

UUU	UUU	EEEEEEEEEEEEEEEE	TTTTTTTTTTTTTTTT	PPPPPPPPPPPPPP	
UUU	UUU	EEEEEEEEEEEEEEEE	TTTTTTTTTTTTTTTT	PPPPPPPPPPPPPP	
UUU	UUU	EEEEEEEEEEEEEEEE	TTTTTTTTTTTTTTTT	PPPPPPPPPPPPPP	
UUU	UUU	EEE	TTT	PPP	PPP
UUU	UUU	EEE	TTT	PPP	PPP
UUU	UUU	EEE	TTT	PPP	PPP
UUU	UUU	EEE	TTT	PPP	PPP
UUU	UUU	EEE	TTT	PPP	PPP
UUU	UUU	EEE	TTT	PPP	PPP
UUU	UUU	EEE	TTT	PPP	PPP
UUU	UUU	EEEEEEEEEEEEEEEE	TTT	PPPPPPPPPPPPPP	
UUU	UUU	EEEEEEEEEEEEEEEE	TTT	PPPPPPPPPPPPPP	
UUU	UUU	EEEEEEEEEEEEEEEE	TTT	PPPPPPPPPPPPPP	
UUU	UUU	EEE	TTT	PPP	
UUU	UUU	EEE	TTT	PPP	
UUU	UUU	EEE	TTT	PPP	
UUU	UUU	EEE	TTT	PPP	
UUU	UUU	EEE	TTT	PPP	
UUU	UUU	EEE	TTT	PPP	
UUU	UUU	EEE	TTT	PPP	
UUUUUUUUUUUUUUUU	UUUUUUUUUUUUUUUU	EEEEEEEEEEEEEEEE	TTT	PPP	
UUUUUUUUUUUUUUUU	UUUUUUUUUUUUUUUU	EEEEEEEEEEEEEEEE	TTT	PPP	
UUUUUUUUUUUUUUUU	UUUUUUUUUUUUUUUU	EEEEEEEEEEEEEEEE	TTT	PPP	

SA
VOA[illegible]

(1)	56	DECLARATIONS
(1)	171	R/W PSECT
(1)	240	SATSSS10
(1)	289	GETMSG TESTS
(1)	342	PUTMSG TESTS
(1)	408	REG_SAVE
(1)	429	REG_CHECK
(1)	472	PRINT_FAIL
(1)	508	CHECK_RESULTS
(1)	564	ACT


```
0000 1 .TITLE SATSSS10 - SATS SYSTEM SERVICE TESTS (SUCC S.C.)
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6
0000 7 *
0000 8 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 9 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 10 * ALL RIGHTS RESERVED.
0000 11 *
0000 12 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 13 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 14 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 15 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 16 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 17 * TRANSFERRED.
0000 18 *
0000 19 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 20 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 21 * CORPORATION.
0000 22 *
0000 23 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 24 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 25 *
0000 26 *****
0000 27
0000 28
0000 29 ++
0000 30 FACILITY: SATS SYSTEM SERVICE TESTS
0000 31
0000 32 ABSTRACT: The SATSSS10 module tests the execution of the following
0000 33 VMS system services:
0000 34
0000 35 $GETMSG
0000 36 $PUTMSG
0000 37
0000 38
0000 39 ENVIRONMENT: User mode image.
0000 40 Needs CMKRN privilege and dynamically acquires other
0000 41 privileges, as needed.
0000 42
0000 43 AUTHOR: Larry D. Jones, CREATION DATE: JULY, 1978
0000 44
0000 45 MODIFIED BY:
0000 46
0000 47 V03-002 LDJ0002 Larry D. Jones, 14-Dec-1981
0000 48 Modified to conform to new $PUTMSG argument modification which
0000 49 made a BR out of range error in routine CHECK_RESULTS.
0000 50
0000 51 V03-001 LDJ0001 Larry D. Jones, 17-Sep-1980
0000 52 Modified to conform to new build command procedures.
0000 53
0000 54 **
0000 55 --
```



```

0000 56 .SBTTL DECLARATIONS
0000 57 ;
0000 58 ; MACRO LIBRARY CALLS
0000 59 ;
0000 60 $PRVDEF ; privilege definitions
0000 61 $SHR_MESSAGES UETP,116,<<TEXT,INFO>> ; UETP$ TEXT definition
0000 62 $STSDEF ; STS definitions
0000 63 $UETPDEF ; UETP message definitions
0000 64 ;
0000 65 ; Equated symbols
0000 66 ;
00000000 0000 67 WARNING = 0 ; warning severity value for msgs
00000001 0000 68 SUCCESS = 1 ; success
00000002 0000 69 ERROR = 2 ; error
00000003 0000 70 INFO = 3 ; information
00000004 0000 71 SEVERE = 4 ; fatal
0000 72 ;
0000 73 ; MACROS
0000 74 ;

```



```
00000000 76 .PSECT RODATA,RD,NOWRT,NOEXE, LONG
0000 77
0000 78 TEST_MOD_NAME:
30 31 53 53 53 54 41 53 00' 0000 79 .ASCIC /SATSSS10/ ; needed for SATSMS message
08 0000
0009 80 TEST_MOD_NAME D:
53 53 53 54 41 53 00000011'010E0000' 0009 81 .ASCID /SATSSS10/ ; module name
30 31 0017
0019 82 TEST_MOD_BEGIN:
6E 75 67 65 62 00' 0019 83 .ASCIC /begun/
05 0019
001F 84 TEST_MOD_SUCC:
6C 75 66 73 73 65 63 63 75 73 00' 001F 85 .ASCIC /successful/
0A 001F
002A 86 TEST_MOD_FAIL:
64 65 6C 69 61 66 00' 002A 87 .ASCIC /failed/
06 002A
0031 88 GETMSG:
47 53 4D 54 45 47 00' 0031 89 .ASCIC /GETMSG/
06 0031
0038 90 PUTMSG:
47 53 4D 54 55 50 00' 0038 91 .ASCIC /PUTMSG/
06 0038
003F 92 CS1:
21 20 74 73 65 54 00000047'010E0000' 003F 93 .ASCID \Test !AC service name !AC step !UL failed.\
6E 20 65 63 69 76 72 65 73 20 43 41 004D
70 65 74 73 20 43 41 21 20 65 6D 61 0059
2E 64 65 6C 69 61 66 20 4C 55 21 20 0065
0071 94 CS2:
74 63 65 70 78 45 00000079'010E0000' 0071 95 .ASCID \Expected !AS = !XL received !AS = !XL\
4C 58 21 20 3D 20 53 41 21 20 64 65 007F
41 21 20 64 65 76 69 65 63 65 72 20 008B
4C 58 21 20 3D 20 53 0097
009E 96 CS3:
74 63 65 70 78 45 000000A6'010E0000' 009E 97 .ASCID \Expected !AS!UB = !XL received !AS!UB = !XL\
20 3D 20 42 55 21 53 41 21 20 64 65 00AC
64 65 76 69 65 63 65 72 20 4C 58 21 00B8
58 21 20 3D 20 42 55 21 53 41 21 20 00C4
4C 00D0
00D1 98 CS4:
74 63 65 70 78 45 000000D9'010E0000' 00D1 99 .ASCID \Expected= !AS!/-UETP-I-TEXT, Received= !AS\
45 55 2D 2F 21 53 41 21 2D 3D 64 65 00DF
52 20 2C 54 58 45 54 2D 49 2D 50 54 00EB
53 41 21 20 3D 64 65 76 69 65 63 65 00F7
0103 100 EXP:
73 75 74 61 74 73 0000010B'010E0000' 0103 101 .ASCID \status\
0111 102 LEN:
68 74 67 6E 65 6C 00000119'010E0000' 0111 103 .ASCID \length\
011F 104 OUT:
52 44 41 54 55 4F 00000127'010E0000' 011F 105 .ASCID \OUTADR\
012D 106 MSGID:
007480E1 012D 107 .LONG UETP$_DDB ; message ID used in the tests
0131 108 MSGVEC:
00000003 0131 109 .LONG 3 ; PUTMSG message vector
00741133 0135 110 .LONG UETP$_TEXT
00000001 0139 111 .LONG 1
000002B3' 013D 112 .ADDRESS MESSAGEL
```



```
00000218'000001EC'000001E7'000001C1' 0141
0000027A'0000024C'00000245'0000021B' 0151
000002E8'000002B7'000002AD'00000280' 0161
0000032B'0000031F'000002F0' 0171
00000384'000001E7'0000035E'00000495' 017D
000003DA'00000245'00000380'00000218' 018D
00000355'000002AD'00000408'0000027A' 019D
00000495'0000031F'00000466'000002E8' 01AD
000004C8' 01BD
01C1 117 MSGTBL:
01C1 114 .ADDRESS M1,M2,M3,M4,M5,M6,M7,M8,M9,M10,M11,M12,M13,M14,M15
01C1 115 MSGTBL1:
01CD 116 .ADDRESS M23,M16,M2,M17,M4,M18,M6,M19,M8,M20,M10,M21,M12,M22,M14,M23,M24
01D9 117 .NLIST BINARY
01E5 118 M1:
01E7 119 .ASCIC /UETINIT00 DDB 0 !AS 00000000 00000000/
01E7 120 M2:
01EC 121 .ASCIC /%DDB/
01EC 122 M3:
01F8 123 .ASCIC /%DDB, UETINIT00 DDB 0 !AS 00000000 00000000/
0204 124 M4:
0210 125 .ASCIC /%S/
0218 126 M5:
0218 127 .ASCIC /%S, UETINIT00 DDB 0 !AS 00000000 00000000/
0218 128 M6:
021B 129 .ASCIC /%S-DDB/
0227 130 M7:
0233 131 .ASCIC /%S-DDB, UETINIT00 DDB 0 !AS 00000000 00000000/
023F 132 M8:
0245 133 .ASCIC /%UETP/
0245 134 M9:
024C 135 .ASCIC /%UETP, UETINIT00 DDB 0 !AS 00000000 00000000/
0258 136 M10:
0264 137 .ASCIC /%UETP-DDB/
0270
027D
027A
027A
027A
0280
0280
028C
0298
02A4
0280
02AD
02AD
02AD
09
```


SATSSS10
V04-000

- SATS SYSTEM SERVICE TESTS (SUCC S.C.) 16-SEP-1984 01:43:51 VAX/VMS Macro V04-00
DECLARATIONS 5-SEP-1984 04:22:41 [UETP.SRC]SATSSS10.MAR;1

Page 5
(1)

20 2C 42 44 44 2D 50 54 45 55 25 00'	02B7	138 M11:	
44 44 20 30 30 54 49 4E 49 54 45 55	02B7	139	.ASCIC /%UETP-DDB, UETINIT00 DDB 0 !AS 00000000 00000000/
30 30 30 30 20 53 41 21 20 30 20 42	02C3		
30 30 30 30 30 30 30 20 30 30 30 30	02CF		
	02DB		
	02E7		
	02B7		
	02E8	140 M12:	
53 2D 50 54 45 55 25 00'	02E8	141	.ASCIC /%UETP-S/
07	02E8		
	02F0	142 M13:	
45 55 20 2C 53 2D 50 54 45 55 25 00'	02F0	143	.ASCIC /%UETP-S, UETINIT00 DDB 0 !AS 00000000 00000000/
20 42 44 44 20 30 30 54 49 4E 49 54	02FC		
30 30 30 30 30 30 20 53 41 21 20 30	0308		
30 30 30 30 30 30 30 30 20 30 30 30	0314		
	2E 02F0		
42 44 44 2D 53 2D 50 54 45 55 25 00'	031F	144 M14:	
	031F	145	.ASCIC /%UETP-S-DDB/
	031F		
	032B	146 M15:	
42 44 44 2D 53 2D 50 54 45 55 25 00'	032B	147	.ASCIC /%UETP-S-DDB, UETINIT00 DDB 0 !AS 00000000 00000000/
20 30 30 54 49 4E 49 54 45 55 20 2C	0337		
30 30 20 53 41 21 20 30 20 42 44 44	0343		
30 30 30 30 30 20 30 30 30 30 30 30	034F		
	035B		
	32 032B		
44 20 30 30 54 49 4E 49 54 45 55 00'	035E	148 M16:	
30 30 30 20 41 44 54 20 30 20 42 44	035E	149	.ASCIC /UETINIT00 DDB 0 TDA 00000000 00000000/
30 30 30 30 30 20 30 30 30 30 30 30	036A		
	0376		
	0382		
	25 035E		
4E 49 54 45 55 20 2C 42 44 44 25 00'	0384	150 M17:	
54 20 30 20 42 44 44 20 30 30 54 49	0384	151	.ASCIC /%DDB, UETINIT00 DDB 0 TDA 00000000 00000000/
20 30 30 30 30 30 30 30 30 30 30 30	0390		
	039C		
	03A8		
	2B 0384		
54 49 4E 49 54 45 55 20 2C 53 25 00'	03B0	152 M18:	
41 44 54 20 30 20 42 44 44 20 30 30	03B0	153	.ASCIC /%S, UETINIT00 DDB 0 TDA 00000000 00000000/
30 30 20 30 30 30 30 30 30 30 30 30	03BC		
	03C8		
	03D4		
	29 03B0		
54 45 55 20 2C 42 44 44 2D 53 25 00'	03DA	154 M19:	
30 20 42 44 44 20 30 30 54 49 4E 49	03DA	155	.ASCIC /%S-DDB, UETINIT00 DDB 0 TDA 00000000 00000000/
30 30 30 30 30 30 30 20 41 44 54 20	03E6		
	03F2		
	03FE		
	2D 03DA		
49 54 45 55 20 2C 50 54 45 55 25 00'	0408	156 M20:	
20 30 20 42 44 44 20 30 30 54 49 4E	0408	157	.ASCIC /%UETP, UETINIT00 DDB 0 TDA 00000000 00000000/
30 30 30 30 30 30 30 30 20 41 44 54	0414		
	0420		
	042C		
	2C 0408		
	0435	158 M21:	

SATSSS10
V04-000

- SATS SYSTEM SERVICE TESTS (SUCC^{E 7} S.C.) 16-SEP-1984 01:43:51 VAX/VMS Macro V04-00
DECLARATIONS 5-SEP-1984 04:22:41 [UETP.SRC]SATSSS10.MAR;1

Page 6
(1)

20 2C 42 44 44 2D 50 54 45 55 25 00' 0435
44 44 20 30 30 54 49 4E 49 54 45 55 0441
30 30 30 30 20 41 44 54 20 30 20 42 044D
30 30 30 30 30 30 20 30 30 30 30 0459
30 0465
30 0435
30 0466

159

.ASCIC /%UETP-DDB, UETINIT00 DDB 0 TDA 00000000 00000000/

45 55 20 2C 53 2D 50 54 45 55 25 00' 0466
20 42 44 44 20 30 30 54 49 4E 49 54 0472
30 30 30 30 30 30 20 41 44 54 20 30 047E
30 30 30 30 30 30 30 30 20 30 30 048A
2E 0466

160 M22:
161

.ASCIC /%UETP-S, UETINIT00 DDB 0 TDA 00000000 00000000/

42 44 44 2D 53 2D 50 54 45 55 25 00' 0495
20 30 30 54 49 4E 49 54 45 55 20 2C 04A1
30 30 20 41 44 54 20 30 20 42 44 44 04AD
30 30 30 30 30 20 30 30 30 30 30 04B9
30 30 30 04C5
32 0495

162 M23:
163

.ASCIC /%UETP-S-DDB, UETINIT00 DDB 0 TDA 00000000 00000000/

42 44 44 2D 53 2D 54 53 45 54 25 00' 04C8
20 30 30 54 49 4E 49 54 45 55 20 2C 04D4
30 30 20 41 44 54 20 30 20 42 44 44 04E0
30 30 30 30 30 20 30 30 30 30 30 04EC
30 30 30 04F8
32 04C8

164 M24:
165

.ASCIC /%TEST-S-DDB, UETINIT00 DDB 0 TDA 00000000 00000000/

54 53 45 54 00000503'010E0000'

04FB
04FB
0507

166 TEST:
167
168

.ASCID /TEST/
.LIST BINARY


```
0507 170 ;
0507 171 .SBTTL R/W PSECT
00000000 172 .PSECT RWDATA,RD,WRT,NOEXE, LONG
0000 173 ;
0000 174 IPID: ; PID for this process
00000000 0000 175 .LONG 0 ; ptr to current test case
00000000 0004 176 CURRENT_IC:
0000 177 .LONG 0 ; register save area
00000044 0008 178 .ALIGN LONG
00000044 0008 179 REG_SAVE_AREA:
0044 180 .BLKL 15 ; test module message code for putmsg
007480D9 0044 181 MOD_MSG_CODE:
0048 182 .LONG UETP$_SATSMS
00000000' 0048 183 TMN_ADDR:
004C 184 .ADDRESS TEST_MOD_NAME
00000019' 004C 185 TMD_ADDR:
004C 186 .ADDRESS TEST_MOD_BEGIN
0050 187 PRVPRT:
00 0050 188 .BYTE 0 ; protection return byte for SETPRT
00000000 00000000 0051 189 PRIVMASK:
0059 190 .QUAD 0 ; priv. mask
00000000 0059 191 CHM_CONT:
005D 192 .LONG 0 ; change mode continue address
00000065 005D 193 RETADR:
0065 194 .BLKL 2 ; returned address's from SETPRT
00000000 0065 195 STATUS:
0069 196 .LONG 0
0069 197 GET:
0069 198 $GETMSG UETP$_DDB,MSGLEN,BUFADR,0,OUTADR ; GETMSG parameter list
00000005' 0069 .ADDRESS 5
007480E1' 006D .ADDRESS UETP$_DDB
000002BF' 0071 .ADDRESS MSGLEN
000002C7' 0075 .ADDRESS BUFADR
00000000' 0079 .ADDRESS 0
000002C3' 007D .ADDRESS OUTADR
0081 199 PUT:
0081 200 $PUTMSG MSGVEC1,ACT,0 ; PUTMSG parameter list
00000004' 0081 .ADDRESS 4
000003DF' 0085 .ADDRESS MSGVEC1
00000355' 0089 .ADDRESS ACT
00000000' 008D .ADDRESS 0
00000000' 0091 .ADDRESS 0
74 73 69 67 65 72 0000009D' 010E0000' 0095 201 REG:
52 20 72 65 00A3 202 .ASCID \register R\
00A7 203 REGNUM:
00000000 00A7 204 .LONG 0 ; register number
00AB 205 MSGL:
00000200 00AB 206 .LONG 512 ; buffer desc.
000000B3' 00AF 207 .ADDRESS BUF
00B3 208 BUF:
000002B3 00B3 209 .BLKB 512
02B3 210 MESSAGEL:
00000000 02B3 211 .LONG 0 ; message desc.
000000B3' 02B7 212 .ADDRESS BUF
02BB 213 SERV_NAME:
00000000 02BB 214 .LONG 0 ; service name pointer
```


00000000	02BF	215	MSGLEN:			
	02BF	216		.LONG	0	; message length parameter
00000000	02C3	217	OUTADR:			
	02C3	218		.LONG	0	; output address parameter
00000100	02C7	219	BUFADR:			
000002CF	02C7	220		.LONG	256	; GETMSG & PUTMSG buffer
000003CF	02CB	221		.ADDRESS	.+4	
	02CF	222		.BLKB	256	; maximum message length
	03CF	223	BADATA:			
00000000	03CF	224		.LONG	0	; bad string desc.
000002CF	03D3	225		.ADDRESS	BUFADR+8	
	03D7	226	GOODATA:			
00000000	03D7	227		.LONG	0	; expected string desc.
00000000	03DB	228		.ADDRESS	0	
	03DF	229	MSGVEC1:			
00000003	03DF	230		.LONG	3	; PUTMSG test message vector
007480E1	03E3	231		.LONG	UETPS_DDB	
00000001	03E7	232		.LONG	1	
000003EF	03EB	233		.ADDRESS	DEV_NAM	
	03EF	234	DEV_NAM:			
41 44 54 000003F7'010E0000'	03EF	235		.ASCID	/TDA/	; test device name
	03FA	236	MSG:			
00000000	03FA	237		.LONG	0	; correct message pointer


```
00000000 239      .PSECT SATSSS10, RD, WRT, EXE, LONG
0000      240      .SBTIL SATSSS10
0000      241      :++
0000      242      : FUNCTIONAL DESCRIPTION:
0000      243      :
0000      244      :     After performing some initial housekeeping, such as
0000      245      :     printing the module begin message and acquiring needed privileges,
0000      246      :     the system services are tested in each of their normal conditions.
0000      247      :     Detected failures are identified and an error message is printed
0000      248      :     on the terminal. Upon completion of the test a success or fail
0000      249      :     message is printed on the terminal.
0000      250      :
0000      251      : CALLING SEQUENCE:
0000      252      :
0000      253      :     $ RUN SATSSS10 ... (DCL COMMAND)
0000      254      :
0000      255      : INPUT PARAMETERS:
0000      256      :
0000      257      :     none
0000      258      :
0000      259      : IMPLICIT INPUTS:
0000      260      :
0000      261      :     none
0000      262      :
0000      263      : OUTPUT PARAMETERS:
0000      264      :
0000      265      :     none
0000      266      :
0000      267      : IMPLICIT OUTPUTS:
0000      268      :
0000      269      :     Messages to SYS$OUTPUT are the only output from SATSSS10.
0000      270      :     They are of the form:
0000      271      :
0000      272      :         %UETP-S-SATSMS, TEST MODULE SATSSS10 BEGUN ... (BEGIN MSG)
0000      273      :         %UETP-S-SATSMS, TEST MODULE SATSSS10 SUCCESSFUL ... (END MSG)
0000      274      :         %UETP-E-SATSMS, TEST MODULE SATSSS10 FAILED ... (END MSG)
0000      275      :         %UETP-I-TEXT, ... (VARIABLE INFORMATION ABOUT A TEST MODULE FAILURE)
0000      276      :
0000      277      : COMPLETION CODES:
0000      278      :
0000      279      :     The SATSSS10 routine terminates with a $EXIT to the
0000      280      :     operating system with a status code defined by UETP$_SATSMS.
0000      281      :
0000      282      : SIDE EFFECTS:
0000      283      :
0000      284      :     none
0000      285      :
0000      286      : --
0000      287      :
0000      288      : TEST_START SATSSS10                ; let the test begin
```



```
0000 0000
0004'CF 00 DD 0002
0000'CF DF 0006
00000000'GF 02 FB 000C
00000000'GF 00 FB 0013
0009'CF 7F 001A
00000000'GF 01 FB 001E
034F 30 0025
004C'CF 001F'CF DE 0028
0044'CF 03 00 01 FO 002F
019B'CF 01 FB 0036
003D 0038
```

.LIST ME

```
.ENTRY SATSSS10,0
CLRL W^CURRENT_TC
PUSHL #0
PUSHAL W^TPID
CALLS #2,G^SYSS$WAKE
CALLS #0,G^SYSS$HIBER
PUSHAQ W^TEST MOD NAME_D
CALLS #1,G^SYSS$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 GETMSG TESTS

\$GETMSG tests

The tests are executed in the following order:

STP #	FLAG value	message content
STP0	0	no message
STP1	1	only text
STP2	2	only mesg ID
STP3	3	text and mesg ID
STP4	4	only severity code
STP5	5	severity code & text
STP6	6	severity code & mesg ID
STP7	7	severity code, mesg ID, & text
STP8	8	only facility name
STP9	9	facility name & text
STP10	10	facility name & mesg ID
STP11	11	facility name, mesg ID, & text
STP12	12	facility name & severity code
STP13	13	facility name, severity code, & text
STP14	14	facility name, severity code, & mesg ID
STP15	15	every thing


```
02BB'CF 0031'CF DE 003D 318      MOVAL W^GETMSG,W^SERV_NAME      ; set service name
56      0141'CF DE 0044 319      MOVAL W^MSGTBL,R6          ; set table pointer
57      01      DO 0049 320      MOVL #1,R7                ; set initial flag value

03FA'CF 86      DO 004C 321      GETMSG_LOOP:              ;
00      00      DD 0051 322      MOVL (R6)+,W^MSG          ; get string pointer from table
019B'CF 01      FB 0053 323      PUSHL #0                  ; save a dummy parameter
0058 324      CALLS #1,W^REG_SAVE                          ; save a register snapshot
0058 325      $GETMSG_S MSGID=W^MSGID,-
0058 326      MSGLEN=W^MSGLEN,-
0058 327      BUFADR=W^BUFADR,-
0058 328      FLAGS=R7,-
0058 329      OUTADR=W^OUTADR
0071 330      FAIL_CHECK SSS_NORMAL                        ; check for success
00000000'8F DD 0071 331      PUSHL #SS$ NORMAL
01A5'CF 01      FB 0077 332      CALLS #1,W^REG_CHECK
0286'CF 00      FB 007C 333      CALLS #0,W^CHECK_RESULTS ; check returned message
0079'CF 57      DO 0081 334      MOVL R7,W^GET+GETMSG$_FLAGS ; set flag value
00      00      DD 0086 335      PUSHL #0                  ; save a dummy
019B'CF 01      FB 0088 336      CALLS #1,W^REG_SAVE        ; save a register snapshot
008D 337      $GETMSG G W^GET                                ; try G form
0096 338      FAIL_CHECK SSS_NORMAL                        ; check for success
00000000'8F DD 0096 339      PUSHL #SS$ NORMAL
01A5'CF 01      FB 009C 340      CALLS #1,W^REG_CHECK
0286'CF 00      FB 00A1 341      CALLS #0,W^CHECK_RESULTS ; check returned message
0004'CF D6      00A6 342      INCL W^CURRENT TC           ; bump the test number
9E 57 0E      F3 00AA 343      AOBLEQ #14,R7,B^GETMSG_LOOP ; bump the flag & do it again
0000000F 00AE 344      STEP=STEP+15
```



```
00AE 342 .SBTTL PUTMSG TESTS
00AE 343 :+
00AE 344 :
00AE 345 $PUTMSG tests
00AE 346 :
00AE 347 The tests are executed if the following order:
00AE 348 :
00AE 349 STP # FLAG value message content
00AE 350 -----
00AE 351 STP17 0 no message
00AE 352 STP18 1 only text
00AE 353 STP19 2 only mesg ID
00AE 354 STP20 3 text and mesg ID
00AE 355 STP21 4 only severity code
00AE 356 STP22 5 severity code & text
00AE 357 STP23 6 severity code & mesg ID
00AE 358 STP24 7 severity code, mesg ID, & text
00AE 359 STP25 8 only facility name
00AE 360 STP26 9 facility name & text
00AE 361 STP27 10 facility name & mesg ID
00AE 362 STP28 11 facility name, mesg ID, & text
00AE 363 STP29 12 facility name & severity code
00AE 364 STP30 13 facility name, severity code, & text
00AE 365 STP31 14 facility name, severity code, & mesg ID
00AE 366 STP32 15 every thing
00AE 367 :
00AE 368 :-
00AE 369 :
00AE 370 NEXT_TEST
00AE 371
00AE 372 STP16:
00AE 373 MOVL #16,W^CURRENT_TC
00AE 374 PUSHL #0
00AE 375 CALLS #1,W^REG_SAVE
00AE 376 MOVAL W^PUTMSG,W^SERV_NAME ; set service name
00AE 377 MOVAL W^MSGTBL1,R6 ; set table pointer
00AE 378 CLRL R7 ; set initial flag value
00AE 379 PUTMSG_LOOP:
00AE 380 MOVL (R6)+,W^MSG ; get string pointer from table
00AE 381 MOVW R7,W^MSGVEC1+10 ; set a new flag value
00AE 382 PUSHL #0 ; save a dummy parameter
00AE 383 CALLS #1,W^REG_SAVE ; save a register snapshot
00AE 384 $PUTMSG_S MSGVEC=W^MSGVEC1,-
00AE 385 ACTRTN=W^ACT ; try it
00AE 386 FAIL_CHECK SS$ _NORMAL ; check for success
00AE 387 PUSHL #SS$ _NORMAL
00AE 388 CALLS #1,W^REG_CHECK
00AE 389 PUSHL #0 ; save a dummy
00AE 390 CALLS #1,W^REG_SAVE ; save a register snapshot
00AE 391 $PUTMSG G W^PUT ; try G form
00AE 392 FAIL_CHECK SS$ _NORMAL ; check for success
00AE 393 PUSHL #SS$ _NORMAL
00AE 394 CALLS #1,W^REG_CHECK
00AE 395 INCL W^CURRENT_TC ; bump the test number
00AE 396 AOBLER #15,R7,B^PUTMSG_LOOP ; bump the flag & do it again
00AE 397 :
00AE 398 :
00AE 399 :
00AE 400 :
```

```
0004'CF 10 DO 00AE
00 00 DD 00B3
019B'CF 01 FB 00B5
02BB'CF 0038'CF DE 00BA
56 017D'CF DE 00C1
57 D4 00C6
00C8
03FA'CF 86 DO 00C8
03E9'CF 57 BO 00CD
00 DD 00D2
019B'CF 01 FB 00D4
00D9
00D9
00EC 381
00000000'8F DD 00EC
01A5'CF 01 FB 00F2
00 DD 00F7
019B'CF 01 FB 00F9
00FE
C107 385
00000000'8F DD 0107
01A5'CF 01 FB 010D
0004'CF D6 0112
AE 57 OF F3 0116
011A 388 :+
011A 389 :
```



```
011A 390 : test the facility name parameter
011A 391 :-
011A 392 :-
011A 393 NEXT_TEST
011A
011A STP17:
0004'CF 11 DD 011A MOVL #17,W^CURRENT_TC
00 DD 011F PUSHL #0
019B'CF 01 FB 0121 CALLS #1,W^REG_SAVE
03FA'CF 86 DD 0126 394 MOVL (R6)+,W^MSG ; get string pointer from table
00 DD 012B 395 PUSHL #0 ; save a dummy
019B'CF 01 FB 012D 396 CALLS #1,W^REG_SAVE ; save a reg snapshot
0132 397 $PUTMSG_S MSGVEC=W^MSGVEC1,-
0132 398 ACTRTN=W^ACT,-
0132 399 FACNAM=W^TEST ; try the _S form
0147 400 FAIL_CHECK SSS_NORMAL ; check for success
00000000'8F DD 0147 PUSHL #SS$ NORMAL
01A5'CF 01 FB 014D CALLS #1,W^REG_CHECK
00 DD 0152 401 PUSHL #0 ; save a dummy
019B'CF 01 FB 0154 402 CALLS #1,W^REG_SAVE ; save a reg snapshot
008D'CF 04FB'CF DE 0159 403 MOVAL W^TEST,W^PUT+PUTMSG$_FACNAM ; set facility name adr
0160 404 $PUTMSG G W^PUT ; try the _G form
0169 405 FAIL_CHECK SSS_NORMAL ; check for success
00000000'8F DD 0169 PUSHL #SS$ NORMAL
01A5'CF 01 FB 016F CALLS #1,W^REG_CHECK
0174 406 TEST_END
004C'CF DD 0174 PUSHL W^TMD_ADDR
0048'CF DD 0178 PUSHL W^TMN_ADDR
02 DD 017C PUSHL #2
0044'CF DD 017E PUSHL W^MOD_MSG_CODE
00000000'GF 04 FB 0182 CALLS #SS$T1,G^LIB$SIGNAL
0044'CF 01 1C 01 FO 0189 INSV #1,#ST$V_INHIB_MSG,#1,W^MOD_MSG_CODE
00000000'GF 01 FB 0190 PUSHL W^MOD_MSG_CODE
0194 CALLS #1,G^SYS$EXIT
```



```
019B 408 .SBTTL REG_SAVE
019B 409 :++
019B 410 : FUNCTIONAL DESCRIPTION:
019B 411 : Subroutine to save R2-R11 in the register save location.
019B 412 :
019B 413 : CALLING SEQUENCE:
019B 414 :     PUSHL #0 ; save a dummy parameter
019B 415 :     CALLS #1,W^REG_SAVE ; save R2-R11
019B 416 :
019B 417 : INPUT PARAMETERS:
019B 418 :     NONE
019B 419 :
019B 420 : OUTPUT PARAMETERS:
019B 421 :     NONE
019B 422 :
019B 423 :--
019B 424 :
019B 425 REG_SAVE:
019B 426 .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
0008'CF 14 AD 28 OFFC 019D 427 MOVCL #4*10,^X14(FP),W^REG_SAVE_AREA ; save the registers in the program
01A4 428 RET
01A5 429 .SBTTL REG_CHECK
01A5 430 :++
01A5 431 : FUNCTIONAL DESCRIPTION:
01A5 432 : Subroutine to test R0 & R2-R11 for proper content after a service
01A5 433 : execution. A snapshot is taken by the REG_SAVE routine at the
01A5 434 : beginning of each step and this routine is executed after the
01A5 435 : services have been executed.
01A5 436 :
01A5 437 : CALLING SEQUENCE:
01A5 438 :     PUSHL #SS$ XXXXXX ; push expected R0 contents
01A5 439 :     CALLS #1,W^REG_CHECK ; execute this routine
01A5 440 :
01A5 441 : INPUT PARAMETERS:
01A5 442 :     expected R0 contents on the stack
01A5 443 :
01A5 444 : OUTPUT PARAMETERS:
01A5 445 :     possible error messages printed using $PUTMSG
01A5 446 :
01A5 447 :--
01A5 448 :
01A5 449 REG_CHECK:
01A5 450 .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
01A7 451 CMPL 4(AP),R0 ; is this the right fail code?
01AB 452 BEQL 10$ ; br if yes
01AD 453 PUSHL R0 ; push received data
01AF 454 PUSHL 4(AP) ; push expected data
01B2 455 PUSHAL W^EXP ; push the string variable
01B6 456 CALLS #3,W^PRINT_FAIL ; print the error message
01BB 457 10$:
01BB 458 CMPC3 #4*10,^X14(FP),W^REG_SAVE_AREA ; check all but R0
01C2 459 BEQL 20$ ; br if O.K.
01C4 460 SUBL3 #REG_SAVE_AREA,R3,R6 ; calculate the register number
01CC 461 DIVL2 #4,R6
01CF 462 ADDB3 #^X2,R6,-(SP) ; set number past R0-R1 and save
01D3 463 BICL2 #3,R1 ; backup to register boundrys
01D6 464 BICL2 #3,R3
```


SATSSS10
V04-000

- SATS SYSTEM SERVICE TESTS (SUCC S.C.)
REG_CHECK

N 7

16-SEP-1984 01:43:51
5-SEP-1984 04:22:41

VAX/VMS Macro V04-00
[UETP.SRC]SATSSS10.MAR;1

Page 15
(1)

01E7	CF	61	DD	01D9	465	PUSHL	(R1)	:	push received data
		63	DD	01DB	466	PUSHL	(R3)	:	push expected data
	0095	CF	DF	01DD	467	PUSHAL	W^REG	:	set string pntr param.
		04	FB	01E1	468	CALLS	#4,W^FRINT_FAIL	:	print the error message
				01E6	469				
			04	01E6	470	RET			

20\$:


```
01E7 472 .SBTTL PRINT_FAIL
01E7 473 :++
01E7 474 : FUNCTIONAL DESCRIPTION:
01E7 475 : Subroutine to report failures using $PUTMSG
01E7 476 :
01E7 477 : CALLING SEQUENCE:
01E7 478 : Mode #1          PUSHL EXPECTED Mode #2          PUSHL REG NUMBER
01E7 479 :                   PUSHL RECEIVED                   PUSHL EXPECTED
01E7 480 :                   PUSHAL STRING VAR                   PUSHL RECEIVED
01E7 481 :                   CALLS #3,W^PRINT_FAIL               PUSHAL STRING VAR
01E7 482 :                                                     CALLS #4,W^PRINT_FAIL
01E7 483 : INPUT PARAMETERS:
01E7 484 : Listed above
01E7 485 :
01E7 486 : OUTPUT PARAMETERS:
01E7 487 : an error message is printed using $PUTMSG
01E7 488 :
01E7 489 :--
01E7 490
01E7 491 PRINT_FAIL:
003C 01E7 492 .WORD ^M<R2,R3,R4,R5>
01E9 493 $FAO S W^CS1,W^MESSAGEL,W^MSGL,#TEST_MOD_NAME,W^SERV_NAME,W^CURRENT_TC
020A 494 $PUTMSG_S W^MSGVEC ; print the message
04 6C 91 021B 495 CMPB (AP),#4 ; is this a register message?
21 13 021E 496 BEQL 10$ ; br if yes
25 11 0220 497 $FAO_S W^CS2,W^MESSAGEL,W^MSGL,4(AP),8(AP),4(AP),12(AP)
023F 498 BRB 20$ ; goto output message
0241 499 10$:
0241 500 $FAO_S W^CS3,W^MESSAGEL,W^MSGL,4(AP),16(AP),8(AP),4(AP),16(AP),12(AP)
0266 501 20$:
0266 502
0266 503 $PUTMSG_S W^MSGVEC ; print the message
0044'CF 03 002A'CF DE 0277 504 MOVAL W^TEST_MOD_FAIL,W^TMD_ADDR ; set failure message address
00 02 FO 027E 505 INSV #ERROR,#0,#3,W^MOD_MSG_CODE ; set severity code
04 0285 506 RET
```



```
0286 508 .SBTTL CHECK_RESULTS
0286 509 :++
0286 510 : FUNCTIONAL DESCRIPTION:
0286 511 : Routine to check message content, message length, and OUTADR.
0286 512 :
0286 513 : CALLING SEQUENCE:
0286 514 : CALLS #0,W^CHECK_RESULTS ; check returned message
0286 515 :
0286 516 : INPUT PARAMETERS:
0286 517 : MSG = pointer to correct resultant counted string
0286 518 : MSGBUF = returned string
0286 519 : OUTADR = outadr parameter returned by the service
0286 520 : MSGLEN = contains returned message length
0286 521 :
0286 522 : OUTPUT PARAMETERS:
0286 523 : NONE
0286 524 :
0286 525 :--
0286 526
0286 527 CHECK_RESULTS:
0286 528 .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
0286 529 MOVZBL @W^MSG,R8 ; get the length
0286 530 CMPB W^MSGLEN,R8 ; is the length OK?
0286 531 BEQL 10$ ; br if yes
0286 532 PUSHL W^MSGLEN ; push received
0286 533 PUSHL R8 ; push expected
0286 534 PUSHAL W^LEN ; push string variable
0286 535 CALLS #3,W^PRINT_FAIL ; print the failure
0286 536 10$:
0286 537 CMPL W^OUTADR,#^X100 ; is OUTADR param OK?
0286 538 BEQL 20$ ; br if OK
0286 539 PUSHL W^OUTADR ; push received
0286 540 PUSHL #^X100 ; push expected
0286 541 PUSHAL W^OUT ; push string variable
0286 542 CALLS #3,W^PRINT_FAIL ; print the failure
0286 543 20$:
0286 544 MOVL W^MSG,R9 ; get message pointer
0286 545 CMPC3 R8,B^1(R9),W^BUFADR+8 ; check the string
0286 546 BNEQ 25$ ; br if not OK
0286 547 BRW 30$ ; BR if OK
0286 548 25$:
0286 549 MOVAL B^1(R9),W^GOODDATA+4 ; fill in good data desc.
0286 550 MOVL R8,W^GOODDATA
0286 551 MOVL W^MSGLEN,W^BADATA ; fill in bad data desc.
0286 552 $FAO_S W^CS1,W^MESSAGEL,W^MSGL,-
0286 553 #TEST_MOD_NAME,W^SERV_NAME,-
0286 554 W^CURRENT_TC ; form 1st line and
0286 555 $PUTMSG S W^MSGVEC ; print it
0286 556 $FAO_S W^CS4,W^MESSAGEL,W^MSGL,-
0286 557 #GOODDATA,#BADATA ; form 2nd line and
0286 558 $PUTMSG S W^MSGVEC ; print it
0286 559 MOVAL W^TEST_MOD_FAIL,W^TMD_ADDR ; set failure message address
0286 560 INSV #ERROR,#0,#3,W^MOD_MSG_CODE ; set severity code
0286 561 30$:
0286 562 RET
```

58 03FA'DF 9A 0288 529
58 02BF'CF 91 028D 530
02BF'CF DD 0292 531
58 DD 0294 532
0111'CF DF 0298 533
FF44 CF 03 FB 029A 534
02C3'CF D1 029E 535
13 13 02A3 536
02C3'CF DD 02AC 537
00000100 8F DD 02AE 538
011F'CF DD 02B2 539
FF26 CF 03 FB 02B8 540
59 03FA'CF D0 02BC 541
02CF'CF 01 A9 58 29 02C1 542
03 12 02C1 543
0082 31 02C6 544
03DB'CF 01 A9 DE 02CD 545
03D7'CF 58 D0 02CF 546
03CF'CF 02BF'CF D0 02D2 547
02D2 548
02D8 549
02DD 550
02E4 551
02E4 552
02E4 553
0305 554
0316 555
0316 556
0335 557
004C'CF 002A'CF DE 0346 558
0044'CF 03 00 02 FO 034D 559
04 0354 560
0354 561
0354 562


```
0355 564 .SBTTL ACT
0355 565 :++
0355 566 : FUNCTIONAL DESCRIPTION:
0355 567 : Routine to take action on a PUTMSG call. The message is checked
0355 568 : for proper content and printing is disabled.
0355 569 :
0355 570 : CALLING SEQUENCE:
0355 571 : Entered via ACTRTN parameter specified with PUTMSG.
0355 572 :
0355 573 : INPUT PARAMETERS:
0355 574 : MSG = pointer to correct resultant counted string
0355 575 : MSGBUF = returned string
0355 576 :
0355 577 : OUTPUT PARAMETERS:
0355 578 : NONE
0355 579 :
0355 580 :--
0355 581
0355 582 ACT:
0355 583 .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
0357 584 MOVL 4(AP),R6 ; get the desc. address
0358 585 MOVL (R6),W^MSGLEN ; set length for CHECK_RESULTS
0360 586 MOVL 4(R6),R7 ; get string pointer
0364 587 MOVC3 (R6),(R7),W^BUFADR+8 ; get string for CHECK_RESULTS
036A 588 CALLS #0,W^CHECK_RESULTS ; check the PUTMSG results
036F 589 MOVL #SS$_ABORT,R0 ; don't really print it
0376 590 RET
```

56	04	AC	DO	0355	583
02BF	'CF	66	DO	0357	584
57	04	A6	DO	0358	585
02CF	'CF	67	28	0360	586
FF17	CF	00	FB	0364	587
50	00000000	'8F	DO	036A	588
			04	036F	589
				0376	590


```
0377 592 MOD_MSG_PRINT:
0377 593 :
0377 594 : *****
0377 595 : *
0377 596 : * PRINTS THE TEST MODULE BEGUN/SUCCESSFUL/FAILED MESSAGES *
0377 597 : * (USING THE PUTMSG MACRO). *
0377 598 : *
0377 599 : *****
0377 600 :
05 0377 601 PUTMSG <MOD_MSG_CODE,#2,TMN_ADDR,TMD_ADDR> ; PRINT MSG
0392 602 RSB ; ... AND RETURN TO CALLER
0393 603 :
0393 604 CHMRTN:
0393 605 : *****
0393 606 : *
0393 607 : * CHANGE MODE ROUTINE. THIS ROUTINE GETS CONTROL WHENEVER *
0393 608 : * A CMKRNL, CMEXEC, OR CMSUP SYSTEM SERVICE IS ISSUED *
0393 609 : * BY THE MODE MACRO ('TO' OPTION). IT MERELY DOES *
0393 610 : * A JUMP INDIRECT ON A FIELD SET UP BY MODE. IT HAS *
0393 611 : * THE EFFECT OF RETURNING TO THE END OF THE MODE *
0393 612 : * MACRO EXPANSION. *
0393 613 : *
0393 614 : *****
0393 615 :
00000059'FF 0000 0393 616 .WORD 0 ; ENTRY MASK
17 0395 617 JMP @CHM_CONT ; RETURN TO MODE MACRO IN NEW MODE
039B 618 :
039B 619 : * RET INSTR WILL BE ISSUED IN EXPANSION OF 'MODE FROM, ....' MACRO
039B 620 :
039B 621 .END SATSSS10
```


SATSSS10
Symbol table

- SATS SYSTEM SERVICE TESTS (SUCC S.C.) 16-SEP-1984 01:43:51 VAX/VMS Macro V04-00
5-SEP-1984 04:22:41 [UETP.SRC]SATSSS10.MAR;1

Page 20
(1)

\$\$ARGS	= 00000004		MOD_MSG_PRINT	00000377 R	04
\$\$T1	= 00000004		MSG	000003FA R	03
\$\$T2	= 00000005		MSGID	0000012D R	02
ACT	00000355 R	04	MSGLEN	000000AB R	03
BADATA	000003CF R	03	MSGTBL	000002BF R	03
BUF	000000B3 R	03	MSGTBL1	00000141 R	02
BUFADR	000002C7 R	03	MSGVEC	0000017D R	02
CHECK_RESULTS	00000286 R	04	MSGVEC1	00000131 R	02
CHMRTN	00000393 R	04	OUT	000003DF R	03
CHM_CONT	00000059 R	03	OUTADR	0000011F R	02
CS1	0000003F R	02	PRINT_FAIL	000002C3 R	03
CS2	00000071 R	02	PRIVMASK	000001E7 R	04
CS3	0000009E R	02	PRVPR	00000051 R	03
CS4	000000D1 R	02	PUT	00000050 R	03
CURRENT_TC	00000004 R	03	PUTMSG	00000081 R	03
DEV_NAM	000003EF R	03	PUTMSG\$	00000038 R	02
ERROR	= 00000002		PUTMSG\$_ACTPRM	= 00000010	
EXP	00000103 R	02	PUTMSG\$_ACTRTN	= 00000008	
GET	00000069 R	03	PUTMSG\$_FACNAM	= 0000000C	
GETMSG	00000031 R	02	PUTMSG\$_MSGVEC	= 00000004	
GETMSG\$_BUFADR	= 0000000C		PUTMSG\$_NARGS	= 00000004	
GETMSG\$_FLAGS	= 00000010		PUTMSG\$_LOOP	000000C8 R	04
GETMSG\$_MSGID	= 00000004		REG	00000095 R	03
GETMSG\$_MSGLEN	= 00000008		REGNUM	000000A7 R	03
GETMSG\$_NARGS	= 00000005		REG_CHECK	000001A5 R	04
GETMSG\$_OUTADR	= 00000014		REG_SAVE	0000019B R	04
GETMSG\$_LOOP	0000004C R	04	REG_SAVE_AREA	00000008 R	03
GOODATA	000003D7 R	03	RETADR	0000005D R	03
INFO	= 00000003		SATSSS10	00000000 RG	04
LEN	00000111 R	02	SERV_NAME	000002BB R	03
LIB\$SIGNAL	***** X	04	SEVERE	= 00000004	
M1	000001C1 R	02	SHR\$K_SHRDEF	= 00000001	
M10	000002AD R	02	SHR\$ TEXT	= 00001130	
M11	000002B7 R	02	SS\$ ABORT	***** X	04
M12	000002E8 R	02	SS\$ NORMAL	***** X	04
M13	000002F0 R	02	STATUS	00000065 R	03
M14	0000031F R	02	STEP	= 00000011	
M15	0000032B R	02	STP0	0000003D R	04
M16	0000035E R	02	STP16	000000AE R	04
M17	00000384 R	02	STP17	0000011A R	04
M18	000003B0 R	02	ST\$V_INHIB_MSG	= 0000001C	
M19	000003DA R	02	SUCCESS	= 00000001	
M2	000001E7 R	02	SYS\$EXIT	***** GX	04
M20	00000408 R	02	SYS\$FAO	***** X	04
M21	00000435 R	02	SYS\$GETMSG	***** GX	04
M22	00000466 R	02	SYS\$HIBER	***** GX	04
M23	00000495 R	02	SYS\$PUTMSG	***** GX	04
M24	000004C8 R	02	SYS\$SETPRN	***** GX	04
M3	000001EC R	02	SYS\$WAKE	***** GX	04
M4	00000218 R	02	TEST	000004FB R	02
M5	0000021B R	02	TEST_MOD_BEGIN	00000019 R	02
M6	00000245 R	02	TEST_MOD_FAIL	0000002A R	02
M7	0000024C R	02	TEST_MOD_NAME	00000000 R	02
M8	0000027A R	02	TEST_MOD_NAME_D	00000009 R	02
M9	00000280 R	02	TEST_MOD_SUCC	0000001F R	02
MESSAGEL	000002B3 R	03	TMD_ADDR	0000004C R	03
MOD_MSG_CODE	00000044 R	03	TMN_ADDR	00000048 R	03

SATSSS10
Symbol table

- SATS SYSTEM SERVICE TESTS (SUCC S.C.) 16-SEP-1984 01:43:51 VAX/VMS Macro V04-00
5-SEP-1984 04:22:41 [UETP.SRC]SATSSS10.MAR;1

Page 21
(1)

TPID 00000000 R 03
UETPS_DDB = 007480E1
UETPS_SATSMS = 007480D9
UETPS_TEXT = 00741133
WARNING = 00000000

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

PSECT name	Allocation	PSECT No.	Attributes
.ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
RODATA	00000507 (1287.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC LONG
RWDATA	000003FE (1022.)	03 (3.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
SATSSS10	0000039B (923.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.10	00:00:00.41
Command processing	107	00:00:00.68	00:00:04.94
Pass 1	263	00:00:06.48	00:00:16.24
Symbol table sort	0	00:00:00.43	00:00:00.54
Pass 2	145	00:00:02.18	00:00:04.37
Symbol table output	15	00:00:00.11	00:00:00.32
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	563	00:00:10.02	00:00:26.86

The working set limit was 1350 pages.
41621 bytes (82 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 349 non-local and 8 local symbols.
621 source lines were read in Pass 1, producing 27 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:[UETP.OBJ]UETP.MLB;1	10
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	21
TOTALS (all libraries)	31

508 GETS were required to define 31 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:SATSSS10/OBJ=OBJ\$:SATSSS10 MSRC\$:SATSSS10/UPDATE=(ENH\$:SATSSS10)+EXECML\$/LIB+LIB\$:UETP/LIB

0410 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY