SEC$M_GBL,-
                       0D7C                  INADR = INADDRESS,-
                       0D7C                   CHAN = SECFAB+FAB$L_STV,-
```

```
                          0D7C                  GSDNAM = GSD_LOGNAM
                          0D9E        FAIL_CHECK SS$_TOOMANYLNAM
     00000374 8F    DD    0D9E                  PUSHL   #SS$_TOOMANYLNAM
     1D36'CF    01  FB    0DA4                  CALLS   #1,W^REG_CHECK
                52  D4    0DA9        CLRL    R2                            ; Reset index variable
        0491'CF     97    0DAB        DECB    NAME_INC1                     ; Back up to a legal name
                          0DAF   40$:
                          0DAF        $DELLOG_S TBLFLG = #2,-
                          0DAF                  LOGNAM = LOGICAL_NAME       ; Clean up process logical name tabl
           07 50    E8    0DBE        BLBS    R0,50$                        ; Skip error report if OK
                50  DD    0DC1        PUSHL   R0                            ; Save error code
     1F15'CF    01  FB    0DC3        CALLS   #1,NONSUB_SSE                 ; Print the failure
                          0DC8   50$:
        0491'CF     97    0DC8        DECB    NAME_INC1                     ; Bump the logical name
      DF 52     0B  F2    0DCC        AOBLSS  #11,R2,40$                    ; Eat all 11 levels
                          0DD0   ;+
                          0DD0   ;
                          0DD0   ; Test PRM GBL section creation without PRMGBL privilege
                          0DD0   ;
                          0DD0   ;-
                          0DD0
                          0DD0        NEXT_TEST
                          0DD0
                          0DD0   STP39:
     0006'CF    27  D0    0DD0                  MOVL    #39,W^CURRENT_TC
                00  DD    0DD5                  PUSHL   #0
     1D2C'CF    01  FB    0DD7                  CALLS   #1,W^REG_SAVE
                          0DDC
                          0DDC        $CRMPSC_S INADR = INADDRESS,-
                          0DDC                  CHAN  = SECFAB+FAB$L_STV,-
                          0DDC                  GSDNAM= GSDNAM,-
                          0DDC                  FLAGS = #SEC$M_PERM!SEC$M_GBL
                          0E02        FAIL_CHECK SS$_NOPRIV
                24  DD    0E02                  PUSHL   #SS$_NOPRIV
     1D36'CF    01  FB    0E04                  CALLS   #1,W^REG_CHECK
                          0E09   ;+
                          0E09   ;
                          0E09   ; Create a one page user PRM GBL P0 local memory read write section.
                          0E09   ;
                          0E09   ;-
                          0E09
                          0E09        NEXT_TEST
                          0E09
                          0E09   STP40:
     0006'CF    28  D0    0E09                  MOVL    #40,W^CURRENT_TC
                00  DD    0E0E                  PUSHL   #0
     1D2C'CF    01  FB    0E10                  CALLS   #1,W^REG_SAVE
                          0E15
  0369'CF  0600'CF  DE    0E15        MOVAL READ_BUF,INADDRESS              ; Set the P0 address up
  036D'CF  0800'CF  DE    0E1C        MOVAL READ_BUF+512,INADDRESS+4
0355'CF  01  18  01  F0   0E23        INSV    #1,#PRV$V_PRMGBL,#1,PRIV_MASK ; Set the PRMGBL priv bit
                          0E2A        $SETPRV_S ENBFLG = #1,-
                          0E2A                  PRVADR = PRIV_MASK          ; Get the PRMGBL priv
                          0E3B        $CRMPSC_S INADR = INADDRESS,-
                          0E3B                  CHAN  = SECFAB+FAB$L_STV,-
                          0E3B                  GSDNAM= GSDNAM,-
                          0E3B                  FLAGS = #SEC$M_GBL!SEC$M_WRT!SEC$M_PERM
```

```
                                   0E61        FAIL_CHECK SS$_CREATED
         00000619 8F        DD     0E61                    PUSHL   #SS$_CREATED
         1D36'CF  01        FB     0E67                    CALLS   #1,W^REG_CHECK
              0369'CF       DF     0E6C        PUSHAL INADDRESS
         FFFFFFFF 8F        DD     0E70        PUSHL   #-1
         1E89'CF  02        FB     0E76        CALLS   #2,DATA_CHECK                          ; Check the section data
                                   0E7B        $DELTVA_S INADR = INADDRESS                    ; Clean up the mess
               07 50        E8     0E8A        BLBS    R0,10$                                 ; Skip error report if OK
                  50        DD     0E8D        PUSHL   R0                                     ; Save error code
         1F15'CF  01        FB     0E8F        CALLS   #1,NONSUB_SSE                          ; Print the failure
                                   0E94   10$:
                                   0E94   ;+
                                   0E94   ;
                                   0E94   ; Map to a one page user PRM GBL P0 local memory read only section.
                                   0E94   ;
                                   0E94   ;-
                                   0E94
                                   0E94        NEXT_TEST
                                   0E94
                                   0E94   STP41:
         0006'CF  29        D0     0E94                    MOVL    #41,W^CURRENT_TC
                  00        DD     0E99                    PUSHL   #0
         1D2C'CF  01        FB     0E9B                    CALLS   #1,W^REG_SAVE
                                   0EA0
0369'CF  00000200 8F        C0     0EA0        ADDL2 #512,INADDRESS                           ; Set the P0 address up to
036D'CF  00000200 8F        C0     0EA9        ADDL2 #512,INADDRESS+4                         ; the next buffer page
                                   0EB2        $CRMPSC_S INADR = INADDRESS,-
                                   0EB2                  CHAN  = SECFAB+FAB$L_STV,-
                                   0EB2                  GSDNAM= GSDNAM,-
                                   0EB2                  FLAGS = #SEC$M_GBL
                                   0ED4        FAIL_CHECK SS$_NORMAL
                  01        DD     0ED4                    PUSHL   #SS$_NORMAL
         1D36'CF  01        FB     0ED6                    CALLS   #1,W^REG_CHECK
              0369'CF       DF     0EDB        PUSHAL INADDRESS
         FFFFFFFF 8F        DD     0EDF        PUSHL   #-1
         1E89'CF  02        FB     0EE5        CALLS   #2,DATA_CHECK                          ; Check the section data
                                   0EEA        $DELTVA_S INADR = INADDRESS                    ; Clean up the mess
               07 50        E8     0EF9        BLBS    R0,10$                                 ; Skip error report if OK
                  50        DD     0EFC        PUSHL   R0                                     ; Save error code
         1F15'CF  01        FB     0EFE        CALLS   #1,NONSUB_SSE                          ; Print the failure
                                   0F03   10$:
                                   0F03        $DGBLSC_S GSDNAM = GSDNAM
               07 50        E8     0F12        BLBS    R0,20$                                 ; Skip error report if OK
                  50        DD     0F15        PUSHL   R0                                     ; Save error code
         1F15'CF  01        FB     0F17        CALLS   #1,NONSUB_SSE                          ; Print the failure
                                   0F1C   20$:
                                   0F1C
                                   0F1C   ;+
                                   0F1C   ;
                                   0F1C   ; Test negative RELPAG
                                   0F1C   ;
                                   0F1C   ;-
                                   0F1C
                                   0F1C        NEXT_TEST
                                   0F1C
                                   0F1C   STP42:
         0006'CF  2A        D0     0F1C                    MOVL    #42,W^CURRENT_TC
```

```
                 00  DD  0F21                    PUSHL   #0
         1D2C'CF 01  FB  0F23                    CALLS   #1,W^REG_SAVE
                         0F28
                         0F28            $CRMPSC_S INADR = INADDRESS,-
                         0F28                      FLAGS = #SEC$M_GBL!SEC$M_EXPREG,-
                         0F28                       CHAN = SECFAB+FAB$L_STV,-
                         0F28                     GSDNAM = GSD_LOGNAM,-
                         0F28                     RELPAG = #-1
                         0F52            FAIL_CHECK SS$_ENDOFFILE
        00000870 8F  DD  0F52                    PUSHL   #SS$_ENDOFFILE
         1D36'CF 01  FB  0F58                    CALLS   #1,W^REG_CHECK
                         0F5D      ;+
                         0F5D      ;
                         0F5D      ; Test RELPAG too large
                         0F5D      ;
                         0F5D      ;-
                         0F5D
                         0F5D            NEXT_TEST
                         0F5D
                         0F5D      STP43:
         0006'CF 2B  D0  0F5D                    MOVL    #43,W^CURRENT_TC
                 00  DD  0F62                    PUSHL   #0
         1D2C'CF 01  FB  0F64                    CALLS   #1,W^REG_SAVE
                         0F69            $CRMPSC_S INADR = INADDRESS,-
                         0F69                      FLAGS = #SEC$M_GBL!SEC$M_EXPREG,-
                         0F69                       CHAN = SECFAB+FAB$L_STV,-
                         0F69                     GSDNAM = GSD_LOGNAM,-
                         0F69                     RELPAG = #5                     ; Only a three page section specifie
                         0F8F            FAIL_CHECK SS$_ENDOFFILE
        00000870 8F  DD  0F8F                    PUSHL   #SS$_ENDOFFILE
         1D36'CF 01  FB  0F95                    CALLS   #1,W^REG_CHECK
                         0F9A      ;+
                         0F9A      ;
                         0F9A      ; Test the offset definitions and the _G format of the system service.
                         0F9A      ;
                         0F9A      ;-
                         0F9A
                         0F9A            NEXT_TEST
                         0F9A
                         0F9A      STP44:
         0006'CF 2C  D0  0F9A                    MOVL    #44,W^CURRENT_TC
                 00  DD  0F9F                    PUSHL   #0
         1D2C'CF 01  FB  0FA1                    CALLS   #1,W^REG_SAVE
            0040 8F  BB  0FA6            PUSHR   #^M<R6>                          ; Save the register contents
      56    02B6'CF  DE  0FAA            MOVAL   CRMPSCG,R6                       ; Save the parameter list address
   04 A6    0369'CF  DE  0FAF            MOVAL   INADDRESS,CRMPSC$_INADR(R6)      ; Set the parameters up
   08 A6    0361'CF  DE  0FB5            MOVAL   OUTADDRESS,CRMPSC$_RETADR(R6)
      0C A6      03  D0  0FBB            MOVL    #PSL$C_USER,CRMPSC$_ACMODE(R6)
      00020001 8F    D0  0FBF            MOVL    #SEC$M_EXPREG!SEC$M_GBL,-
             10 A6       0FC5                    CRMPSC$_FLAGS(R6)
   14 A6    0346'CF  DE  0FC7            MOVAL   GSDNAM,CRMPSC$_GSDNAM(R6)
            033E'CF  7C  0FCD            CLRQ    AN_IDENT
   18 A6    033E'CF  DE  0FD1            MOVAL   AN_IDENT,CRMPSC$_IDENT(R6)
      1C A6      01  D0  0FD7            MOVL    #1,CRMPSC$_RELPAG(R6)
   20 A6    160C'CF  D0  0FDB            MOVL    SECFAB+FAB$L_STV,CRMPSC$_CHAN(R6)
      24 A6      02  D0  0FE1            MOVL    #2,CRMPSC$_PAGCNT(R6)
      28 A6      01  D0  0FE5            MOVL    #1,CRMPSC$_VBN(R6)
```

```
        2C A6    00    D0  0FE9           MOVL    #0,CRMPSC$_PROT(R6)
        30 A6    02    D0  0FED           MOVL    #2,CRMPSC$_PFC(R6)
         0040 8F       BA  0FF1           POPR    #^M<R6>                               ; Restore register
                           0FF5           $CRMPSC_G CRMPSCG
                           0FFE           FAIL_CHECK SS$_CREATED                        ; Check the status
     00000619 8F       DD  0FFE                   PUSHL   #SS$_CREATED
      1D36'CF    01    FB  1004                   CALLS   #1,W^REG_CHECK
        0361'CF        DF  1009           PUSHAL  OUTADDRESS                            ; Check the data
     FFFFFF7F 8F       DD  100D           PUSHL   #-<<512/4>+1>
      1E89'CF    02    FB  1013           CALLS   #2,DATA_CHECK
                           1018           $DELTVA_S INADR = OUTADDRESS                  ; Clean up
           07 50       E8  1027           BLBS    R0,10$                                ; Skip error report if OK
              50       DD  102A           PUSHL   R0                                    ; Save error code
      1F15'CF    01    FB  102C           CALLS   #1,NONSUB_SSE                         ; Print the failure
                           1031      10$:
                           1031           $DGBLSC_S GSDNAM = GSDNAM
           07 50       E8  1040           BLBS    R0,20$                                ; Skip error report if OK
              50       DD  1043           PUSHL   R0                                    ; Save error code
      1F15'CF    01    FB  1045           CALLS   #1,NONSUB_SSE                         ; Print the failure
                           104A      20$:
                           104A      ;+
                           104A      ;
                           104A      ; Create a one page PRM GBL read only section in P0.
                           104A      ;
                           104A      ;-
                           104A           NEXT_TEST
                           104A
                           104A      STP45:
      0006'CF    2D    D0  104A                   MOVL    #45,W^CURRENT_TC
                 00    DD  104F                   PUSHL   #0
      1D2C'CF    01    FB  1051                   CALLS   #1,W^REG_SAVE
                           1056
0369'CF    0600'CF     DE  1056           MOVAL   READ_BUF,INADDRESS                    ; Set buffer address
036D'CF    07FF'CF     DE  105D           MOVAL   READ_BUF+511,INADDRESS+4
                           1064           $CRMPSC_S INADR = INADDRESS,-
                           1064                      CHAN = SECFAB+FAB$L_STV,-
                           1064                    GSDNAM = GSDNAM,-
                           1064                     FLAGS = #SEC$M_GBL!SEC$M_PERM,-
                           1064                    PAGCNT = #2,-
                           1064                    RELPAG = #1                          ; Create the section
                           108C           FAIL_CHECK SS$_CREATED
     00000619 8F       DD  108C                   PUSHL   #SS$_CREATED
      1D36'CF    01    FB  1092                   CALLS   #1,W^REG_CHECK
        0369'CF        DF  1097           PUSHAL  INADDRESS                             ; Push address spec
     FFFFFF7F 8F       DD  109B           PUSHL   #-<<512/4>+1>                         ; Push expected data pattern
      1E89'CF    02    FB  10A1           CALLS   #2,DATA_CHECK                         ; Check the data
                           10A6           $DELTVA_S INADR = INADDRESS                   ; Leave section but remove the mappi
           07 50       E8  10B5           BLBS    R0,10$                                ; Check the status return
              50       DD  10B8           PUSHL   R0                                    ; Push the error code
      1F15'CF    01    FB  10BA           CALLS   #1,NONSUB_SSE                         ; Report the failure
                           10BF      10$:
023A'CF    0041'CF     DE  10BF           MOVAL   MGBLSC,SERV_NAME                      ; Set the new service name
                           10C6           $MGBLSC_S GSDNAM = GSDNAM,-
                           10C6                     INADR = INADDRESS                   ; Remap to the permanent global sect
                           10DF           FAIL_CHECK SS$_NORMAL                         ; Check the status
                 01    DD  10DF                   PUSHL   #SS$_NORMAL
      1D36'CF    01    FB  10E1                   CALLS   #1,W^REG_CHECK
```

```
          0369'CF   DF  10E6          PUSHAL  INADDRESS                         ; Push the address specifier
     FFFFFFFF 8F    DD  10EA          PUSHL   #-1                               ; Push the data pattern specifier
       1E89'CF  02  FB  10F0          CALLS   #2,DATA_CHECK                     ; Check the data
                        10F5          $DELTVA_S INADR = INADDRESS               ; Delete the mapping
            07 50   E8  1104          BLBS    R0,20$                            ; Check the status return
               50   DD  1107          PUSHL   R0                                ; Push the error code
       1F15'CF  01  FB  1109          CALLS   #1,NONSUB_SSE                     ; Report the failure
                        110E    20$:
  023A'CF   003A'CF DE  110E          MOVAL   DGBLSC,SERV_NAME                  ; Set the new service name
                        1115          $DGBLSC_S GSDNAM = GSDNAM                 ; Delete the section
                        1124          FAIL_CHECK SS$_NORMAL                     ; Check the status
               01   DD  1124                  PUSHL   #SS$_NORMAL
       1D36'CF  01  FB  1126                  CALLS   #1,W^REG_CHECK
  023A'CF   0033'CF DE  112B          MOVAL   CRMPSC,SERV_NAME                  ; Set the new service name
                        1132    ;+
                        1132    ;
                        1132    ; Create a one page temp system global read only section in P1 space.
                        1132    ;
                        1132    ;-
                        1132          NEXT_TEST
                        1132
                        1132    STP46:
       0006'CF  2E  D0  1132                  MOVL    #46,W^CURRENT_TC
               00   DD  1137                  PUSHL   #0
       1D2C'CF  01  FB  1139                  CALLS   #1,W^REG_SAVE
                        113E
0355'CF  01  19  01 F0  113E          INSV    #1,#PRV$V_SYSGBL,#1,PRIV_MASK     ; Set SYSGBL priv. bit
0355'CF  01  02  01 F0  1145          INSV    #1,#PRV$V_SYSNAM,#1,PRIV_MASK     ; Set SYSNAM priv. bit
                        114C          $SETPRV_S ENBFLG = #1,-
                        114C                  PRVADR = PRIV_MASK                ; Get the SYSGBL and SYSNAM priv
 0369'CF  70000000'EF DE 115D         MOVAL   ^X70000000,INADDRESS              ; Set a P1 space address
  036D'CF   0369'CF D0  1166          MOVL    INADDRESS,INADDRESS+4
                        116D          $CRMPSC_S INADR = INADDRESS,-
                        116D                  RETADR = OUTADDRESS,-
                        116D                    CHAN = SECFAB+FAB$L_STV,-
                        116D                  GSDNAM = GSDNAM,-
                        116D                   FLAGS = #SEC$M_GBL!SEC$M_SYSGBL!SEC$M_EXPREG,-
                        116D                  PAGCNT = #2,-
                        116D                  RELPAG = #1                       ; Create the section
                        1197          FAIL_CHECK SS$_CREATED
     00000619 8F    DD  1197                  PUSHL   #SS$_CREATED
       1D36'CF  01  FB  119D                  CALLS   #1,W^REG_CHECK
          0361'CF   DF  11A2          PUSHAL  OUTADDRESS                        ; Push address spec
     FFFFFF7F 8F    DD  11A6          PUSHL   #-<<512/4>+1>                     ; Push expected data pattern
       1E89'CF  02  FB  11AC          CALLS   #2,DATA_CHECK                     ; Check the data
                        11B1          $DELTVA_S INADR = OUTADDRESS              ; Leave section but remove the mappi
            07 50   E8  11C0          BLBS    R0,10$                            ; Check the status return
               50   DD  11C3          PUSHL   R0                                ; Push the error code
       1F15'CF  01  FB  11C5          CALLS   #1,NONSUB_SSE                     ; Report the failure
                        11CA    10$:
                        11CA    ;+
                        11CA    ;
                        11CA    ; Create a one page perm system global read only section in P0.
                        11CA    ;
                        11CA    ;-
                        11CA          NEXT_TEST
                        11CA
```

```
                          11CA         STP47:
        0006'CF     2F  D0  11CA                        MOVL    #47,W^CURRENT_TC
                    00  DD  11CF                        PUSHL   #0
        1D2C'CF     01  FB  11D1                        CALLS   #1,W^REG_SAVE
                            11D6
 0369'CF     0600'CF    DE  11D6             MOVAL   READ_BUF,INADDRESS                  ; Set buffer address
 036D'CF     07FF'CF    DE  11DD             MOVAL   READ_BUF+511,INADDRESS+4
                            11E4             $CRMPSC_S INADR = INADDRESS,-
                            11E4                     CHAN = SECFAB+FAB$L_STV,-
                            11E4                   GSDNAM = GSDNAM,-
                            11E4                     FLAGS = #SEC$M_GBL!SEC$M_PERM!SEC$M_SYSGBL,-
                            11E4                   PAGCNT = #2,-
                            11E4                   RELPAG = #1                           ; Create the section
                            120C             FAIL_CHECK SS$_CREATED
          00000619 8F   DD  120C                        PUSHL   #SS$_CREATED
        1D36'CF     01  FB  1212                        CALLS   #1,W^REG_CHECK
             0369'CF    DF  1217             PUSHAL  INADDRESS                           ; Push address spec
          FFFFFF7F 8F   DD  121B             PUSHL   #-<<512/4>+1>                       ; Push expected data pattern
        1E89'CF     02  FB  1221             CALLS   #2,DATA_CHECK                       ; Check the data
                            1226             $DELTVA_S INADR = INADDRESS                 ; Leave section but remove the mappi
              07 50     E8  1235             BLBS    R0,10$                              ; Check the status return
                 50     DD  1238             PUSHL   R0                                  ; Push the error code
        1F15'CF     01  FB  123A             CALLS   #1,NONSUB_SSE                       ; Report the failure
                            123F   10$:
 023A'CF     0041'CF    DE  123F             MOVAL   MGBLSC,SERV_NAME                    ; Set the new service name
                            1246             $MGBLSC_S GSDNAM = GSDNAM,-
                            1246                     FLAGS = #SEC$M_SYSGBL,-
                            1246                     INADR = INADDRESS                   ; Remap to the permanent global sect
                            1263             FAIL_CHECK SS$_NORMAL                       ; Check the status
                    01  DD  1263                        PUSHL   #SS$_NORMAL
        1D36'CF     01  FB  1265                        CALLS   #1,W^REG_CHECK
             0369'CF    DF  126A             PUSHAL  INADDRESS                           ; Push the address specifier
          FFFFFFFF 8F   DD  126E             PUSHL   #-1                                 ; Push the data pattern specifier
        1E89'CF     02  FB  1274             CALLS   #2,DATA_CHECK                       ; Check the data
                            1279             $DELTVA_S INADR = INADDRESS                 ; Delete the mapping
              07 50     E8  1288             BLBS    R0,20$                              ; Check the status return
                 50     DD  128B             PUSHL   R0                                  ; Push the error code
        1F15'CF     01  FB  128D             CALLS   #1,NONSUB_SSE                       ; Report the failure
                            1292   20$:
 023A'CF     003A'CF    DE  1292             MOVAL   DGBLSC,SERV_NAME                    ; Set the new service name
                            1299             $DGBLSC_S GSDNAM = GSDNAM,-
                            1299                     FLAGS = #SEC$M_SYSGBL               ; Delete the section
                            12AC             FAIL_CHECK SS$_NORMAL                       ; Check the status
                    01  DD  12AC                        PUSHL   #SS$_NORMAL
        1D36'CF     01  FB  12AE                        CALLS   #1,W^REG_CHECK
 023A'CF     0033'CF    DE  12B3             MOVAL   CRMPSC,SERV_NAME                    ; Set the new service name
                            12BA   ;+
                            12BA   ;
                            12BA   ;         PFNMAP tests
                            12BA   ;
                            12BA   ; Create a one page temp local memory P0 PFNMAP section
                            12BA   ;
                            12BA   ;-
                            12BA             NEXT_TEST
                            12BA
                            12BA   STP48:
        0006'CF     30  D0  12BA                        MOVL    #48,W^CURRENT_TC
```

```
                        00  DD  12BF              PUSHL   #0
          1D2C'CF       01  FB  12C1              CALLS   #1,W^REG_SAVE
                                12C6
    0355'CF   01   1A   01  F0  12C6              INSV    #1,#PRV$V_PFNMAP,#1,PRIV_MASK   ; Set PFNMAP bit
    0355'CF   01   0C   01  F0  12CD              INSV    #1,#PRV$V_PSWAPM,#1,PRIV_MASK   ; Set PSWAPM bit
    0355'CF   01   01   01  F0  12D4              INSV    #1,#PRV$V_CMEXEC,#1,PRIV_MASK   ; Set CMEXEC bit
                                12DB              $SETPRV_S ENBFLG = #1,-                 ; Get the PFNMAP, PSWAPM and
                                12DB                      PRVADR = PRIV_MASK              ; CMEXEC privilege
      0369'CF    0000'CF    DE  12EC              MOVAL   WRITE_BUF,INADDRESS             ; Set up INADR
      036D'CF    01FF'CF    DE  12F3              MOVAL   WRITE_BUF+511,INADDRESS+4
                                12FA              $LCKPAG_S INADR = INADDRESS             ; Keep the target still
                                1309              $CMEXEC_S ROUTIN = 10$                  ; Get the physical address
                    0039    31  1316              BRW     20$
                                1319        10$:
                        003C    1319              .WORD   ^M<R2,R3,R4,R5>
     55    00000000'9F      D0  131B              MOVL    @#CTL$GL_PHD,R5                 ; Set up Process Header Address
        53    00C8 C5       DE  1322              MOVAL   PHD$L_P0BR(R5),R3               ; Get base register address
        52    0000'CF       DE  1327              MOVAL   WRITE_BUF,R2                    ; Set up virtual address
     51   52   15   09      EF  132C              EXTZV   #VA$V_VPN,#VA$S_VPN,R2,R1       ; Extract the virtual page number
        50    00 B341       D0  1331              MOVL    @(R3)[R1],R0                    ; Get the page table entry
     50   50   15   00      EF  1336              EXTZV   #PTE$V_PFN,#PTE$S_PFN,R0,R0     ; Get the page frame number
     50    00000000'GF      D1  133B              CMPL    G^MMG$GL_MAXPFN,R0              ; Can this be right?
                        05  1E  1342              BGEQU   12$                             ; BR if it is right
                   50   14  D0  1344              MOVL    #SS$_BADPARAM,R0                ; Signal a bad return from the exec
                        08  11  1347              BRB     14$                             ; and bail out of EXEC
                                1349        12$:
          035D'CF       50  D0  1349              MOVL    R0,VBN                          ; Save the good PFN
                   50   01  D0  134E              MOVL    #SS$_NORMAL,R0                  ; Clean exit
                                1351        14$:
                            04  1351              RET
                                1352        20$:
                     0D 50  E8  1352              BLBS    R0,25$                          ; BR if the PFN was aquired OK
                   009E'CF  DF  1355              PUSHAL  BAD_PFN                         ; Report error message
          1D9A'CF       01  FB  1359              CALLS   #1,PRINT_FAIL
                   1504'CF  17  135E              JMP     NO_PFN_TEST                     ; Skip PFN testing
                                1362        25$:
      0369'CF    0600'CF    DE  1362              MOVAL   READ_BUF,INADDRESS              ; Set up INADR
      036D'CF    07FF'CF    DE  1369              MOVAL   READ_BUF+511,INADDRESS+4
                                1370              $CRMPSC_S INADR = INADDRESS,-
                                1370                        FLAGS = #SEC$M_PFNMAP,-
                                1370                          VBN = VBN,-
                                1370                       PAGCNT = #1
                                1396              FAIL_CHECK SS$_CREATED                  ; Check the status return
             00000619 8F    DD  1396                      PUSHL   #SS$_CREATED
          1D36'CF       01  FB  139C                      CALLS   #1,W^REG_CHECK
                  0369'CF   DF  13A1              PUSHAL  INADDRESS                       ; Push address of data
             FFFFFFFF 8F    DD  13A5              PUSHL   #-1                             ; Push data pattern indicator
          1E89'CF       02  FB  13AB              CALLS   #2,DATA_CHECK                   ; Check the data
                                13B0              $DELTVA_S INADR = INADDRESS             ; Delete the section
                     07 50  E8  13BF              BLBS    R0,30$                          ; Check the status return
                        50  DD  13C2              PUSHL   R0                              ; Push the error code
          1F15'CF       01  FB  13C4              CALLS   #1,NONSUB_SSE                   ; Report the failure
                                13C9        30$:
                                13C9
                                13C9        ;+
                                13C9        ;
                                13C9        ; Create a one page PERM local memory GBL P1 PFNMAP section
```

```
                                  13C9      ;
                                  13C9      ;-
                                  13C9              NEXT_TEST
                                  13C9
                                  13C9      STP49:
        0006'CF      31    D0     13C9                      MOVL    #49,W^CURRENT_TC
                     00    DD     13CE                      PUSHL   #0
        1D2C'CF      01    FB     13D0                      CALLS   #1,W^REG_SAVE
                                  13D5
    0369'CF    70000000'EF  DE    13D5              MOVAL   ^X70000000,INADDRESS          ; Set up INADR
    036D'CF    70000000'EF  DE    13DE              MOVAL   ^X70000000,INADDRESS+4
                                  13E7              $CRMPSC_S INADR = INADDRESS,-
                                  13E7                      RETADR = OUTADDRESS,-
                                  13E7                       FLAGS = #SEC$M_PFNMAP!SEC$M_GBL!SEC$M_EXPREG!SEC$M_PERM,-
                                  13E7                      GSDNAM = GSDNAM,-
                                  13E7                         VBN = VBN,-
                                  13E7                      PAGCNT = #1
                                  1411              FAIL_CHECK SS$_CREATED                ; Check the status return
       00000619 8F         DD     1411                      PUSHL   #SS$_CREATED
        1D36'CF      01    FB     1417                      CALLS   #1,W^REG_CHECK
            0361'CF        DF     141C              PUSHAL  OUTADDRESS                    ; Push address of data
       FFFFFFFF 8F         DD     1420              PUSHL   #-1                           ; Push data pattern indicator
        1E89'CF      02    FB     1426              CALLS   #2,DATA_CHECK                 ; Check the data
                                  142B              $DELTVA_S INADR = OUTADDRESS          ; Delete the section
                  07 50    E8     143A              BLBS    R0,10$                        ; Check the status return
                     50    DD     143D              PUSHL   R0                            ; Push the error code
        1F15'CF      01    FB     143F              CALLS   #1,NONSUB_SSE                 ; Report the failure
                                  1444      10$:
    023A'CF     003A'CF    DE     1444              MOVAL   DGBLSC,SERV_NAME              ; Set the new service name
                                  144B              $DGBLSC_S GSDNAM = GSDNAM             ; Delete the section
                                  145A              FAIL_CHECK SS$_NORMAL                 ; Check the status
                     01    DD     145A                      PUSHL   #SS$_NORMAL
        1D36'CF      01    FB     145C                      CALLS   #1,W^REG_CHECK
    023A'CF     0033'CF    DE     1461              MOVAL   CRMPSC,SERV_NAME              ; Set the new service name
                                  1468
                                  1468      ;+
                                  1468      ;
                                  1468      ; Create a one page PERM local memory SYSTEM P1 PFNMAP section
                                  1468      ;
                                  1468      ;-
                                  1468              NEXT_TEST
                                  1468
                                  1468      STP50:
        0006'CF      32    D0     1468                      MOVL    #50,W^CURRENT_TC
                     00    DD     146D                      PUSHL   #0
        1D2C'CF      01    FB     146F                      CALLS   #1,W^REG_SAVE
                                  1474
    0369'CF    70000000'EF  DE    1474              MOVAL   ^X70000000,INADDRESS          ; Set up INADR
    036D'CF    70000000'EF  DE    147D              MOVAL   ^X70000000,INADDRESS+4
                                  1486              $CRMPSC_S INADR = INADDRESS,-
                                  1486                      RETADR = OUTADDRESS,-
                                  1486                       FLAGS = #SEC$M_PFNMAP!SEC$M_SYSGBL!SEC$M_EXPREG!SEC$M_PERM!SEC$M_G
                                  1486                      GSDNAM = GSDNAM,-
                                  1486                         VBN = VBN,-
                                  1486                      PAGCNT = #1
                                  14B0              FAIL_CHECK SS$_CREATED                ; Check the status return
       00000619 8F         DD     14B0                      PUSHL   #SS$_CREATED
```

```
        1D36'CF   01    FB  14B6                   CALLS   #1,W^REG_CHECK
            0361'CF     DF  14BB          PUSHAL  OUTADDRESS                     ; Push address of data
        FFFFFFFF 8F     DD  14BF          PUSHL   #-1                            ; Push data pattern indicator
        1E89'CF   02    FB  14C5          CALLS   #2,DATA_CHECK                  ; Check the data
                            14CA          $DELTVA_S INADR = OUTADDRESS           ; Delete the section
                07 50   E8  14D9          BLBS    R0,10$                         ; Check the status return
                   50   DD  14DC          PUSHL   R0                             ; Push the error code
        1F15'CF   01    FB  14DE          CALLS   #1,NONSUB_SSE                  ; Report the failure
                            14E3     10$:
023A'CF   003A'CF       DE  14E3          MOVAL   DGBLSC,SERV_NAME               ; Set the new service name
                            14EA          $DGBLSC_S GSDNAM = GSDNAM,-
                            14EA                  FLAGS = #SEC$M_SYSGBL          ; Delete the section
                            14FD          FAIL_CHECK SS$_NORMAL                  ; Check the status
                   01   DD  14FD                   PUSHL   #SS$_NORMAL
        1D36'CF   01    FB  14FF                   CALLS   #1,W^REG_CHECK
                            1504
                            1504     NO_PFN_TEST:
                            1504
023A'CF   0033'CF       DE  1504          MOVAL   CRMPSC,SERV_NAME               ; Set the new service name
```

```
                              150B          .SBTTL  MGBLSC TESTS
                              150B   ;+
                              150B   ;
                              150B   ; $MGBLSC tests
                              150B   ;
                              150B   ; Few steps exist in this section because the system code to create and map
                              150B   ; a section is the same code used for $MGBLSC. Only the MGBLSC unique parts
                              150B   ; are tested.
                              150B   ;
                              150B   ; Test simplest case
                              150B   ;
                              150B   ; Create the simplest section. One page, user, temporary, P0, global, local memory,
                              150B   ; read only section. This will be used as a target section to map to by the
                              150B   ; $MGBLSC.
                              150B   ;
                              150B   ;-
                              150B
                              150B          NEXT_TEST
                              150B
                              150B   STP51:
        0006'CF     33   D0   150B                  MOVL    #51,W^CURRENT_TC
                    00   DD   1510                  PUSHL   #0
        1D2C'CF     01   FB   1512                  CALLS   #1,W^REG_SAVE
                              1517
   0369'CF    0600'CF    DE   1517          MOVAL   READ_BUF,INADDRESS               ; Set up inadr
   036D'CF    0800'CF    DE   151E          MOVAL   READ_BUF+512,INADDRESS+4
                              1525          $CRMPSC_S INADR = INADDRESS,-
                              1525                  CHAN  = SECFAB+FAB$L_STV,-
                              1525                  GSDNAM= GSDNAM,-
                              1525                  FLAGS = #SEC$M_GBL               ; Create a target section
                              1547          FAIL_CHECK SS$_CREATED
      00000619 8F        DD   1547                  PUSHL   #SS$_CREATED
        1D36'CF     01   FB   154D                  CALLS   #1,W^REG_CHECK
             0369'CF     DF   1552          PUSHAL INADDRESS
          FFFFFFFF 8F    DD   1556          PUSHL   #-1
        1E89'CF     02   FB   155C          CALLS   #2,DATA_CHECK                    ; Check the section data
   000A'CF    01DC'CF    DE   1561          MOVAL   UM,MODE                          ; Set the mode
   023A'CF    0041'CF    DE   1568          MOVAL   MGBLSC,SERV_NAME                 ; Set the service name
0800 8F   00   0E00'CF   00   2C   156F     MOVC5   #0,READ_BUF1,#0,#512*4,READ_BUF1 ; Init the second read buffer
             0E00'CF          1578
                    00   DD   157B          PUSHL   #0                               ; Push the argument
        1D2C'CF     01   FB   157D          CALLS   #1,REG_SAVE                      ; Save the register contents
   0369'CF    0E00'CF    DE   1582          MOVAL READ_BUF1,INADDRESS                ; Set the P0 address up
   036D'CF    1000'CF    DE   1589          MOVAL READ_BUF1+512,INADDRESS+4
                              1590          $MGBLSC_S GSDNAM = GSDNAM,-
                              1590                  INADR = INADDRESS
                              15A9          FAIL_CHECK SS$_NORMAL
                    01   DD   15A9                  PUSHL   #SS$_NORMAL
        1D36'CF     01   FB   15AB                  CALLS   #1,W^REG_CHECK
             0369'CF     DF   15B0          PUSHAL INADDRESS
          FFFFFFFF 8F    DD   15B4          PUSHL   #-1
        1E89'CF     02   FB   15BA          CALLS   #2,DATA_CHECK                    ; Check the section data
                              15BF          $DELTVA_S INADR = INADDRESS              ; Clean up
                 07 50   E8   15CE          BLBS    R0,10$                           ; Skip error report if OK
                    50   DD   15D1          PUSHL   R0                               ; Save error code
        1F15'CF     01   FB   15D3          CALLS   #1,NONSUB_SSE                    ; Print the failure
                              15D8   10$:
```

```
                          15D8
                          15D8   ;+
                          15D8   ;
                          15D8   ; Test mapping to a non-existant global section.
                          15D8   ;
                          15D8   ;-
                          15D8
                          15D8           NEXT_TEST
                          15D8
                          15D8   STP52:
     0006'CF     34   D0  15D8                   MOVL    #52,W^CURRENT_TC
                 00   DD  15DD                   PUSHL   #0
     1D2C'CF     01   FB  15DF                   CALLS   #1,W^REG_SAVE
                          15E4
 034A'DF     4C 8F    90  15E4           MOVB    #^A/L/,@<GSDNAM+4>                  ; Futs up the name
                          15EA           $MGBLSC_S GSDNAM = GSDNAM,-
                          15EA                   INADR = INADDRESS
                          1603           FAIL_CHECK SS$_NOSUCHSEC
     00000978 8F      DD  1603                   PUSHL   #SS$_NOSUCHSEC
     1D36'CF     01   FB  1609                   CALLS   #1,W^REG_CHECK
 034A'DF     53 8F    90  160E           MOVB    #^A/S/,@<GSDNAM+4>                  ; Fix up the name
                          1614
                          1614   ;+
                          1614   ;
                          1614   ; Test match control > SEC$K_MATLEQ
                          1614   ;
                          1614   ;-
                          1614
                          1614           NEXT_TEST
                          1614
                          1614   STP53:
     0006'CF     35   D0  1614                   MOVL    #53,W^CURRENT_TC
                 00   DD  1619                   PUSHL   #0
     1D2C'CF     01   FB  161B                   CALLS   #1,W^REG_SAVE
                          1620
                          1620           $MGBLSC_S INADR = INADDRESS,-
                          1620                   GSDNAM = GSDNAM,-
                          1620                   IDENT = AN_ILLEGAL_IDENT
                          163B           FAIL_CHECK SS$_IVSECIDCTL
     000002E4 8F      DD  163B                   PUSHL   #SS$_IVSECIDCTL
     1D36'CF     01   FB  1641                   CALLS   #1,W^REG_CHECK
                          1646
                          1646   ;+
                          1646   ;
                          1646   ; Test match control SEC$K_MATEQU
                          1646   ;
                          1646   ;-
                          1646
                          1646           NEXT_TEST
                          1646
                          1646   STP54:
     0006'CF     36   D0  1646                   MOVL    #54,W^CURRENT_TC
                 00   DD  164B                   PUSHL   #0
     1D2C'CF     01   FB  164D                   CALLS   #1,W^REG_SAVE
                          1652
 023A'CF    003A'CF   DE  1652           MOVAL   DGBLSC,SERV_NAME                    ; Set the service name
                          1659           $DGBLSC_S GSDNAM = GSDNAM                   ; Delete old global section
```

```
                                1668       FAIL_CHECK SS$_NORMAL
                    01    DD    1668                  PUSHL    #SS$_NORMAL
           1D36'CF  01    FB    166A                  CALLS    #1,W^REG_CHECK
     0369'CF  0600'CF     DE    166F       MOVAL    READ_BUF,INADDRESS                 ; Set up inadr
     036D'CF  0800'CF     DE    1676       MOVAL    READ_BUF+512,INADDRESS+4
                                167D       $DELTVA_S INADR = INADDRESS
                 07 50    E8    168C       BLBS     R0,10$                             ; Skip error report if OK
                    50    DD    168F       PUSHL    R0                                 ; Save error code
           1F15'CF  01    FB    1691       CALLS    #1,NONSUB_SSE                      ; Print the failure
                                1696  10$:
   0342'CF    08080000 8F   D0  1696       MOVL     #^X8080000,AN_IDENT+4              ; Set up an ident of MSB in
     023A'CF  0033'CF     DE    169F       MOVAL    CRMPSC,SERV_NAME                   ; Set the service name
                                16A6       $CRMPSC_S INADR = INADDRESS,-
                                16A6                  CHAN   = SECFAB+FAB$L_STV,-
                                16A6                  GSDNAM= GSDNAM,-
                                16A6                  IDENT = AN_IDENT,-
                                16A6                  FLAGS = #SEC$M_GBL               ; Create a target section with
                                16CA                                                   ; an ident specified
                                16CA       FAIL_CHECK SS$_CREATED
           00000619 8F    DD    16CA                  PUSHL    #SS$_CREATED
           1D36'CF  01    FB    16D0                  CALLS    #1,W^REG_CHECK
              0369'CF     DF    16D5       PUSHAL INADDRESS
           FFFFFFFF 8F    DD    16D9       PUSHL    #-1
           1E89'CF  02    FB    16DF       CALLS    #2,DATA_CHECK                      ; Check the section data
     023A'CF  0041'CF     DE    16E4       MOVAL    MGBLSC,SERV_NAME                   ; Set the service name
     0369'CF  0E00'CF     DE    16EB       MOVAL READ_BUF1,INADDRESS                   ; Set the P0 address up
     036D'CF  1000'CF     DE    16F2       MOVAL READ_BUF1+512,INADDRESS+4
           033E'CF  01    D0    16F9       MOVL     #SEC$K_MATEQU,AN_IDENT             ; Set the match criteria
                                16FE       $MGBLSC_S GSDNAM = GSDNAM,-
                                16FE                    IDENT = AN_IDENT,-
                                16FE                    INADR = INADDRESS
                                1719       FAIL_CHECK SS$_NORMAL
                    01    DD    1719                  PUSHL    #SS$_NORMAL
           1D36'CF  01    FB    171B                  CALLS    #1,W^REG_CHECK
              0369'CF     DF    1720       PUSHAL INADDRESS
           FFFFFFFF 8F    DD    1724       PUSHL    #-1
           1E89'CF  02    FB    172A       CALLS    #2,DATA_CHECK                      ; Check the section data
                                172F       $DELTVA_S INADR = INADDRESS                 ; Clean up
                 07 50    E8    173E       BLBS     R0,20$                             ; Skip error report if OK
                    50    DD    1741       PUSHL    R0                                 ; Save error code
           1F15'CF  01    FB    1743       CALLS    #1,NONSUB_SSE                      ; Print the failure
                                1748  20$:
                                1748
                                1748  ;+
                                1748  ;
                                1748  ; Test match control SEC$K_MATEQL failure to map cause of minor ident mismatch.
                                1748  ;
                                1748  ;-
                                1748
                                1748       NEXT_TEST
                                1748
                                1748  STP55:
           0006'CF  37    D0    1748                  MOVL     #55,W^CURRENT_TC
                    00    DD    174D                  PUSHL    #0
           1D2C'CF  01    FB    174F                  CALLS    #1,W^REG_SAVE
                                1754
   0342'CF    08040000 8F   D0  1754       MOVL     #^X8040000,AN_IDENT+4              ; Set up an ident with minor mismatc
```

```
                          175D  1708          $MGBLSC_S INADR = INADDRESS,-
                          175D  1709                  GSDNAM = GSDNAM,-
                          175D  1710                   IDENT = AN_IDENT
                          1778  1711          FAIL_CHECK SS$_NOSUCHSEC
        00000978 8F   DD  1778                         PUSHL   #SS$_NOSUCHSEC
        1D36'CF   01  FB  177E                         CALLS   #1,W^REG_CHECK
                          1783  1712
                          1783  1713  ;+
                          1783  1714  ;
                          1783  1715  ; Test match control SEC$K_MATEQL failure to map cause of major ident mismatch.
                          1783  1716  ;
                          1783  1717  ;-
                          1783  1718
                          1783  1719          NEXT_TEST
                          1783
                          1783        STP56:
        0006'CF   38  D0  1783                         MOVL    #56,W^CURRENT_TC
                  00  DD  1788                         PUSHL   #0
        1D2C'CF   01  FB  178A                         CALLS   #1,W^REG_SAVE
                          178F  1720
0342'CF 04080000 8F   D0  178F  1721          MOVL    #^X4080000,AN_IDENT+4          ; Set up an ident with major
                          1798  1722                                                 ; ident less
                          1798  1723          $MGBLSC_S INADR = INADDRESS,-
                          1798  1724                  GSDNAM = GSDNAM,-
                          1798  1725                   IDENT = AN_IDENT
                          17B3  1726          FAIL_CHECK SS$_NOSUCHSEC
        00000978 8F   DD  17B3                         PUSHL   #SS$_NOSUCHSEC
        1D36'CF   01  FB  17B9                         CALLS   #1,W^REG_CHECK
                          17BE  1727
                          17BE  1728  ;+
                          17BE  1729  ;
                          17BE  1730  ; Test match control SEC$K_MATLEQ map with minor ident <.
                          17BE  1731  ;
                          17BE  1732  ;-
                          17BE  1733
                          17BE  1734          NEXT_TEST
                          17BE
                          17BE        STP57:
        0006'CF   39  D0  17BE                         MOVL    #57,W^CURRENT_TC
                  00  DD  17C3                         PUSHL   #0
        1D2C'CF   01  FB  17C5                         CALLS   #1,W^REG_SAVE
                          17CA  1735
        033E'CF   02  D0  17CA  1736          MOVL    #SEC$K_MATLEQ,AN_IDENT         ; Set the match criteria
0342'CF 0803FFFF 8F   D0  17CF  1737          MOVL    #^X803FFFF,AN_IDENT+4          ; Set up an ident with minor
                          17D8  1738                                                 ; ident less
                          17D8  1739          $MGBLSC_S INADR = INADDRESS,-
                          17D8  1740                  GSDNAM = GSDNAM,-
                          17D8  1741                   IDENT = AN_IDENT
                          17F3  1742          FAIL_CHECK SS$_NORMAL
                  01  DD  17F3                         PUSHL   #SS$_NORMAL
        1D36'CF   01  FB  17F5                         CALLS   #1,W^REG_CHECK
           0369'CF    DF  17FA  1743          PUSHAL INADDRESS
        FFFFFFFF 8F   DD  17FE  1744          PUSHL   #-1
        1E89'CF   02  FB  1804  1745          CALLS   #2,DATA_CHECK                  ; Check the section data
                          1809  1746          $DELTVA_S INADR = INADDRESS            ; Clean up
               07 50  E8  1818  1747          BLBS    R0,10$                         ; Skip error report if OK
                  50  DD  181B  1748          PUSHL   R0                             ; Save error code
```

```
          1F15'CF   01    FB 181D            CALLS   #1,NONSUB_SSE                   ; Print the failure
                             1822    10$:
                             1822
                             1822    ;+
                             1822    ;
                             1822    ; Test match control SEC$K_MATLEQ map with minor ident =.
                             1822    ;
                             1822    ;-
                             1822
                             1822            NEXT_TEST
                             1822
                             1822    STP58:
          0006'CF   3A    D0 1822                    MOVL    #58,W^CURRENT_TC
                    00    DD 1827                    PUSHL   #0
          1D2C'CF   01    FB 1829                    CALLS   #1,W^REG_SAVE
                             182E
  0342'CF   08040000 8F   D0 182E            MOVL    #^X8040000,AN_IDENT+4           ; Set up an ident with minor
                             1837                                                    ; ident equal
                             1837            $MGBLSC_S INADR = INADDRESS,-
                             1837                    GSDNAM = GSDNAM,-
                             1837                    IDENT = AN_IDENT
                             1852            FAIL_CHECK SS$_NORMAL
                    01    DD 1852                    PUSHL   #SS$_NORMAL
          1D36'CF   01    FB 1854                    CALLS   #1,W^REG_CHECK
             0369'CF      DF 1859            PUSHAL INADDRESS
           FFFFFFFF 8F    DD 185D            PUSHL   #-1
          1E89'CF   02    FB 1863            CALLS   #2,DATA_CHECK                   ; Check the section data
                             1868            $DELTVA_S INADR = INADDRESS             ; Clean up
                 07 50    E8 1877            BLBS    R0,10$                          ; Skip error report if OK
                    50    DD 187A            PUSHL   R0                              ; Save error code
          1F15'CF   01    FB 187C            CALLS   #1,NONSUB_SSE                   ; Print the failure
                             1881    10$:
                             1881    ;+
                             1881    ;
                             1881    ; Test the offset definitions and the _G format of the system service.
                             1881    ;
                             1881    ;-
                             1881
                             1881            NEXT_TEST
                             1881
                             1881    STP59:
          0006'CF   3B    D0 1881                    MOVL    #59,W^CURRENT_TC
                    00    DD 1886                    PUSHL   #0
          1D2C'CF   01    FB 1888                    CALLS   #1,W^REG_SAVE
             0040 8F      BB 188D            PUSHR   #^M<R6>                         ; Save the register contents
          56  02EA'CF     DE 1891            MOVAL   MGBLSCG,R6                      ; Save the parameter list address
       04 A6  0369'CF     DE 1896            MOVAL   INADDRESS,MGBLSC$_INADR(R6)     ; Set the parameters up
       08 A6  0361'CF     DE 189C            MOVAL   OUTADDRESS,MGBLSC$_RETADR(R6)
          0C A6   03      D0 18A2            MOVL    #PSL$C_USER,MGBLSC$_ACMODE(R6)
                  01      D0 18A6            MOVL    #SEC$M_GBL,-
               10 A6         18A8                    MGBLSC$_FLAGS(R6)
       14 A6  0346'CF     DE 18AA            MOVAL   GSDNAM,MGBLSC$_GSDNAM(R6)
       18 A6  033E'CF     DE 18B0            MOVAL   AN_IDENT,MGBLSC$_IDENT(R6)
          1C A6   01      D0 18B6            MOVL    #1,MGBLSC$_RELPAG(R6)
             0040 8F      BA 18BA            POPR    #^M<R6>                         ; Restore register
                             18BE            $MGBLSC_G MGBLSCG
                             18C7            FAIL_CHECK SS$_NORMAL                   ; Check the status
```

```
                  01   DD  18C7             PUSHL   #SS$_NORMAL
         1D36'CF  01   FB  18C9             CALLS   #1,W^REG_CHECK
            0369'CF    DF  18CE  1794       PUSHAL  INADDRESS                          ; Check the data
          FFFFFF7F 8F  DD  18D2  1795       PUSHL   #<-1-128>
         1E89'CF  02   FB  18D8  1796       CALLS   #2,DATA_CHECK
                           18DD  1797       $DELTVA_S INADR = INADDRESS                ; Clean up
               07 50   E8  18EC  1798       BLBS    R0,10$                             ; Skip error report if OK
                  50   DD  18EF  1799       PUSHL   R0                                 ; Save error code
         1F15'CF  01   FB  18F1  1800       CALLS   #1,NONSUB_SSE                      ; Print the failure
                           18F6  1801 10$:
                           18F6  1802
                           18F6  1803 ;+
                           18F6  1804 ;
                           18F6  1805 ; Create a Supervisor section
                           18F6  1806 ;
                           18F6  1807 ;-
                           18F6  1808
                           18F6  1809       NEXT_TEST
                           18F6
                           18F6       STP60:
         0006'CF  3C   D0  18F6                     MOVL    #60,W^CURRENT_TC
                  00   DD  18FB                     PUSHL   #0
         1D2C'CF  01   FB  18FD                     CALLS   #1,W^REG_SAVE
                           1902  1810
 000A'CF   01E8'CF     DE  1902  1811       MOVAL   SM,MODE                    ; Set the mode ID address
                           1909  1812
                           1909  1813 ;+
                           1909  1814 ;
                           1909  1815 ; Create a Executive section
                           1909  1816 ;
                           1909  1817 ;-
                           1909  1818
                           1909  1819       NEXT_TEST
                           1909
                           1909       STP61:
         0006'CF  3D   D0  1909                     MOVL    #61,W^CURRENT_TC
                  00   DD  190E                     PUSHL   #0
         1D2C'CF  01   FB  1910                     CALLS   #1,W^REG_SAVE
                           1915  1820
 000A'CF   01F5'CF     DE  1915  1821       MOVAL   EM,MODE                    ; Set the mode ID address
                           191C  1822
                           191C  1823 ;+
                           191C  1824 ;
                           191C  1825 ; Create a Kernal section
                           191C  1826 ;
                           191C  1827 ;-
                           191C  1828
                           191C  1829       NEXT_TEST
                           191C
                           191C       STP62:
         0006'CF  3E   D0  191C                     MOVL    #62,W^CURRENT_TC
                  00   DD  1921                     PUSHL   #0
         1D2C'CF  01   FB  1923                     CALLS   #1,W^REG_SAVE
                           1928  1830
 000A'CF   0206'CF     DE  1928  1831       MOVAL   KM,MODE                    ; Set the mode ID address
 023A'CF   003A'CF     DE  192F  1832       MOVAL   DGBLSC,SERV_NAME                   ; Set the service name
                           1936  1833       $DGBLSC_S GSDNAM = GSDNAM                  ; Clean up the mess
```

```
                         1945 1834          FAIL_CHECK SS$_NORMAL                    ; Check the status
                 01  DD  1945                       PUSHL   #SS$_NORMAL
        1D36'CF  01  FB  1947                       CALLS   #1,W^REG_CHECK
  0369'CF   0600'CF  DE  194C 1835          MOVAL   READ_BUF,INADDRESS               ; Set up inadr
  036D'CF   0800'CF  DE  1953 1836          MOVAL   READ_BUF+512,INADDRESS+4
                         195A 1837          $DELTVA_S INADR = INADDRESS
              07 50  E8  1969 1838          BLBS    R0,10$                           ; Skip error report if OK
                 50  DD  196C 1839          PUSHL   R0                               ; Save error code
        1F15'CF  01  FB  196E 1840          CALLS   #1,NONSUB_SSE                    ; Print the failure
                         1973 1841 10$:
```

```
                              1973          .SBTTL  UPDSEC TESTS
                              1973    ;+
                              1973    ;
                              1973    ; $UPDSEC tests
                              1973    ;
                              1973    ; Test simplest case SS$_NOTMODIFIED
                              1973    ;
                              1973    ;-
                              1973
                              1973          NEXT_TEST
                              1973
                              1973    STP63:
         0006'CF    3F    D0  1973                  MOVL    #63,W^CURRENT_TC
                    00    DD  1978                  PUSHL   #0
         1D2C'CF    01    FB  197A                  CALLS   #1,W^REG_SAVE
                              197F
  000A'CF    01DC'CF      DE  197F          MOVAL   UM,MODE                                 ; Set the mode
  023A'CF    0033'CF      DE  1986          MOVAL   CRMPSC,SERV_NAME                        ; Set the service name
  0369'CF    0600'CF      DE  198D          MOVAL   READ_BUF,INADDRESS                      ; Set the map limits
  036D'CF    0BFF'CF      DE  1994          MOVAL   READ_BUF+<512*3>-1,INADDRESS+4
                              199B          $CRMPSC_S CHAN = SECFAB+FAB$L_STV,-
                              199B                  FLAGS = #SEC$M_WRT,-
                              199B                  INADR = INADDRESS,-
                              199B                  RETADR = OUTADDRESS,-
                              199B                  PAGCNT = #3
                              19BF          FAIL_CHECK SS$_CREATED
      00000619 8F         DD  19BF                  PUSHL   #SS$_CREATED
      1D36'CF    01       FB  19C5                  CALLS   #1,W^REG_CHECK
  023A'CF    0048'CF      DE  19CA          MOVAL   UPDSEC,SERV_NAME                        ; Set the service name
                              19D1          $UPDSEC_S INADR = INADDRESS,-
                              19D1                  RETADR = OUTADDRESS,-
                              19D1                  UPDFLG = #0
                              19E8          FAIL_CHECK SS$_NOTMODIFIED
      00000659 8F         DD  19E8                  PUSHL   #SS$_NOTMODIFIED
      1D36'CF    01       FB  19EE                  CALLS   #1,W^REG_CHECK
                              19F3    ;+
                              19F3    ;
                              19F3    ; Test EFN with page modification
                              19F3    ;
                              19F3    ;-
                              19F3          NEXT_TEST
                              19F3
                              19F3    STP64:
  0006'CF    00000040 8F  D0  19F3                  MOVL    #64,W^CURRENT_TC
                    00    DD  19FC                  PUSHL   #0
      1D2C'CF    01       FB  19FE                  CALLS   #1,W^REG_SAVE
                              1A03
         0800'CF          D4  1A03          CLRL    READ_BUF + 512                          ; Modify a page
                              1A07          $UPDSEC_S INADR = INADDRESS,-
                              1A07                  RETADR = OUTADDRESS,-
                              1A07                  UPDFLG = #0,-
                              1A07                  EFN = #1,-
                              1A07                  IOSB = UPD_IOSB
                              1A22          FAIL_CHECK SS$_NORMAL
                    01    DD  1A22                  PUSHL   #SS$_NORMAL
      1D36'CF       01    FB  1A24                  CALLS   #1,W^REG_CHECK
              2E 50       E9  1A29          BLBC    R0,10$                                  ; Skip wait if failure occured
```

```
                                 1A2C   1884            $WAITFR_S EFN = #1                      ; Wait for I/O complete
            50   02AE'CF    3C   1A35   1885            MOVZWL  UPD_IOSB,R0                     ; Get the return status code
                                 1A3A   1886            FAIL_CHECK SS$_NORMAL
                      01    DD   1A3A                           PUSHL   #SS$_NORMAL
            1D36'CF   01    FB   1A3C                           CALLS   #1,W^REG_CHECK
                                 1A41   1887            $DELTVA_S INADR = INADDRESS             ; Unmap the section
                   07 50    E8   1A50   1888            BLBS    R0,10$                          ; Skip error report if OK
                      50    DD   1A53   1889            PUSHL   R0                              ; Save error code
            1F15'CF   01    FB   1A55   1890            CALLS   #1,NONSUB_SSE                   ; Print the failure
                                 1A5A   1891 10$:
     023A'CF   0033'CF      DE   1A5A   1892            MOVAL   CRMPSC,SERV_NAME                ; Set the service name
                                 1A61   1893            $CRMPSC_S CHAN = SECFAB+FAB$L_STV,-     ; Remap the section
                                 1A61   1894                      FLAGS = #SEC$M_WRT,-
                                 1A61   1895                      INADR = INADDRESS,-
                                 1A61   1896                      PAGCNT = #3
                                 1A83   1897            FAIL_CHECK SS$_CREATED
              00000619 8F   DD   1A83                           PUSHL   #SS$_CREATED
            1D36'CF   01    FB   1A89                           CALLS   #1,W^REG_CHECK
     023A'CF   0048'CF      DE   1A8E   1898            MOVAL   UPDSEC,SERV_NAME                ; Set the service name
               0800'CF      D5   1A95   1899            TSTL    READ_BUF + 512                  ; Check for the modification
                      09    13   1A99   1900            BEQL    20$
               006C'CF      DF   1A9B   1901            PUSHAL  UPDSEC_FAILED                   ; Push the failure message
            1D9A'CF   01    FB   1A9F   1902            CALLS   #1,PRINT_FAIL                   ; And print it
                                 1AA4   1903 20$:
                                 1AA4   1904 ;+
                                 1AA4   1905 ;
                                 1AA4   1906 ; Test ASTADR and ASTPRM arguments
                                 1AA4   1907 ;
                                 1AA4   1908 ;-
                                 1AA4   1909            NEXT_TEST
                                 1AA4
                                 1AA4        STP65:
 0006'CF    00000041 8F     D0   1AA4                           MOVL    #65,W^CURRENT_TC
                      00    DD   1AAD                           PUSHL   #0
            1D2C'CF   01    FB   1AAF                           CALLS   #1,W^REG_SAVE
                                 1AB4   1910
 0800'CF    FFFFFF7F 8F     D0   1AB4   1911            MOVL    #<-1-128>,READ_BUF+512          ; Modify a page
                                 1ABD   1912            $SETAST_S ENBFLG = #0                   ; Disable AST's for error checking
                                 1AC6   1913            $UPDSEC_S INADR = INADDRESS,-
                                 1AC6   1914                      RETADR = OUTADDRESS,-
                                 1AC6   1915                      UPDFLG = #0,-
                                 1AC6   1916                         EFN = #1,-
                                 1AC6   1917                      ASTADR = 20$,-
                                 1AC6   1918                      ASTPRM = #3,-
                                 1AC6   1919                        IOSB = UPD_IOSB
                                 1AE5   1920            FAIL_CHECK SS$_NORMAL
                      01    DD   1AE5                           PUSHL   #SS$_NORMAL
            1D36'CF   01    FB   1AE7                           CALLS   #1,W^REG_CHECK
                                 1AEC   1921            $SETAST_S ENBFLG = #1                   ; OK checking done go ahead and AST
                   03 50    E8   1AF5   1922            BLBS    R0,10$                          ; Skip to wait if no failure occured
                    00A2    31   1AF8   1923            BRW     50$                             ; Else skip the wait
                                 1AFB   1924 10$:
                                 1AFB   1925            $HIBER_S                                ; Wait for AST
                    0098    31   1B02   1926            BRW     50$                             ; Skip over AST routine
                                 1B05   1927 20$:
                         0FFC    1B05   1928            .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                      00    DD   1B07   1929            PUSHL   #0                              ; Push a dummy parameter
```

```
        1D2C'CF   01   FB  1B09  1930         CALLS   #1,REG_SAVE                      ; Save the registers
          03   04 AC   D1  1B0E  1931         CMPL    4(AP),#3                         ; Did the parameter get here?
                  0E   13  1B12  1932         BEQL    30$                              ; BR if yes
             04 AC     DD  1B14  1933         PUSHL   4(AP)                            ; Push received
                  03   DD  1B17  1934         PUSHL   #3                               ; Push expected
            0289'CF    DF  1B19  1935         PUSHAL  DATA                             ; Push string address
        1D9A'CF   03   FB  1B1D  1936         CALLS   #3,PRINT_FAIL                    ; Report the failure
                           1B22  1937 30$:
     50    02AE'CF     3C  1B22  1938         MOVZWL  UPD_IOSB,R0                      ; Get the return status code
                           1B27  1939         FAIL_CHECK SS$_NORMAL
                  01   DD  1B27                       PUSHL   #SS$_NORMAL
        1D36'CF   01   FB  1B29                       CALLS   #1,W^REG_CHECK
                           1B2E  1940         $DELTVA_S INADR = INADDRESS              ; Unmap the section
               07 50   E8  1B3D  1941         BLBS    R0,40$                           ; Skip error report if OK
                  50   DD  1B40  1942         PUSHL   R0                               ; Save error code
        1F15'CF   01   FB  1B42  1943         CALLS   #1,NONSUB_SSE                    ; Print the failure
                           1B47  1944 40$:
  023A'CF    0033'CF   DE  1B47  1945         MOVAL   CRMPSC,SERV_NAME                 ; Set the service name
                           1B4E  1946         $CRMPSC_S CHAN = SECFAB+FAB$L_STV,-      ; Remap the section
                           1B4E  1947                 FLAGS = #SEC$M_WRT,-
                           1B4E  1948                 INADR = INADDRESS,-
                           1B4E  1949                 PAGCNT = #3
                           1B70  1950         FAIL_CHECK SS$_CREATED
         00000619 8F   DD  1B70                       PUSHL   #SS$_CREATED
        1D36'CF   01   FB  1B76                       CALLS   #1,W^REG_CHECK
  023A'CF    0048'CF   DE  1B7B  1951         MOVAL   UPDSEC,SERV_NAME                 ; Set the service name
            0369'CF    DF  1B82  1952         PUSHAL  INADDRESS                        ; Check the data
         FFFFFFFF 8F   DD  1B86  1953         PUSHL   #-1
        1E89'CF   02   FB  1B8C  1954         CALLS   #2,DATA_CHECK
                           1B91  1955         $WAKE_S                                  ; Carry on
                  04       1B9C  1956         RET
                           1B9D  1957 50$:
                           1B9D  1958 ;+
                           1B9D  1959 ;
                           1B9D  1960 ; Test the offset definitions and the _G format of the system service.
                           1B9D  1961 ;
                           1B9D  1962 ;-
                           1B9D  1963
                           1B9D  1964         NEXT_TEST
                           1B9D
                           1B9D       STP66:
  0006'CF   00000042 8F    D0  1B9D                       MOVL    #66,W^CURRENT_TC
                  00   DD  1BA6                       PUSHL   #0
        1D2C'CF   01   FB  1BA8                       CALLS   #1,W^REG_SAVE
             0040 8F   BB  1BAD  1965         PUSHR   #^M<R6>                          ; Save the register contents
     56    031A'CF     DE  1BB1  1966         MOVAL   UPDSECG,R6                       ; Save the parameter list address
  0600'CF   FFFFFFFF 8F    D0  1BB6  1967     MOVL    #-1,READ_BUF                     ; Modify a page
     04 A6   0369'CF   DE  1BBF  1968         MOVAL   INADDRESS,UPDSEC$_INADR(R6)      ; Set the parameters up
     08 A6   0361'CF   DE  1BC5  1969         MOVAL   OUTADDRESS,UPDSEC$_RETADR(R6)
          0C A6   03   D0  1BCB  1970         MOVL    #PSL$C_USER,UPDSEC$_ACMODE(R6)
          10 A6   01   D0  1BCF  1971         MOVL    #1,UPDSEC$_UPDFLG(R6)
          14 A6   01   D0  1BD3  1972         MOVL    #EFN1,UPDSEC$_EFN(R6)
     18 A6   02AE'CF   DE  1BD7  1973         MOVAL   UPD_IOSB,UPDSEC$_IOSB(R6)
     1C A6   1C21'CF   DE  1BDD  1974         MOVAL   20$,UPDSEC$_ASTADR(R6)
  20 A6   00000064 8F  D0  1BE3  1975         MOVL    #100,UPDSEC$_ASTPRM(R6)
             0040 8F   BA  1BEB  1976         POPR    #^M<R6>                          ; Restore register
                           1BEF  1977         $SETAST_S ENBFLG = #0                    ; Disable AST's for fail check
```

```
                          1BF8            $UPDSEC_G UPDSECG
                          1C01            FAIL_CHECK SS$_NORMAL                 ; Check the status
              01    DD    1C01                      PUSHL   #SS$_NORMAL
      1D36'CF 01    FB    1C03                      CALLS   #1,W^REG_CHECK
                          1C08            $SETAST_S ENBFLG = #1                 ; OK done checking go ahead and AST
           03 50    E8    1C11            BLBS    R0,10$                        ; BR if service request OK...
            004F    31    1C14            BRW     50$                           ; ...else skip the I/O wait
                          1C17    10$:
                          1C17            $HIBER_S                              ; Wait for I/O complete
            0045    31    1C1E            BRW     50$                           ; Skip the AST routine
                          1C21    20$:
            0000          1C21            .WORD   0                             ; AST entry point for $UPDSEC
              00    DD    1C23            PUSHL   #0                            ; Push a dummy argument
      1D2C'CF 01    FB    1C25            CALLS   #1,REG_SAVE                   ; Save the register contents
00000064 8F  04 AC  D1    1C2A            CMPL    4(AP),#100                    ; Correct AST parmeter?
              12    13    1C32            BEQL    30$                           ; BR if OK
           04 AC    DD    1C34            PUSHL   4(AP)                         ; Push received
     00000064 8F    DD    1C37            PUSHL   #100                          ; Push expected
         0289'CF    DF    1C3D            PUSHAL  DATA                          ; Push string address
      1D9A'CF 03    FB    1C41            CALLS   #3,PRINT_FAIL                 ; Report the failure
                          1C46    30$:
      0F 02AE'CF    E8    1C46            BLBS    UPD_IOSB,40$                  ; Check IO status returned
         02AE'CF    DD    1C4B            PUSHL   UPD_IOSB                      ; Push received
              01    DD    1C4F            PUSHL   #SS$_NORMAL                   ; Push expected
         0295'CF    DF    1C51            PUSHAL  STATUS                        ; Push string address
      1D9A'CF 03    FB    1C55            CALLS   #3,PRINT_FAIL                 ; Report the failure
                          1C5A    40$:
                          1C5A            $WAKE_S                               ; Exit the HIBER
              04          1C65            RET
                          1C66    50$:
                          1C66            $DELTVA_S INADR = INADDRESS           ; Delete the address space
           07 50    E8    1C75            BLBS    R0,60$                        ; Skip error report if OK
              50    DD    1C78            PUSHL   R0                            ; Save error code
      1F15'CF 01    FB    1C7A            CALLS   #1,NONSUB_SSE                 ; Print the failure
                          1C7F    60$:
                          1C7F
                          1C7F    ;+
                          1C7F    ;
                          1C7F    ; Create a Supervisor section
                          1C7F    ;
                          1C7F    ;-
                          1C7F
                          1C7F            NEXT_TEST
                          1C7F
                          1C7F    STP67:
0006'CF  00000043 8F  D0  1C7F                      MOVL    #67,W^CURRENT_TC
              00    DD    1C88                      PUSHL   #0
      1D2C'CF 01    FB    1C8A                      CALLS   #1,W^REG_SAVE
                          1C8F
000A'CF   01E8'CF   DE    1C8F            MOVAL   SM,MODE                       ; Set the mode ID address
                          1C96
                          1C96    ;+
                          1C96    ;
                          1C96    ; Create a Executive section
                          1C96    ;
                          1C96    ;-
                          1C96
```

```
                               1C96  2028          NEXT_TEST
                               1C96
                               1C96        STP68:
    0006'CF   00000044 8F   D0  1C96                    MOVL    #68,W^CURRENT_TC
                       00   DD  1C9F                    PUSHL   #0
           1D2C'CF     01   FB  1CA1                    CALLS   #1,W^REG_SAVE
                               1CA6  2029
      000A'CF   01F5'CF     DE  1CA6  2030          MOVAL   EM,MODE                 ; Set the mode ID address
                               1CAD  2031
                               1CAD  2032  ;+
                               1CAD  2033  ;
                               1CAD  2034  ; Create a Kernal section
                               1CAD  2035  ;
                               1CAD  2036  ;-
                               1CAD  2037
                               1CAD  2038          NEXT_TEST
                               1CAD
                               1CAD        STP69:
    0006'CF   00000045 8F   D0  1CAD                    MOVL    #69,W^CURRENT_TC
                       00   DD  1CB6                    PUSHL   #0
           1D2C'CF     01   FB  1CB8                    CALLS   #1,W^REG_SAVE
                               1CBD  2039
      000A'CF   0206'CF     DE  1CBD  2040          MOVAL   KM,MODE                 ; Set the mode ID address
                               1CC4  2041
```

```
                                   1CC4        .SBTTL  DGBLSC TESTS
                                   1CC4  ;+
                                   1CC4  ;
                                   1CC4  ; $DGBLSC tests
                                   1CC4  ;
                                   1CC4  ; Few cases exist in this section because most of the cases are tested in
                                   1CC4  ; the $CRMPSC section of this program.
                                   1CC4  ;
                                   1CC4  ; Test SS$_NOSUCHSEC
                                   1CC4  ;
                                   1CC4  ;-
                                   1CC4
                                   1CC4        NEXT_TEST
                                   1CC4
                                   1CC4  STP70:
     0006'CF   00000046 8F   D0    1CC4              MOVL    #70,W^CURRENT_TC
                         00   DD    1CCD              PUSHL   #0
          1D2C'CF        01   FB    1CCF              CALLS   #1,W^REG_SAVE
                                   1CD4
     000A'CF    01DC'CF       DE    1CD4        MOVAL   UM,MODE                        ; Set the mode
     023A'CF    003A'CF       DE    1CDB        MOVAL   DGBLSC,SERV_NAME               ; Set the service name
                                   1CE2        $DGBLSC_S GSDNAM = GSD_LOGNAM          ; Try a non-existent section
                                   1CF1        FAIL_CHECK SS$_NOSUCHSEC               ; Check failure code
              00000978 8F   DD    1CF1              PUSHL   #SS$_NOSUCHSEC
          1D36'CF        01   FB    1CF7              CALLS   #1,W^REG_CHECK
                                   1CFC        TEST_END
                    0282'CF   DD    1CFC              PUSHL   W^TMD_ADDR
                    027E'CF   DD    1D00              PUSHL   W^TMN_ADDR
                         02   DD    1D04              PUSHL   #2
                    027A'CF   DD    1D06              PUSHL   W^MOD_MSG_CODE
          00000000'GF    04   FB    1D0A              CALLS   #$$T1,G^LIB$SIGNAL
 027A'CF    01    1C     01   F0    1D11              INSV    #1,#STS$V_INHIB_MSG,#1,W^MOD_MSG_CODE
                    027A'CF   DD    1D18              PUSHL   W^MOD_MSG_CODE
          00000000'GF    01   FB    1D1C              CALLS   #1,G^SYS$EXIT
                                   1D23        $EXIT_S
```

```
                                  1D2C        .SBTTL REG_SAVE
                                  1D2C ;++
                                  1D2C ; FUNCTIONAL DESCRIPTION:
                                  1D2C ;        Subroutine to save R2-R11 in the register save location.
                                  1D2C ;
                                  1D2C ; CALLING SEQUENCE:
                                  1D2C ;        PUSHL   #0                  ; save a dummy parameter
                                  1D2C ;        CALLS   #1,W^REG_SAVE       ; save R2-R11
                                  1D2C ;
                                  1D2C ; INPUT PARAMETERS:
                                  1D2C ;        NONE
                                  1D2C ;
                                  1D2C ; OUTPUT PARAMETERS:
                                  1D2C ;        NONE
                                  1D2C ;
                                  1D2C ;--
                                  1D2C
                                  1D2C REG_SAVE:
                             0FFC 1D2C        .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
         023E'CF   14 AD   28   28 1D2E        MOVC3   #4*10,^X14(FP),W^REG_SAVE_AREA  ; save the registers in the program
                               04 1D35        RET
```

```
                               1D36          .SBTTL  REG_CHECK
                               1D36  ;++
                               1D36  ; FUNCTIONAL DESCRIPTION:
                               1D36  ;       Subroutine to test R0 & R2-R11 for proper content after a service
                               1D36  ;       execution. A snapshot is taken by the REG_SAVE routine at the
                               1D36  ;       beginning of each step and this routine is executed after the
                               1D36  ;       services have been executed.
                               1D36  ;
                               1D36  ; CALLING SEQUENCE:
                               1D36  ;       PUSHL   #SS$_XXXXXX      ; push expected R0 contents
                               1D36  ;       CALLS   #1,W^REG_CHECK   ; execute this routine
                               1D36  ;
                               1D36  ; INPUT PARAMETERS:
                               1D36  ;       expected R0 contents on the stack
                               1D36  ;
                               1D36  ; OUTPUT PARAMETERS:
                               1D36  ;       possible error messages printed using $PUTMSG
                               1D36  ;
                               1D36  ;--
                               1D36
                               1D36  REG_CHECK:
                       0FFC    1D36          .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
          50    04 AC    D1    1D38          CMPL    4(AP),R0                              ; is this the right fail code?
                   1C    13    1D3C          BEQL    10$                                   ; br if yes
                   50    DD    1D3E          PUSHL   R0                                    ; push received data
                04 AC    DD    1D40          PUSHL   4(AP)                                 ; push expected data
             0214'CF    DF    1D43          PUSHAL  W^EXP                                 ; push the string variable
      1D9A'CF   03    FB    1D47          CALLS   #3,W^PRINT_FAIL                       ; print the error message
    0282'CF   002C'CF    DE    1D4C          MOVAL   W^TEST_MOD_FAIL,W^TMD_ADDR            ; set failure message address
027A'CF   03   00   02    F0    1D53          INSV    #ERROR,#0,#3,W^MOD_MSG_CODE           ; set severity code
                               1D5A  10$:
  023E'CF    14 AD    28   29    1D5A          CMPC3   #4*10,^X14(FP),W^REG_SAVE_AREA        ; check all but R0
                   36    13    1D61          BEQL    20$                                   ; br if O.K.
  56   53   0000023E'8F    C3    1D63          SUBL3   #REG_SAVE_AREA,R3,R6                  ; calculate the register number
              56   04    C6    1D6B          DIVL2   #4,R6
     000E'CF   56   02    81    1D6E          ADDB3   #^X2,R6,W^REGNUM                      ; put it in the string
              51   03    CA    1D74          BICL2   #3,R1                                 ; backup to register boundrys
              53   03    CA    1D77          BICL2   #3,R3
             000E'CF    DD    1D7A          PUSHL   W^REGNUM                                  ; push register number
                   61    DD    1D7E          PUSHL   (R1)                                  ; push received data
                   63    DD    1D80          PUSHL   (R3)                                  ; push expected data
             0268'CF    DF    1D82          PUSHAL  W^REG                                 ; set string pntr param.
      1D9A'CF   04    FB    1D86          CALLS   #4,W^PRINT_FAIL                       ; print the error message
    0282'CF   002C'CF    DE    1D8B          MOVAL   W^TEST_MOD_FAIL,W^TMD_ADDR            ; set failure message address
027A'CF   03   00   02    F0    1D92          INSV    #ERROR,#0,#3,W^MOD_MSG_CODE           ; set severity code
                               1D99  20$:
                         04    1D99          RET
```

```
                         1D9A         .SBTTL  PRINT_FAIL
                         1D9A  ;++
                         1D9A  ; FUNCTIONAL DESCRIPTION:
                         1D9A  ;      Subroutine to report failures using $PUTMSG
                         1D9A  ;
                         1D9A  ; CALLING SEQUENCE:
                         1D9A  ; Mode  #1     PUSHL EXPECTED  Mode    #2      PUSHL REG_NUMBER
                         1D9A  ;              PUSHL RECEIVED                  PUSHL EXPECTED
                         1D9A  ;              PUSHAL STRING_VAR               PUSHL RECEIVED
                         1D9A  ;              CALLS #3,W^PRINT_FAIL           PUSHAL STRING_VAR
                         1D9A  ;                                              CALLS #4,W^PRINT_FAIL
                         1D9A  ; Mode  #3     PUSHAL STRING_VAR
                         1D9A  ;              CALLS #1,W^PRINT_FAIL
                         1D9A  ;
                         1D9A  ; INPUT PARAMETERS:
                         1D9A  ;      listed above
                         1D9A  ;
                         1D9A  ; OUTPUT PARAMETERS:
                         1D9A  ;      an error message is printed using $PUTMSG
                         1D9A  ;
                         1D9A  ;--
                         1D9A
                         1D9A  PRINT_FAIL:
                  003C   1D9A         .WORD   ^M<R2,R3,R4,R5>
                         1D9C         $FAO_S  CTRSTR = CS1,-
                         1D9C                 OUTLEN = MESSAGEL,-
                         1D9C                 OUTBUF = MSGL,-
                         1D9C                 P1     = #TEST_MOD_NAME,-
                         1D9C                 P2     = SERV_NAME,-
                         1D9C                 P3     = CURRENT_TC
                         1DBD         $PUTMSG_S MSGVEC = MSGVEC               ; print the message
          04    6C    91 1DCE         CMPB    (AP),#4                         ; is this a register message?
                26    13 1DD1         BEQL    10$                             ; br if yes
          01    6C    91 1DD3         CMPB    (AP),#1                         ; is this just a message?
                48    13 1DD6         BEQL    20$                             ; br if yes
                         1DD8         $FAO_S  CTRSTR = CS2,-
                         1DD8                 OUTLEN = MESSAGEL,-
                         1DD8                 OUTBUF = MSGL,-
                         1DD8                 P1     = 4(AP),-
                         1DD8                 P2     = 8(AP),-
                         1DD8                 P3     = 4(AP),-
                         1DD8                 P4     = 12(AP)
                40    11 1DF7         BRB     30$                             ; goto output message
                         1DF9  10$:
                         1DF9         $FAO_S  CTRSTR = CS3,-
                         1DF9                 OUTLEN = MESSAGEL,-
                         1DF9                 OUTBUF = MSGL,-
                         1DF9                 P1     = 4(AP),-
                         1DF9                 P2     = 16(AP),-
                         1DF9                 P3     = 8(AP),-
                         1DF9                 P4     = 4(AP),-
                         1DF9                 P5     = 16(AP),-
                         1DF9                 P6     = 12(AP)
                19    11 1E1E         BRB     30$                             ; goto output message
                         1E20  20$:
  029A'CF    04 AC    D0 1E20         MOVL    4(AP),MSGVEC1+12                ; save string address
                         1E26         $PUTMSG_S MSGVEC = MSGVEC1              ; print the message
```

```
                 11    11  1E37  2191          BRB     40$                            ; skip the other message
                           1E39  2192 30$:
                           1E39  2193          $PUTMSG_S MSGVEC = MSGVEC              ; print the message
                           1E4A  2194 40$:
        1E5E'CF    00  FB  1E4A  2195          CALLS   #0,MODE_ID                     ; identify the mode
    0282'CF  002C'CF   DE  1E4F  2196          MOVAL   TEST_MOD_FAIL,TMD_ADDR         ; set failure message address
027A'CF   03     00    02  F0  1E56  2197      INSV    #ERROR,#0,#3,MOD_MSG_CODE      ; set severity code
                       04  1E5D  2198          RET
```

```
        1E5E         .SBTTL  MODE_ID
        1E5E  ;++
        1E5E  ; FUNCTIONAL DESCRIPTION:
        1E5E  ;      Subroutine to identify the mode that an exit handler is in.
        1E5E  ;
        1E5E  ; CALLING SEQUENCE:
        1E5E  ;      CALLS   #0,W^MODE_ID
        1E5E  ;
        1E5E  ; INPUT PARAMETERS:
        1E5E  ;      MODE contains an address pointing to an ascii string desc.
        1E5E  ;      of the current CPU mode.
        1E5E  ;
        1E5E  ; OUTPUT PARAMETERS:
        1E5E  ;      NONE
        1E5E  ;
        1E5E  ;--
        1E5E
        1E5E  MODE_ID:
  003C  1E5E         .WORD   ^M<R2,R3,R4,R5>
        1E60         $FAO_S  CTRSTR = CS5,-
        1E60                 OUTLEN = MESSAGEL,-
        1E60                 OUTBUF = MSGL,-
        1E60                 P1     = MODE           ; format the error message
        1E77         $PUTMSG_S MSGVEC = MSGVEC       ; print the mode message
    04  1E88         RET
```

```
                         1E89          .SBTTL  DATA_CHECK
                         1E89   ;++
                         1E89   ; FUNCTIONAL DESCRIPTION:
                         1E89   ;       Routine to check the data pattern used in the sections.
                         1E89   ;
                         1E89   ; CALLING SEQUENCE:
                         1E89   ;       PUSHAL  INADR
                         1E89   ;       PUSHL   FIRST_VALUE
                         1E89   ;       CALLS   #2,DATA_CHECK
                         1E89   ;
                         1E89   ; INPUT PARAMETERS:
                         1E89   ;       8(AP) = A quad word containing the first address and the last address
                         1E89   ;               of the buffer to be checked.
                         1E89   ;       4(AP) = First value which should be stored in the buffer. If this value
                         1E89   ;               is non-zero the data pattern is assumed to be a decrementing
                         1E89   ;               longword count pattern.  If it is zero the data is assumed to
                         1E89   ;               be DZRO.
                         1E89   ;
                         1E89   ; IMPLICIT INPUTS:
                         1E89   ;       NONE
                         1E89   ;
                         1E89   ; OUTPUT PARAMETERS:
                         1E89   ;       NONE
                         1E89   ;
                         1E89   ; IMPLICIT OUTPUTS:
                         1E89   ;       NONE
                         1E89   ;
                         1E89   ; COMPLETION CODES:
                         1E89   ;       R0 = SS$_NORMAL if data was correct.
                         1E89   ;       R0 = 0 if data was incorrect.
                         1E89   ;
                         1E89   ; SIDE EFFECTS:
                         1E89   ;       Error message is printed if a failure occured.
                         1E89   ;
                         1E89   ;--
                         1E89
                         1E89   DATA_CHECK:
                   0FFC  1E89          .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; Entry mask
   53    08 AC     D0    1E8B          MOVL    8(AP),R3                 ; Get the address of the address quadword
   54    04 A3     D0    1E8F          MOVL    4(R3),R4                 ; Get the ending address
      53    63     D0    1E93          MOVL    (R3),R3                  ; Get the starting address
      54    53     D1    1E96          CMPL    R3,R4                    ; Which order are the address in
            09     15    1E99          BLEQ    10$                      ; Br if inorder or equal
      55    53     D0    1E9B          MOVL    R3,R5                    ; Otherwise swap them around
      53    54     D0    1E9E          MOVL    R4,R3                    ; To get the right end data
      54    55     D0    1EA1          MOVL    R5,R4
                         1EA4   10$:
0286'CF     53     D0    1EA4          MOVL    R3,START_ADDRESS         ; Save the lower address
   52    04 AC     D0    1EA9          MOVL    4(AP),R2                 ; Get the starting data value
            1B     13    1EAD          BEQL    30$                      ; BR if data pattern of zero
55    54    53     C3    1EAF          SUBL3   R3,R4,R5                 ; Calculate the byte difference
      55    04     C6    1EB3          DIVL2   #4,R5                    ; Calculate the longword difference
55    52    55     C3    1EB6          SUBL3   R5,R2,R5                 ; Subtract the start value bias
                         1EBA   20$:
      52    83     D1    1EBA          CMPL    (R3)+,R2                 ; Check the data
            16     12    1EBD          BNEQ    40$                      ; Br if data is bad
            52     D7    1EBF          DECL    R2                       ; Do all of the longwords
```

```
            55    52    D1  1EC1              CMPL    R2,R5                   ; All done?
                  F4    12  1EC4              BNEQ    20$                     ; BR if not
            50    01    D0  1EC6              MOVL    #SS$_NORMAL,R0          ; Set success status if OK
                        04  1EC9              RET
                            1ECA     30$:
            00    83    D1  1ECA              CMPL    (R3)+,#0                ; Check for zeroed data
                  06    12  1ECD              BNEQ    40$                     ; BR if bad
            54    53    D1  1ECF              CMPL    R3,R4                   ; End of data to check?
                  F6    19  1ED2              BLSS    30$                     ; BR if not
                        04  1ED4              RET
                            1ED5     40$:
               FC A3    DD  1ED5              PUSHL   -4(R3)                  ; Push bad data value
                  52    DD  1ED8              PUSHL   R2                      ; Push good data value
            0289'CF     DF  1EDA              PUSHAL  DATA                    ; Push failure type name
       FEB7 CF    03    FB  1EDE              CALLS   #3,PRINT_FAIL           ; Report the failure
53     53   0286'CF     C3  1EE3              SUBL3   START_ADDRESS,R3,R3     ; Get the buffer offset
            53    04    C2  1EE9              SUBL2   #4,R3                   ; Compensate for auto inc
                            1EEC              $FAO_S  CTRSTR = CS4,-
                            1EEC                      OUTLEN = MESSAGEL,-
                            1EEC                      OUTBUF = MSGL,-
                            1EEC                      P1     = R3             ; format the error message
                            1F01              $PUTMSG_S MSGVEC = MSGVEC       ; print the offset message
            50          D4  1F12              CLRL    R0                      ; Set failure return
                        04  1F14              RET
                            1F15
```

```
                              1F15        .SBTTL NONSUB_SSE
                              1F15  ;++
                              1F15  ; FUNCTIONAL DESCRIPTION:
                              1F15  ;        Subroutine to report the failure of a system service which is not the
                              1F15  ;        subject system service.
                              1F15  ;
                              1F15  ; CALLING SEQUENCE:
                              1F15  ;        PUSHL   R0                ; Save the failure status
                              1F15  ;        CALLS   #1,NONSUB_SSE     ; Print the failure message
                              1F15  ;
                              1F15  ; INPUT PARAMETERS:
                              1F15  ;        4(AP) = Status code of failing system service
                              1F15  ;
                              1F15  ; OUTPUT PARAMETERS:
                              1F15  ;        NONE
                              1F15  ;
                              1F15  ;--
                              1F15
                              1F15  NONSUB_SSE:
                        0FFC  1F15        .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
                              1F17        $GETMSG_S MSGID  = 4(AP),-
                              1F17                  MSGLEN = BUFFER_PTR,-
                              1F17                  BUFADR = BUFFER_PTR,-
                              1F17                  FLAGS  = #1                       ; Get just the text of the message
                              1F2D        $FAO_S  CTRSTR = NSSSF,-
                              1F2D                OUTLEN = MESSAGE1L,-
                              1F2D                OUTBUF = MSG1L,-
                              1F2D                P1     = #BUFFER_PTR              ; Format the failure message
0379'CF   0100 8F      B0     1F46        MOVW    #TEXT_BUFFER,BUFFER_PTR           ; Reset the descriptor length
          0232'CF      DF     1F4D        PUSHAL  MESSAGE1L                         ; Push the string address
   FE44 CF    01       FB     1F51        CALLS   #1,PRINT_FAIL                     ; Print the failure
                       04     1F56        RET
```

```
                              1F57          .SBTTL  RMS Error Handler
                              1F57  ;++
                              1F57  ; FUNCTIONAL DESCRIPTION:
                              1F57  ;       This routine handles error returns from RMS calls.
                              1F57  ;
                              1F57  ; CALLING SEQUENCE:
                              1F57  ;       Called by RMS when a file processing error is found.
                              1F57  ;
                              1F57  ; INPUT PARAMETERS:
                              1F57  ;       NONE
                              1F57  ;
                              1F57  ; IMPLICIT INPUTS:
                              1F57  ;       The FAB or RAB associated with the RMS call.
                              1F57  ;
                              1F57  ; OUTPUT PARAMETERS:
                              1F57  ;       NONE
                              1F57  ;
                              1F57  ; IMPLICIT OUTPUTS:
                              1F57  ;       Error message
                              1F57  ;
                              1F57  ; COMPLETION CODES:
                              1F57  ;       NONE
                              1F57  ;
                              1F57  ; SIDE EFFECTS:
                              1F57  ;       Program may exit, depending on severity of the error.
                              1F57  ;
                              1F57  ;--
                              1F57
                              1F57  RMS_ERROR:
                        0FFC  1F57          .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; Entry mask
                              1F59
        56   04 AC    D0  1F59          MOVL    4(AP),R6                ; See whether we're dealing with...
           66   03    91  1F5D          CMPB    #FAB$C_BID,FAB$B_BID(R6) ; ...a FAB or a RAB
                11    12  1F60          BNEQ    10$                     ; BR if it's a RAB
           58   56    D0  1F62          MOVL    R6,R8                   ; ...address of FAB...
             0C A6    DD  1F65          PUSHL   FAB$L_STV(R6)           ; ...STV field for error...
             08 A6    DD  1F68          PUSHL   FAB$L_STS(R6)           ; ...STS field for error...
     0012'CF   08 A6    D0  1F6B          MOVL    FAB$L_STS(R6),STATUS_CODE ; ...and save the error code
                10    11  1F71          BRB     COMMON                  ; FAB and RAB share other code
                          1F73  10$:
        58   3C A6    D0  1F73          MOVL    RAB$L_FAB(R6),R8        ; ...address of associated FAB...
             0C A6    DD  1F77          PUSHL   RAB$L_STV(R6)           ; ...STV field for error...
             08 A6    DD  1F7A          PUSHL   RAB$L_STS(R6)           ; ...STS field for error...
     0012'CF   08 A6    D0  1F7D          MOVL    RAB$L_STS(R6),STATUS_CODE ; ...and save the error code
                          1F83  COMMON:
        5A   34 A8    9A  1F83          MOVZBL  FAB$B_FNS(R8),R10       ; Get the file name size
                          1F87          $FAO_S  CTRSTR = RMS_ERR_MSG,- ; Common code, prepare error message...
                          1F87                  OUTLEN = BUFFER_PTR,-
                          1F87                  OUTBUF = FAO_BUF,-
                          1F87                  P2     = R10,-
                          1F87                  P3     = FAB$L_FNA(R8)
         0379'CF      DF  1F9F          PUSHAL  BUFFER_PTR              ; Push the string address
      FDF2 CF   01    FB  1FA3          CALLS   #1,PRINT_FAIL           ; Print the failure
                      04  1FA8          RET
                          1FA9
                          1FA9  MOD_MSG_PRINT:
                          1FA9  ;
```

```
         1FA9  ;   *******************************************************************
         1FA9  ;   *                                                                 *
         1FA9  ;   *  PRINTS THE TEST MODULE BEGUN/SUCCESSFUL/FAILED MESSAGES        *
         1FA9  ;   *       (USING THE PUTMSG MACRO).                                 *
         1FA9  ;   *                                                                 *
         1FA9  ;   *******************************************************************
         1FA9  ;
         1FA9           PUTMSG  <MOD_MSG_CODE,#2,TMN_ADDR,TMD_ADDR> ; PRINT MSG
     05  1FBE           RSB                               ; ... AND RETURN TO CALLER
```

```
           1FBF         .SBTTL  Exit Handler
           1FBF  ;++
           1FBF  ; FUNCTIONAL DESCRIPTION:
           1FBF  ;      This routine handles cleanup on exits.
           1FBF  ;
           1FBF  ; CALLING SEQUENCE:
           1FBF  ;      Invoked automatically by $EXIT System Service.
           1FBF  ;
           1FBF  ; INPUT PARAMETERS:
           1FBF  ;      Location STATUS_CODE contains the exit status
           1FBF  ;
           1FBF  ; IMPLICIT INPUTS:
           1FBF  ;      NONE
           1FBF  ;
           1FBF  ; OUTPUT PARAMETERS:
           1FBF  ;      NONE
           1FBF  ;
           1FBF  ; IMPLICIT OUTPUTS:
           1FBF  ;      Erase the UETPSECT.DAT file even if the test is aborted.
           1FBF  ;
           1FBF  ; COMPLETION CODES:
           1FBF  ;      NONE
           1FBF  ;
           1FBF  ; SIDE EFFECTS:
           1FBF  ;      NONE
           1FBF  ;
           1FBF  ;--
           1FBF
           1FBF  EXIT_HANDLER:
     0FFC  1FBF         .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11> ; Entry mask
           1FC1
           1FC1         $DASSGN_S CHAN = SECFAB+FAB$L_STV       ; Deassign the channel
           1FCD         $ERASE  FAB = FILEFAB,-
           1FCD                 ERR = RMS_ERROR                 ; Erase the old section file
           1FDB         $SETPRV_S ENBFLG = #0,-
           1FDB                 PRVADR = PRIV_MASK              ; Remove all temp privileges
       04  1FEC         RET                                     ; That's all folks!
           1FED         .END UETSSMM00
```

```
$$.TAB                = 000016A0 R  04     EXIT_HANDLER             00001FBF R  06
$$.TABEND             = 000016E4 R  04     EXP                      00000214 R  02
$$.TMP                = 00000000           FAB$B_BID              = 00000000
$$.TMP1               = 00000002           FAB$B_FAC              = 00000016
$$.TMP2               = 000000CF           FAB$B_FNS              = 00000034
$$.TMPX               = 0000000C R  05     FAB$C_BID              = 00000003
$$.TMPX1              = 0000000C           FAB$C_BLN              = 00000050
$$ARGS                = 00000008           FAB$C_SEQ              = 00000000
$$T1                  = 00000004           FAB$C_VAR              = 00000002
$$T2                  = 00000005           FAB$L_ALQ              = 00000010
A                     = FFFFFE7F           FAB$L_FNA              = 0000002C
ADR                     0000027A R  02     FAB$L_FOP              = 00000004
AN_IDENT                0000033E R  03     FAB$L_STS              = 00000008
AN_ILLEGAL_IDENT        000002BF R  02     FAB$L_STV              = 0000000C
ARG_COUNT               00000016 R  03     FAB$M_PUT              = 00000001
BAD_PFN                 0000009E R  02     FAB$V_BIO              = 00000005
BUF                     00000022 R  03     FAB$V_CHAN_MODE        = 00000002
BUF1                    00000132 R  03     FAB$V_CR               = 00000001
BUFFER                  00000381 R  03     FAB$V_FILE_MODE        = 00000004
BUFFER_PTR              00000379 R  03     FAB$V_GET              = 00000001
COMMON                  00001F83 R  06     FAB$V_LNM_MODE         = 00000000
CRMPSC                  00000033 R  02     FAB$V_PUT              = 00000000
CRMPSC$_ACMODE        = 0000000C           FAB$V_UFO              = 00000011
CRMPSC$_CHAN          = 00000020           FAB$W_GBC              = 00000048
CRMPSC$_FLAGS         = 00000010           FAO_BUF                  00000371 R  03
CRMPSC$_GSDNAM        = 00000014           FILEFAB                  00001650 R  04
CRMPSC$_IDENT         = 00000018           FILERAB                  000016A0 R  04
CRMPSC$_INADR         = 00000004           GSDNAM                   00000346 R  03
CRMPSC$_NARGS         = 0000000C           GSD_LOGNAM               0000049F R  03
CRMPSC$_PAGCNT        = 00000024           INADDRESS                00000369 R  03
CRMPSC$_PFC           = 00000030           KM                       00000206 R  02
CRMPSC$_PROT          = 0000002C           LIB$SIGNAL               ********   X  06
CRMPSC$_RELPAG        = 0000001C           LOGICAL_NAME             00000481 R  03
CRMPSC$_RETADR        = 00000008           MESSAGETL                00000232 R  03
CRMPSC$_VBN           = 00000028           MESSAGEL                 00000122 R  03
CRMPSCG                 000002B6 R  03     MGBLSC                   00000041 R  02
CS1                     000000DB R  02     MGBLSC$_ACMODE         = 0000000C
CS2                     0000010D R  02     MGBLSC$_FLAGS          = 00000010
CS3                     0000013A R  02     MGBLSC$_GSDNAM         = 00000014
CS4                     0000016D R  02     MGBLSC$_IDENT          = 00000018
CS5                     0000018A R  02     MGBLSC$_INADR          = 00000004
CTL$GL_PHD              ********   X  06   MGBLSC$_NARGS          = 00000007
CURRENT_TC              00000006 R  03     MGBLSC$_RELPAG         = 0000001C
DATA                    00000289 R  02     MGBLSC$_RETADR         = 00000008
DATA_CHECK              00001E89 R  06     MGBLSCG                  000002EA R  03
DGBLSC                  0000003A R  02     MMG$GL_MAXPFN            ********   X  06
DGBLSC$_FLAGS         = 00000004           MODE                     0000000A R  03
DGBLSC$_GSDNAM        = 00000008           MODE_ID                  00001E5E R  06
DGBLSC$_IDENT         = 0000000C           MOD_MSG_CODE             0000027A R  03
DGBLSC$_NARGS         = 00000003           MOD_MSG_PRINT            00001FA9 R  06
DGBLSCG                 0000030A R  03     MS                       000002A3 R  02
EF2                   = 00000002           MSG1L                    0000012A R  03
EFN1                  = 00000001           MSGL                     0000001A R  03
EM                      000001F5 R  02     MSGVEC                   000002AF R  02
EQUIV_NAME              00000492 R  03     MSGVEC1                  0000028E R  03
ERROR                 = 00000002           NAME_INC1                00000491 R  03
EXIT_DESC               0000029E R  03     NAME_INC2                0000049E R  03
```

```
NONSUB_SSE                 00001F15 R     06     SS$_IVSECFLG              = 0000016C
NO_PFN_TEST                00001504 R     06     SS$_IVSECIDCTL            = 000002E4
NSSSF                      0000019F R     02     SS$_NOPRIV                = 00000024
OUTADDRESS                 00000361 R     03     SS$_NORMAL                = 00000001
PHD$L_P0BR               = 000000C8              SS$_NOSUCHSEC             = 00000978
PRINT_FAIL                 00001D9A R     06     SS$_NOTFILEDEV            = 000001CC
PRIV_MASK                  00000355 R     03     SS$_NOTMODIFIED           = 00000659
PRT$C_NA                 = 00000000              SS$_NOWRT                 = 000003FC
PRT$C_UW                 = 00000004              SS$_TOOMANYLNAM           = 00000374
PRV$V_CMEXEC             = 00000001              SS$_VASFULL               = 00000244
PRV$V_PFNMAP             = 0000001A              START_ADDRESS               00000286 R     03
PRV$V_PRMGBL             = 00000018              STATUS                      00000295 R     02
PRV$V_PSWAPM             = 0000000C              STATUS_CODE                 00000012 R     03
PRV$V_SYSGBL             = 00000019              STEP                      = 00000046
PRV$V_SYSNAM             = 00000002              STP0                        0000003D R     06
PSL$C_USER               = 00000003              STP1                        0000003D R     06
PTE$S_PFN                = 00000015              STP10                       00000310 R     06
PTE$V_PFN                = 00000000              STP11                       00000343 R     06
RAB$B_RAC                = 0000001E              STP12                       000003B7 R     06
RAB$C_BID                = 00000001              STP13                       00000414 R     06
RAB$C_BLN                = 00000044              STP14                       00000487 R     06
RAB$C_SEQ                = 00000000              STP15                       00000501 R     06
RAB$L_CTX                = 00000018              STP16                       00000566 R     06
RAB$L_FAB                = 0000003C              STP17                       000005EF R     06
RAB$L_ROP                = 00000004              STP18                       000006C4 R     06
RAB$L_STS                = 00000008              STP19                       0000073B R     06
RAB$L_STV                = 0000000C              STP2                        00000083 R     06
RAB$W_RSZ                = 00000022              STP20                       000007B2 R     06
READ_BUF                   00000600 R     04     STP21                       0000082F R     06
READ_BUF1                  00000E00 R     04     STP22                       00000894 R     06
REG                        00000268 R     02     STP23                       00000905 R     06
REGNUM                     0000000E R     03     STP24                       00000968 R     06
REG_CHECK                  00001D36 R     06     STP25                       000009E7 R     06
REG_SAVE                   00001D2C R     06     STP26                       00000A5C R     06
REG_SAVE_AREA              0000023E R     03     STP27                       00000AC3 R     06
RMS_ERROR                  00001F57 R     06     STP28                       00000B3C R     06
RMS_ERR_MSG                0000004F R     02     STP29                       00000BB1 R     06
SEC$K_MATEQU             = 00000001              STP3                        000000B2 R     06
SEC$K_MATLEQ             = 00000002              STP30                       00000BF0 R     06
SEC$M_CRF                = 00000002              STP31                       00000C03 R     06
SEC$M_DZRO               = 00000004              STP32                       00000C16 R     06
SEC$M_EXPREG             = 00020000              STP33                       00000C30 R     06
SEC$M_GBL                = 00000001              STP34                       00000C69 R     06
SEC$M_PERM               = 00004000              STP35                       00000CA2 R     06
SEC$M_PFNMAP             = 00010000              STP36                       00000CD5 R     06
SEC$M_SYSGBL             = 00008000              STP37                       00000D0A R     06
SEC$M_WRT                = 00000008              STP38                       00000D41 R     06
SECFAB                     00001600 R     04     STP39                       00000DD0 R     06
SERV_NAME                  0000023A R     03     STP4                        00000168 R     06
SHR$_TEXT                = 00001130              STP40                       00000E09 R     06
SM                         000001E8 R     02     STP41                       00000E94 R     06
SS$_ACCVIO               = 0000000C              STP42                       00000F1C R     06
SS$_BADPARAM             = 00000014              STP43                       00000F5D R     06
SS$_CREATED              = 00000619              STP44                       00000F9A R     06
SS$_ENDOFFILE            = 00000870              STP45                       0000104A R     06
SS$_IVCHAN               = 0000013C              STP46                       00001132 R     06
SS$_IVLOGNAM             = 00000154              STP47                       000011CA R     06
```

```
STP48                 000012BA R      06      SYS$WAKE                ********  GX   06
STP49                 000013C9 R      06      SYS$WRITE               ********  GX   06
STP5                  000001AB R      06      TEST_MOD_BEGIN          0000001B R     02
STP50                 00001468 R      06      TEST_MOD_FAIL           0000002C R     02
STP51                 0000150B R      06      TEST_MOD_NAME           00000000 R     02
STP52                 000015D8 R      06      TEST_MOD_NAME_D         0000000A R     02
STP53                 00001614 R      06      TEST_MOD_SUCC           00000021 R     02
STP54                 00001646 R      06      TEXT_BUFFER           = 00000100
STP55                 00001748 R      06      TMD_ADDR                00000282 R     03
STP56                 00001783 R      06      TMN_ADDR                0000027E R     03
STP57                 000017BE R      06      TOOLONGSTRING           0000022A R     02
STP58                 00001822 R      06      TPID                    00000000 R     03
STP59                 00001881 R      06      TTCHAN                  00000004 R     03
STP6                  00000200 R      06      TTNAM                   0000025E R     02
STP60                 000018F6 R      06      UETP$_SATSMS          = 007480D9
STP61                 00001909 R      06      UETP$_TEXT            = 00741130
STP62                 0000191C R      06      UETSSMM00               00000000 RG    06
STP63                 00001973 R      06      UM                      000001DC R     02
STP64                 000019F3 R      06      UPDSEC                  00000048 R     02
STP65                 00001AA4 R      06      UPDSEC$_ACMODE        = 0000000C
STP66                 00001B9D R      06      UPDSEC$_ASTADR        = 0000001C
STP67                 00001C7F R      06      UPDSEC$_ASTPRM        = 00000020
STP68                 00001C96 R      06      UPDSEC$_EFN           = 00000014
STP69                 00001CAD R      06      UPDSEC$_INADR         = 00000004
STP7                  00000235 R      06      UPDSEC$_IOSB          = 00000018
STP70                 00001CC4 R      06      UPDSEC$_NARGS         = 00000008
STP8                  000002A2 R      06      UPDSEC$_RETADR        = 00000008
STP9                  000002D5 R      06      UPDSEC$_UPDFLG        = 00000010
STS$V_INHIB_MSG     = 0000001C                UPDSECG                 0000031A R     03
SUCCESS             = 00000001                UPDSEC_FAILED           0000006C R     02
SYS$CLOSE             ********  GX    06      UPD_IOSB                000002AE R     03
SYS$CMEXEC            ********  GX    06      VA$S_VPN              = 00000015
SYS$CONNECT           ********  GX    06      VA$V_VPN              = 00000009
SYS$CREATE            ********  GX    06      VBN                     0000035D R     03
SYS$CRELOG            ********  GX    06      WORK                    0000028A R     03
SYS$CREMBX            ********  GX    06      WRITE_BUF               00000000 R     04
SYS$CRMPSC            ********  GX    06      ZEROSTRING              00000222 R     02
SYS$DASSGN            ********  GX    06
SYS$DCLEXH            ********  GX    06
SYS$DELLOG            ********  GX    06
SYS$DELTVA            ********  GX    06
SYS$DGBLSC            ********  GX    06
SYS$ERASE             ********  GX    06
SYS$EXIT              ********  GX    06
SYS$FAO               ********   X    06
SYS$GETMSG            ********  GX    06
SYS$HIBER             ********  GX    06
SYS$LCKPAG            ********  GX    06
SYS$MGBLSC            ********  GX    06
SYS$OPEN              ********  GX    06
SYS$PUTMSG            ********  GX    06
SYS$SETAST            ********  GX    06
SYS$SETPRN            ********  GX    06
SYS$SETPRT            ********  GX    06
SYS$SETPRV            ********  GX    06
SYS$UPDSEC            ********  GX    06
SYS$WAITFR            ********  GX    06
```

```
                                        +----------------+
                                        ! 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                           000002C7  (   711.)     02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR NOEXE   RD   NOWRT NOVEC LONG
RWDATA                           000004AC  (  1196.)     03 (  3.)  NOPIC   USR   CON   REL   LCL NOSHR NOEXE   RD     WRT NOVEC LONG
RMS                              000016E4  (  5860.)     04 (  4.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC PAGE
$RMSNAM                          00000018  (    24.)     05 (  5.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
UETSSMM00                        00001FED  (  8173.)     06 (  6.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC LONG

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

Phase                        Page faults    CPU Time       Elapsed Time
-----                        -----------    --------       ------------
Initialization                        36    00:00:00.10    00:00:00.38
Command processing                   135    00:00:00.63    00:00:02.52
Pass 1                               781    00:00:37.14    00:01:08.36
Symbol table sort                      4    00:00:02.24    00:00:02.59
Pass 2                               452    00:00:10.02    00:00:13.37
Symbol table output                    2    00:00:00.26    00:00:00.72
Psect synopsis output                  2    00:00:00.03    00:00:00.03
Cross-reference output                 0    00:00:00.00    00:00:00.00
Assembler run totals                1414    00:00:50.43    00:01:27.97

The working set limit was 2000 pages.
242043 bytes (473 pages) of virtual memory were used to buffer the intermediate code.
There were 80 pages of symbol table space allocated to hold 1498 non-local and 78 local symbols.
2446 source lines were read in Pass 1, producing 60 object records in Pass 2.
85 pages of virtual memory were used to define 77 macros.

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

Macro library name                                 Macros defined
------------------                                 --------------
_$255$DUA28:[SHRLIB]UETP.MLB;1                                  8
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                                  3
_$255$DUA28:[SYSLIB]STARLET.MLB;2                              63
TOTALS (all libraries)                                         74

1943 GETS were required to define 74 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:UETSSMM00/OBJ=OBJ$:UETSSMM00 MSRC$:UETSSMM00/UPDATE=(ENH$:UETSSMM00)+EXECML$/LIB+SHRLIB$:UETP/LIB
```

0428 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

DISKMON
MAP

UETMEMY01
LIS

UETSSMM00
LIS

DCLDIET
MAP

DISKQ
MAP

UTIL32

CHECKSUM
MAP