


```

SSSSSSSS  AAAAAA  TTTTTTTTTT  SSSSSSSS  SSSSSSSS  SSSSSSSS  5555555555  222222
SSSSSSSS  AAAAAA  TTTTTTTTTT  SSSSSSSS  SSSSSSSS  SSSSSSSS  5555555555  222222
SS        AA      AA      TT        SS        SS        SS        55        22        22
SS        AA      AA      TT        SS        SS        SS        55        22        22
SS        AA      AA      TT        SS        SS        SS        555555    22        22
SS        AA      AA      TT        SS        SSSSSS    SSSSSS    555555    22        22
SSSSSSS   AA      AA      TT        SSSSSS    SSSSSS    SSSSSS    55        22        22
SS        AA      AA      TT        SS        SS        SS        55        55        22
SS        AA      AA      TT        SS        SS        SS        55        55        22
SS        AA      AA      TT        SS        SS        SS        55        55        22
SS        AA      AA      TT        SSSSSS    SSSSSS    SSSSSS    55        55        22
SSSSSSSS  AA      AA      TT        SSSSSSSS  SSSSSSSS  SSSSSSSS  555555    2222222222
SSSSSSSS  AA      AA      TT        SSSSSSSS  SSSSSSSS  SSSSSSSS  555555    2222222222

```

```

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)	56	DECLARATIONS
(1)	105	CONDITION TABLES
(1)	130	TM SETUP, TM CLEANUP
(1)	201	CONDITION SUBROUTINES - SETUP AND CLEANUP
(1)	271	FORM CONDS
(1)	364	VERIFY
(1)	612	VFY_CLEANUP
(1)	669	BUILD_CLUST SUBROUTINE
(1)	723	READ_DACEFC SUBROUTINE

```
0000 1 .TITLE SATSSS52 SATS SYSTEM SERVICE TESTS $DLCEFC (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 SATSSS52 TO TEST SUCCESSFUL
0000 36 : OPERATION OF THE $DLCEFC SYSTEM SERVICE. THE SERVICE IS INVOKED
0000 37 : UNDER VARIOUS INPUT CONDITIONS WITH VARYING INPUT PARAMETERS. ONLY
0000 38 : SUCCESSFUL STATUS CODES ARE EXPECTED IN THIS TEST MODULE. CORRECT
0000 39 : OPERATION OF THE SERVICE FOR EACH OF ITS ISSUANCES IS VERIFIED BY
0000 40 : CHECKING FOR AN SSS NORMAL STATUS CODE, EXPECTED RETURN ARGUMENTS
0000 41 : AND EXPECTED FUNCTIONALITY PERFORMED.
0000 42 :
0000 43 : ENVIRONMENT: USER MODE IMAGE; NEEDS CMKRNL PRIVILEGE,
0000 44 : DYNAMICALLY ACQUIRES OTHER PRIVILEGES, AS NEEDED.
0000 45 :
0000 46 : AUTHOR: THOMAS L. CAFARELLA, CREATION DATE: JAN, 1978
0000 47 :
0000 48 : MODIFIED BY:
0000 49 :
0000 50 : VERSION 1.5 : 25-MAY-79
0000 51 :
0000 52 : 01 LDJ 10/11/79 Fixed bug caused by DIB$K_LENGTH change ACG052.RNO mem
0000 53 :
0000 54 :--
```

```
0000 56 .SBTTL DECLARATIONS
0000 57 :
0000 58 : INCLUDE FILES:
0000 59 :
0000 60 $PRVDEF ; PRIVILEGE BIT DEFINITIONS
0000 61 $PHDDEF ; PROCESS HEADER OFFSETS
0000 62 $PQLDEF ; PROCESS QUOTA CODES
0000 63 $DIBDEF ; DEVICE INFO BLOCK OFFSETS
0000 64 :
0000 65 : MACROS:
0000 66 :
0000 67 :
0000 68 : EQUATED SYMBOLS:
0000 69 :
0000 70 :
0000 71 : OWN STORAGE:
0000 72 :
```

```
00000000 74 .PSECT RODATA,RD,NOWRT,NOEXE,LONG
0000 75 TEST_MOD_NAME:: STRING C,<SATSS52> ; TEST MODULE NAME
0009 76 TEST_MOD_NAME_D: STRING I,<SATSS52> ; TEST MODULE NAME DESCRIPTOR
0019 77 MSG1_INP_CTL: STRING I,<SSDEF!4ZW: CONDITIONS:>
0039 78 ; FAO CTL STRING FOR MSG1 IN SUCCOMMON.MAR
0039 79 MSG3_ERR_CTL:: STRING I,<*SSDEF!4ZW: !AS>
0051 80 ; FAO CTL STRING FOR MSG3 IN SUCCOMMON.MAR
0051 81 CREPRN: STRING I,<SATSS52_CRE> ; CREATED PROCESS NAME
0065 82 CLUS_NAME: STRING I,<SATSS52-CLUS> ; SUBJECT CLUSTER NAME
007A 83 IMAGNAM: STRING I,<SYSTST$RES:SATSUT10.EXE> ; IMAGE NAME FOR CREATED PROC
0099 84 QUOTALIST: $QUOTA CPULM,0 ; INFINITE CPU
009E 85 $QUOTA BYTLM,512 ; BYTE LIMIT FOR BUFFERED I/O
00A3 86 $QUOTA FILLM,2 ; OPEN FILE COUNT LIMIT
00A8 87 $QUOTA PGFLQUOTA,10 ; PAGING FILE QUOTA
00AD 88 $QUOTA PRCLM,2 ; SUBPROCESS QUOTA
00B2 89 $QUOTA TQELM,3 ; TIMER QUEUE ENTRY QUOTA
00B7 90 $QUOTA LISTEND ; DEFINES END OF LIST
```

00000000	0000	92	.PSECT	RWDATA,RD,WRT,NOEXE,LONG		
00000008	0000	93	PRIVMASK:	.BLKQ 1	:	ADDR OF PRIVILEGE MASK (IN PHD)
0000000C	0008	94	MBXCHAN:	.BLKL 1	:	CHAN NO. FOR MAILBOX FOR CREATED PROCESS
	000C	95	MBXCHANINFO:		:	CHANNEL INFO RETURNED BY GETCHN
00000074	000C	96		.LONG DIB\$K_LENGTH		
00000014	0010	97		.ADDRESS +4		
00000088	0014	98		.BLKB DIB\$K_LENGTH		
0000008C	0088	99	MBXUNIT:	.BLKL 1	:	SAVE AREA FOR MAILBOX UNIT NUMBER
	008C	100	MBXBUFF:	STRING 0,120	:	MAILBOX BUFFER FOR CREATED PROCESS
00000110	010C	101	CLUS_MASK:	.BLKL 1	:	CLUSTER MASK; USED TO SET SUBJECT CLUSTER
00000114	0110	102	CLUS_STATE:	.BLKL 1	:	STATE OF SUBJECT CLUSTER
00000118	0114	103	EFN_REFCT1:	.BLKL 1	:	SAVE AREA FOR EFN WHEN REF CT = 1

```
0118 105 .SBTTL CONDITION TABLES
0118 106 :
0118 107 :
0118 108 :
0118 109 :
0118 110 ***** CONDITION TABLES FOR DLCEFC SYSTEM SERVICE *****
0118 111 COND 1,NOTARG,<CLUSTER'S PERM/TEMP SETTING AT TIME OF DLCEFC>,-
0118 112 <TEMPORARY>,-
0163 113 <PERMANENT>,-
016B 114 .LONG 0,1 ; TEMPORARY/PERMANENT
016B 115 COND 2,NOTARG,<CLUSTER'S REFERENCE COUNT AT TIME OF DLCEFC>,-
016B 116 <ZERO>,-
016B 117 <ONE (E.F. GROUP 2)>,-
016B 118 <ONE (E.F. GROUP 3)>,-
016B 119 <TWO (BOTH E.F. GROUPS)>,-
016B 120 <FOUR (BOTH E.F. GROUPS IN TWO PROCESSES)>,-
0217 121 COND 3,NULL
0218 122 COND 4,NULL
0218 123 COND 5,NULL
0219 124
0219 125
021A 126
00000000 127
00000000 128 .PSECT SATSSS52,RD,WRT,EXE
```



```

0000 130 .SBTTL TM_SETUP, TM_CLEANUP
0000 131 :++
0000 132 : FUNCTIONAL DESCRIPTION:
0000 133 :
0000 134 : TM SETUP AND TM CLEANUP ARE CALLED TO PERFORM
0000 135 : REQUIRED HOUSEKEEPING AT THE BEGINNING AND END, RESPECTIVELY, OF
0000 136 : TEST MODULE EXECUTION.
0000 137 :
0000 138 : CALLING SEQUENCE:
0000 139 :
0000 140 : BSBW TM_SETUP BSBW TM_CLEANUP
0000 141 :
0000 142 : INPUT PARAMETERS:
0000 143 :
0000 144 : NONE
0000 145 :
0000 146 : IMPLICIT INPUTS:
0000 147 :
0000 148 : NONE
0000 149 :
0000 150 : OUTPUT PARAMETERS:
0000 151 :
0000 152 : NONE
0000 153 :
0000 154 : IMPLICIT OUTPUTS:
0000 155 :
0000 156 : TM_SETUP: COND TABLE INDEX REGISTERS (R2,3,4,5,6) CLEARED;
0000 157 : ALL PRIVILEGES ACQUIRED.
0000 158 :
0000 159 : COMPLETION CODES:
0000 160 :
0000 161 : EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
0000 162 :
0000 163 : SIDE EFFECTS:
0000 164 :
0000 165 : SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
0000 166 : (VIA RSB) IF ERROR ENCOUNTERED.
0000 167 :
0000 168 :--
0000 169 :
0000 170 :
0000 171 :
0000 172 TM_SETUP::
0000 173 CLRL R2 ; INITIALIZE
0000 174 CLRL R3 ; .. CONDITION
0000 175 CLRL R4 ; .... TABLE
0000 176 CLRL R5 ; ..... INDEX
0000 177 CLRL R6 ; ..... REGISTERS
0000 178 BSBW MOD MSG PRINT ; PRINT TEST MODULE BEGIN MSG
0000 179 MOVAL TEST_MOD_SUCC,TMD_ADDR ; ASSUME END MSG WILL SHOW SUCCESS
0000 180 INSV #SUCCESS,#0,#3,MOD_MSG_CODE ; ADJUST STATUS CODE FOR SUCCESS
0000 181
0000 181 MODE TO,5$,KRNL ; KERNEL MODE TO ACCESS PHD
0000 182 MOVL @#CTL$GL,PHD,R9 ; GET PROCESS HEADER ADDRESS
0000 183 MOVAL PHD$Q,PRIVMSK(R9),PRIVMASK ; GET PRIV MASK ADDRESS
0000 184 MODE FROM,5$ ; BACK TO USER MODE
0000 185 PRIV ADD,ALL ; GET ALL PRIVILEGES

```

```

52 D4 0000
53 D4 0002
54 D4 0004
55 D4 0006
56 D4 0008
FFF3' 30 000A
00000000'EF 00000000'EF DE 000D
03 00 00000000'8F FO 0018
00000000'EF 0020
59 00000000'9F D0 0048
00000000'EF 69 DE 004F
0056
0057

```

			0077	186	\$SETPRN_S TEST MOD_NAME_D	; SET PROCESS NAME
			0084	187	SS CHECK NORMAL	; CHECK STATUS CODE RETURNED FROM SETPRN
			00B2	188	\$CREMBX_S CHAN=MBXCHAN, LOGNAM=CREPRN, -	; GET MAILBOX FOR PROCESS
			00B2	189	MAXMSG=#120, PROMSK=#0, BUFQUO=#240	
			00D7	190	SS CHECK NORMAL	; CHECK NORMAL COMPLETION
			0105	191	\$GETCHN_S CHAN=MBXCHAN, -	; GET CHAN INFO (UNIT NUMBER)
			0105	192	PRIBUF=MBXCHANINFO	
			011F	193	SS CHECK NORMAL	; CHECK NORMAL COMPLETION
0000088'EF	0000020'EF	3C	014D	194	MOVZWL MBXCHANINFO+8+DIBSW_UNIT, MBXUNIT	; SAVE MAILBOX UNIT NUMBER
		05	0158	195	RSB	; RETURN TO MAIN ROUTINE
			0159	196	TM_CLEANUP::	
			0159	197	\$DELMBX_S MBXCHAN	; DELETE TERMINATION MAILBOX
	FE96'	30	0167	198	BSBW MOD_MSG_PRINT	; PRINT TEST MODULE END MSG
		05	016A	199	RSB	; RETURN TO MAIN ROUTINE

```

016B 201 .SBTTL CONDITION SUBROUTINES - SETUP AND CLEANUP
016B 202 :++
016B 203 : FUNCTIONAL DESCRIPTION:
016B 204 :
016B 205 : COND1 AND COND2 CLEANUP ARE SUBROUTINES WHICH ARE EXECUTED
016B 206 : BEFORE AND AFTER THE VERIFY SUBROUTINE, RESPECTIVELY, WHENEVER A NEW
016B 207 : CONDITION X VALUE IS SELECTED (SEE FUNCTIONAL DESCRIPTION OF SUCCOMMON
016B 208 : ROUTINE IN SUCCOMMON.MAR). ANY SETUP FUNCTION PARTICULAR TO THE
016B 209 : CONDITION X TABLE IS INCLUDED IN THE COND1 SUBROUTINE AND CLEANED
016B 210 : UP, IF NECESSARY, IN THE COND2 CLEANUP SUBROUTINE. THIS INCLUDES,
016B 211 : ESPECIALLY, CODE TO DETECT CONFLICTS AMONG CURRENT ENTRIES IN TWO
016B 212 : OR MORE CONDITION TABLES. IF A CONFLICT IS DETECTED, A NON-ZERO
016B 213 : VALUE IS STORED INTO CONFLICT, WHICH CAUSES THE CALLING ROUTINE
016B 214 : (SUCCOMMON) TO SKIP THE CURRENT ENTRY IN THE CONDITION X TABLE.
016B 215 :
016B 216 : CALLING SEQUENCE:
016B 217 :
016B 218 : BSBW COND1 BSBW COND2_CLEANUP
016B 219 : WHERE X = 1,2,3,4,5
016B 220 :
016B 221 : INPUT PARAMETERS:
016B 222 :
016B 223 : CONFLICT = 0
016B 224 :
016B 225 : IMPLICIT INPUTS:
016B 226 :
016B 227 : R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
016B 228 : FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
016B 229 :
016B 230 : OUTPUT PARAMETERS:
016B 231 :
016B 232 : CONFLICT SET TO NON-ZERO IF COND TABLE CONFLICT DETECTED.
016B 233 :
016B 234 : IMPLICIT OUTPUTS:
016B 235 :
016B 236 : R2,3,4,5,6 PRESERVED
016B 237 :
016B 238 : COMPLETION CODES:
016B 239 :
016B 240 : NONE
016B 241 :
016B 242 : SIDE EFFECTS:
016B 243 :
016B 244 : NONE
016B 245 :
016B 246 : --
016B 247 :
016B 248 :
016B 249 :
05 016B 250 COND1::
016B 251 RSB ; RETURN TO MAIN ROUTINE
016C 252 COND1_CLEANUP::
05 016C 253 RSB ; RETURN TO MAIN ROUTINE
016D 254 COND2::
05 016D 255 RSB ; RETURN TO MAIN ROUTINE
016E 256 COND2_CLEANUP::
05 016E 257 RSB ; RETURN TO MAIN ROUTINE

```

```
05 016F 258 COND3::
05 016F 259 RSB ; RETURN TO MAIN ROUTINE
05 0170 260 COND3_CLEANUP::
05 0170 261 RSB ; RETURN TO MAIN ROUTINE
05 0171 262 COND4::
05 0171 263 RSB ; RETURN TO MAIN ROUTINE
05 0172 264 COND4_CLEANUP::
05 0172 265 RSB ; RETURN TO MAIN ROUTINE
05 0173 266 COND5::
05 0173 267 RSB ; RETURN TO MAIN ROUTINE
05 0174 268 COND5_CLEANUP::
05 0174 269 RSB ; RETURN TO MAIN ROUTINE
```

```

0175 271 .SBTTL FORM_CONDS
0175 272 :++
0175 273 : FUNCTIONAL DESCRIPTION:
0175 274 :
0175 275 :         FORM_CONDS FORMATS AND PRINTS INFORMATION ABOUT
0175 276 : THE CURRENT ELEMENT IN EACH OF THE CONDITION TABLES.
0175 277 :
0175 278 : CALLING SEQUENCE:
0175 279 :
0175 280 :         BSBW FORM_CONDS
0175 281 :
0175 282 : INPUT PARAMETERS:
0175 283 :
0175 284 :         NONE
0175 285 :
0175 286 : IMPLICIT INPUTS:
0175 287 :
0175 288 :         R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
0175 289 :         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
0175 290 :         FOR X = 1,2,3,4,5 :
0175 291 :             CONDX_T - TITLE TEXT FOR CONDX TABLE
0175 292 :             CONDX_TAB - ELEMENT TEXT FOR CONDX TABLE
0175 293 :             CONDX_C - CONTEXT OF THE CONDX TABLE
0175 294 :             CONDX_E - DATA ELEMENTS OF THE CONDX TABLE
0175 295 :
0175 296 : OUTPUT PARAMETERS:
0175 297 :
0175 298 :         NONE
0175 299 :
0175 300 : IMPLICIT OUTPUTS:
0175 301 :
0175 302 :         NONE
0175 303 :
0175 304 : COMPLETION CODES:
0175 305 :
0175 306 :         NONE
0175 307 :
0175 308 : SIDE EFFECTS:
0175 309 :
0175 310 :         NONE
0175 311 :
0175 312 : --
0175 313 :
0175 314 :
0175 315 :
0175 316 FORM_CONDS::
0175 317 $FAO_S MSG1_INP_CTL,FAO_LEN,FAO_DESC,TESTNUM
0194 318 : FORMAT CONDITIONS HEADER MSG
0194 319 BSBW OUTPUT_MSG : ... AND PRINT IT
14 00 91 0197 CMPB #COND1_C,#NULL : IS CONDITION 1 NULL ?
00BF 31 019A 321 BNEQU 10$ : NO -- CONTINUE
019F 322 BRW FORM_CONDSX : YES -- SUBROUTINE IS FINISHED
00000000'EF 00000118'EF DE 019F 323 10$:
00000000'EF 00000147'EF42 DO 01AA 324 MOVAL COND1_T,MSG_A : SAVE ADDRESS OF CONDITION 1 TITLE FOR FAO
00000000'EF 00000000'EF 00 90 0186 325 MOVL COND1_TAB[R2],MSG_B : SAVE ADDR OF COND 1 CURR TEXT ELT FOR FAO
018D 326 MOVB #COND1_C,MSG_CTXT : SAVE CONDITION 1 CONTEXT FOR FAO
018D 327 MOV_VAL COND1_C,COND1_E[R2],MSG_DATA1 : GIVE COND 1 DATA VALUE TO FAO

```

```

      FE40' 30 01BD 328      BSBW  WRITE_MSG2      ; FORMAT AND WRITE CONDITION 1 MSG
14  00 91 01C0 329      CMPB  #COND2_C,#NULL      ; IS CONDITION 2 NULL ?
      03 12 01C3 330      BNEQU 20$              ; NO -- CONTINUE
      0096 31 01C5 331      BRW    FORM_CONDSX      ; YES -- SUBROUTINE IS FINISHED
      01C9 332
00000000'EF 0000016B'EF DE 01C8 333      20$:  MOVAL  COND2_T,MSG_A      ; SAVE ADDRESS OF CONDITION 2 TITLE FOR FAO
00000000'EF 00000198'EF43 DO 01D3 334      MOVL  COND2_TAB[R3],MSG_B      ; SAVE ADDR OF COND 2 CURR TEXT ELT FOR FAO
      00000000'EF 00 90 01DF 335      MOVB  #COND2_C,MSG_CTXT      ; SAVE CONDITION 2 CONTEXT FOR FAO
      01E6 336      MOV_VAL COND2_C,COND2_E[R3],MSG_DATA1 ; GIVE COND 2 DATA VALUE TO FAO
      FE17' 30 01E6 337      BSBW  WRITE_MSG2      ; FORMAT AND WRITE CONDITION 2 MSG
14  14 91 01E9 338      CMPB  #COND3_C,#NULL      ; IS CONDITION 3 NULL ?
      03 12 01EC 339      BNEQU 30$              ; NO -- CONTINUE
      006D 31 01EE 340      BRW    FORM_CONDSX      ; YES -- SUBROUTINE IS FINISHED
      01F1 341
00000000'EF 00000217'EF DE 01F1 342      30$:  MOVAL  COND3_T,MSG_A      ; SAVE ADDRESS OF CONDITION 3 TITLE FOR FAO
00000000'EF 00000217'EF44 DO 01FC 343      MOVL  COND3_TAB[R4],MSG_B      ; SAVE ADDR OF COND 3 CURR TEXT ELT FOR FAO
      00000000'EF 14 90 0208 344      MOVB  #COND3_C,MSG_CTXT      ; SAVE CONDITION 3 CONTEXT FOR FAO
      020F 345      MOV_VAL COND3_C,COND3_E[R4],MSG_DATA1 ; GIVE COND 3 DATA VALUE TO FAO
      FDEE' 30 020F 346      BSBW  WRITE_MSG2      ; FORMAT AND WRITE CONDITION 3 MSG
14  14 91 0212 347      CMPB  #COND4_C,#NULL      ; IS CONDITION 4 NULL ?
      47 13 0215 348      BEQLU FORM_CONDSX      ; YES -- SUBROUTINE IS FINISHED
00000000'EF 00000218'EF DE 0217 349      MOVAL  COND4_T,MSG_A      ; SAVE ADDRESS OF CONDITION 4 TITLE FOR FAO
00000000'EF 00000218'EF45 DO 0222 350      MOVL  COND4_TAB[R5],MSG_B      ; SAVE ADDR OF COND 4 CURR TEXT ELT FOR FAO
      00000000'EF 14 90 022E 351      MOVB  #COND4_C,MSG_CTXT      ; SAVE CONDITION 4 CONTEXT FOR FAO
      0235 352      MOV_VAL COND4_C,COND4_E[R5],MSG_DATA1 ; GIVE COND 4 DATA VALUE TO FAO
      FDC8' 30 0235 353      BSBW  WRITE_MSG2      ; FORMAT AND WRITE CONDITION 4 MSG
14  14 91 0238 354      CMPB  #COND5_C,#NULL      ; IS CONDITION 5 NULL ?
      21 13 023B 355      BEQLU FORM_CONDSX      ; YES -- SUBROUTINE IS FINISHED
00000000'EF 00000219'EF DE 023D 356      MOVAL  COND5_T,MSG_A      ; SAVE ADDRESS OF CONDITION 5 TITLE FOR FAO
00000000'EF 00000219'EF46 DO 0248 357      MOVL  COND5_TAB[R6],MSG_B      ; SAVE ADDR OF COND 5 CURR TEXT ELT FOR FAO
      00000000'EF 14 90 0254 358      MOVB  #COND5_C,MSG_CTXT      ; SAVE CONDITION 5 CONTEXT FOR FAO
      025B 359      MOV_VAL COND5_C,COND5_E[R6],MSG_DATA1 ; GIVE COND 5 DATA VALUE TO FAO
      FDA2' 30 025B 360      BSBW  WRITE_MSG2      ; FORMAT AND WRITE CONDITION 5 MSG
      025E 361 FORM_CONDSX:
      05 025E 362      RSB    ; RETURN TO CALLER

```

```

025F 364 .SBTTL VERIFY
025F 365 :++
025F 366 : FUNCTIONAL DESCRIPTION:
025F 367 :
025F 368 : VERIFY IS CALLED ONCE FOR EACH COMBINATION OF CONDITION
025F 369 : TABLE VALUES (AS DETERMINED BY THE INDEX REGISTERS R2,3,4,5,6 FOR
025F 370 : COND TABLES 1,2,3,4,5, RESPECTIVELY). VERIFY ESTABLISHES THE CONDITIONS
025F 371 : SPECIFIED BY THE COND TABLES AND ISSUES THE SUBJECT SYSTEM SERVICE
025F 372 : ($DLCEFC). THEN, THE SUCCESSFUL OPERATION OF THE SERVICE IS VERIFIED
025F 373 : BY EXAMINING THE STATUS CODE RETURNED, THE VALUES FOR RETURN ARGUMENTS
025F 374 : AND THE FUNCTIONALITY PERFORMED. THE EXAMINATIONS TAKE THE FORM OF
025F 375 : COMPARISONS AGAINST EXPECTED VALUES. ANY FAILING COMPARISON CAUSES AN
025F 376 : ERR_EXIT MACRO TO BE EXECUTED (EITHER DIRECTLY, OR INDIRECTLY,
025F 377 : THROUGH THE SS_CHECK MACRO); ERR_EXIT SETS EFLAG TO NON-ZERO,
025F 378 : PRINTS ERROR MESSAGES AND CAUSES AN IMMEDIATE RSB TO CALLER.
025F 379 : WHEN ERR_EXIT IS EXECUTED, FURTHER CALLS TO VERIFY ARE SUPPRESSED,
025F 380 : AND, AFTER EXECUTING CLEANUP SUBROUTINES, THE IMAGE EXITS.
025F 381 :
025F 382 : CALLING SEQUENCE:
025F 383 :
025F 384 : BSBW VERIFY
025F 385 :
025F 386 : INPUT PARAMETERS:
025F 387 :
025F 388 : NONE
025F 389 :
025F 390 : IMPLICIT INPUTS:
025F 391 :
025F 392 : R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
025F 393 : FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
025F 394 : FOR X = 1,2,3,4,5 :
025F 395 : CONDX_E - ADDRESS OF TABLE OF DATA VALUES FOR CONDX
025F 396 : TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
025F 397 : ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
025F 398 : FOR CONDX_E.
025F 399 :
025F 400 : OUTPUT PARAMETERS:
025F 401 :
025F 402 : NONE
025F 403 :
025F 404 : IMPLICIT OUTPUTS:
025F 405 :
025F 406 : VERIFY HAS NO OUTPUT. SINCE ITS PURPOSE IS TO TEST FOR ERRORS,
025F 407 : IT MERELY RETURNS TO CALLER NORMALLY AFTER THE TESTS, PROVIDING
025F 408 : ALL WERE SUCCESSFUL; IF AN ERROR IS DISCOVERED, RETURN IS VIA
025F 409 : AN ERR_EXIT OR SS_CHECK MACRO, BOTH OF WHICH DOCUMENT DETECTED
025F 410 : ERRORS.
025F 411 :
025F 412 : COMPLETION CODES:
025F 413 :
025F 414 : EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
025F 415 :
025F 416 : SIDE EFFECTS:
025F 417 :
025F 418 : SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
025F 419 : (VIA RSB) IF ERROR ENCOUNTERED.
025F 420 :

```

```

025F 421 :--
025F 422
025F 423
025F 424
025F 425 VERIFY::
00000000'EF 95 025F 426 TSTB CFLAG ; SHOULD CONDITIONS BE PRINTED ?
03 13 0265 427 BEQL 5$ ; NO -- CONTINUE
FF0B 30 0267 428 BSBW FORM_CONDS ; YES -- FMT & PRINT ALL CONDS FOR THIS T.C.
026A 429 5$:
026A 430
026A 431 : CREATE A CLUSTER AND GET IT BUILT AT LEAST ONCE, THEN DELETE
026A 432 : IT WITH A DACEFC.
026A 433
026A 434 $ASCEFC S EFN=#64, NAME=CLUS_NAME ; CREATE A NEW CLUSTER
0281 435 SS_CHECK NORMAL ; CHECK ITS COMPLETION
5A 40 8F 9A 02AF 436 MOVZBL #64,R10 ; IDENTIFY EVENT FLAG GROUP TO BUILD_CLUSTER
04EA 30 0283 437 BSBW BUILD_CLUSTER ; BUILD THE SUBJECT CLUSTER
00000000'EF 95 0286 438 TSTB EFLAG ; IS AN ERROR BEING PROCESSED ?
03 13 028C 439 BEQL 10$ ; NO -- CONTINUE
04A2 31 02BE 440 BRW VERIFYX ; YES -- RETURN IMMEDIATELY
02C1 441 10$:
02C1 442 $SETEF S EFN=#95 ; ENSURE CLUSTER HAS AT LEAST ONE FLAG ON
02CE 443 SS_CHECK WASCLR ; FLAG 95 SHOULD HAVE BEEN CLEAR PREVIOUSLY
02FC 444 $DACEFC S EFN=#64 ; NOW GET RID OF TEMPORARY CLUSTER
0309 445 SS_CHECK NORMAL ; ... CHECK ITS COMPLETION
0337 446
0337 447 : THE FOLLOWING CASE INSTRUCTION AND SUBSEQUENT CODING
0337 448 : ISSUES AS MANY ASCEFC'S AS NECESSARY TO ACHIEVE THE
0337 449 : SPECIFIED REFERENCE COUNT FOR THIS TEST CASE.
0337 450
00000114'EF 40 8F 9A 0337 451 MOVZBL #64,EFN,REFCT1 ; ASSUME EVENT FLAG GROUP 2
04 00 53 8F 033F 452 CASEB R3,#0,#4 ; ISSU CORRECT ASCEFC'S PER COND 2 INDEX REG
0343 453 15$: ; START OF CASE WORD DISPLACEMENTS
000D' 0343 454 .WORD 20$-15$ ; REF COUNT 0
0010' 0345 455 .WORD 30$-15$ ; REF COUNT 1, EVENT FLAG GROUP 2
005D' 0347 456 .WORD 40$-15$ ; REF COUNT 1, EVENT FLAG GROUP 3
00B2' 0349 457 .WORD 50$-15$ ; REF COUNT 2
00B2' 034B 458 .WORD 50$-15$ ; REF COUNT 4
01A5 31 034D 459 BRW 55$ ; BRANCH PAST CASE ROUTINES
0350 460 20$:
0350 461
0350 462 : REF COUNT 0, NO ASCEFC'S TO BE ISSUED
0350 463
01B7 31 0350 464 BRW 60$ ; GO ON TO ISSUE SUBJECT DLCEFC
0353 465 30$:
0353 466
0353 467 : REF COUNT 1, EVENT FLAG GROUP 2
0353 468
0353 469 $ASCEFC_S EFN=EFN,REFCT1, - ; INCREMENT REF COUNT; EFN SET UP ABOVE
0353 470 NAME=CLUS_NAME, -
0353 471 PERM=COND_T_E[R2]
036F 472 SS_CHECK NORMAL ; CHECK FOR NORMAL COMPLETION
0155 31 039D 473 BRW 55$ ; GO BUILD THE CLUSTER JUST CREATED
03A0 474 40$:
03A0 475
03A0 476 : REF COUNT 1, EVENT FLAG GROUP 3
03A0 477

```



```

00000114'EF 60 8F 9A 03A0 478 MOVZBL #96,EFN,REFCT1 ; EST EFN FOR E.F. GROUP 3
03A8 479 $ASCEFC_S EFN=EFN,REFCT1, - ; INCREMENT REF COUNT
03A8 480 NAME=CLUS_NAME, -
03A8 481 PERM=CONDT_E[R2]
0100 31 03C4 482 SS CHECK NORMAL ; CHECK FOR NORMAL STATUS CODE
03F2 483 BRW 55$ ; GO BUILD CLUSTER JUST CREATED
03F5 484 50$:
03F5 485 :
03F5 486 : REF COUNT 2 OR 4, BOTH EVENT FLAG GROUPS IN THIS PROCESS
03F5 487 :
03F5 488 $ASCEFC_S EFN=#64, - ; INCREMENT REF COUNT
03F5 489 NAME=CLUS_NAME, -
03F5 490 PERM=CONDT_E[R2]
0411 491 SS CHECK NORMAL ; CHECK FOR NORMAL COMPLETION
043F 492 $ASCEFC_S EFN=#96, - ; INCREMENT REF COUNT
043F 493 NAME=CLUS_NAME, -
043F 494 PERM=CONDT_E[R2]
045B 495 SS CHECK NORMAL ; CHECK FOR NORMAL COMPLETION
0489 496 CMPL R3,#4 ; FIFTH COND 2 ELEMENT (REF COUNT 4) ?
048C 497 BNEQU 55$ ; NO -- REF COUNT IS 2 -- NO MORE ASCEFC'S
048E 498 ; YES -- CREATED PROC ISSUE 2 MORE ASCEFC'S
048E 499 $CREPRC_S PRCNAM=CREPRN, IMAGE=IMAGNAM, -
048E 500 MBXUNT=MBXUNIT, QUOTA=QUOTALIST
04C0 501 SS CHECK NORMAL ; CHECK CREPRC COMPLETION STATUS
04EE 502 $HTBER_S ; SLEEP UNTIL CREATED PROCESS DOES ASCEFC'S
04F5 503 55$:
04F5 504 :
04F5 505 : END OF CASE ROUTINES
04F5 506 :
04F5 507 : AT THIS POINT, AT LEAST ONE ASCEFC HAS BEEN ISSUED; CALL
04F5 508 : BUILD_CLUSTER TO GET THE CREATED CLUSTER BUILT.
04F5 509 :
SA 00000114'EF D0 04F5 510 MOVL EFN,REFCT1,R10 ; ESTABLISH EFN FOR BUILD_CLUSTER
02A1 30 04FC 511 BSBW BUIED_CLUSTER ; BUILD CLUSTER
00000000'EF 95 04FF 512 TSTB EFLAG ; IS AN ERROR BEING PROCESSED ?
03 13 0505 513 BEQL 60$ ; NO -- CONTINUE
0259 31 0507 514 BRW VERIFYX ; YES -- RETURN IMMEDIATELY
050A 515 60$:
050A 516 :
050A 517 : ***** SYSTEM SERVICE CALL WHICH IS THE SUBJECT OF THIS TEST CASE *****
050A 518 :
050A 519 $DLCEFC_S NAME=CLUS_NAME ; CLEAR PERMANENT INDICATOR
00000000'8F 50 D1 0517 520 CMPL RO,#SS$_NORMAL ; CODE RECEIVED = CODE EXPECTED ?
61 13 051E 521 BEQLU 70$ ; YES -- CONTINUE
00000000'EF 00000000'8F D0 0520 522 MOVL #SS$_NORMAL,EXPV ; LOAD UP EXPECTED AND
00000000'EF 50 D0 052B 523 MOVL RO,RECV ; ... RECEIVED VALUES, THEN EXIT
0532 524 ERR_EXIT LONG,<INCORRECT STATUS CODE RETURNED FROM DLCEFC>
0581 525 70$:
0581 526 :
0581 527 : TO VERIFY THE SUBJECT DLCEFC, THE CLUSTER'S REFERENCE COUNT
0581 528 : WILL BE DROPPED TO 0 (BY ISSUING DLCEFC'S) AND THEN AN
0581 529 : ADDITIONAL ASCEFC WILL BE ISSUED TO CREATE A NEW CLUSTER
0581 530 : WHICH WILL BE VERIFIED TO BE ALL 0'S. BEFORE EACH DLCEFC
0581 531 : IS ISSUED, A READEF OF THE CLUSTER WILL BE DONE TO ENSURE
0581 532 : THAT THE CLUSTER REMAINS EQUAL TO THE CLUSTER MASK; THIS
0581 533 : GUARANTEES THAT THE REFERENCE COUNT WAS CORRECT AND, IN
0581 534 : FACT, THAT THE CLUSTER IS NOT DELETED UNTIL THE COUNT

```

```

0581 535 : GOES TO 0.
0581 536 :
0581 537 :
0581 538 : THE FOLLOWING CASE STATEMENT AND SUBSEQUENT CODING
0581 539 : DECREMENTS THE REFERENCE COUNT BY ISSUING THE CORRECT
0581 540 : SEQUENCE OF DACEFC'S, BASED ON THE REFERENCE COUNT
0581 541 : FOR THIS TEST CASE.
0581 542 :
04 00 53 8F 0581 543 : CASEB R3,#0,#4 ; ISSU CORRECT DACEFC'S PER COND 2 INDEX REG
0585 544 75$: ; CASE INSTRUCTION WORD DISPLACEMENTS
000D' 0585 545 : .WORD 80$-75$ ; REF COUNT 0
0010' 0587 546 : .WORD 90$-75$ ; REF COUNT 1, EVENT FLAG GROUP 2
0010' 0589 547 : .WORD 90$-75$ ; REF COUNT 1, EVENT FLAG GROUP 3
0028' 058B 548 : .WORD 100$-75$ ; REF COUNT 2
0028' 058D 549 : .WORD 100$-75$ ; REF COUNT 4
00DB 31 058F 550 : BRW 130$ ; BRANCH PAST CASE ROUTINES
0592 551 80$:
0592 552 :
0592 553 : REF COUNT 0, NO DACEFC NECESSARY
0592 554 :
00DB 31 0592 555 : BRW 130$ ; GO ON TO CHECK CLUSTER FOR 0'S
0595 556 90$:
0595 557 :
0595 558 : REF COUNT 1, ISSUE ONE DACEFC
0595 559 :
5A 00000114'EF D0 0595 560 : MOVL EFN,REFCT1,R10 ; SET UP CORRECT EFN FOR READ DACEFC SUBRTN
02A7 30 059C 561 : BSBW READ DACEFC ; CHECK THE CLUSTER AND DISASSOCIATE
00000000'EF 95 059F 562 : TSTB EFLAG ; IS AN ERROR BEING PROCESSED ?
03 13 05A5 563 : BEQL 95$ ; NO -- CONTINUE
01B9 31 05A7 564 : BRW VERIFYX ; YES -- RETURN IMMEDIATELY
00C0 31 05AA 565 95$:
05AD 566 : BRW 130$ ; GO ON TO CHECK CLUSTER FOR 0'S
05AD 567 100$:
05AD 568 :
05AD 569 : REF COUNT 2 OR 4, ISSUE 2 OR 4 DACEFC'S
05AD 570 :
04 53 D1 05AD 571 : .C MPL R3,#4 ; FIFTH COND 2 ELEMENT (REF COUNT 4) ?
03 13 05B0 572 : BEQLU 105$ ; YES -- CONTINUE
0094 31 05B2 573 : BRW 110$ ; NO -- REF COUNT MUST BE 2
05B5 574 105$:
05B5 575 : $WAKE_S PRCNAM=CREPRN ; WAKE PROCESS TO GET DACEFC'S ISSUED
05C4 576 : SS_CHECK NORMAL ; CHECK FOR NORMAL STATUS CODE
05F2 577 : $QIOW_S CHAN=MBXCHAN, FUNC=#IOS$ READVBLK, -
05F2 578 : P1=MBXBUFF+8, P2=MBXBUFF ; ... AND WAIT FOR IT TO SEND MAIL
061B 579 :
061B 580 : SS_CHECK NORMAL ; CHECK FOR NORMAL STATUS CODE
0649 581 110$:
5A 40 8F 9A 0649 582 : MOVZBL #64,R10 ; SET UP CORRECT EFN FOR READ DACEFC SUBRTN
01F6 30 064D 583 : BSBW READ DACEFC ; CHECK THE CLUSTER AND DISASSOCIATE
00000000'EF 95 0650 584 : TSTB EFLAG ; IS AN ERROR BEING PROCESSED ?
03 13 0656 585 : BEQL 120$ ; NO -- CONTINUE
0108 31 0658 586 : BRW VERIFYX ; YES -- RETURN IMMEDIATELY
065B 587 120$:
5A 60 8F 9A 065B 588 : MOVZBL #96,R10 ; SET UP CORRECT EFN FOR READ DACEFC SUBRTN
01E4 30 065F 589 : BSBW READ DACEFC ; CHECK THE CLUSTER AND DISASSOCIATE
00000000'EF 95 0662 590 : TSTB EFLAG ; IS AN ERROR BEING PROCESSED ?
03 13 0668 591 : BEQL 130$ ; NO -- CONTINUE
    
```

```

00F6 31 066A 592 BRW VERIFYX ; YES -- RETURN IMMEDIATELY
      066D 593 130$:
      066D 594 :
      066D 595 : REFERENCE COUNT SHOULD BE DOWN TO 0 AND CLUSTER DELETED.
      066D 596 : DO ONE MORE ASCEFC AND EXPECT A CLUSTER OF 0'S.
      066D 597 :
      066D 598 : $ASCEFC S EFN=#64, NAME=CLUS_NAME ; RE-ASSOCIATE SAME CLUSTER
      0684 599 : SS_CHECK NORMAL ; CHECK COMPLETION STATUS
      06B2 600 : $READEF S EFN=#64, STATE=CLUS_STATE ; READ CLUSTER
2E 50 E8 06C5 601 BLBS RO,140$ ; CONTINUE IF NORMAL COMPLETION
      06C8 602 SS_CHECK NORMAL ; USE SS_CHECK TO TERMINATE TEST MODULE
      06F6 603 140$:
00000110'EF D5 06F6 604 T_TL CLUS STATE ; IS CLUSTER INITIALIZED TO 0'S ?
      65 13 06FC 605 BEQLU VERIFYX ; YES -- THIS TEST CASE COMPLETE
00000000'EF D4 06FE 606 CLRL EXPV ; NO -- LOAD EXPECTED AND ...
00000110'EF D0 0704 607 MOVL CLUS STATE,RCV ; ... RECEIVED VALUES, THEN EXIT
      070F 608 ERR_EXIT LONG,<DELETED AND RE-ASSOCIATED CLUSTER NOT RE-INIT'D>
      0763 609 VERIFYX:
05 0763 610 RSB ; RETURN TO CALLER

```

```

0764 612 .SBTTL VFY_CLEANUP
0764 613 :++
0764 614 : FUNCTIONAL DESCRIPTION:
0764 615 :
0764 616 : VFY_CLEANUP EXECUTES SYSTEM SERVICES TO UNDO THE
0764 617 : EFFECT OF THOSE ISSUED IN THE VERIFY SUBROUTINE. VFY_CLEANUP MUST
0764 618 : ASSUME THAT VERIFY MAY NOT HAVE EXECUTED IN ITS ENTIRETY (IF AN
0764 619 : ERROR IS FOUND). ALSO, VFY_CLEANUP MAY ISSUE SS_CHECK OR ERR_EXIT
0764 620 : ONLY AFTER PERFORMING ALL OF ITS CLEANUP OPERATIONS; THIS IS REQUIRED
0764 621 : IN THE EVENT THAT VFY_CLEANUP IS CALLED DURING ERROR PROCESSING,
0764 622 : WHEN PERFORMING THE REQUIRED CLEANUP IS MORE IMPORTANT THAN
0764 623 : POSSIBLY DISCOVERING A SECOND ERROR.
0764 624 :
0764 625 : CALLING SEQUENCE:
0764 626 :
0764 627 : BSBW VFY_CLEANUP
0764 628 :
0764 629 : INPUT PARAMETERS:
0764 630 :
0764 631 : NONE
0764 632 :
0764 633 : IMPLICIT INPUTS:
0764 634 :
0764 635 : R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
0764 636 : FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
0764 637 : FOR X = 1,2,3,4,5 :
0764 638 : COND_X_E - ADDRESS OF TABLE OF DATA VALUES FOR COND_X
0764 639 : TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
0764 640 : ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
0764 641 : FOR COND_X_E.
0764 642 :
0764 643 : OUTPUT PARAMETERS:
0764 644 :
0764 645 : NONE
0764 646 :
0764 647 : IMPLICIT OUTPUTS:
0764 648 :
0764 649 : NONE
0764 650 :
0764 651 : COMPLETION CODES:
0764 652 :
0764 653 : EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
0764 654 :
0764 655 : SIDE EFFECTS:
0764 656 :
0764 657 : SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
0764 658 : (VIA RSB) IF ERROR ENCOUNTERED.
0764 659 :
0764 660 :--
0764 661 :
0764 662 :
0764 663 :
0764 664 VFY_CLEANUP::
0764 665 $DLCEFC S NAME=CLUS_NAME : CLEAR PERM INDICATOR IF PRESENT ...
0771 666 SS_CHECK NORMAL : ... AND CHECK COMPLETION
05 079F 667 RSB : RETURN TO CALLER

```

```

07A0 669 .SBTTL BUILD_CLUST SUBROUTINE
07A0 670 :
07A0 671 : *****
07A0 672 : *
07A0 673 : * BUILD_CLUST SUBROUTINE
07A0 674 : *
07A0 675 : * THIS SUBROUTINE CREATES A 32-BIT CLUSTER MASK BY
07A0 676 : * CONCATENATING THE LOW-ORDER BYTES OF REGS R2-R5.
07A0 677 : * IT THEN SETS CLUSTER A EQUAL TO THE MASK BY
07A0 678 : * ISSUING THE PROPER COMBINATION OF 32 SETEF/CLREF'S.
07A0 679 : *
07A0 680 : * INPUTS:
07A0 681 : *
07A0 682 : * R2,R3,R4,R5 - CONDITION TABLE INDEX VALUES
07A0 683 : *
07A0 684 : * R10 - ANY EFN IN CLUSTER A
07A0 685 : *
07A0 686 : * OUTPUTS:
07A0 687 : *
07A0 688 : * CLUS_MASK - LONGWORD CONTAINING THE CREATED
07A0 689 : * CLUSTER MASK.
07A0 690 : *
07A0 691 : * SUBJECT CLUSTER - UPDATED TO LOOK LIKE CLUS_MASK.
07A0 692 : *
07A0 693 : * VOLATILE REGISTERS:
07A0 694 : *
07A0 695 : * R0, R1, R8, R9
07A0 696 : *
07A0 697 : *****
07A0 698 :
07A0 699 BUILD_CLUST:
0000010C'EF 55 90 07A0 700 MOVB R5,CLUS_MASK ; BUILD
0000010D'EF 54 90 07A7 701 MOVB R4,CLUS_MASK+1 ; .. CLUSTER
0000010E'EF 53 90 07AE 702 MOVB R3,CLUS_MASK+2 ; ..... MASK
0000010F'EF 52 90 07B5 703 MOVB R2,CLUS_MASK+3 ; .....
07BC 704 :
07BC 705 : THE FOLLOWING CODE SETS SUBJECT CLUSTER EQUAL TO CLUS_MASK
07BC 706 :
58 SA D0 07BC 707 MOVL R10,R8 ; ESTABLISH FIRST EFN (EVENT FLAG NO.)
59 D4 07BF 708 CLRL R9 ; INIT OFFSET INTO CLUS_MASK
3A 0000010C'EF 59 E0 07C1 709 20$: BBS R9,CLUS_MASK,30$ ; ISSUE $SETEF IF BIT FOR THIS FLAG IS SET
68 50 E8 07C9 711 $CLREF_S EFN=R8 ; ... OTHERWISE, ISSUE $CLREF
2E 50 E8 07D2 712 BLBS R0,40$ ; IF NORMAL STATUS, PROCESS NEXT EVENT FLAG
0803 713 SS_CHECK NORMAL ; USE SS_CHECK TO TERMINATE TEST MODULE
0803 714 30$: $SETEF_S EFN=R8 ; SET CURRENT EVENT FLAG
080C 716 BLBS R0,40$ ; IF NORMAL STATUS, PROCESS NEXT EVENT FLAG
080F 717 SS_CHECK NORMAL ; USE SS_CHECK TO TERMINATE TEST MODULE
FF7C 59 01 58 B6 083D 718 40$: INCW R8 ; GET NEXT EFN
01 1F 9D 083F 719 ACBB #31,#1,R9,20$ ; GO DO NEXT EVENT FLAG
05 0845 720 RSB ; RETUR' TO CALLER
07A0 721

```

```

0846 723      .SBTTL READ_DACEFC SUBROUTINE
0846 724      :
0846 725      :*****
0846 726      :
0846 727      : READ_DACEFC SUBROUTINE
0846 728      :
0846 729      : THIS SUBROUTINE COMPARES THE CLUSTER AGAINST THE MASK
0846 730      : AND THEN DISASSOCIATES THE EVENT FLAG GROUP SPECIFIED
0846 731      : BY R10 FROM THE CLUSTER.
0846 732      :
0846 733      : INPUTS:
0846 734      :
0846 735      :     R10      -      ANY EFN IN THE EVENT FLAG GROUP
0846 736      :                   TO BE DISASSOCIATED.
0846 737      :
0846 738      :     CLUS_MASK -      LONGWORD CONTAINING THE IMAGE
0846 739      :                   OF THE SUBJECT CLUSTER.
0846 740      : OUTPUTS:
0846 741      :
0846 742      :     NONE.
0846 743      :
0846 744      : EXITS:
0846 745      :
0846 746      :     NORMAL   -      RSB TO CALLER IF CLUSTER EQUALS
0846 747      :                   MASK AND SYSTEM SERVICES
0846 748      :                   FINISH NORMALLY.
0846 749      :
0846 750      :     ERROR    -      ERR_EXIT MACRO EXECUTED IF ABOVE
0846 751      :                   NORMAL CONDITIONS ARE NOT MET.
0846 752      :
0846 753      : VOLATILE REGISTERS, AREAS:
0846 754      :
0846 755      :     R0, R1, R8, R9, CLUS_STATE, EXPV, RECV
0846 756      :
0846 757      :*****
0846 758      :
0846 759      READ_DACEFC:
0846 760      $REDEF S EFN=R10, STATE=CLUS_STATE ; READ CLUSTER
2E 50  E8 0855 761      BLBS -R0,10$ ; CONTINUE IF NORMAL COMPLETION
0858 762      SS_CHECK NORMAL ; USE SS_CHECK TO TERMINATE TEST MODULE
0886 763 10$:
0886 764      CML CLUS_STATE,CLUS_MASK ; DOES CLUSTER STILL = MASK ?
0891 765      BEQLU 20$ ; YES -- ISSUE DACEFC AND GET OUT
0893 766      MOVL CLUS_MASK,EXPV ; NO -- LOAD EXPECTED AND ...
089E 767      MOVL CLUS_STATE,RECV ; ... RECEIVED VALUES, THEN EXIT
08A9 768      ERR_EXIT LONG,<CLUSTER STATE NOT MAINTAINED ACROSS DLCEFC OR DACEFC>
0902 769 20$:
0902 770      $DACEFC S EFN=R10 ; DISASSOCIATE (DECR REF CT BY 1)
0908 771      SS_CHECK NORMAL ; CHECK FOR NORMAL COMPLETION
0939 772 READ_DACEFCX:
05 0939 773      RSB ; RETURN TO CALLER
093A 774      .END

```

0000010C'EF 00000110'EF D1 0886 764
 6F 13 0891 765
 00000000'EF 0000010C'EF D0 0893 766
 00000000'EF 00000110'EF D0 089E 767

\$\$\$\$	= 000008B3	R	04	DIBSW_UNIT	= 0000000C		
\$\$\$CHARS	= 00000034			EFLAG	*****	X	04
\$\$\$CHARS1	= 00000004			EFN_REFCT1	00000114	R	03
\$\$\$CHARS2	= 00000012			EXPV	*****	X	04
\$\$\$CHARS3	= 00000012			FAO_DESC	*****	X	04
\$\$\$CHARS4	= 00000016			FAO_LEN	*****	X	04
\$\$\$CHARS5	= 00000028			FORM_CONDS	00000175	RG	04
\$\$\$COND_A	= 00000004			FORM_CONDSX	0000025E	R	04
\$\$\$STRINGS	= 00000001			IMAGNAM	0000007A	R	02
\$\$\$STRINGS2	= 00000005			IOS_READVBLK	*****	X	04
\$\$T1	= 00000000			LONG	= 00000004	G	
\$\$T2	= 00000004			MBXBUFF	0000008C	R	03
BUILD_CLUST	000007A0	R	04	MBXCHAN	00000008	R	03
BYTE	= 00000001			MBXCHANINFO	0000000C	R	03
CFLAG	*****	X	04	MBXUNIT	00000088	R	03
CHMRTN	*****	X	04	MOD_MSG_CODE	*****	X	04
CHM_CONT	*****	X	04	MOD_MSG_PRINT	*****	X	04
CLUS_MASK	0000010C	R	03	MSGT_INP_CTL	00000019	R	02
CLUS_NAME	00000065	R	02	MSG3_ERR_CTL	00000039	RG	02
CLUS_STATE	00000110	R	03	MSG_A	*****	X	04
COMP_SC	*****	X	04	MSG_B	*****	X	04
COND1	0000016B	RG	04	MSG_CTXT	*****	X	04
COND1_C	= 00000000			NOTARG	= 00000000	G	
COND1_CLEANUP	0000016C	RG	04	NULL	= 00000014	G	
COND1_E	00000163	R	03	OUTPUT_MSG	*****	X	04
COND1_H	00000146	RG	03	PCV	*****	X	04
COND1_T	00000118	R	03	PHDSQ_PRIVMSK	= 00000000		
COND1_TAB	00000147	R	03	PQL\$_BYTLM	= 00000003		
COND2	0000016D	RG	04	PQL\$_CPULM	= 00000004		
COND2_C	= 00000000			PQL\$_FILLM	= 00000006		
COND2_CLEANUP	0000016E	RG	04	PQL\$_LISTEND	= 00000000		
COND2_E	00000217	R	03	PQL\$_PGFLQUOTA	= 00000007		
COND2_H	00000197	RG	03	PQL\$_PRCLM	= 00000008		
COND2_T	0000016B	R	03	PQL\$_TQELM	= 00000009		
COND2_TAB	00000198	R	03	PRIVMASK	= 00000000	R	03
COND3	0000016F	RG	04	PRIV_ARGS	= 00000002		
COND3_C	= 00000014			PROCESS_ERR	*****	X	04
COND3_CLEANUP	00000170	RG	04	QUAD	= 00000008	G	
COND3_H	00000217	RG	03	QUOTALIST	00000099	R	02
COND3_T	00000217	R	03	READ_DACEFC	00000846	R	04
COND3_TAB	00000217	R	03	READ_DACEFCX	00000939	R	04
COND4	00000171	RG	04	RECV	*****	X	04
COND4_C	= 00000014			REST_REGS	*****	X	04
COND4_CLEANUP	00000172	RG	04	SAVE_REGS	*****	X	04
COND4_H	00000218	RG	03	SS\$_NORMAL	*****	X	04
COND4_T	00000218	R	03	SS\$_WASCLR	*****	X	04
COND4_TAB	00000218	R	03	SUCCESS	*****	X	04
COND5	00000173	RG	04	SYSSASCEFC	*****	GX	04
COND5_C	= 00000014			SYSSCLREF	*****	GX	04
COND5_CLEANUP	00000174	RG	04	SYSSCMKRNL	*****	GX	04
COND5_H	00000219	RG	03	SYSSCREMBX	*****	GX	04
COND5_T	00000219	R	03	SYSSCREPRC	*****	GX	04
COND5_TAB	00000219	R	03	SYSSDACEFC	*****	GX	04
CREPRN	00000051	R	02	SYSSDELMBX	*****	GX	04
CTL\$GL_PHD	*****	X	04	SYSSDLCEFC	*****	GX	04
DESC	= 00000010	G		SYSSFAO	*****	X	04
DIBSK_LENGTH	= 00000074			SYSSGETCHN	*****	GX	04

```

SYSSHIBER      ***** GX 04
SYSSQIOW       ***** GX 04
SYSSREADEF     ***** GX 04
SYSSSETEF      ***** GX 04
SYSSSETPRN     ***** GX 04
SYSSSETPRV     ***** GX 04
SYSSWAKE       ***** GX 04
TESTNUM        ***** X 04
TEST_MOD_NAME  00000000 RG 02
TEST_MOD_NAME_D 00000009 R 02
TEST_MOD_SUCC  ***** X 04
TMD_ADDR       ***** X 04
TM_CLEANUP     00000159 RG 04
TM_SETUP       00000000 RG 04
VERIFY         0000025F RG 04
VERIFYX        00000763 R 04
VFY_CLEANUP    00000764 RG 04
WORD           = 00000002 G
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	000000BC (188.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC LONG
RWDATA	0000021A (538.)	03 (3.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
SATSS52	0000093A (2362.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.07	00:00:00.31
Command processing	140	00:00:00.68	00:00:02.93
Pass 1	292	00:00:08.83	00:00:16.59
Symbol table sort	0	00:00:00.60	00:00:00.63
Pass 2	158	00:00:02.39	00:00:03.13
Symbol table output	16	00:00:00.10	00:00:00.13
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	645	00:00:12.70	00:00:23.76

The working set limit was 1650 pages.
 47530 bytes (93 pages) of virtual memory were used to buffer the intermediate code.
 There were 30 pages of symbol table space allocated to hold 357 non-local and 80 local symbols.
 774 source lines were read in Pass 1, producing 28 object records in Pass 2.
 50 pages of virtual memory were used to define 40 macros.

! Macro library statistics !

Macro library name	Macros defined
-----	-----
\$255\$DUA28:[SHRLIB]UETP.MLB;1	9
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	1
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	27
TOTALS (all libraries)	37

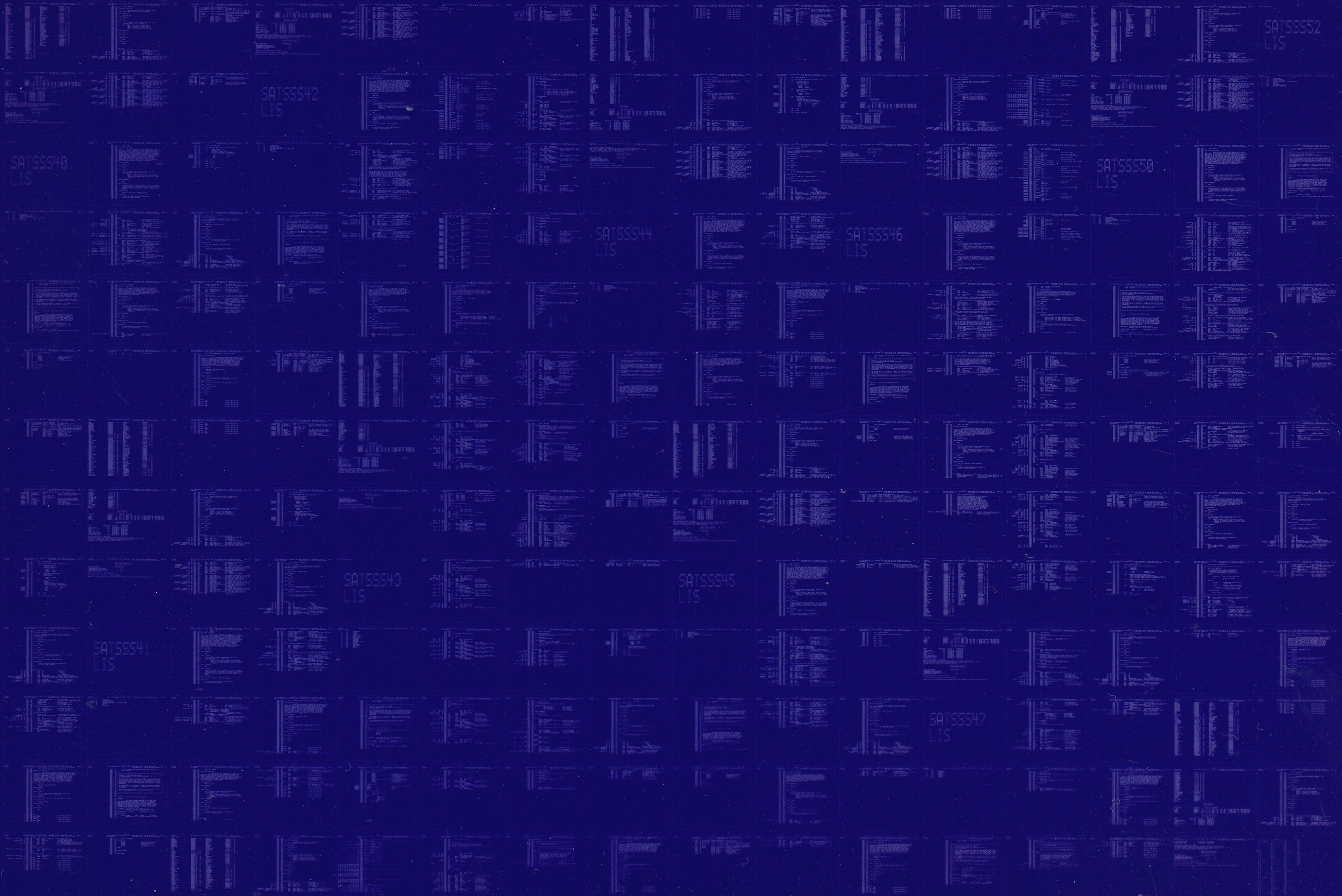
775 GETS were required to define 37 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:SATSS52/OBJ=OBJ\$:SATSS52 MSRC\$:SATSS52/UPDATE=(ENH\$:SATSS52)+EXECML\$/LIB+SHRLIB\$:UETP/LIB

0423 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 fragments, likely representing a data table or a collection of micro-panels. The fragments are arranged in a regular pattern across the page. Several fragments are clearly legible and contain the text "SATSS553 LIS", "SATSS554 LIS", "SATSS555 LIS", "SATSS556 LIS", "SATSS560 LIS", "SATSS570 LIS", "SATSS571 LIS", "SATSS572 LIS", and "SATSS573 LIS". These fragments appear to be part of a larger dataset or a series of related items. The overall appearance is that of a high-resolution scan of a document with very small text.