


```

SSSSSSSS  AAAAAA  TTTTTTTTTT  SSSSSSSS  SSSSSSSS  SSSSSSSS  44  44  77777777
SSSSSSSS  AAAAAA  TTTTTTTTTT  SSSSSSSS  SSSSSSSS  SSSSSSSS  44  44  77777777
SS          AA      AA      TT          SS          SS          SS          44  44  77
SS          AA      AA      TT          SS          SS          SS          44  44  77
SS          AA      AA      TT          SS          SS          SS          44  44  77
SS          AA      AA      TT          SS          SS          SS          44  44  77
SSSSSSS    AA      AA      TT          SSSSSSS    SSSSSSS    SSSSSSS    44  44  77
SSSSSSS    AA      AA      TT          SSSSSSS    SSSSSSS    SSSSSSS    44  44  77
          SS    AAAAAAAAAA  TT          SS          SS          SS          44  44  77
          SS    AAAAAAAAAA  TT          SS          SS          SS          44  44  77
          SS    AA      AA      TT          SS          SS          SS          44  44  77
          SS    AA      AA      TT          SS          SS          SS          44  44  77
SSSSSSSS    AA      AA      TT          SSSSSSSSS  SSSSSSSSS  SSSSSSSSS  44  44  77
SSSSSSSS    AA      AA      TT          SSSSSSSSS  SSSSSSSSS  SSSSSSSSS  44  44  77

```

```

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
LLLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLLL IIIIII  SSSSSSSS

```

(1)	50	DECLARATIONS
(1)	108	R/W PSECT
(1)	190	SATSSS47
(1)	239	SETPRV TESTS
(2)	379	REG_SAVE
(2)	400	REG_CHECK
(2)	442	PRINT_FAIL
(2)	489	MODE_ID

```
0000 1 .TITLE SATSSS47 - SATS SYSTEM SERVICE TESTS (SUCC S.C.)
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 :*****
0000 6 :*
0000 7 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 :* ALL RIGHTS RESERVED.
0000 10 :*
0000 11 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 :* TRANSFERRED.
0000 17 :*
0000 18 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 :* CORPORATION.
0000 21 :*
0000 22 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 :*
0000 25 :*
0000 26 :*****
0000 27 :
0000 28 :
0000 29 :++
0000 30 : FACILITY: SATS SYSTEM SERVICE TESTS
0000 31 :
0000 32 : ABSTRACT: The SATSSS47 module tests the execution of the following
0000 33 : VMS system services:
0000 34 :
0000 35 : $SETPRV
0000 36 :
0000 37 : ENVIRONMENT: User mode image.
0000 38 : Needs CMKRNL privilege and dynamically acquires other
0000 39 : privileges, as needed.
0000 40 :
0000 41 : AUTHOR: Larry D. Jones, CREATION DATE: OCTOBER, 1979
0000 42 :
0000 43 : MODIFIED BY:
0000 44 :
0000 45 : V03-001 LDJ0001 Larry D. Jones, 17-Sep-1980
0000 46 : Modified to conform to new build command procedures.
0000 47 :**
0000 48 :--
```

```

0000 50      .SBTTL  DECLARATIONS
0000 51      :
0000 52      : MACRO LIBRARY CALLS
0000 53      :
0000 54      .LIBRARY /SYSS$LIBRARY:STARLET.MLB/
0000 55      $JPIDEF      ; GETJPI definitions
0000 56      $SHR MESSAGES UETP,116,<<TEXT,INFO>> ; UETP$ TEXT definition
0000 57      $SFDEF      ; stack frame definitions
0000 58      $STSDEF     ; STS definitions
0000 59      $UETPDEF    ; UETP message definitions
0000 60      :
0000 61      : Equated symbols
0000 62      :
00000000 0000 63 WARNING      = 0      ; warning severity value for msgs
00000001 0000 64 SUCCESS     = 1      ; success
00000002 0000 65 ERROR       = 2      ; error
00000003 0000 66 INFO        = 3      ; information
00000004 0000 67 SEVERE      = 4      ; fatal
0000 68      :
0000 69      :
0000 70      : MACROS
0000 71      :

```

SAT
Sym
\$\$A
\$\$T
\$\$T
BUF
CHM
CHM
CS1
CS2
CS3
CS5
CUR
ERR
EXP
GET
GET
INF
JPI
JPI
LIB
MES
ML
MOD
MOD
MOD
MOD
MSG
MSG
MSG
PRI
PRI
PRI
PRI
PRI
PRV
RO
REG
REG
REG
REG
REG
RET
SAT
SER
SET
SET
SET
SET
SET
SET
SEV
SHR
SHR
SS\$
STA
STE

```

00000000 73 .PSECT RODATA, RD, NOWRT, NOEXE, PAGE
0000 74
0000 75 TEST_MOD_NAME:
37 34 53 53 53 54 41 53 00' 0000 76 .ASCIC /SATSSS47/ ; needed for SATSMS message
08 0000
0009 77 TEST_MOD_NAME D:
53 53 53 54 41 53 00000011'010E0000' 0009 78 .ASCID /SATSSS47/ ; module name
37 34 0017
0019 79 TEST_MOD_BEGIN: ; start end and fail messages
6E 69 67 65 62 00' 0019 80 .ASCIC /begin/
05 0019
001F 81 TEST_MOD_SUCC:
6C 75 66 73 73 65 63 63 75 73 00' 001F 82 .ASCIC /successful/
0A 001F
002A 83 TEST_MOD_FAIL:
64 65 6C 69 61 66 00' 002A 84 .ASCIC /failed/
06 002A
0031 85 CS1: ; failure messages
21 20 74 73 65 54 00000039'010E0000' 0031 86 .ASCID \Test !AC service name !AC step !UL failed.\
6E 20 65 63 69 76 72 65 73 20 43 41 003F
70 65 74 73 20 43 41 21 20 65 6D 61 004B
2E 64 65 6C 69 61 66 20 4C 55 21 20 0057
0063 87 CS2:
74 63 65 70 78 45 0000006B'010E0000' 0063 88 .ASCID \Expected !AS = !XL received !AS = !XL\
4C 58 21 20 3D 20 53 41 21 20 64 65 0071
41 21 20 64 65 76 69 65 63 65 72 20 007D
4C 58 21 20 3D 20 53 0089
0090 89 CS3:
74 63 65 70 78 45 00000098'010E0000' 0090 90 .ASCID \Expected !AS!UB = !XL received !AS!UB = !XL\
20 3D 20 42 55 21 53 41 21 20 64 65 009E
64 65 76 69 65 63 65 72 20 4C 58 21 00AA
58 21 20 3D 20 42 55 21 53 41 21 20 00B6
4C 00C2
00C3 91 CS5:
69 20 65 64 6F 4D 000000CB'010E0000' 00C3 92 .ASCID \Mode is !AS.\
2E 53 41 21 20 73 00D1
00D7 93 EXP:
73 75 74 61 74 73 000000DF'010E0000' 00D7 94 .ASCID \status\
00E5 95 UM: ; mode messages
72 65 73 75 000000ED'010E0000' 00E5 96 .ASCID \user\
00F1 97 UNEXPRVCHNG:
65 70 78 65 6E 55 000000F9'010E0000' 00F1 98 .ASCID \Unexpected privilege change.\
65 6C 69 76 69 72 70 20 64 65 74 63 00FF
2E 65 67 6E 61 68 63 20 65 67 010B
0115 99 MSGVEC:
00000003 0115 100 .LONG 3 ; PUTMSG message vector
00741133 0119 101 .LONG UETPS_TEXT
00000001 011D 102 .LONG 1
00000173' 0121 103 .ADDRESS MESSAGEL
0125 104 SETPRV:
56 52 50 54 45 53 00' 0125 105 .ASCIC \SETPRV\ ; SETPRV service name
06 0125

```

```

012C 107 ;
012C 108 .SBTTL R/W PSECT
00000000 109 .PSECT RWDATA,RD,WRT,NOEXE,PAGE
0000 110 ;
0000 111 fPID: ;
00000000 0000 112 .LONG 0 ; PID for this process
00000000 0004 113 CURRENT_IC: ;
00000000 0004 114 .LONG 0 ; ptr to current test case
0008 115 .ALIGN LONG ; put it on a long word boundry
00000044 0008 116 REG_SAVE_AREA: ;
00000044 0008 117 .BLKL 15 ; register save area
00000000 0044 118 RO_SAVE: ;
00000000 0044 119 .LONG 0 ; special case save of R0
007480D9 0048 120 MOD_MSG_CODE: ;
00000000 004C 121 .LONG UETPS_SATSMS ; test module message code for putmsg
00000000 004C 122 TMN_ADDR: ;
00000019 0050 123 .ADDRESS TEST_MOD_NAME
00000019 0050 124 TMD_ADDR: ;
00000019 0050 125 .ADDRESS TEST_MOD_BEGIN
00 0054 126 PRVPRT: ;
00 0054 127 .BYTE 0 ; protection return byte for SETPRT
00000000 00000000 0055 128 PRIVMASK: ;
00000000 0055 129 .QUAD 0 ; priv. mask
00000000 005D 130 CHM_CONT: ;
00000000 005D 131 .LONG 0 ; change mode continue address
00000069 0061 132 RETADR: ;
00000069 0061 133 .BLKL 2 ; returned address's from SETPRT
00000000 0069 134 STATUS: ;
00000000 0069 135 .LONG 0
00000000 006D 136 MODE: ;
00000000 006D 137 .LONG 0 ; current mode string pointer
74 73 69 67 65 72 00000079 010E0000 0071 138 REG: ;
52 20 72 65 0071 139 .ASCID \register R\
00000000 0083 140 REGNUM: ;
00000000 0083 141 .LONG 0 ; register number
00000050 0087 142 MSGL: ;
0000008F 0087 143 .LONG 80 ; buffer desc.
0000008F 008B 144 .ADDRESS BUF
000000DF 008F 145 BUF: ;
000000DF 008F 146 .BLKB 0
00000000 00DF 147 ML: ;
00000000 00DF 148 .LONG 0 ; desc. for BUF_CHECK routine
000000EF 00E3 149 .ADDRESS GETBUF+8
00000084 00E7 150 GETBUF: ;
000000EF 00E7 151 .LONG 132
000000EF 00EB 152 .ADDRESS +4
00000173 00EF 153 .BLKB 132
0173 154 MESSAGEL: ;
00000000 0173 155 .LONG 0 ; message desc.
0000008F 0177 156 .ADDRESS BUF
00000000 017B 157 SERV_NAME: ;
00000000 017B 158 .LONG 0 ; service name pointer
00000003 017F 159 MSGVEC1: ;
00000003 017F 160 .LONG 3 ; PUTMSG message vector
00741133 0183 161 .LONG UETPS_TEXT
00000001 0187 162 .LONG 1

```

```

00000000 018B 163 .LONG 0
018F 164 GET_LIST:
0008 018F 165 .WORD 8
0400 0191 166 .WORD JPI$_CURPRIV ; GETJPI item list
000001AB' 0193 167 .LONG PRIV_LIST
00000000 0197 168 .LONG 0
0008 019B 169 .WORD 8
0204 019D 170 .WORD JPI$_PROCPRIV
000001B3' 019F 171 .LONG PRIV_LIST+8
00000000 01A3 172 .LONG 0
00000000 01A7 173 .LONG 0
01AB 174 PRIV_LIST:
00000000 00000000 01AB 175 .QUAD 0 ; resultant CURPRIV
00000000 00000000 01B3 176 .QUAD 0 ; resultant PROCPRIV
01BB 177 PRIV_TEST:
00000000 00000000 01BB 178 .QUAD 0 ; privileges for SETPRV to set
01C3 179 PRIV_SAVE:
00000000 00000000 01C3 180 .QUAD 0 ; saved initial image privileges
00000000 00000000 01CB 181 .QUAD 0 ; saved initial process privileges
01D3 182 PRIV_MOD:
00000000 00000000 01D3 183 .QUAD 0 ; expected current image privileges
00000000 00000000 01DB 184 .QUAD 0 ; expected current process privileges
01E3 185 SET:
01E3 186 $SETPRV 0,0,0,PRIV_TEST ; SETPRV parameter list
01F7 187

```



```
00000000 189 .PSECT SATSSS47, RD, WRT, EXE, PAGE
0000 190 .SBTTL SATSSS47
0000 191 :++
0000 192 : FUNCTIONAL DESCRIPTION:
0000 193 :
0000 194 : After performing some initial housekeeping, such as
0000 195 : printing the module begin message and acquiring needed privileges,
0000 196 : the system services are tested in each of their normal conditions.
0000 197 : Detected failures are identified and an error message is printed
0000 198 : on the terminal. Upon completion of the test a success or fail
0000 199 : message is printed on the terminal.
0000 200 :
0000 201 : CALLING SEQUENCE:
0000 202 :
0000 203 : $ RUN SATSSS47 ... (DCL COMMAND)
0000 204 :
0000 205 : INPUT PARAMETERS:
0000 206 :
0000 207 : none
0000 208 :
0000 209 : IMPLICIT INPUTS:
0000 210 :
0000 211 : none
0000 212 :
0000 213 : OUTPUT PARAMETERS:
0000 214 :
0000 215 : none
0000 216 :
0000 217 : IMPLICIT OUTPUTS:
0000 218 :
0000 219 : Messages to SYS$OUTPUT are the only output from SATSSS47.
0000 220 : They are of the form:
0000 221 :
0000 222 : %UETP-S-SATSMS, TEST MODULE SATSSS47 BEGUN ... (BEGIN MSG)
0000 223 : %UETP-S-SATSMS, TEST MODULE SATSSS47 SUCCESSFUL ... (END MSG)
0000 224 : %UETP-E-SATSMS, TEST MODULE SATSSS47 FAILED ... (END MSG)
0000 225 : %UETP-I-TEXT, ... (VARIABLE INFORMATION ABOUT A TEST MODULE FAILURE)
0000 226 :
0000 227 : COMPLETION CODES:
0000 228 :
0000 229 : The SATSSS47 routine terminates with a $EXIT to the
0000 230 : operating system with a status code defined by UETPS_SATSMS.
0000 231 :
0000 232 : SIDE EFFECTS:
0000 233 :
0000 234 : none
0000 235 :
0000 236 : --
0000 237 :
0000 238 : TEST_START SATSSS47 ; let the test begin
```

```

0000 0000
0004'CF D4 0002
00 DD 0006
0000'CF DF 0008
00000000'GF G2 FB 000C
00000000'GF 00 FB 0013
00000000'GF 01 FB 001A
00000000'GF 01 FB 001E
0050'CF 001F'CF 30 0025
0048'CF 03 00 01 FO 002F
00 DD 0036
0330'CF 01 FB 0038
003D
003D 239
003D 240
003D 241
003D 242
003D 243
003D 244
003D 245
003D 246
017B'CF 0125'CF DE 003D 247
006D'CF 00E5'CF DE 0044 248
004B 249
01C3'CF 01AB'CF 7D 0060 250
01CB'CF 01B3'CF 7D 0067 251
00 DD 006E 252
0330'CF 01 FB 0070 253
0075 254
0084 255
00000000'8F DD 0084
033A'CF 01 FB 008A
008F 256
01AB'CF 01C3'CF 10 29 00A4 257
009 13 00AC 258
00F1'CF DF 00AE 259
037C'CF 01 FB 00B2 260
00B7 261
00B7 262
00B7 263
00B7 264
00B7 265
00B7 266
00B7 267
0004'CF 01 D0 00B7
00 DD 00BC
0330'CF 01 FB 00BE
00C3 268
00CC 269
00000000'8F DD 00CC
033A'CF 01 FB 00D2
01BB'CF 01C3'CF 08 29 00D7 270
009 13 00DF 271

```

```

.ENTRY SATSSS47,0
CLRL W^CURRENT_TC
PUSHL #0
PUSHAL W^TPID
CALLS #2,G^SYSSWAKE
CALLS #0,G^SYSSHIBER
PUSHAQ W^TEST MOD NAME_D
CALLS #1,G^SYS$SETPRN
BSBW W^MOD MSG PRINT
MOVAL W^TEST MOD_SUCC,W^TMD_ADDR
INSV #SUCCESS,#0,#3,W^MOD_MSG_CODE
PUSHL #0
CALLS #1,W^REG_SAVE

STP0:
.SBTTL SETPRV TESTS
239
240 :+
241 :
242 : $SETPRV tests
243 :
244 : test _S form with a complete default parameter list
245 :
246 :-
247 MOVAL W^SETPRV,W^SERV_NAME ; set service name
248 MOVAL W^UM,W^MODE ; set the mode
249 $GETJPI _S ITMLST=W^GET LIST ; get fresh copy of privileges
250 MOVQ W^PRIV_LIST,W^PRIV_SAVE ; save current privileges
251 MOVQ W^PRIV_LIST+8,W^PRIV_SAVE+8 ; save process privileges
252 PUSHL #0 ; push a dummy parameter
253 CALLS #1,W^REG_SAVE ; save a reg snapshot
254 $SETPRV _S ; try total default
255 FAIL_CHECK SSS_NORMAL ; check success
PUSHL #SS$ NORMAL
CALLS #1,W^REG_CHECK
256 $GETJPI _S ITMLST=W^GET LIST ; get the current priv.
257 CMPC3 #16,W^PRIV_SAVE,W^PRIV_LIST ; check for changes
258 BEQL 10$ ; br if OK
259 PUSHAL W^UNEXPRVCHNG ; push string variable
260 CALLS #1,W^PRINT_FAIL ; print the failure
261 10$:
262 :+
263 :
264 : test the PRVPRV parameter _G
265 :
266 :-
267 NEXT_TEST

STP1:
MOVL #1,W^CURRENT_TC
PUSHL #0
CALLS #1,W^REG_SAVE
268 $SETPRV G W^SET ; try _G with PRVPRV
269 FAIL_CHECK SSS_NORMAL ; check for success
PUSHL #SS$ NORMAL
CALLS #1,W^REG_CHECK
270 CMPC3 #8,W^PRIV_SAVE,W^PRIV_TEST ; check for changes
271 BEQL 20$ ; br if OK

```

```

00F1'CF 01 DF 00E1 272 PUSHAL W^UNEXPRVCHNG ; push string variable
037C'CF 01 FB 00E5 273 CALLS #1,W^PRINT_FAIL ; print the failure
00EA 274 20$:
00EA 275 :+
00EA 276 :
00EA 277 : test temp clr of one priv _G
00EA 278 :
00EA 279 :-
00EA 280 NEXT_TEST

0004'CF 02 DO 00EA STP2:
0004'CF 00 DD 00EF MOVL #2,W^CURRENT_TC
0330'CF 01 FB 00F1 PUSHL #0
01BB'CF 01C3'CF 7D 00F6 281 MOVQ W^PRIV_SAVE,W^PRIV_TEST ; get current image priv.
01D3'CF 01C3'CF 7D 00FD 282 MOVQ W^PRIV_SAVE,W^PRIV_MOD ; make a copy of the priv.
01E7'CF D4 0104 283 CLRL W^SET+SETPRV$_ENBFLG ; set for disable
01F3'CF D4 0108 284 CLRL W^SET+SETPRV$_PRVPRV ; disable previous priv
52 01EB'CF 01BB'CF DE 010C 285 MOVAL W^PRIV_TEST,W^SET+SETPRV$_PRVADR ; set priv. address
01BB'CF 1F 00 EA 0113 286 FFS #0,#31,W^PRIV_TEST,R2 ; find a priv
01BB'CF 01BB'CF D4 011A 287 CLRL W^PRIV_TEST ; clear off a space to work
01BB'CF 01 52 01 FO 011E 288 INSV #1,R2,#1,W^PRIV_TEST ; set a bit for the priv to remove
01D3'CF 01 52 00 FO 0125 289 INSV #0,R2,#1,W^PRIV_MOD ; set expected results
0330'CF 00 DD 012C 290 PUSHL #0 ; push a dummy parameter
0330'CF 01 FB 012E 291 CALLS #1,W^REG_SAVE ; save a register snapshot
0133 292 $SETPRV G W^SET ; try G
013C 293 FAIL_CHECK SSS_NORMAL ; check results
00000000'8F DD 013C PUSHL #SS$_NORMAL
033A'CF 01 FB 0142 CALLS #1,W^REG_CHECK
01AB'CF 01D3'CF 08 29 0147 294 $GETJPI _S ITMLST=W^GET_LIST ; get new priv.
01D3'CF 09 13 0164 295 CMPC3 #8,W^PRIV_MOD,W^PRIV_LIST ; check the results
00F1'CF DF 0166 296 BEQL 30$ ; br if OK
037C'CF 01 FB 016A 297 PUSHAL W^UNEXPRVCHNG ; push str var
016F 298 CALLS #1,W^PRINT_FAIL ; print the failure
016F 299 30$:
016F 300 :+
016F 301 :
016F 302 : test temp adding of one priv _S
016F 303 :
016F 304 :-
016F 305 NEXT_TEST

0004'CF 03 DO 016F STP3:
0004'CF 00 DD 0174 MOVL #3,W^CURRENT_TC
0330'CF 01 FB 0176 PUSHL #0
01D3'CF 01AB'CF 7D 017B 306 MOVQ W^PRIV_LIST,W^PRIV_MOD ; save a copy of the privs
01BB'CF 01AB'CF 7D 0182 307 MOVQ W^PRIV_LIST,W^PRIV_TEST ;
52 01BB'CF 1F 00 EB 0189 308 FFC #0,#31,W^PRIV_TEST,R2 ; find a missing priv
01BB'CF 01BB'CF 7C 0190 309 CLRQ W^PRIV_TEST ; clean out the bits
01BB'CF 01 52 01 FO 0194 310 INSV #1,R2,#1,W^PRIV_TEST ; enable that priv.
01D3'CF 01 52 01 FO 019B 311 INSV #1,R2,#1,W^PRIV_MOD ; make expected results
0330'CF 00 DD 01A2 312 PUSHL #0 ; push a dummy parameter
01A4 313 CALLS #1,W^REG_SAVE ; save a reg snapshot
01A9 314 $SETPRV _S ENBFLG=#1, - ; try S
01A9 315 PRVADR=W^PRIV_TEST ;
01BA 316 FAIL_CHECK SSS_NORMAL ; check success

```

```

00000000'8F DD 01BA          PUSHL  #SS$ NORMAL
033A'CF 01 FB 01C0          CALLS  #1,W^REG_CHECK
01AB'CF 01D3'CF 08 29 01C5 317 $GETJPI -S ITMLST=W^GET_LIST ; get current priv
01D3'CF 09 13 01DA 318 CMPC3  #8,W^PRIV_MOD,W^PRIV_LIST ; check for the change
00F1'CF DF 01E2 319 BEQL  40$ ; br if OK
037C'CF 01 FB 01E4 320 PUSHAL W^UNEXPRVCHNG ; push str var
01ED 321 CALLS  #1,W^PRINT_FAIL ; print the failure
01ED 322 40$:
01ED 323 :+
01ED 324 :
01ED 325 : test the perm clearing of one privilege _G
01ED 326 :
01ED 327 :-
01ED 328 NEXT_TEST

0004'CF 04 DO 01ED          STP4:
0330'CF 01 DD 01F2          MOVL  #4,W^CURRENT_TC
01BB'CF 01CB'CF 01 FB 01F4          PUSHL #0
01D3'CF 01CB'CF 7D 01F9 329 CALLS  #1,W^REG_SAVE
52 01BB'CF 1F 00 7D 0200 330 MOVQ  W^PRIV_SAVE+8,W^PRIV_TEST ; get process priv.
01BB'CF 01 52 01 7D 0200 330 MOVQ  W^PRIV_SAVE+8,W^PRIV_MOD
01D3'CF 01 52 00 FO 0207 331 FFS  #0,#31,W^PRIV_TEST,R2 ; find a priv
01EF'CF 01 DO 020E 332 CLRL  W^PRIV_TEST ; clear off a space to work
01D3'CF 01 52 00 FO 0212 333 INSV  #1,R2,#1,W^PRIV_TEST ; set a bit for the priv to remove
01EF'CF 01 DO 0219 334 INSV  #0,R2,#1,W^PRIV_MOD ; set expected results
0330'CF 01 FB 0220 335 MOVL  #1,W^SET^SETPRV$ _PRMFLG ; set the perm ilag
0225 336 PUSHL #0 ; push a dummy parameter
0227 337 CALLS  #1,W^REG_SAVE ; save a reg snapshot
022C 338 $SETPRV G W^SET ; try G
0235 339 FAIL_CHECK SS$ NORMAL ; check for success
00000000'8F DD 0235          PUSHL #SS$ NORMAL
033A'CF 01 FB 023B          CALLS  #1,W^REG_CHECK
01B3'CF 01D3'CF 08 29 0240 340 $GETJPI -S ITMLST=W^GET_LIST ; get current priv.
01D3'CF 09 13 0255 341 CMPC3  #8,W^PRIV_MOD,W^PRIV_LIST+8 ; check the priv.'s
00F1'CF DF 025D 342 BEQL  60$ ; br if OK
037C'CF 01 FB 025F 343 PUSHAL W^UNEXPRVCHNG ; push string variable
0263 344 CALLS  #1,W^PRINT_FAIL ; print the failure
0268 345 60$:
0330'CF 01 FB 0268 346 PUSHL #0 ; push a dummy parameter
026A 347 CALLS  #1,W^REG_SAVE ; save a reg snapshot
026F 348 $SETPRV _S ENBFLG=#1,-
026F 349 PRVADR=W^PRIV_SAVE+8,-
026F 350 PRMFLG=#1 ; reset perm priv to original
0280 351 FAIL_CHECK SS$ NORMAL ; check for failure
00000000'8F DD 0280          PUSHL #SS$ NORMAL
033A'CF 01 FB 0286          CALLS  #1,W^REG_CHECK
028B 352 :+
028B 353 :
028B 354 : test perm add one priv _S
028B 355 :
028B 356 :-
028B 357 NEXT_TEST

0004'CF 05 DO 028B          STP5:
0330'CF 01 FB 0290          MOVL  #5,W^CURRENT_TC
0292          PUSHL #0
0292          CALLS  #1,W^REG_SAVE

```

```

01D3'CF 01B3'CF 7D 0297 358
01BB'CF 01B3'CF 7D 029E 359
52 01BB'CF 1F 00 EB 02A5 360
01BB'CF 01 52 01 D4 02AC 361
01D3'CF 01 52 01 F0 02B0 362
01D3'CF 01 52 01 F0 02B7 363
0330'CF 00 DD 02BE 364
0330'CF 01 FB 02C0 365
02C5 366
02C5 367
02C5 368
02D6 369
00000000'8F DD 02D6
033A'CF 01 FB 02DC
01D3'CF 01B3'CF 08 29 02E1 370
01D3'CF 01B3'CF 09 13 02F6 371
00F1'CF DF 02FE 372
037C'CF 01 FB 0300 373
0309 374
0309 375 50$:
0309 376
0050'CF DD 0309
004C'CF DD 030D
02 DD 0311
0048'CF DD 0313
00000000'GF 04 FB 0317
0048'CF 01 1C 01 F0 031E
0048'CF DD 0325
00000000'GF 01 FB 0329

```

```

MOVQ W^PRIV_LIST+8,W^PRIV_MOD ; save a copy of the privs
MOVQ W^PRIV_LIST+8,W^PRIV_TEST
FFC #0,#31,W^PRIV_TEST,R2 ; find a missing priv
CLRL W^PRIV_TEST ; clean up the bits
INSV #1,R2,#1,W^PRIV_TEST ; add the missing priv
INSV #1,R2,#1,W^PRIV_MOD ; make expected results
PUSHL #0 ; push a dummy parameter
CALLS #1,W^REG_SAVE ; save a reg snapshot
$SETPRV_S ENBFLG=#1,-
PRVADR=W^PRIV_TEST,-
PRMFLG=#1 ; try S
FAIL_CHECK $$$NORMAL ; check for success
PUSHL #$$$NORMAL
CALLS #1,W^REG_CHECK
$GETJPI_S ITMLST=W^GET_LIST ; get the current priv.
CMPC3 #8,W^PRIV_LIST+8,W^PRIV_MOD ; check for change
BEQL 50$ ; br if OK
PUSHAL W^UNEXPRVCHNG ; push the str variable
CALLS #1,W^PRINT_FAIL ; print the failure

TEST_END
PUSHL W^TMD_ADDR
PUSHL W^TMN_ADDR
PUSHL #2
PUSHL W^MOD_MSG_CODE
CALLS $$$T1-G^LIB$SIGNAL
INSV #1,#ST$SV_INHIB_MSG,#1,W^MOD_MSG_CODE
PUSHL W^MOD_MSG_CODE
CALLS #1,G^SYS$EXIT

```

```

0330 379      .SBTTL REG_SAVE
0330 380      :++
0330 381      : FUNCTIONAL DESCRIPTION:
0330 382      : Subroutine to save R2-R11 in the register save location.
0330 383      :
0330 384      : CALLING SEQUENCE:
0330 385      :   PUSHL   #0           ; save a dummy parameter
0330 386      :   CALLS   #1,W^REG_SAVE ; save R2-R11
0330 387      :
0330 388      : INPUT PARAMETERS:
0330 389      :   NONE
0330 390      :
0330 391      : OUTPUT PARAMETERS:
0330 392      :   NONE
0330 393      :
0330 394      :--
0330 395
0330 396 REG_SAVE:
0330 397      .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
0008'CF 14 AD 28 OFFC 0332 398      MOVCS   #4*10,^X14(FP),W^REG_SAVE_AREA ; save the registers in the program
0330 399      RET
033A 400      .SBTTL REG_CHECK
033A 401      :++
033A 402      : FUNCTIONAL DESCRIPTION:
033A 403      : Subroutine to test R0 & R2-R11 for proper content after a service
033A 404      : execution. A snapshot is taken by the REG_SAVE routine at the
033A 405      : beginning of each step and this routine is executed after the
033A 406      : services have been executed.
033A 407      :
033A 408      : CALLING SEQUENCE:
033A 409      :   PUSHL   #SS$ XXXXXX   ; push expected R0 contents
033A 410      :   CALLS   #1,W^REG_CHECK ; execute this routine
033A 411      :
033A 412      : INPUT PARAMETERS:
033A 413      :   expected R0 contents on the stack
033A 414      :
033A 415      : OUTPUT PARAMETERS:
033A 416      :   possible error messages printed using $PUTMSG
033A 417      :
033A 418      :--
033A 419
033A 420 REG_CHECK:
033A 421      .WORD   ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
033C 422      CMPL   4(AP),R0           ; is this the right fail code?
0340 423      BEQL   10$,R0             ; br if yes
0342 424      PUSHL  R0               ; push received data
0344 425      PUSHL  4(AP)            ; push expected data
0347 426      PUSHAL W^EXP           ; push the string variable
034B 427      CALLS  #3,W^PRINT_FAIL ; print the error message
0350 428      10$:
0350 429      CMPCS  #4*10,^X14(FP),W^REG_SAVE_AREA ; check all but R0
0357 430      BEQL   20$,R0             ; br if O.K.
0359 431      SUBL3  #REG_SAVE_AREA,R3,R6 ; calculate the register number
0361 432      DIVL2  #4,R8              ;
0364 433      ADDB3  #^X2,R6,-(SP)      ; set number past R0-R1 and save
0368 434      BICL2  #3,R1             ; backup to register boundrys
036B 435      BICL2  #3,R3

```



```
0440 489 .SBTTL MODE_ID
0440 490 :++
0440 491 : FUNCTIONAL DESCRIPTION:
0440 492 : Subroutine to identify the mode that an exit handler is in.
0440 493 :
0440 494 : CALLING SEQUENCE:
0440 495 : CALLS #0,W^MODE_ID
0440 496 :
0440 497 : INPUT PARAMETERS:
0440 498 : MODE contains an address pointing to an ascii string desc.
0440 499 : of the current CPU mode.
0440 500 :
0440 501 : OUTPUT PARAMETERS:
0440 502 : NONE
0440 503 :
0440 504 :--
0440 505 :
003C 0440 506 MODE_ID:
0442 507 .WORD ^M<R2,R3,R4,R5>
045B 508 $FAO S W^CSS,W^MESSAGEL,W^MSGL,MODE ; format the error message
04 046C 509 $PUTMSG_S W^MSGVEC ; print the mode message
510 RET
```



```
046D 513 MOD_MSG_PRINT:
046D 514 :
046D 515 : *****
046D 516 : *
046D 517 : * PRINTS THE TEST MODULE BEGUN/SUCCESSFUL/FAILED MESSAGES *
046D 518 : * (USING THE PUTMSG MACRO). *
046D 519 : *
046D 520 : *****
046D 521 :
05 046D 522 PUTMSG <MOD_MSG_CODE,#2,TMN_ADDR,TMD_ADDR> ; PRINT MSG
0488 523 RSB ; ... AND RETURN TO CALLER
0489 524 :
0489 525 CHMRTN:
0489 526 : *****
0489 527 : *
0489 528 : * CHANGE MODE ROUTINE. THIS ROUTINE GETS CONTROL WHENEVER *
0489 529 : * A CMKRNL, CMEXEC, OR CMSUP SYSTEM SERVICE IS ISSUED *
0489 530 : * BY THE MODE MACRO ('TO' OPTION). IT MERELY DOES *
0489 531 : * A JUMP INDIRECT ON A FIELD SET UP BY MODE. IT HAS *
0489 532 : * THE EFFECT OF RETURNING TO THE END OF THE MODE *
0489 533 : * MACRO EXPANSION. *
0489 534 : *
0489 535 : *****
0489 536 :
000005D'FF 0000 0489 537 .WORD 0 ; ENTRY MASK
17 0488 538 JMP @CHM_CONT ; RETURN TO MODE MACRO IN NEW MODE
0491 539 :
0491 540 : * RET INSTR WILL BE ISSUED IN EXPANSION OF 'MODE FROM, ....' MACRO
0491 541 :
0491 542 TEST_END:
0491 543 .END SATSSS47
```

\$\$ARGS = 00000004
\$\$T1 = 00000004
\$\$T2 = 00000004
BUF 0000008F R 03
CHMRTN 00000489 R R 04
CHM_CONT 0000005D R R 03
CS1 00000031 R R 02
CS2 00000063 R R 02
CS3 00000090 R R 02
CS5 000000C3 R R 02
CURRENT_TC 00000004 R 03
ERROR = 00000002
EXP 000000D7 R 02
GETBUF 000000E7 R 03
GET_LIST 0000018F R 03
INFO = 00000003
JPIS_CURPRIV = 00000400
JPIS_PROCPRIV = 00000204
LIB\$SIGNAL ***** X 04
MESSAGEL 00000173 R 03
ML 000000DF R 03
MODE 0000006D R 03
MODE_ID 00000440 R 04
MOD_MSG_CODE 00000046 R 03
MOD_MSG_PRINT 0000046D R 04
MSGC 00000C87 R 03
MSGVEC 00000115 R 02
MSGVEC1 0000017F R 03
PRINT_FAIL 0000037C R 04
PRIVMASK 00000055 R 03
PRIV_LIST 000001AB R 03
PRIV_MOD 000001D3 R 03
PRIV_SAVE 000001C3 R 03
PRIV_TEST 000001BB R 03
PRVPRV 00000054 R 03
RO_SAVE 00000044 R 03
REG 00000071 R 03
REGNUM 00000083 R 03
REG_CHECK 0000033A R 04
REG_SAVE 00000330 R 04
REG_SAVE_AREA 00000008 R 03
RETADR 00000061 R 03
SATSSS47 00000000 RG 04
SERV_NAME 00000178 R 03
SET 000001E3 R 03
SETPRV 00000125 R 02
SETPRV\$_ENBFLG = 00000004
SETPRV\$_NARGS = 00000004
SETPRV\$_PRMFLG = 0000000C
SETPRV\$_PRVADR = 00000008
SETPRV\$_PRVPRV = 00000010
SEVERE = 00000004
SHR\$K_SHRDEF = 00000001
SHR\$ TEXT = 00001130
SS\$ NORMAL ***** X 04
STATUS 00000069 R 03
STEP = 00000005

STP0 000C003D R 04
STP1 000000B7 R 04
STP2 000000EA R 04
STP3 0000016F R 04
STP4 000001ED R 04
STP5 0000028B R 04
ST\$V_INHIB_MSG = 0000001C
SUCCESS = 00000001
SYS\$EXIT ***** GX 04
SYS\$FAO ***** X 04
SYS\$GETJPI ***** GX 04
SYS\$HIBER ***** GX 04
SYS\$PUTMSG ***** GX 04
SYS\$SETPRN ***** GX 04
SYS\$SETPRV ***** GX 04
SYS\$WAKE ***** GX 04
TEST_END 00000491 R 04
TEST_MOD_BEGIN 00000019 R R 02
TEST_MOD_FAIL 0000002A R R 02
TEST_MOD_NAME 00000000 R R 02
TEST_MOD_NAME_D 00000009 R R 02
TEST_MOD_SUCC 0000001F R R 02
TMD_ADDR 00000050 R 03
TMN_ADDR 0000004C R 03
TPID 00000000 R 03
UETPS_SATSMS = 007480D9
UETPS_TEXT = 00741133
UM 000000E5 R 02
UNEXPRVCHNG 000000F1 R 02
WARNING = 00000000

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

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
RODATA	0000012C (300.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC PAGE
RWDATA	000001F7 (503.)	03 (3.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC PAGE
SATSSS47	00000491 (1169.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC PAGE

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.08	00:00:00.37
Command processing	137	00:00:00.80	00:00:04.21
Pass 1	263	00:00:07.19	00:00:13.92
Symbol table sort	0	00:00:00.2	00:00:00.59
Pass 2	119	00:00:01.92	00:00:03.38
Symbol table output	10	00:00:00.08	00:00:00.09
Psect synopsis output	3	00:00:00.02	00:00:00.04
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	569	00:00:00.61	00:00:22.60

The working set limit was 1350 pages.
42200 bytes (83 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 378 non-local and 12 local symbols.
543 source lines were read in Pass 1, producing 25 object records in Pass 2.
38 pages of virtual memory were used to define 34 macros.

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

Macro library name	Macros defined
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	21
_\$255\$DUA28:[SHRLIB]UETP.MLB;1	10
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	0
TOTALS (all libraries)	31

582 GETS were required to define 31 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:SATSSS47/OBJ=OBJ\$:SATSSS47 MSRC\$:SATSSS47/UPDATE=(ENH\$:SATSSS47)+EXECMLS/LIB+SHRLIB\$:UETP/LIB

0423 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

