


```

SSSSSSSS  AAAAAA  TTTTTTTTTT  SSSSSSSS  SSSSSSSS  SSSSSSSS  999999  000000
SSSSSSSS  AAAAAA  TTTTTTTTTT  SSSSSSSS  SSSSSSSS  SSSSSSSS  999999  000000
SS        AA      AA      TT        SS        SS        SS        99      99      00      00
SS        AA      AA      TT        SS        SS        SS        99      99      00      00
SS        AA      AA      TT        SS        SS        SS        99      99      00      0000
SS        AA      AA      TT        SS        SS        SS        99      99      00      0000
SSSSSSS   AA      AA      TT        SSSSSS   SSSSSS   SSSSSS   99999999  00  00  00
SSSSSSS   AA      AA      TT        SSSSSS   SSSSSS   SSSSSS   99999999  00  00  00
SS        AA      AA      TT        SS        SS        SS        99      99      0000  00
SS        AA      AA      TT        SS        SS        SS        99      99      0000  00
SS        AA      AA      TT        SS        SS        SS        99      99      00      00
SS        AA      AA      TT        SS        SS        SS        99      99      00      00
SSSSSSSS  AA      AA      TT        SSSSSSSS  SSSSSSSS  SSSSSSSS  999999  000000
SSSSSSSS  AA      AA      TT        SSSSSSSS  SSSSSSSS  SSSSSSSS  999999  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

```

(1)	54	DECLARATIONS
(1)	95	CONDITION TABLES
(1)	123	TM SETUP, TM CLEANUP
(1)	189	CONDITION SUBROUTINES - SETUP AND CLEANUP
(1)	259	FORM CONDS
(1)	352	VERIFY
(1)	465	VFY CLEANUP
(1)	520	KERNEL OR EXEC MODE ROUTINE

```
0000 1 .TITLE SATSSS90 SATS SYST SERV TESTS $CMKRNL/EXEC (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: SYSTST (SATS SYSTEM SERVICE TESTS)
0000 31 :
0000 32 : ABSTRACT:
0000 33 :
0000 34 : THIS MODULE CONTAINS SUBROUTINES WHICH, WHEN LINKED
0000 35 : WITH SUCCOMMON.OBJ, FORM TEST MODULE SATSSS90 TO TEST SUCCESSFUL
0000 36 : OPERATION OF THE $CMKRNL/EXEC SYSTEM SERVICES. EACH SERVICE IS INVOKED
0000 37 : UNDER VARIOUS INPUT CONDITIONS WITH VARYING INPUT PARAMETERS. ONLY
0000 38 : SUCCESSFUL STATUS CODES ARE EXPECTED IN THIS TEST MODULE. CORRECT
0000 39 : OPERATION OF EACH SERVICE FOR EACH OF ITS ISSUANCES IS VERIFIED BY
0000 40 : CHECKING FOR AN SSS NORMAL STATUS CODE, EXPECTED RETURN ARGUMENTS
0000 41 : AND EXPECTED FUNCTIONALITY PERFORMED.
0000 42 :
0000 43 : ENVIRONMENT: USER MODE IMAC., NEEDS CMKRNL PRIVILEGE,
0000 44 : DYNAMICALLY ACQUIRES OTHER PRIVILEGES, AS NEEDED.
0000 45 :
0000 46 : AUTHOR: THOMAS L. CAFARELLA, CREATION DATE: MAY, 1978
0000 47 :
0000 48 : MODIFIED BY:
0000 49 :
0000 50 : : VERSION
0000 51 : 01 -
0000 52 : --
```

```
0000 54 .SBTTL DECLARATIONS
0000 55 :
0000 56 : INCLUDE FILES:
0000 57 :
0000 58 $PRVDEF ; PRIVILEGE BIT DEFINITIONS
0000 59 $PHDDEF ; PROCESS HEADER OFFSETS
0000 60 $PSLDEF ; MODE SYMBOL DEFINITIONS
0000 61 $LOGDEF ; LOG NAME TABLE DEFINITIONS
0000 62 :
0000 63 : MACROS:
0000 64 :
0000 65 :
0000 66 : EQUATED SYMBOLS:
0000 67 :
0000000A 0000 68 RLEN = 10 ; NO. OF RANDOM LWORDS FOR ARG LIST
0000003C 0000 69 STRING_MASK = ^M<R2,R3,R4,R5> ; REGISTER SAVE MASK FOR CMPC INSTRUCTION
0000 70 :
0000 71 : OWN STORAGE:
0000 72 :
```

SATSSS90
V04-000

```
00000000 74 .PSECT RODATA, RD, NOWRT, NOEXE, LONG
0000 75 TEST_MOD_NAME:: STRING C, <SATSSS90> : TEST MODULE NAME
0009 76 TEST_MOD_NAME_D: STRING I, <SATSSS90> : TEST MODULE NAME DESCRIPTOR
0019 77 MSG1_INP_CTL: STRING I, <SSCHM!4ZW: CONDITIONS:>
0039 78 : FAO CTL STRING FOR MSG1 IN SUCCOMMON.MAR
0039 79 MSG3_ERR_CTL:: STRING I, <*SSCHM!4ZW: AS>
0051 80 : FAO CTL STRING FOR MSG3 IN SUCCOMMON.MAR
00000000 0051 81 ARGLSTO: .LONG 0 : ZERO LENGTH ARG LIST
00000002 0055 82 PMODE_LOG: .LONG 2 : UNIQUE LOGICAL NAME ...
00000000 0059 83 .ADDRESS TESTNUM : ... FOR EACH TEST CASE
```

00000000	85	.PSECT	RWDATA,RD,WRT,NOEXE,LONG		
00000008	0000	86 PRIVMASK:	.BLKQ 1	:	ADDR OF PRIVILEGE MASK (IN PHD)
0000000C	0008	87 ARGLST:	.BLKL 1	:	HOLDS CURRENT COND 1 TABLE ENTRY
0000000A	000C	88 ARGLSTR:	.LONG RLEN	:	ARG LIST TO CONTAIN ...
00000038	0010	89	.BLKL RLEN	:	... RANDOM DATA
00000040	0038	90 ARGLSTDESC1:	.BLKL 2	:	DESCRIPTOR FOR ARG LIST (BEFORE RTN CALL)
	0040	91 ARGLSTDESC2:	STRING 0,4*<RLEN+1>	:	DESCR'R FOR ARG LIST (DURING RTN CALL)
00000075	0074	92 RTNMODE:	.BLKB 1	:	MODE IN WHICH PRIV MODE RTN GETS CONTROL
	0075	93 PMODE_EQL:	STRING 0,1	:	EQUIVALENCE NAME STRING

```
007E 95 .SBTTL CONDITION TABLES
007E 96 :
007E 97 :
007E 98 :
007E 99 :
007E 100 ***** CONDITION TABLES FOR CMKRNL/CMEXEC SYSTEM SERVICES *****
007E 101 COND 1,NOTARG,<ARGUMENT LIST TYPE>,-
007E 102 <ARG LIST NOT SPECIFIED>,-
007E 103 <ARG LIST SPECIFIED WITH 0 ARGS>,-
007E 104 <ARG LIST SPECIFIED WITH SEVERAL ARGS>,-
00000000 00F9 .LONG 0 ; ARG LIST NOT SPECIFIED
00000051' 00FD .ADDRESS ARGLSTO ; ARG LIST WITH 0 ARGS
0000000C' 0101 .ADDRESS ARGLSTR ; ARG LIST WITH SEVERAL ARGS
0105 107 :
0105 108 COND 2,NOTARG,<PRIVILEGED MODE SERVICE>,-
0105 109 <$CMKRNL>,-
0105 110 <$CMEXEC>,-
0105 111
00 0136 112 .BYTE PSL$C_KERNEL ; $CMKRNL SERVICE
01 0137 113 .BYTE PSL$C_EXEC ; $CMEXEC SERVICE
0138 114 :
0138 115 COND 3,NULL
0139 116 COND 4,NULL
0139 117 COND 5,NULL
013A 118
013A 119
013B 120
00000000 121 .PSECT SATSSS90,RD,WRT,EXE
```



```

0000 123 .SBTTL TM_SETUP, TM_CLEANUP
0000 124 :++
0000 125 : FUNCTIONAL DESCRIPTION:
0000 126 :
0000 127 : TM SETUP AND TM CLEANUP ARE CALLED TO PERFORM
0000 128 : REQUIRED HOUSEKEEPING AT THE BEGINNING AND END, RESPECTIVELY, OF
0000 129 : TEST MODULE EXECUTION.
0000 130 :
0000 131 : CALLING SEQUENCE:
0000 132 :
0000 133 : BSBW TM_SETUP BSBW TM_CLEANUP
0000 134 :
0000 135 : INPUT PARAMETERS:
0000 136 :
0000 137 : NONE
0000 138 :
0000 139 : IMPLICIT INPUTS:
0000 140 :
0000 141 : NONE
0000 142 :
0000 143 : OUTPUT PARAMETERS:
0000 144 :
0000 145 : NONE
0000 146 :
0000 147 : IMPLICIT OUTPUTS:
0000 148 :
0000 149 : TM_SETUP: COND TABLE INDEX REGISTERS (R2,3,4,5,6) CLEARED;
0000 150 : ALL PRIVILEGES ACQUIRED.
0000 151 :
0000 152 : COMPLETION CODES:
0000 153 :
0000 154 : EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
0000 155 :
0000 156 : SIDE EFFECTS:
0000 157 :
0000 158 : SS CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
0000 159 : (VIA RSB) IF ERROR ENCOUNTERED.
0000 160 :
0000 161 :--
0000 162 :
0000 163 :
0000 164 :
0000 165 :

```

```

00000000'EF 00000000'EF 52 D4 0000 166 TM_SETUP:: CLRL R2 ; INITIALIZE
03 00 00000000'8F 53 D4 0002 167 CLRL R3 ; .. CONDITION
00000000'EF 54 D4 0004 168 CLRL R4 ; .... TABLE
00000000'EF 55 D4 0006 169 CLRL R5 ; ..... INDEX
00000000'EF 56 D4 0008 170 CLRL R6 ; ..... REGISTERS
00000000'EF 59 00000000'9F 30 000A 171 BSBW MOD MSG PRINT ; PRINT TEST MODULE BEGIN MSG
00000000'EF 69 00000000'9F DE 000D 172 MOVAL TEST_MOD_SUCC,TMD_ADDR ; ASSUME END MSG WILL SHOW SUCCESS
00000000'EF 69 00000000'9F FO 0018 173 INSV #SUCCESS,#0,#3,MOD_MSG_CODE ; ADJUST STATUS CODE FOR SUCCESS
00000000'EF 69 00000000'9F 0020 174
00000000'EF 69 00000000'9F 0025 174 MODE TO,5$,KRNL ; KERNEL MODE TO ACCESS PHD
00000000'EF 69 00000000'9F DO 0048 175 MOVL @#CTL$GL,PHD,R9 ; GET PROCESS HEADER ADDRESS
00000000'EF 69 00000000'9F DE 004F 176 MOVAL PHD$Q,PRIVMSK(R9),PRIVMASK ; GET PRIV MASK ADDRESS
00000000'EF 69 00000000'9F 0056 177 MODE FROM,5$ ; BACK TO USER MODE
00000000'EF 69 00000000'9F 0057 178 PRIV ADD,ALL ; GET ALL PRIVILEGES

```

00000010'EF	FF47 CF	3C	BB	0077	179	\$SETPRN S TEST MOD_NAME_D	; SET PROCESS NAME
		28	28	0084	180	SS CHECK NORMAL	; CHECK STATUS CODE RETURNED FROM SETPRN
		3C	BA	00B2	181	PUSHR #STRING_MASK	; SAVE REGS R2-R5 FOR MOV C3
			05	00B4	182	MOV C3 #4*RLEN, TM_SETUP, ARG LSTR+4	; GET 'RLEN' LWORDS OF RANDOM DATA
				00BE	183	POPR #STRING_MASK	; RESTORE REGS R2-R5
				00C0	184	RSB	; RETURN TO MAIN ROUTINE
				00C1	185	TM_CLEANUP::	
	FF3C'		30	00C1	186	BSBW MOD_MSG_PRINT	; PRINT TEST MODULE END MSG
			05	00C4	187	RSB	; RETURN TO MAIN ROUTINE

```

00C5 189          .SBTTL  CONDITION SUBROUTINES - SETUP AND CLEANUP
00C5 190          :++
00C5 191          : FUNCTIONAL DESCRIPTION:
00C5 192          :
00C5 193          :           CONDX AND CONDX CLEANUP ARE SUBROUTINES WHICH ARE EXECUTED
00C5 194          : BEFORE AND AFTER THE VERIFY SOBROUTINE, RESPECTIVELY, WHENEVER A NEW
00C5 195          : CONDITION X VALUE IS SELECTED (SEE FUNCTIONAL DESCRIPTION OF SUCCOMMON
00C5 196          : ROUTINE IN SUCCOMMON.MAR). ANY SETUP FUNCTION PARTICULAR TO THE
00C5 197          : CONDITION X TABLE IS INCLUDED IN THE CONDX SUBROUTINE AND CLEANED
00C5 198          : UP, IF NECESSARY, IN THE CONDX CLEANUP SUBROUTINE. THIS INCLUDES,
00C5 199          : ESPECIALLY, CODE TO DETECT CONFLICTS AMONG CURRENT ENTRIES IN TWO
00C5 200          : OR MORE CONDITION TABLES. IF A CONFLICT IS DETECTED, A NON-ZERO
00C5 201          : VALUE IS STORED INTO CONFLICT, WHICH CAUSES THE CALLING ROUTINE
00C5 202          : (SUCCOMMON) TO SKIP THE CURRENT ENTRY IN THE CONDITION X TABLE.
00C5 203          :
00C5 204          : CALLING SEQUENCE:
00C5 205          :
00C5 206          :           BSBW CONDX   BSBW CONDX_CLEANUP
00C5 207          :           WHERE X = 1,2,3,4,5
00C5 208          :
00C5 209          : INPUT PARAMETERS:
00C5 210          :
00C5 211          :           CONFLICT = 0
00C5 212          :
00C5 213          : IMPLICIT INPUTS:
00C5 214          :
00C5 215          :           R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
00C5 216          :           FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
00C5 217          :
00C5 218          : OUTPUT PARAMETERS:
00C5 219          :
00C5 220          :           CONFLICT SET TO NON-ZERO IF COND TABLE CONFLICT DETECTED.
00C5 221          :
00C5 222          : IMPLICIT OUTPUTS:
00C5 223          :
00C5 224          :           R2,3,4,5,6 PRESERVED
00C5 225          :
00C5 226          : COMPLETION CODES:
00C5 227          :
00C5 228          :           NONE
00C5 229          :
00C5 230          : SIDE EFFECTS:
00C5 231          :
00C5 232          :           NONE
00C5 233          :
00C5 234          : --
00C5 235          :
00C5 236          :
00C5 237          :
05 00C5 238 COND1::          RSB          ; RETURN TO MAIN ROUTINE
00C5 239          :
05 00C6 240 COND1_CLEANUP::  RSB          ; RETURN TO MAIN ROUTINE
00C5 241          :
00C5 242 COND2::          RSB          ; RETURN TO MAIN ROUTINE
05 00C7 243          :
00C5 244 COND2_CLEANUP::  RSB          ; RETURN TO MAIN ROUTINE
05 00C8 245          :

```

```

05 00C9 246 COND3::
05 00C9 247 RSB ; RETURN TO MAIN ROUTINE
05 00CA 248 COND3_CLEANUP::
05 00CA 249 RSB ; RETURN TO MAIN ROUTINE
05 00CB 250 COND4::
05 00CB 251 RSB ; RETURN TO MAIN ROUTINE
05 00CC 252 COND4_CLEANUP::
05 00CC 253 RSB ; RETURN TO MAIN ROUTINE
05 00CD 254 COND5::
05 00CD 255 RSB ; RETURN TO MAIN ROUTINE
05 00CE 256 COND5_CLEANUP::
05 00CE 257 RSB ; RETURN TO MAIN ROUTINE
```

```

00CF 259 .SBTTL FORM_CONDS
00CF 260 :++
00CF 261 : FUNCTIONAL DESCRIPTION:
00CF 262 :
00CF 263 :           FORM_CONDS FORMATS AND PRINTS INFORMATION ABOUT
00CF 264 :           THE CURRENT ELEMENT IN EACH OF THE CONDITION TABLES.
00CF 265 :
00CF 266 : CALLING SEQUENCE:
00CF 267 :
00CF 268 :           BSBW FORM_CONDS
00CF 269 :
00CF 270 : INPUT PARAMETERS:
00CF 271 :
00CF 272 :           NONE
00CF 273 :
00CF 274 : IMPLICIT INPUTS:
00CF 275 :
00CF 276 :           R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
00CF 277 :           FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
00CF 278 :           FOR X = 1,2,3,4,5 :
00CF 279 :           CONDX_T - TITLE TEXT FOR CONDX TABLE
00CF 280 :           CONDX_TAB - ELEMENT TEXT FOR CONDX TABLE
00CF 281 :           CONDX_C - CONTEXT OF THE CONDX TABLE
00CF 282 :           CONDX_E - DATA ELEMENTS OF THE CONDX TABLE
00CF 283 :
00CF 284 : OUTPUT PARAMETERS:
00CF 285 :
00CF 286 :           NONE
00CF 287 :
00CF 288 : IMPLICIT OUTPUTS:
00CF 289 :
00CF 290 :           NONE
00CF 291 :
00CF 292 : COMPLETION CODES:
00CF 293 :
00CF 294 :           NCNE
00CF 295 :
00CF 296 : SIDE EFFECTS:
00CF 297 :
00CF 298 :           NONE
00CF 299 :
00CF 300 :--
00CF 301 :
00CF 302 :
00CF 303 :
00CF 304 FORM_CONDS::
00CF 305 $FAO_S MSG1_INP_CTL,FAO_LEN,FAO_DESC,TESTNUM
00EE 306 : FORMAT CONDITIONS HEADER MSG
14 FF0F' 30 00EE 307 BSBW OUTPUT_MSG : ... AND PRINT IT
00 91 00F1 308 CMPB #COND1_C,#NULL : IS CONDITION 1 NULL ?
03 12 00F4 309 BNEQU 10$ : NO -- CONTINUE
00BF 31 00F6 310 BR^ FORM_CONDSX : YES -- SUBROUTINE IS FINISHED
00F9 311 10$:
00F9 312 MOVAL COND1_T,MSG_A : SAVE ADDRESS OF CONDITION 1 TITLE FOR FAO
00000000'EF 0000007E'EF DE 00F9 313 MOVL COND1_TAB[R2],MSG_B : SAVE ADDR OF COND 1 CURR TEXT ELT FOR FAO
00000000'EF 00000092'EF42 DO 0104 314 MOVB #COND1_C,MSG_TXT : SAVE CONDITION 1 CONTEXT FOR FAO
00000000'EF 00 90 0110 314 MOV_VAL COND1_C,COND1_E[R2],MSG_DATA1 : GIVE COND 1 DATA VALUE TO FAO
0117 315

```

	FEE6'	30	0117	316	BSBW	WRITE_MSG2	; FORMAT AND WRITE CONDITION 1 MSG
14	00	91	011A	317	CMPB	#COND2_C,#NULL	; IS CONDITION 2 NULL ?
	03	12	011D	318	BNEQU	20\$; NO -- CONTINUE
	0096	31	011F	319	BRW	FORM_CONDSX	; YES -- SUBROUTINE IS FINISHED
			0122	320			
			0122	321	MOVAL	COND2_T,MSG_A	; SAVE ADDRESS OF CONDITION 2 TITLE FOR FAO
00000000'EF	00000105'EF	DE	0122	321	MOVL	COND2_TAB[R3],MSG_B	; SAVE ADDR OF COND 2 CURR TEXT ELT FOR FAO
00000000'EF	0000011E'EF43	D0	012D	322	MOVB	#COND2_C,MSG_CTXT	; SAVE CONDITION 2 CONTEXT FOR FAO
	00000000'EF 00	90	0139	323	MOV_VAL	COND2_C,COND2_E[R3],MSG_DATA1	; GIVE COND 2 DATA VALUE TO FAO
			0140	324			
	FEBD'	30	0140	325	BSBW	WRITE_MSG2	; FORMAT AND WRITE CONDITION 2 MSG
14	14	91	0143	326	CMPB	#COND3_C,#NULL	; IS CONDITION 3 NULL ?
	03	12	0146	327	BNEQU	30\$; NO -- CONTINUE
	006D	31	0148	328	BRW	FORM_CONDSX	; YES -- SUBROUTINE IS FINISHED
			0148	329			
			0148	330	MOVAL	COND3_T,MSG_A	; SAVE ADDRESS OF CONDITION 3 TITLE FOR FAO
00000000'EF	00000138'EF	DE	0148	330	MOVL	COND3_TAB[R4],MSG_B	; SAVE ADDR OF COND 3 CURR TEXT ELT FOR FAO
00000000'EF	00000138'EF44	D0	0156	331	MOVB	#COND3_C,MSG_CTXT	; SAVE CONDITION 3 CONTEXT FOR FAO
	00000000'EF 14	90	0162	332	MOV_VAL	COND3_C,COND3_E[R4],MSG_DATA1	; GIVE COND 3 DATA VALUE TO FAO
			0169	333			
	FE94'	30	0169	334	BSBW	WRITE_MSG2	; FORMAT AND WRITE CONDITION 3 MSG
14	14	91	016C	335	CMPB	#COND4_C,#NULL	; IS CONDITION 4 NULL ?
	47	13	016F	336	BEQU	FORM_CONDSX	; YES -- SUBROUTINE IS FINISHED
			0171	337	MOVAL	COND4_T,MSG_A	; SAVE ADDRESS OF CONDITION 4 TITLE FOR FAO
00000000'EF	00000139'EF	DE	0171	337	MOVL	COND4_TAB[R5],MSG_B	; SAVE ADDR OF COND 4 CURR TEXT ELT FOR FAO
00000000'EF	00000139'EF45	D0	017C	338	MOVB	#COND4_C,MSG_CTXT	; SAVE CONDITION 4 CONTEXT FOR FAO
	00000000'EF 14	90	0188	339	MOV_VAL	COND4_C,COND4_E[R5],MSG_DATA1	; GIVE COND 4 DATA VALUE TO FAO
			018F	340			
	FE6E'	30	018F	341	BSBW	WRITE_MSG2	; FORMAT AND WRITE CONDITION 4 MSG
14	14	91	0192	342	CMPB	#COND5_C,#NULL	; IS CONDITION 5 NULL ?
	21	13	0195	343	BEQU	FORM_CONDSX	; YES -- SUBROUTINE IS FINISHED
			0197	344	MOVAL	COND5_T,MSG_A	; SAVE ADDRESS OF CONDITION 5 TITLE FOR FAO
00000000'EF	0000013A'EF	DE	0197	344	MOVL	COND5_TAB[R6],MSG_B	; SAVE ADDR OF COND 5 CURR TEXT ELT FOR FAO
00000000'EF	0000013A'EF46	D0	01A2	345	MOVB	#COND5_C,MSG_CTXT	; SAVE CONDITION 5 CONTEXT FOR FAO
	00000000'EF 14	90	01AE	346	MOV_VAL	COND5_C,COND5_E[R6],MSG_DATA1	; GIVE COND 5 DATA VALUE TO FAO
			01B5	347			
	FE48'	30	01B5	348	BSBW	WRITE_MSG2	; FORMAT AND WRITE CONDITION 5 MSG
			01B8	349			
		05	01B8	350	FORM_CONDSX:		
			01B8	350	RSB		; RETURN TO CALLER

```
0189 352 .SBTTL VERIFY
0189 353 :++
0189 354 : FUNCTIONAL DESCRIPTION:
0189 355 :
0189 356 : VERIFY IS CALLED ONCE FOR EACH COMBINATION OF CONDITION
0189 357 : TABLE VALUES (AS DETERMINED BY THE INDEX REGISTERS R2,3,4,5,6 FOR
0189 358 : COND TABLES 1,2,3,4,5, RESPECTIVELY). VERIFY ESTABLISHES THE CONDITIONS
0189 359 : SPECIFIED BY THE COND TABLES AND ISSUES THE SUBJECT SYSTEM SERVICE
0189 360 : ($CMKRNL OR $CMEXEC). THEN, THE SUCCESSFUL OPERATION OF THE SERVICE IS
0189 361 : VERIFIED BY EXAMINING THE STATUS CODE RETURNED, THE VALUES FOR RETURN ARGS
0189 362 : AND THE FUNCTIONALITY PERFORMED. THE EXAMINATIONS TAKE THE FORM OF
0189 363 : COMPARISONS AGAINST EXPECTED VALUES. ANY FAILING COMPARISON CAUSES AN
0189 364 : ERR_EXIT MACRO TO BE EXECUTED (EITHER DIRECTLY, OR INDIRECTLY,
0189 365 : THROUGH THE SS_CHECK MACRO); ERR_EXIT SETS EFLAG TO NON-ZERO,
0189 366 : PRINTS ERROR MESSAGES AND CAUSES AN IMMEDIATE RSB TO CALLER.
0189 367 : WHEN ERR_EXIT IS EXECUTED, FURTHER CALLS TO VERIFY ARE SUPPRESSED,
0189 368 : AND, AFTER EXECUTING CLEANUP SUBROUTINES, THE IMAGE EXITS.
0189 369 :
0189 370 : CALLING SEQUENCE:
0189 371 :
0189 372 : BSBW VERIFY
0189 373 :
0189 374 : INPUT PARAMETERS:
0189 375 :
0189 376 : NONE
0189 377 :
0189 378 : IMPLICIT INPUTS:
0189 379 :
0189 380 : R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
0189 381 : FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
0189 382 : FOR X = 1,2,3,4,5 :
0189 383 : COND_X E - ADDRESS OF TABLE OF DATA VALUES FOR COND_X
0189 384 : TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
0189 385 : ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
0189 386 : FOR COND_X_E.
0189 387 :
0189 388 : OUTPUT PARAMETERS:
0189 389 :
0189 390 : NONE
0189 391 :
0189 392 : IMPLICIT OUTPUTS:
0189 393 :
0189 394 : VERIFY HAS NO OUTPUT. SINCE ITS PURPOSE IS TO TEST FOR ERRORS,
0189 395 : IT MERELY RETURNS TO CALLER NORMALLY AFTER THE TESTS, PROVIDING
0189 396 : ALL WERE SUCCESSFUL; IF AN ERROR IS DISCOVERED, RETURN IS VIA
0189 397 : AN ERR_EXIT OR SS_CHECK MACRO, BOTH OF WHICH DOCUMENT DETECTED
0189 398 : ERRORS.
0189 399 :
0189 400 : COMPLETION CODES:
0189 401 :
0189 402 : EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
0189 403 :
0189 404 : SIDE EFFECTS:
0189 405 :
0189 406 : SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
0189 407 : (VIA RSB) IF ERROR ENCOUNTERED.
0189 408 :
```

```

01B9 409 ;--
01B9 410
01B9 411
01B9 412
01B9 413 VERIFY::
00000000'EF 95 01B9 414 TSTB CFLAG ; SHOULD CONDITIONS BE PRINTED ?
03 13 01BF 415 BEQL 5$ ; NO -- CONTINUE
FF0B 30 01C1 416 BSBW FORM_CONDS ; YES -- FMT & PRINT ALL CONDS FOR THIS T.C.
01C4 417 5$:
00000008'EF 00000F9'EF42 D0 01C4 418 MOVL COND1_E[R2],ARGLST ; GET CURRENT ENTRY OUT OF TABLE
00000074'EF 00000000'EF 90 01D0 419 MOVB ONES,RTNMODE ; INIT MODE FOR EACH TST CASE
00000010'EF 00000000'EF B0 01DB 420 MOVW TESTNUM,ARGLSTR+4 ; MAKE RANDOM DATA UNIQUE FOR EACH T.C.
01E6 421
01E6 422 : ***** SYSTEM SERVICE CALL WHICH IS THE SUBJECT OF THIS TEST CASE *****
01E6 423 :
00 00000136'EF43 91 01E6 424 CMPB COND2_E[R3],#PSL$C_KERNEL ; IS THIS A CMKRNL REQUEST ?
15 13 01EE 425 BEQLU 10$ ; YES -- GO DO IT
01F0 426 $CMEXEC_S PRIVMODERTN,@ARGLST ; ISSUE CMEXEC REQUEST
13 11 0203 427 BRB 15$ ; ... AND GO CHECK RESULTS
0205 428 10$:
0205 429 $CMKRNL_S PRIVMODERTN,@ARGLST ; ISSUE KERNEL REQUEST
0218 430 15$:
57 00000000'EF D0 0218 431 MOVL ONES,R7 ; PREPARE FOR R0 COMPARISON
57 00000000'EF B0 021F 432 MOVW TESTNUM,R7 ;
57 50 D1 0226 433 CML R0,R7 ; DID PRIV MODE RTN RETURN EXPECTED STATUS ?
00000000'EF 73 D0 0229 434 BEQLU 20$ ; YES -- CONTINUE
00000000'EF 50 D0 022B 435 MOVL R7,EXPV ; NO -- LOAD EXPECTED AND ...
0232 436 MOVL R0,RECV ; ... RETURNED VALUES, THEN EXIT
0239 437 ERR_EXIT LONG,<INCORRECT USER-GENERATED STATUS CODE>, -
0239 438 < RETURNED FROM PRIV MODE RTN>
029E 439 20$:
00000136'EF43 00000074'EF 91 029E 440 CMPB RTNMODE,COND2_E[R3] ; DID PRIV MODE RTN REALLY GET EXP MODE ?
74 13 02AA 441 BEQLU 30$ ; YES -- CONTINUE
00000000'EF 00000136'EF43 90 02AC 442 MOVB COND2_E[R3],EXPV ; NO -- LOAD UP EXPECTED AND
00000000'EF 00000074'EF 90 02B8 443 MOVB RTNMODE,RECV ; ... RECEIVED VALUES, THEN EXIT
02C3 444 ERR_EXIT BYTE,<PRIVILEGED MODE SERVICE GAVE USER>, -
02C3 445 < ROUTINE INCORRECT MODE>
0320 446 30$:
00000038'EF 00000040'EF D0 0320 447 MOVL ARGLSTDESC2,ARGLSTDESC1 ; GET LENGTH OF DESCRIPTOR
0000003C'EF 00000008'EF D0 032B 448 MOVL ARGLST,ARGLSTDESC1+4 ; AND ITS ADDRESS
52 D5 0336 449 TSTL R2 ; 1ST CONDITION 1 ELEMENT ?
0B 12 0338 450 BNEQ 40$ ; NO -- CONTINUE
0000003C'EF 00000051'EF DE 033A 451 MOVAL ARGLST0,ARGLSTDESC1+4 ; YES -- USE 0 ARG LIST
0345 452 40$:
0000003C'FF 00000038'EF BB 0345 453 PUSHR #STRING_MASK ; SAVE REGS R2-R5 FOR CMPC
00000044'FF 29 0347 454 CMPC3 ARGLSTDESC1,@ARGLSTDESC1+4,@ARGLSTDESC2+4
0352 455
0357 456 : IS EXPECTED LIST = ACTUAL LIST ?
3C BA 0357 456 POPR #STRING_MASK ; RESTORE REGS R2-R5
70 13 0359 457 BEQLU VERIFYX ; YES -- ALL FINISHED
00000000'EF 00000038'EF 7D 035B 458 MOVQ ARGLSTDESC1,EXPV ; NO -- LOAD UP EXPECTED AND
00000000'EF 00000040'EF 7D 0366 459 MOVQ ARGLSTDESC2,RECV ; ... RECEIVED VALUES, THEN EXIT
0371 460 ERR_EXIT DESC,<ARG LIST IN PRIV MODE ROUTINE >, -
0371 461 <CONTAINS INCORRECT DATA>
03CB 462 VERIFYX:
05 03CB 463 RSB ; RETURN TO CALLER

```



```

03CC 465 .SBTTL VFY_CLEANUP
03CC 466 :++
03CC 467 : FUNCTIONAL DESCRIPTION:
03CC 468 :
03CC 469 : VFY_CLEANUP EXECUTES SYSTEM SERVICES TO UNDO THE
03CC 470 : EFFECT OF THOSE ISSUED IN THE VERIFY SUBROUTINE. VFY_CLEANUP MUST
03CC 471 : ASSUME THAT VERIFY MAY NOT HAVE EXECUTED IN ITS ENTIRETY (IF AN
03CC 472 : ERROR IS FOUND). ALSO, VFY_CLEANUP MAY ISSUE SS CHECK OR ERR_EXIT
03CC 473 : ONLY AFTER PERFORMING ALL OF ITS CLEANUP OPERATIONS; THIS IS REQUIRED
03CC 474 : IN THE EVENT THAT VFY_CLEANUP IS CALLED DURING ERROR PROCESSING,
03CC 475 : WHEN PERFORMING THE REQUIRED CLEANUP IS MORE IMPORTANT THAN
03CC 476 : POSSIBLY DISCOVERING A SECOND ERROR.
03CC 477 :
03CC 478 : CALLING SEQUENCE:
03CC 479 :
03CC 480 : BSBW VFY_CLEANUP
03CC 481 :
03CC 482 : INPUT PARAMETERS:
03CC 483 :
03CC 484 : NONE
03CC 485 :
03CC 486 : IMPLICIT INPUTS:
03CC 487 :
03CC 488 : R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
03CC 489 : FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
03CC 490 : FOR X = 1,2,3,4,5 :
03CC 491 : CONDX_E - ADDRESS OF TABLE OF DATA VALUES FOR CONDX
03CC 492 : TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
03CC 493 : ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
03CC 494 : FOR CONDX_E.
03CC 495 :
03CC 496 : OUTPUT PARAMETERS:
03CC 497 :
03CC 498 : NONE
03CC 499 :
03CC 500 : IMPLICIT OUTPUTS:
03CC 501 :
03CC 502 : NONE
03CC 503 :
03CC 504 : COMPLETION CODES:
03CC 505 :
03CC 506 : EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
03CC 507 :
03CC 508 : SIDE EFFECTS:
03CC 509 :
03CC 510 : SS CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
03CC 511 : (VIA RSB) IF ERROR ENCOUNTERED.
03CC 512 :
03CC 513 : --
03CC 514 :
03CC 515 :
03CC 516 :
05 03CC 517 VFY_CLEANUP::
03CC 518 RSB ; RETURN TO CALLER

```

```

03CD 520      .SBTTL KERNEL OR EXEC MODE ROUTINE
03CD 521      :
03CD 522      : THIS ROUTINE IS INVOKED IN THE VERIFY SUBROUTINE BY A $CMKRNL
03CD 523      : OR $CMEXEC SYSTEM SERVICE. ITS FUNCTION IS TO ASCERTAIN
03CD 524      : ITS EXECUTION MODE, STORING ITS VALUE IN RTNMODE;
03CD 525      : SET A RETURN STATUS VALUE IN R0; AND, STORE THE
03CD 526      : CONTENTS OF ITS ARGUMENT LIST IN ARGLSTDESC2.
03CD 527      : ALL THREE OF THESE DATA BASES (RTNMODE, R0,
03CD 528      : ARGLSTDESC2) WILL BE EXAMINED FOR EXPECTED VALUES
03CD 529      : IN THE VERIFY SUBROUTINE.
03CD 530      :
OFFC 03CD 531 PRIVMODERTN:
03CD 532      .WORD  *M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
03CF 533      :
03CF 534      : FIND CURRENT EXECUTION MODE BY MAKING A LOG NAME AND TRANSLATING IT.
03CF 535      :
03CF 536      $CRELOG_S TBLFLG=#LOG$C_PROCESS, - ; DEFINE A LOG NAME
03CF 537      LOGNAM=PMODE_LOG, -
03CF 538      EQLNAM=PMODE_EQL
03E6 539      $STRNLOG_S LOGNAM=PMODE_LOG, - ; TRAN IT TO RECEIVE MODE
03E6 540      RSLBUF=PMODE_EQL, -
03E6 541      ACMODE=RTNMODE
0405 542      $DELLOG_S TBLFLG=#LOG$C_PROCESS, - ; NO LONGER NEED IT
0405 543      LOGNAM=PMODE_LOG
5C 0416 544      TSTL AP ; IS THERE AN ARGUMENT LIST ?
0B 12 0418 545      BNEQ 10$ ; YES -- CONTINUE
58 04 9A 041A 546      MOVZBL #4,R8 ; NO -- ESTABLISH LENGTH OF NULL LIST
00000044'FF D4 041D 547      CLRL @ARGLSTDESC2+4 ; ... AND SET ZERO ARGUMENT COUNT
OF 11 0423 548      BRB 20$ ; JOIN COMMON CODE
0425 549 10$:
58 04 6C C5 0425 550      MULL3 (AP),#4,R8 ; GET NUMBER OF BYTES IN ARG LIST
58 04 C0 0429 551      ADDL2 #4,R8 ; ... ADD IN LENGTH OF LIST HEADER
00000044'FF 6C 58 28 042C 552      MOVCL R8,(AP),@ARGLSTDESC2+4 ; SAVE ENTIRE ARG LIST INCL HDR
0434 553 20$:
00000040'EF 58 D0 0434 554      MOVL R8,ARGLSTDESC2 ; PUT LENGTH IN DESCRIPTOR
50 00000000'EF D0 043B 555      MOVL ONES,R0 ; SAVE RETURN DATA IN R0
50 00000000'EF B0 0442 556      MOVW TESTNUM,R0 ; ... MAKE IT UNIQUE TO TEST CASE
04 0449 557      RET ; EXIT FROM PRIV MODE RTN
044A 558      .END
  
```

SSSS	= 0000037B	R	04	EXPV	*****	X	04
SSSCHARS	= 00000035			FAO_DESC	*****	X	04
SSSCHARS1	= 00000007			FAO_LEN	*****	X	04
SSSCHARS2	= 00000007			FORM_CONDS	000000CF	RG	04
SSSCHARS3	= 00000000			FORM_CONDSX	000001B8	R	04
SSSCHARS'	= 00000000			LOG\$C_PROCESS	= 00000002		
SSSCHARS5	= 00000000			LONG	= 00000004	G	
SSSCOND A	= 00000001			MOD_MSG_CODE	*****	X	04
SSSTRINGS	= 00000001			MOD_MSG_PRINT	*****	X	04
SSSTRINGS2	= 00000005			MSGT_INP_CTL	00000019	R	02
SST1	= 00000000			MSG3_ERR_CTL	00000039	RG	02
SST2	= 00000004			MSG_A	*****	X	04
ARGLST	00000008	R	03	MSG_B	*****	X	04
ARGLSTO	00000051	R	02	MSG_CTXT	*****	X	04
ARGLSTDESC1	00000038	R	03	NOTARG	= 00000000	G	
ARGLSTDESC2	00000040	R	03	NULL	= 00000014	G	
ARGLSTR	0000000C	R	03	ONES	*****	X	04
BYTE	= 00000001	G		OUTPUT_MSG	*****	X	04
CFLAG	*****	X	04	PCV	*****	X	04
CHMRTN	*****	X	04	PHDSQ_PRIVMSK	= 00000000		
CHM_CONT	*****	X	04	PMODE_EQL	00000075	R	03
COMP_SC	*****	X	04	PMODE_LOG	00000055	R	02
COND1	000000C5	RG	04	PRIVMASK	00000000	R	03
COND1_C	= 00000000			PRIVMODERTN	000003CD	R	04
COND1_CLEANUP	000000C6	RG	04	PRIV_ARGS	= 00000002		
COND1_E	000000F9	R	03	PROCESS_ERR	*****	X	04
COND1_H	00000091	RG	03	PSL\$C_EXEC	= 00000001		
COND1_T	0000007E	R	03	PSL\$C_KERNEL	= 00000000		
COND1_TAB	00000092	R	03	QUAD	= 00000008	G	
COND2	000000C7	RG	04	RECV	*****	X	04
COND2_C	= 00000000			REST_REGS	*****	X	04
COND2_CLEANUP	000000C8	RG	04	RLEN	= 0000000A		
COND2_E	00000136	R	03	RTNMODE	00000074	R	03
COND2_H	0000011D	RG	03	SAVE_REGS	*****	X	04
COND2_T	00000105	R	03	SS\$NORMAL	*****	X	04
COND2_TAB	0000011E	R	03	STRING_MASK	= 0000003C		
COND3	000000C9	RG	04	SUCCESS	*****	X	04
COND3_C	= 00000014			SYSSCMEXEC	*****	GX	04
COND3_CLEANUP	000000CA	RG	04	SYSSCMKRNL	*****	GX	04
COND3_H	00000138	RG	03	SYSSCRELOG	*****	GX	04
COND3_T	00000138	R	03	SYSDDELLOG	*****	GX	04
COND3_TAB	00000138	R	03	SYSSFAO	*****	X	04
COND4	000000CB	RG	04	SYSSSETPRN	*****	GX	04
COND4_C	= 00000014			SYSSSETPRV	*****	GX	04
COND4_CLEANUP	000000CC	RG	04	SYSSTRNLOG	*****	GX	04
COND4_H	00000139	RG	03	TESTNUM	*****	X	02
COND4_T	00000139	R	03	TEST_MOD_NAME	00000000	RG	02
COND4_TAB	00000139	R	03	TEST_MOD_NAME_D	00000009	R	02
COND5	000000CD	RG	04	TEST_MOD_SUCC	*****	X	04
COND5_C	= 00000014			TMD_ADDR	*****	X	04
COND5_CLEANUP	000000CE	RG	04	TM_CLEANUP	000000C1	RG	04
COND5_H	0000013A	RG	03	TM_SETUP	00000000	RG	04
COND5_T	0000013A	R	03	VERIFY	000001B9	RG	04
COND5_TAB	0000013A	R	03	VERIFYX	000003CB	R	04
CTL\$GC_PHD	*****	X	04	VFY_CLEANUP	000003CC	RG	04
DESC	= 00000010	G		WORD	= 00000002	G	
EFLAG	*****	X	04	WRITE_MSG2	*****	X	04

! 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	0000005D (93.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC LONG
RWDATA	0000013B (315.)	03 (3.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
SATSSS90	0000044A (1098.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.08	00:00:00.42
Command processing	107	00:00:00.74	00:00:04.97
Pass 1	250	00:00:06.85	00:00:13.17
Symbol table sort	0	00:00:00.51	00:00:00.54
Pass 2	122	00:00:01.82	00:00:02.70
Symbol table output	13	00:00:00.08	00:00:00.08
Psect synopsis output	3	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	527	00:00:10.12	00:00:21.92

The working set limit was 1200 pages.
34758 bytes (68 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 360 non-local and 23 local symbols.
558 source lines were read in Pass 1, producing 23 object records in Pass 2.
37 pages of virtual memory were used to define 28 macros.

! Macro library statistics !

Macro library name	Macros defined
_\$255\$DUA28:[SHRLIB]UETP.MLB;1	8
-\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	2
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	15
TOTALS (all libraries)	25

690 GETS were required to define 25 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:SATSSS90/OBJ=OBJ\$:SATSSS90 MSRC\$:SATSSS90/UPDATE=(ENH\$:SATSSS90)+EXECML\$/LIB+SHRLIB\$:UETP/LIB

0425 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

The image displays a grid of 140 small terminal windows arranged in 10 rows and 14 columns. Each window contains text-based data, likely representing a list of system components or configurations. The text is mostly illegible due to the low resolution and dim lighting. Several windows are highlighted with larger, clearer text labels:

- Top-left corner: SAT55574 LIS
- Row 2, Column 12: SATSUT01 LIS
- Row 2, Column 14: SATSUT05 LIS
- Row 3, Column 9: SAT55582 LIS
- Row 3, Column 11: SAT55590 LIS
- Row 4, Column 3: SAT55578 LIS
- Row 5, Column 6: SAT55580 LIS
- Row 8, Column 9: SAT55581 LIS
- Row 8, Column 11: SAT55591 LIS
- Row 10, Column 3: SAT55579 LIS
- Row 10, Column 10: SAT55583 LIS