





(1)	54	DECLARATIONS
(1)	82	CONDITION TABLES
(1)	107	TM_SETUP, TM_CLEANUP
(1)	170	CONDITION SUBROUTINES - SETUP AND CLEANUP
(1)	240	FORM_CONDS
(1)	333	VERIFY
(1)	455	VFY_CLEANUP

```

0000 1 .TITLE SATSS53 SATS SYSTEM SERVICE TESTS $SETEF (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 SATSS53 TO TEST SUCCESSFUL
0000 36 : OPERATION OF THE $SETEF SYSTEM SERVICE. THE 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 THE 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 IMAGE; NEEDS CMKRNL PRIVILEGE,
0000 44 : DYNAMICALLY ACQUIRES OTHER PRIVILEGES, AS NEEDED.
0000 45 :
0000 46 : AUTHOR: THOMAS L. CAFARELLA, CREATION DATE: SEP, 1977
0000 47 :
0000 48 : MODIFIED :
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 :
0000 61 : MACROS:
0000 62 :
0000 63 :
0000 64 : EQUATED SYMBOLS:
0000 65 :
0000 66 :
0000 67 : OWN STORAGE:
0000 68 :
```



SATSS53  
V04-000

SATS SYSTEM SERVICE TESTS \$SETEF<sup>F 2</sup> (SUCC 16-SEP-1984 00:57:38 VAX/VMS Macro V04-00  
DECLARATIONS 5-SEP-1984 04:32:17 [UETPSY.SRC]SATSS53.MAR;1

00000000	78	.PSECT	RWDATA, RD, WRT, NOEXE, LONG	
00000008 C000	79	PRIVMASK:	.BLKQ 1	; ADDR OF PRIVILEGE MASK (IN PHD)
0000000C 00C8	80	CLUSTER:	.BLKL 1	; STATE ARGUMENT ON REDEF SERVICE

SAT  
Pse

PSE  
---

\$AB  
ROD  
RWD  
SAT

Pha  
---

Ini  
Com  
Pas  
Sym  
Pas  
Sym  
Pse  
Cro  
Ass

The  
285  
The  
509  
35

Mac  
---

\$2  
- \$2  
- \$2  
TOT

620

The

MAC

```
000C 82 .SBTTL CONDITION TABLES
000C 83 :
000C 84 :
000C 85 :
000C 86 :
000C 87 :
000C 88 :
000C 89 :
000C 90 :
000C 91 :
00 0086 92 .BYTE 0 ; CLUSTER NUMBER 0
01 0087 93 .BYTE 1 ; CLUSTER NUMBER 1
02 0088 94 .BYTE 2 ; CLUSTER NUMBER 2
03 0089 95 .BYTE 3 ; CLUSTER NUMBER 3
008A 96 :
008A 97 COND 2,NULL
008B 98 COND 3,NULL
008B 99 COND 4,NULL
008C 100 COND 5,NULL
008C 101
008D 102
008D 103
008E 104
00000000 105 .PSECT SATSS53,RD,WRT,EXE
```



```

0000 107 .SBTTL TM_SETUP, TM_CLEANUP
0000 108 :++
0000 109 : FUNCTIONAL DESCRIPTION:
0000 110 :
0000 111 : TM_SETUP AND TM_CLEANUP ARE CALLED TO PERFORM
0000 112 : REQUIRED HOUSEKEEPING AT THE BEGINNING AND END, RESPECTIVELY, OF
0000 113 : TEST MODULE EXECUTION.
0000 114 :
0000 115 : CALLING SEQUENCE:
0000 116 :
0000 117 : BSBW TM_SETUP BSBW TM_CLEANUP
0000 118 :
0000 119 : INPUT PARAMETERS:
0000 120 :
0000 121 : NONE
0000 122 :
0000 123 : IMPLICIT INPUTS:
0000 124 :
0000 125 : NONE
0000 126 :
0000 127 : OUTPUT PARAMETERS:
0000 128 :
0000 129 : NONE
0000 130 :
0000 131 : IMPLICIT OUTPUTS:
0000 132 :
0000 133 : TM_SETUP: COND TABLE INDEX REGISTERS (R2,3,4,5,6) CLEARED;
0000 134 : ALL PRIVILEGES ACQUIRED.
0000 135 :
0000 136 : COMPLETION CODES:
0000 137 :
0000 138 : EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
0000 139 :
0000 140 : SIDE EFFECTS:
0000 141 :
0000 142 : SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
0000 143 : (VIA RSB) IF ERROR ENCOUNTERED.
0000 144 :
0000 145 :--
0000 146 :
0000 147 :
0000 148 :
0000 149 TM_SETUP::
52 D4 0000 150 CLRL R2 ; INITIALIZE
53 D4 0002 151 CLRL R3 ; .. CONDITION
54 D4 0004 152 CLRL R4 ; .... TABLE
55 D4 0006 153 CLRL R5 ; ..... INDEX
56 D4 0008 154 CLRL R6 ; ..... REGISTERS
FFF3' 30 000A 155 BSBW MOD MSG PRINT ; PRINT TEST MODULE BEGIN MSG
00000000'EF 00000000'EF DE 000D 156 MOVAL TEST_MOD_SUCC,TMD_ADDR ; ASSUME END MSG WILL SHOW SUCCESS
03 00 00000000'8F FO 0018 157 INSV #SUCCESS,#0,#3,MOD_MSG_CODE ; ADJUST STATUS CODE FOR SUCCESS
00000000'EF 00000000'EF 0020
0025 158 MODE TO,5$,KRNL ; KERNEL MODE TO ACCESS PHD
59 00000000'9F DO 0048 159 MOVL @#CTL$GL PHD,R9 ; GET PROCESS HEADER ADDRESS
00000000'EF 69 DE 004F 160 MOVAL PHD$Q PRIVMSK(R9),PRIVMSK ; GET PRIV MASK ADDRESS
0056 161 MODE FROM,5$ ; BACK TO USER MODE
0057 162 PRIV ADD,ALL ; GET ALL PRIVILEGES

```

	0077	163	\$SETPRN S TEST MOD_NAME_D	:	SET PROCESS NAME
	0084	164	SS CHECK NORMAL	:	CHECK STATUS CODE RETURNED FROM SETPRN
05	0082	165	RSB	:	RETURN TO MAIN ROUTINE
	0083	166	TM_CLEANUP::		
FF4A'	30	0083	BSBW MOD_MSG_PRINT	:	PRINT TEST MODULE END MSG
	05	0086	RSB	:	RETURN TO MAIN ROUTINE

```

00B7 170 .SBTTL CONDITION SUBROUTINES - SETUP AND CLEANUP
00B7 171 :++
00B7 172 : FUNCTIONAL DESCRIPTION:
00B7 173 :
00B7 174 : CONDX AND CONDX CLEANUP ARE SUBROUTINES WHICH ARE EXECUTED
00B7 175 : BEFORE AND AFTER THE VERIFY SUBROUTINE, RESPECTIVELY, WHENEVER A NEW
00B7 176 : CONDITION X VALUE IS SELECTED (SEE FUNCTIONAL DESCRIPTION OF SUCCOMMON
00B7 177 : ROUTINE IN SUCCOMMON.MAR). ANY SETUP FUNCTION PARTICULAR TO THE
00B7 178 : CONDITION X TABLE IS INCLUDED IN THE CONDX SUBROUTINE AND CLEANED
00B7 179 : UP, IF NECESSARY, IN THE CONDX CLEANUP SUBROUTINE. THIS INCLUDES,
00B7 180 : ESPECIALLY, CODE TO DETECT CONFLICTS AMONG CURRENT ENTRIES IN TWO
00B7 181 : OR MORE CONDITION TABLES. IF A CONFLICT IS DETECTED, A NON-ZERO
00B7 182 : VALUE IS STORED INTO CONFLICT, WHICH CAUSES THE CALLING ROUTINE
00B7 183 : (SUCCOMMON) TO SKIP THE CURRENT ENTRY IN THE CONDITION X TABLE.
00B7 184 :
00B7 185 : CALLING SEQUENCE:
00B7 186 :
00B7 187 : BSBW CONDX BSBW CONDX_CLEANUP
00B7 188 : WHERE X = 1,2,3,4,5
00B7 189 :
00B7 190 : INPUT PARAMETERS:
00B7 191 :
00B7 192 : CONFLICT = 0
00B7 193 :
00B7 194 : IMPLICIT INPUTS:
00B7 195 :
00B7 196 : R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
00B7 197 : FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
00B7 198 :
00B7 199 : OUTPUT PARAMETERS:
00B7 200 :
00B7 201 : CONFLICT SET TO NON-ZERO IF COND TABLE CONFLICT DETECTED.
00B7 202 :
00B7 203 : IMPLICIT OUTPUTS:
00B7 204 :
00B7 205 : R2,3,4,5,6 PRESERVED
00B7 206 :
00B7 207 : COMPLETION CODES:
00B7 208 :
00B7 209 : NONE
00B7 210 :
00B7 211 : SIDE EFFECTS:
00B7 212 :
00B7 213 : NONE
00B7 214 :
00B7 215 : --
00B7 216 :
00B7 217 :
00B7 218 :
05 00B7 219 COND1:: ; RETURN TO MAIN ROUTINE
00B7 220 RSB
00B8 221 COND1_CLEANUP:: ; RETURN TO MAIN ROUTINE
05 00B8 222 RSB
00B9 223 COND2:: ; RETURN TO MAIN ROUTINE
05 00B9 224 RSB
00BA 225 COND2_CLEANUP:: ; RETURN TO MAIN ROUTINE
05 00BA 226 RSB

```

```
05 00BB 227 COND3::  
05 00BB 228 RSB ; RETURN TO MAIN ROUTINE  
05 00BC 229 COND3_CLEANUP::  
05 00BC 230 RSB ; RETURN TO MAIN ROUTINE  
05 00BD 231 COND4::  
05 00BD 232 RSB ; RETURN TO MAIN ROUTINE  
05 00BE 233 COND4_CLEANUP::  
05 00BE 234 RSB ; RETURN TO MAIN ROUTINE  
05 00BF 235 COND5::  
05 00BF 236 RSB ; RETURN TO MAIN ROUTINE  
05 00C0 237 COND5_CLEANUP::  
05 00C0 238 RSB ; RETURN TO MAIN ROUTINE
```

```

OOC1 240 .SBTTL FORM_CONDS
OOC1 241 :++
OOC1 242 : FUNCTIONAL DESCRIPTION:
OOC1 243 :
OOC1 244 : FORM_CONDS FORMATS AND PRINTS INFORMATION ABOUT
OOC1 245 : THE CURRENT ELEMENT IN EACH OF THE CONDITION TABLES.
OOC1 246 :
OOC1 247 : CALLING SEQUENCE:
OOC1 248 :
OOC1 249 : BSBW FORM_CONDS
OOC1 250 :
OOC1 251 : INPUT PARAMETERS:
OOC1 252 :
OOC1 253 : NONE
OOC1 254 :
OOC1 255 : IMPLICIT INPUTS:
OOC1 256 :
OOC1 257 : R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
OOC1 258 : FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
OOC1 259 : FOR X = 1,2,3,4,5 :
OOC1 260 : COND_X_T - TITLE TEXT FOR CONDX TABLE
OOC1 261 : COND_X_TAB - ELEMENT TEXT FOR CONDX TABLE
OOC1 262 : COND_X_C - CONTEXT OF THE CONDX TABLE
OOC1 263 : COND_X_E - DATA ELEMENTS OF THE CONDX TABLE
OOC1 264 :
OOC1 265 : OUTPUT PARAMETERS:
OOC1 266 :
OOC1 267 : NONE
OOC1 268 :
OOC1 269 : IMPLICIT OUTPUTS:
OOC1 270 :
OOC1 271 : NONE
OOC1 272 :
OOC1 273 : COMPLETION CODES:
OOC1 274 :
OOC1 275 : NONE
OOC1 276 :
OOC1 277 : SIDE EFFECTS:
OOC1 278 :
OOC1 279 : NONE
OOC1 280 :
OOC1 281 : --
OOC1 282 :
OOC1 283 :
OOC1 284 :
OOC1 285 FORM_CONDS::
OOC1 286 $FAO_S MSG1_INP_CTL,FAO_LEN,FAO_DESC,TESTNUM
OOC1 287 : FORMAT CONDITIONS HEADER MSG
OOC1 288 BSBW OUTPUT_MSG : ... AND PRINT IT
OOC1 289 CMPB #COND1_C,#NULL : IS CONDITION 1 NULL ?
OOC1 290 BNEQU 10$ : NO -- CONTINUE
OOC1 291 BRW FORM_CONDSX : YES -- SUBROUTINE IS FINISHED
OOC1 292 10$:
OOC1 293 MOVAL COND1_T,MSG_A : SAVE ADDRESS OF CONDITION 1 TITLE FOR FAO
OOC1 294 MOVL COND1_TAB[R2],MSG_B : SAVE ADDR OF COND 1 CURR TEXT ELI FOR FAO
OOC1 295 MOVB #COND1_C,MSG_CTXT : SAVE CONDITION 1 CONTEXT FOR FAO
OOC1 296 MOV_VAL COND1_C,COND1_E[R2],MSG_DATA1 : GIVE COND 1 DATA VALUE TO FAO

```

```

FF1D' 30
14 00 91
03 12
00BF 31
DE 00EB
DO 00F6
90 0102
0109

```

```

      FEF4' 30 0109 297      BSBW WRITE_MSG2      ; FORMAT AND WRITE CONDITION 1 MSG
      14 14 91 010C 298      CMPB #COND2_C,#NULL      ; IS CONDITION 2 NULL ?
      03 12 010F 299      BNEQU 20$      ; NO -- CONTINUE
      0096 31 0111 300      BRW FORM_CONDSX      ; YES -- SUBROUTINE IS FINISHED
      0114 301 20$:
00000000'EF 0000008A'EF DE 0114 302      MOVAL COND2_T,MSG_A      ; SAVE ADDRESS OF CONDITION 2 TITLE FOR FAO
00000000'EF 0000008A'EF43 DO 011F 303      MOVL COND2_TAB[R3],MSG_B      ; SAVE ADDR OF COND 2 CURR TEXT ELT FOR FAO
      00000000'EF 14 90 012B 304      MOVB #COND2_C,MSG_CTXT      ; SAVE CONDITION 2 CONTEXT FOR FAO
      0132 305      MOV_VAL COND2_C,COND2_E[R3],MSG_DATA1 ; GIVE COND 2 DATA VALUE TO FAO
      FECB' 30 0132 306      BSBW WRITE_MSG2      ; FORMAT AND WRITE CONDITION 2 MSG
      14 14 91 0135 307      CMPB #COND3_C,#NULL      ; IS CONDITION 3 NULL ?
      03 12 0138 308      BNEQU 30$      ; NO -- CONTINUE
      006D 31 013A 309      BRW FORM_CONDSX      ; YES -- SUBROUTINE IS FINISHED
      013D 310 30$:
00000000'EF 0000008B'EF DF 013D 311      MOVAL COND3_T,MSG_A      ; SAVE ADDRESS OF CONDITION 3 TITLE FOR FAO
00000000'EF 0000008B'EF44 DO 0148 312      MOVL COND3_TAB[R4],MSG_B      ; SAVE ADDR OF COND 3 CURR TEXT ELT FOR FAO
      00000000'EF 14 90 0154 313      MOVB #COND3_C,MSG_CTXT      ; SAVE CONDITION 3 CONTEXT FOR FAO
      015B 314      MOV_VAL COND3_C,COND3_E[R4],MSG_DATA1 ; GIVE COND 3 DATA VALUE TO FAO
      FEA2' 30 015B 315      BSBW WRITE_MSG2      ; FORMAT AND WRITE CONDITION 3 MSG
      14 14 91 015E 316      CMPB #COND4_C,#NULL      ; IS CONDITION 4 NULL ?
      47 13 0161 317      BEQLU FORM_CONDSX      ; YES -- SUBROUTINE IS FINISHED
      00000000'EF 0000008C'EF DE 0163 318      MOVAL COND4_T,MSG_A      ; SAVE ADDRESS OF CONDITION 4 TITLE FOR FAO
00000000'EF 0000008C'EF45 DO 016E 319      MOVL COND4_TAB[R5],MSG_B      ; SAVE ADDR OF COND 4 CURR TEXT ELT FOR FAO
      00000000'EF 14 90 017A 320      MOVB #COND4_C,MSG_CTXT      ; SAVE CONDITION 4 CONTEXT FOR FAO
      0181 321      MOV_VAL COND4_C,COND4_E[R5],MSG_DATA1 ; GIVE COND 4 DATA VALUE TO FAO
      FE7C' 30 0181 322      BSBW WRITE_MSG2      ; FORMAT AND WRITE CONDITION 4 MSG
      14 14 91 0184 323      CMPB #COND5_C,#NULL      ; IS CONDITION 5 NULL ?
      21 13 0187 324      BEQLU FORM_CONDSX      ; YES -- SUBROUTINE IS FINISHED
      00000000'EF 0000008D'EF DE 0189 325      MOVAL COND5_T,MSG_A      ; SAVE ADDRESS OF CONDITION 5 TITLE FOR FAO
00000000'EF 0000008D'EF46 DO 0194 326      MOVL COND5_TAB[R6],MSG_B      ; SAVE ADDR OF COND 5 CURR TEXT ELT FOR FAO
      00000000'EF 14 90 01A0 327      MOVB #COND5_C,MSG_CTXT      ; SAVE CONDITION 5 CONTEXT FOR FAO
      FE56' 30 01A7 328      MOV_VAL COND5_C,COND5_E[R6],MSG_DATA1 ; GIVE COND 5 DATA VALUE TO FAO
      01A7 329      BSBW WRITE_MSG2      ; FORMAT AND WRITE CONDITION 5 MSG
      01AA 330 FORM_CONDSX:
      05 01AA 331      RSB      ; RETURN TO CALLER
```

```
01AB 333 .SBTTL VERIFY
01AB 334 :++
01AB 335 : FUNCTIONAL DESCRIPTION:
01AB 336 :
01AB 337 : VERIFY IS CALLED ONCE FOR EACH COMBINATION OF CONDITION
01AB 338 : TABLE VALUES (AS DETERMINED BY THE INDEX REGISTERS R2,3,4,5,6 FOR
01AB 339 : COND TABLES 1,2,3,4,5, RESPECTIVELY). VERIFY ESTABLISHES THE CONDITIONS
01AB 340 : SPECIFIED BY THE COND TABLES AND ISSUES THE SUBJECT SYSTEM SERVICE
01AB 341 : ($SETEF). THEN, THE SUCCESSFUL OPERATION OF THE SERVICE IS VERIFIED
01AB 342 : BY EXAMINING THE STATUS CODE RETURNED, THE VALUES FOR RETURN ARGUMENTS
01AB 343 : AND THE FUNCTIONALITY PERFORMED. THE EXAMINATIONS TAKE THE FORM OF
01AB 344 : COMPARISONS AGAINST EXPECTED VALUES. ANY FAILING COMPARISON CAUSES AN
01AB 345 : ERR_EXIT MACRO TO BE EXECUTED (EITHER DIRECTLY, OR INDIRECTLY,
01AB 346 : THROUGH THE SS_CHECK MACRO); ERR_EXIT SETS EFLAG TO NON-ZERO,
01AB 347 : PRINTS ERROR MESSAGES AND CAUSES AN IMMEDIATE RSB TO CALLER.
01AB 348 : WHEN ERR_EXIT IS EXECUTED, FURTHER CALLS TO VERIFY ARE SUPPRESSED,
01AB 349 : AND, AFTER EXECUTING CLEANUP SUBROUTINES, THE IMAGE EXITS.
01AB 350 :
01AB 351 : CALLING SEQUENCE:
01AB 352 :
01AB 353 : BSBW VERIFY
01AB 354 :
01AB 355 : INPUT PARAMETERS:
01AB 356 :
01AB 357 : NONE
01AB 358 :
01AB 359 : IMPLICIT INPUTS:
01AB 360 :
01AB 361 : R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
01AB 362 : FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
01AB 363 : FOR X = 1,2,3,4,5 :
01AB 364 : CONDX_E - ADDRESS OF TABLE OF DATA VALUES FOR CONDX
01AB 365 : TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
01AB 366 : ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
01AB 367 : FOR CONDX_E.
01AB 368 :
01AB 369 : OUTPUT PARAMETERS:
01AB 370 :
01AB 371 : NONE
01AB 372 :
01AB 373 : IMPLICIT OUTPUTS:
01AB 374 :
01AB 375 : VERIFY HAS NO OUTPUT. SINCE ITS PURPOSE IS TO TEST FOR ERRORS,
01AB 376 : IT MERELY RETURNS TO CALLER NORMALLY AFTER THE TESTS, PROVIDING
01AB 377 : ALL WERE SUCCESSFUL; IF AN ERROR IS DISCOVERED, RETURN IS VIA
01AB 378 : AN ERR_EXIT OR SS_CHECK MACRO, BOTH OF WHICH DOCUMENT DETECTED
01AB 379 : ERRORS.
01AB 380 :
01AB 381 : COMPLETION CODES:
01AB 382 :
01AB 383 : EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
01AB 384 :
01AB 385 : SIDE EFFECTS:
01AB 386 :
01AB 387 : SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
01AB 388 : (VIA RSB) IF ERROR ENCOUNTERED.
01AB 389 :
```

			01AB	390	;	--			
			01AB	391					
			01AB	392					
			01AB	393					
			01AB	394	VERIFY::				
	00000000	'EF	95	01AB	395	TSTB	CFLAG		; SHOULD CONDITIONS BE PRINTED ?
		03	13	01B1	396	BEQL	5\$		; NO -- CONTINUE
		FF0B	30	01B3	397	BSBW	FORM_CONDS		; YES -- FMT & PRINT ALL CONDS FOR THIS T.C.
				01B6	398	5\$:			
58	00000086	'EF42	9A	01B6	399	MOVZBL	COND1_E[R2],R8		; GET CLUSTER NO. INTO REGISTER
	58	58	05	78	01BE	400	ASHL	#5,R8,R8	; MULT BY 32 TO GET 1ST EVENT FLAG NO.
			59	D4	01C2	401	CLRL	R9	; ESTAB OFFSET INTO CLUSTER FOR 1ST FLAG
			5A	D4	01C4	402	CLRL	R10	; INIT REG TO HOLD EVENT FLAG NO. (EFN)
02	00000086	'EF42	91	01C6	403	CMPB	COND1_E[R2],#2		; COMMON CLUSTER ?
		4B	19	01CE	404	BLSS	15\$		; NO -- BYPASS THE ASSOCIATE SERVICE
				01D0	405	\$ASCEFC	S EFN=R8, NAME=TEST_MOD_NAME_D		; YES -- ASSOCIATE
				01E3	406	SS_CHECK	NORMAL		; ... CLUSTER & CHECK STATUS CODE
02	00000086	'EF42	91	0211	407	CMPB	COND1_E[R2],#2		; CLUSTER 2 ?
		4D	13	0219	408	BEQL	30\$		; YES -- CLUSTER 2 INITIALLY CLEAR
				021B	409	15\$:			
			5A	58	D0	021B	410	MOVL	R8,R10
			5B	5A	1F	C1	021E	411	ADDL3 #31,R10,R11
							0222	412	20\$:
							0222	413	\$CLREF_S EFN=R10
	00000000	'8F	50	D	022B	414	CMPB	R0,#SS\$_WASSET	; CLEAR CURRENT EVENT FLAG
			2E	13	0232	415	BEQLU	25\$	; WASSET STATUS CODE ?
					0234	416	SS_CHECK	WASCLR	; YES -- GO LOOP FOR ANOTHER CLREF
					0262	417	25\$:		; NO -- BETTER BE WASCLR, THEN
	FFBA	5A	01	5B	3D	0262	418	ACBW	R11,#1,R10,20\$
						0268	419	30\$:	
			5A	59	58	81	0268	420	ADDB3 R8,R9,R10
							026C	421	; COMPUTE EVENT FLAG NUMBER
							026C	422	; ***** SYSTEM SERVICE CALL WHICH IS THE SUBJECT OF THIS TEST CASE *****
							026C	423	; *****
							026C	424	
	00000000	'8F	50	D1	0275	425	\$SETEF_S	EFN=R10	; SET EVENT FLAG
			60	13	027C	426	CMPB	R0,#SS\$_WASCLR	; CODE RECEIVED = CODE EXPECTED ?
	00000000	'EF	00000000	'8F	D0	027E	427	BEQLU	40\$
	00000000	'EF	00000000	'EF	D0	0289	428	MOVL	#SS\$ WASCLR,EXPV
			50	D0	0289	428	MOVL	R0,RECV	; LOAD UP EXPECTED AND ...
						0290	429	ERR_EXIT	LONG,<INCORRECT STATUS CODE RETURNED FROM SETEF>
						02DE	430	40\$:	
						02DE	431	\$READEF_S	EFN=R8, STATE=CLUSTER
						02ED	432	SS_CHECK	WASSET
						031B	433	CLRL	R11
			5B	D4	031B	433	CLRL	R7	; CLEAR REGISTERS TO ALLOW ...
			57	D4	031D	434	CLRL	R7	; ... BYTE OPERATIONS ON THEM
			58	59	01	81	031F	435	ADDB3 #1,R9,R11
							0323	436	CLRL EXPV
5B	00	00000000	'EF	D4	0323	436	CLRL	EXPV	; ESTAB EXPECTED VALUE FOR ...
		00000000	'EF	F0	0329	437	INSV	ONES,#0,R11,EXPV	; ... POSSIBLE ERR_EXIT
		00000000	'EF		0331	437			
	00000000	'EF	00000008	'EF	D0	0336	438	MOVL	CLUSTER,RECV
	00000008	'EF	5B	00	EC	0341	439	CMPV	#0,R11,CLUSTER,ONES
			00000000	'EF		0349	439		; ESTAB RECEIVED VALUE AS WELL
			4B	13	034E	440	BEQL	50\$	; ARE ALL EXPECTED EVENT FLAGS SET ?
						0350	441	ERR_EXIT	LONG,<EVENT FLAG(S) IN CLUSTER SHOULD BE SET>
						039B	442		; YES -- GO LOOK AT CLEAR FLAGS
						039B	443		; NO -- GENERATE ERROR & EXIT
			59	1F	91	039B	444	50\$:	
									; IS CURRENT EFN HIGHEST IN CLUSTER ?



00	00000008'EF	57	1F	57	5E	13	039E	445	BEQL	60\$	:	YES -- THEN CLUSTER IS ALL ONES
					59	83	03A0	446	SUBB3	R9,#31,R7	:	NO -- COMPUTE NO. OF 0-BITS TO COMPARE
					5B	EC	03A4	447	CMPV	R11,R7,CLUSTER,#0	:	ARE ALL EV FLAGS NOT YET SET STILL CLEAR ?
					4F	13	03AD	448	BEQL	60\$	:	YES -- GO LOOK AT NEXT EVENT FLAG
							03AF	449	ERR_EXIT	LONG,<EVENT FLAG(S) IN	:	CLUSTER SHOULD NOT BE SET>
							03FE	450			:	NO -- GENERATE ERROR & EXIT
							03FE	451				
	FE64	59	01	1F	9D	03FE	452	60\$:	ACBB	#31,#1,R9,30\$	:	INCR TO NEXT EFN IN THIS CLUSTER & LOOP
					05	0404	453		RSB		:	RETURN TO CALLER

```

0405 455 .SBTTL VFY_CLEANUP
0405 456 :++
0405 457 : FUNCTIONAL DESCRIPTION:
0405 458 :
0405 459 : VFY_CLEANUP EXECUTES SYSTEM SERVICES TO UNDO THE
0405 460 : EFFECT OF THOSE ISSUED IN THE VERIFY SUBROUTINE. VFY_CLEANUP MUST
0405 461 : ASSUME THAT VERIFY MAY NOT HAVE EXECUTED IN ITS ENTIRETY (IF AN
0405 462 : ERROR IS FOUND). ALSO, VFY_CLEANUP MAY ISSUE SS_CHECK OR ERR_EXIT
0405 463 : ONLY AFTER PERFORMING ALL OF ITS CLEANUP OPERATIONS; THIS IS REQUIRED
0405 464 : IN THE EVENT THAT VFY_CLEANUP IS CALLED DURING ERROR PROCESSING,
0405 465 : WHEN PERFORMING THE REQUIRED CLEANUP IS MORE IMPORTANT THAN
0405 466 : POSSIBLY DISCOVERING A SECOND ERROR.
0405 467 :
0405 468 : CALLING SEQUENCE:
0405 469 :
0405 470 : BSBW VFY_CLEANUP
0405 471 :
0405 472 : INPUT PARAMETERS:
0405 473 :
0405 474 : NONE
0405 475 :
0405 476 : IMPLICIT INPUTS:
0405 477 :
0405 478 : R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
0405 479 : FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
0405 480 : FOR X = 1,2,3,4,5 :
0405 481 : CONDX_E - ADDRESS OF TABLE OF DATA VALUES FOR CONDX
0405 482 : TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
0405 483 : ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
0405 484 : FOR CONDX_E.
0405 485 :
0405 486 : OUTPUT PARAMETERS:
0405 487 :
0405 488 : NONE
0405 489 :
0405 490 : IMPLICIT OUTPUTS:
0405 491 :
0405 492 : NONE
0405 493 :
0405 494 : COMPLETION CODES:
0405 495 :
0405 496 : EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
0405 497 :
0405 498 : SIDE EFFECTS:
0405 499 :
0405 500 : SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
0405 501 : (VIA RSB) IF ERROR ENCOUNTERED.
0405 502 :
0405 503 :--
0405 504 :
0405 505 :
0405 506 :
05 0405 507 VFY_CLEANUP::
0405 508 RSB ; RETURN TO CALLER
0406 509 .END

```

SSSS	= 000003B9	R	04	LONG	= 00000004	G	
SSSCHARS	= 0000002A			MOD_MSG_CODE	*****	X	04
SSSCHARS1	= 00000019			MOD_MSG_PRINT	*****	X	04
SSSCHARS2	= 00000019			MSGT_INP_CTL	00000019	R	02
SSSCHARS3	= 00000012			MSG3_ERR_CTL	00000039	RG	02
SSSCHARS4	= 00000012			MSG_A	*****	X	04
SSSCHARS5	= 00000000			MSG_B	*****	X	04
SSSCOND_A	= 00000003			MSG_CTXT	*****	X	04
SSSTRINGS	= 00000001			NOTARG	= 00000000	G	
SSSTRINGS2	= 00000005			NULL	= 00000014	G	
SST1	= 00000000			ONES	*****	X	04
SST2	= 00000004			OUTPUT_MSG	*****	X	04
BYTE	= 00000001	G		PCV	*****	X	04
CFLAG	*****	X	04	PHD\$Q_PRIVMSK	= 00000000		
CHMRTN	*****	X	04	PRIVMSK	00000000	R	03
CHM_CONT	*****	X	04	PRIV_ARGS	= 00000002		
CLUSTER	00000008	R	03	PROCESS_ERR	*****	X	04
COMP_SC	*****	X	04	QUAD	= 00000008	G	
CONDT	000000B7	RG	04	RECV	*****	X	04
COND1_C	= 00000000			REST_REGS	*****	X	04
COND1_CLEANUP	000000B8	RG	04	SAVE_REGS	*****	X	04
COND1_E	000000B6	R	03	SS\$NORMAL	*****	X	04
COND1_H	0000001B	RG	03	SS\$WASCLR	*****	X	04
COND1_T	0000000C	R	03	SS\$WASSET	*****	X	04
COND1_TAB	0000001C	R	03	SUCCESS	*****	X	04
COND2	000000B9	RG	04	SYSSASCEFC	*****	GX	04
COND2_C	= 00000014			SYSSCLREF	*****	GX	04
COND2_CLEANUP	000000BA	RG	04	SYSSCMKRN	*****	GX	04
COND2_H	0000008A	RG	03	SYSSFAO	*****	X	04
COND2_T	0000008A	R	03	SYSSREADEF	*****	GX	04
COND2_TAB	0000008A	R	03	SYSSSETEF	*****	GX	04
COND3	000000BB	RG	04	SYSSSETPRN	*****	GX	04
COND3_C	= 00000014			SYSSSETPRV	*****	GX	04
COND3_CLEANUP	000000BC	RG	04	TESTNUM	*****	X	04
COND3_H	0000008B	RG	03	TEST_MOD_NAME	00000000	RG	02
COND3_T	0000008B	R	03	TEST_MOD_NAME_D	00000009	R	02
COND3_TAB	0000008B	R	03	TEST_MOD_SUCC_D	*****	X	04
COND4	000000BD	RG	04	TMD_ADDR	*****	X	04
COND4_C	= 00000014			TM_CLEANUP	000000B3	RG	04
COND4_CLEANUP	000000BE	RG	04	TM_SETUP	00000000	RG	04
COND4_H	0000008C	RG	03	VERIFY	000001AB	RG	04
COND4_T	0000008C	R	03	VFY_CLEANUP	00000405	RG	04
COND4_TAB	0000008C	R	03	WORD	= 00000002	G	
COND5	000000BF	RG	04	WRITE_MSG2	*****	X	04
COND5_C	= 00000014						
COND5_CLEANUP	000000C0	RG	04				
COND5_H	0000008D	RG	03				
COND5_T	0000008D	R	03				
COND5_TAB	0000008D	R	03				
CTL\$GC_PHD	*****	X	04				
DESC	= 00000010	G					
EFLAG	*****	X	04				
EXPV	*****	X	04				
FAO_DESC	*****	X	04				
FAO_LEN	*****	X	04				
FORM_CONDS	000000C1	RG	04				
FORM_CONDSX	000001AA	R	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	00000051 ( 81.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC LONG
RWDATA	0000008E ( 142.)	03 ( 3.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
SATSS53	00000406 ( 1030.)	04 ( 4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	30	00:00:00.07	00:00:00.36
Command processing	107	00:00:00.62	00:00:02.03
Pass 1	228	00:00:05.56	00:00:10.82
Symbol table sort	0	00:00:00.43	00:00:00.42
Pass 2	110	00:00:01.49	00:00:02.41
Symbol table output	13	00:00:00.07	00:00:00.07
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	492	00:00:08.27	00:00:16.16

The working set limit was 1500 pages.  
28587 bytes (56 pages) of virtual memory were used to buffer the intermediate code.  
There were 20 pages of symbol table space allocated to hold 295 non-local and 28 local symbols.  
509 source lines were read in Pass 1, producing 22 object records in Pass 2.  
35 pages of virtual memory were used to define 26 macros.

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

Macro library name	Macros defined
_\$255\$DUA28:[SHRLIB]UETP.MLB;1	8
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	1
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	14
TOTALS (all libraries)	23

620 GETS were required to define 23 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:SATSS53/OBJ=OBJ\$:SATSS53 MSRC\$:SATSS53/UPDATE=(ENH\$:SATSS53)+EXECML\$/LIB+SHRLIB\$:UETP/LIB

0424 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

The image displays a grid of 14 columns and 12 rows of small, illegible text screens, likely representing a data table or a series of test results. Several screens are highlighted with larger text labels:

- SATSS553 LIS (bottom left)
- SATSS554 LIS (row 4, column 3)
- SATSS555 LIS (row 10, column 3)
- SATSS556 LIS (row 3, column 5)
- SATSS560 LIS (row 4, column 7)
- SATSS561 LIS (row 12, column 7)
- SATSS570 LIS (row 8, column 10)
- SATSS571 LIS (row 3, column 11)
- SATSS572 LIS (row 8, column 12)
- SATSS573 LIS (row 5, column 14)