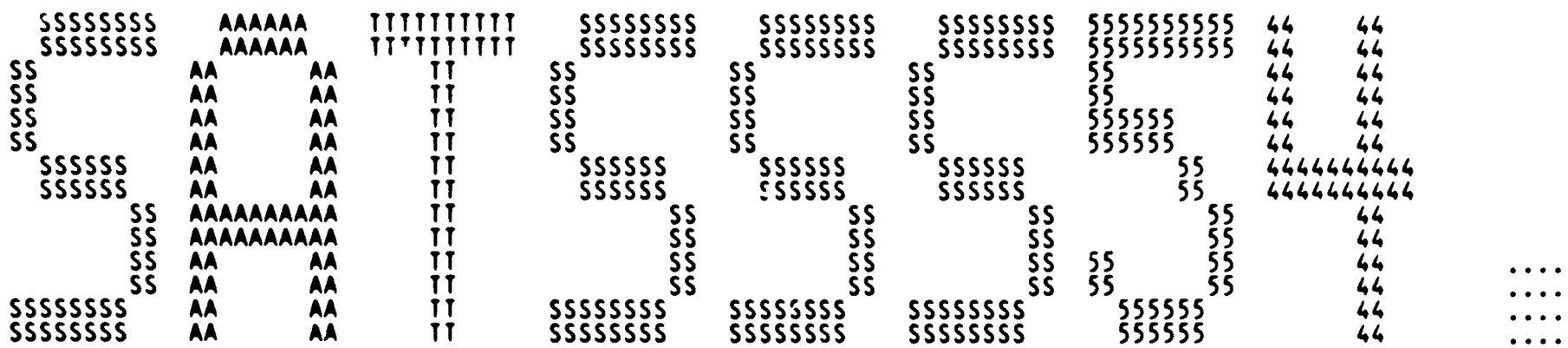
```
UUU          UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP        SSSSSSSSSSS  YYY          YYY
UUU          UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP        SSSSSSSSSSS  YYY          YYY
UUU          UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP        SSSSSSSSSSS  YYY          YYY
UUU          UUU  EEE                    TTT        PPP          PPP  SSS              YYY          YYY
UUU          UUU  EEE                    TTT        PPP          PPP  SSS              YYY          YYY
UUU          UUU  EEE                    TTT        PPP          PPP  SSS              YYY          YYY
UUU          UUU  EEE                    TTT        PPP          PPP  SSS                 YYY    YYY
UUU          UUU  EEE                    TTT        PPP          PPP  SSS                 YYY    YYY
UUU          UUU  EEE                    TTT        PPP          PPP  SSS                 YYY    YYY
UUU          UUU  EEEEEEEEEEEE           TTT        PPPPPPPPPPPP       SSSSSSSSS              YYY
UUU          UUU  EEEEEEEEEEEE           TTT        PPPPPPPPPPPP       SSSSSSSSS              YYY
UUU          UUU  EEEEEEEEEEEE           TTT        PPPPPPPPPPPP       SSSSSSSSS              YYY
UUU          UUU  EEE                    TTT        PPP                        SSS              YYY
UUU          UUU  EEE                    TTT        PPP                        SSS              YYY
UUU          UUU  EEE                    TTT        PPP                        SSS              YYY
UUU          UUU  EEE                    TTT        PPP                        SSS              YYY
UUU          UUU  EEE                    TTT        PPP                        SSS              YYY
UUUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE        TTT        PPP                SSSSSSSSSSS              YYY
UUUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE        TTT        PPP                SSSSSSSSSSS              YYY
UUUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE        TTT        PPP                SSSSSSSSSSS              YYY
```


SSSSSSSS AAAAAA TTTTTTTTTT SSSSSSSS SSSSSSSS SSSSSSSS 5555555555 44 44
SSSSSSSS AAAAAA TTTTTTTTTT SSSSSSSS SSSSSSSS SSSSSSSS 5555555555 44 44
SS AA AA TT SS SS SS 55 44 44
SS AA AA TT SS SS SS 55 44 44
SS AA AA TT SS SS SS 555555 44 44
SS AA AA TT SS SS SS 555555 44 44
SSSSSS AA AA TT SSSSSS SSSSSS SSSSSS 55 4444444444
SSSSSS AA AA TT SSSSSS SSSSSS SSSSSS 55 4444444444
SS AAAAAAAAAA TT SS SS SS 55 44
SS AAAAAAAAAA TT SS SS SS 55 44
SS AA AA TT SS SS SS 55 55 44
SS AA AA TT SS SS SS 55 55 44
SSSSSSSS AA AA TT SSSSSSSS SSSSSSSS SSSSSSSS 555555 44
SSSSSSSS AA AA TT SSSSSSSS SSSSSSSS SSSSSSSS 555555 44
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

```
0000     1          .TITLE  SATSSS54 SATS SYSTEM SERVICE TESTS  $CLREF (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 SATSSS54 TO TEST SUCCESSFUL
0000    36  ; OPERATION OF THE $CLREF SYSTEM SERVICE. THE SERVICE IS INVOKED
0000    37  ; UNDER VARIOUS INPUT CONDITIONS WITH VARYING INPUT PARAMETERS. ONLY
0000    38  ; SUCCESSFUL STATUS CODES ARE EXPECTED IN THIS TEST MODULE.  CORRECT
0000    39  ; OPERATION OF THE SERVICE FOR EACH OF ITS ISSUANCES IS VERIFIED BY
0000    40  ; CHECKING FOR AN SS$_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 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 ;
0000     61 ; MACROS:
0000     62 ;
0000     63 ;
0000     64 ; EQUATED SYMBOLS:
0000     65 ;
0000     66 ;
0000     67 ; OWN STORAGE:
0000     68 ;
```

```
00000000     70          .PSECT  RODATA,RD,NOWRT,NOEXE,LONG
     0000    71 TEST_MOD_NAME.: STRING  C,<SATSSS54>     ; TEST MODULE NAME
     0009    72 TEST_MOD_NAME_D: STRING I,<SATSSS54>     ; TEST MODULE NAME DESCRIPTOR
     0019    73 MSG1_INP_CTL:   STRING  I,<  SSCEF!4ZW: CONDITIONS:>
     0039    74                                          ; FAO CTL STRING FOR MSG1 IN SUCCOMMON.MAR
     0039    75 MSG3_ERR_CTL::  STRING  I,< *SSCEF!4ZW: !AS>
     0051    76                                          ; FAO CTL STRING FOR MSG3 IN SUCCOMMON.MAR
```

```
                00000000     78          .PSECT  RWDATA,RD,WRT,NOEXE,LONG
00000008  0000  0000     79 PRIVMASK:        .BLKQ   1               ; ADDR OF PRIVILEGE MASK (IN PHD)
0000000C  0008  0008     80 CLUSTER:         .BLKL   1               ; STATE ARGUMENT ON READEF SERVICE
```

```
                                                M 3
                        000C     82          .SBTTL  CONDITION TABLES
                        000C     83 ;
                        000C     84 ;       ***** CONDITION TABLES FOR CLREF SYSTEM SERVICE *****
                        000C     85 ;
                        000C     86          COND    1,NOTARG,<CLUSTER NUMBER>,-
                        000C     87                    <CLUSTER 0 (PROCESS-LOCAL)>,-
                        000C     88                    <CLUSTER 1 (PROCESS-LOCAL)>,-
                        000C     89                    <CLUSTER 2 (COMMON)>,-
                        000C     90                    <CLUSTER 3 (COMMON)>,-
                        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
                        008B     99          COND    3,NULL
                        008C    100
                        008C    101          COND    4,NULL
                        008D    102
                        008D    103          COND    5,NULL
                        008E    104
                  00000000      105          .PSECT  SATSSS54,RD,WRT,EXE
```

```
                                                     .SBTTL  TM_SETUP, TM_CLEANUP
                                                ;++
                                                ; FUNCTIONAL DESCRIPTION:
                                                ;
                                                ;               TM_SETUP AND TM_CLEANUP ARE CALLED TO PERFORM
                                                ; REQUIRED HOUSEKEEPING AT THE BEGINNING AND END, RESPECTIVELY, OF
                                                ; TEST MODULE EXECUTION.
                                                ;
                                                ; CALLING SEQUENCE:
                                                ;
                                                ;       BSBW TM_SETUP   BSBW TM_CLEANUP
                                                ;
                                                ; INPUT PARAMETERS:
                                                ;
                                                ;       NONE
                                                ;
                                                ; IMPLICIT INPUTS:
                                                ;
                                                ;       NONE
                                                ;
                                                ; OUTPUT PARAMETERS:
                                                ;
                                                ;       NONE
                                                ;
                                                ; IMPLICIT OUTPUTS:
                                                ;
                                                ;       TM_SETUP:  COND TABLE INDEX REGISTERS (R2,3,4,5,6) CLEARED;
                                                ;                  ALL PRIVILEGES ACQUIRED.
                                                ;
                                                ; COMPLETION CODES:
                                                ;
                                                ;       EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
                                                ;
                                                ; SIDE EFFECTS:
                                                ;
                                                ;       SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
                                                ;       (VIA RSB) IF ERROR ENCOUNTERED.
                                                ;
                                                ;--


                                                TM_SETUP::
                                52      D4      CLRL    R2                      ; INITIALIZE
                                53      D4      CLRL    R3                      ; .. CONDITION
                                54      D4      CLRL    R4                      ; .... TABLE
                                55      D4      CLRL    R5                      ; ...... INDEX
                                56      D4      CLRL    R6                      ; ........ REGISTERS
                            FFF3'       30      BSBW    MOD_MSG_PRINT           ; PRINT TEST MODULE BEGIN MSG
00000000'EF     00000000'EF     DE      MOVAL   TEST_MOD_SUCC,TMD_ADDR  ; ASSUME END MSG WILL SHOW SUCCESS
    03  00      00000000'8F     F0      INSV    #SUCCESS,#0,#3,MOD_MSG_CODE ; ADJUST STATUS CODE FOR SUCCESS
                00000000'EF             
                                        MODE    TO,5$,KRNL              ; KERNEL MODE TO ACCESS PHD
            59  00000000'9F     D0      MOVL    @#CTL$GL_PHD,R9         ; GET PROCESS HEADER ADDRESS
            00000000'EF     69  DE      MOVAL   PHD$Q_PRIVMSK(R9),PRIVMASK ; GET PRIV MASK ADDRESS
                                        MODE    FROM,5$ ; BACK TO USER MODE
                                        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 00B2   165          RSB                             ; RETURN TO MAIN ROUTINE
                     00B3   166 TM_CLEANUP::
                FF4A' 30 00B3   167          BSBW    MOD_MSG_PRINT       ; PRINT TEST MODULE END MSG
                  05 00B6   168          RSB                             ; RETURN TO MAIN ROUTINE
```

```
                        .SBTTL  CONDITION SUBROUTINES - SETUP AND CLEANUP
                ;++
                ; FUNCTIONAL DESCRIPTION:
                ;
                ;               CONDX AND CONDX_CLEANUP ARE SUBROUTINES WHICH ARE EXECUTED
                ; BEFORE AND AFTER THE VERIFY SUBROUTINE, RESPECTIVELY, WHENEVER A NEW
                ; CONDITION X VALUE IS SELECTED (SEE FUNCTIONAL DESCRIPTION OF SUCCOMMON
                ; ROUTINE IN SUCCOMMON.MAR). ANY SETUP FUNCTION PARTICULAR TO THE
                ; CONDITION X TABLE IS INCLUDED IN THE CONDX SUBROUTINE AND CLEANED
                ; UP, IF NECESSARY, IN THE CONDX_CLEANUP SUBROUTINE. THIS INCLUDES,
                ; ESPECIALLY, CODE TO DETECT CONFLICTS AMONG CURRENT ENTRIES IN TWO
                ; OR MORE CONDITION TABLES. IF A CONFLICT IS DETECTED, A NON-ZERO
                ; VALUE IS STORED INTO CONFLICT, WHICH CAUSES THE CALLING ROUTINE
                ; (SUCCOMMON) TO SKIP THE CURRENT ENTRY IN THE CONDITION X TABLE.
                ;
                ; CALLING SEQUENCE:
                ;
                ;       BSBW CONDX   BSBW CONDX_CLEANUP
                ;         WHERE X = 1,2,3,4,5
                ;
                ; INPUT PARAMETERS:
                ;
                ;       CONFLICT = 0
                ;
                ; IMPLICIT INPUTS:
                ;
                ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
                ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
                ;
                ; OUTPUT PARAMETERS:
                ;
                ;       CONFLICT SET TO NON-ZERO IF COND TABLE CONFLICT DETECTED.
                ;
                ; IMPLICIT OUTPUTS:
                ;
                ;       R2,3,4,5,6 PRESERVED
                ;
                ; COMPLETION CODES:
                ;
                ;       NONE
                ;
                ; SIDE EFFECTS:
                ;
                ;       NONE
                ;
                ;--


                COND1::
05              RSB                             ; RETURN TO MAIN ROUTINE
                COND1_CLEANUP::
05              RSB                             ; RETURN TO MAIN ROUTINE
                COND2::
05              RSB                             ; RETURN TO MAIN ROUTINE
                COND2_CLEANUP::
05              RSB                             ; RETURN TO MAIN ROUTINE
```

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

```
                        00C1    240          .SBTTL  FORM_CONDS
                        00C1    241 ;++
                        00C1    242 ; FUNCTIONAL DESCRIPTION:
                        00C1    243 ;
                        00C1    244 ;               FORM_CONDS FORMATS AND PRINTS INFORMATION ABOUT
                        00C1    245 ;  THE CURRENT ELEMENT IN EACH OF THE CONDITION TABLES.
                        00C1    246 ;
                        00C1    247 ; CALLING SEQUENCE:
                        00C1    248 ;
                        00C1    249 ;       BSBW FORM_CONDS
                        00C1    250 ;
                        00C1    251 ; INPUT PARAMETERS:
                        00C1    252 ;
                        00C1    253 ;       NONE
                        00C1    254 ;
                        00C1    255 ; IMPLICIT INPUTS:
                        00C1    256 ;
                        00C1    257 ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
                        00C1    258 ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
                        00C1    259 ;       FOR X = 1,2,3,4,5 :
                        00C1    260 ;               CONDX_T - TITLE TEXT FOR CONDX TABLE
                        00C1    261 ;               CONDX_TAB - ELEMENT TEXT FOR CONDX TABLE
                        00C1    262 ;               CONDX_C - CONTEXT OF THE CONDX TABLE
                        00C1    263 ;               CONDX_E - DATA ELEMENTS OF THE CONDX TABLE
                        00C1    264 ;
                        00C1    265 ; OUTPUT PARAMETERS:
                        00C1    266 ;
                        00C1    267 ;       NONE
                        00C1    268 ;
                        00C1    269 ; IMPLICIT OUTPUTS:
                        00C1    270 ;
                        00C1    271 ;       NONE
                        00C1    272 ;
                        00C1    273 ; COMPLETION CODES:
                        00C1    274 ;
                        00C1    275 ;       NONE
                        00C1    276 ;
                        00C1    277 ; SIDE EFFECTS:
                        00C1    278 ;
                        00C1    279 ;       NONE
                        00C1    280 ;
                        00C1    281 ;--
                        00C1    282
                        00C1    283
                        00C1    284
                        00C1    285 FORM_CONDS::
                        00C1    286         $FAO_S  MSG1_INP_CTL,FAO_LEN,FAO_DESC,TESTNUM
                        00E0    287                                         ; FORMAT CONDITIONS HEADER MSG
                FF1D'   30      00E0    288         BSBW    OUTPUT_MSG      ; ... AND PRINT IT
             14   00    91      00E3    289         CMPB    #COND1_C,#NULL  ; IS CONDITION 1 NULL ?
                  03    12      00E6    290         BNEQU   10$             ; NO -- CONTINUE
                00BF    31      00E8    291         BRW     FORM_CONDSX     ; YES -- SUBROUTINE IS FINISHED
                        00EB    292 10$:
  00000000'EF   0000000C'EF     DE      00EB    293         MOVAL   COND1_T,MSG_A           ; SAVE ADDRESS OF CONDITION 1 TITLE FOR FAO
00000000'EF   0000001C'EF42     D0      00F6    294         MOVL    COND1_TAB[R2],MSG_B     ; SAVE ADDR OF COND 1 CURR TEXT ELT FOR FAO
        00000000'EF   00        90      0102    295         MOVB    #COND1_C,MSG_CTXT       ; SAVE CONDITION 1 CONTEXT FOR FAO
                              0109      296         MOV_VAL COND1_C,COND1_E[R2],MSG_DATA1 ; GIVE COND 1 DATA VALUE TO FAO
```

```
                     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$:
  00000000'EF   0000008A'EF   DE  0114   302          MOVAL   COND2_T,MSG_A                   ; SAVE ADDRESS OF CONDITION 2 TITLE FOR FAO
00000000'EF   0000008A'EF43   D0  011F   303          MOVL    COND2_TAB[R3],MSG_B             ; SAVE ADDR OF COND 2 CURR TEXT ELT FOR FAO
        00000000'EF    14   90  012B   304          MOVB    #COND2_C,MSG_CTXT               ; SAVE CONDITION 2 CONTEXT FOR FAO
                            0132   305          MOV_VAL COND2_C,COND2_E[R3],MSG_DATA1 ; GIVE COND 2 DATA VALUE TO FAO
                     FECB'  30  0132   306          BSBW    WRITE_MSG2                      ; FORMAT AND WRITE CONDITION 2 MSG
                 14    14   91  0135   307          CMPB    #COND3_C,#NULL                  ; IS CONDITION 3 NULL ?
                       03   12  0138   308          BNEQU   30$                             ; NO -- CONTINUE
                     006D   31  013A   309          BRW     FORM_CONDSX                     ; YES -- SUBROUTINE IS FINISHED
                            013D   310 30$:
  00000000'EF   0000008B'EF   DE  013D   311          MOVAL   COND3_T,MSG_A                   ; SAVE ADDRESS OF CONDITION 3 TITLE FOR FAO
00000000'EF   0000008B'EF44   D0  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   D0  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   D0  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
                            01A7   328          MOV_VAL COND5_C,COND5_E[R6],MSG_DATA1 ; GIVE COND 5 DATA VALUE TO FAO
                     FE56'  30  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 ; ($CLREF). 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$:
     58   00000086'EF42   9A    01B6   399         MOVZBL  COND1_E[R2],R8          ; GET CLUSTER NO. INTO REGISTER
          58   58   05   78    01BE   400         ASHL    #5,R8,R8                ; MULT BY 32 TO GET 1ST EVENT FLAG NO.
                         59   D4  01C2   401         CLRL    R9                      ; ESTAB OFFSET INTO CLUSTER FOR 1ST FLAG
     02   00000086'EF42   91    01C4   402         CMPB    COND1_E[R2],#2          ; COMMON CLUSTER ?
                         41   19  01CC   403         BLSS    15$                     ; NO -- BYPASS THE ASSOCIATE SERVICE
                                01CE   404         $ASCEFC_S EFN=R8, NAME=TEST_MOD_NAME_D  ; YES -- ASSOCIATE ....
                                01E1   405         SS_CHECK NORMAL                 ; ... CLUSTER & CHECK STATUS CODE
                                020F   406 15$:
               5A   58   D0    020F   407         MOVL    R8,R10                  ; ESTAB CURRENT EFN IN REG 10
          5B   5A   1F   C1    0212   408         ADDL3   #31,R10,R11             ; ESTAB HIGH EFN FOR THIS CLUSTER
                                0216   409 20$:
                                0216   410         $SETEF_S EFN=R10                ; SET CURRENT EVENT FLAG
      00000000'8F   50   D1    021F   411         CMPL    R0,#SS$_WASSET          ; WASSET STATUS CODE ?
                         2E   13  0226   412         BEQLU   25$                     ; YES -- GO LOOP FOR ANOTHER SETEF
                                0228   413         SS_CHECK WASCLR                 ; NO -- BETTER BE WASCLR, THEN
                                0256   414 25$:
     FFBA 5A   01   5B   3D    0256   415         ACBW    R11,#1,R10,20$          ; INCREMENT TO NEXT EFN & LOOP
                                025C   416 30$:
               5A   59   58   81  025C   417         ADDB3   R8,R9,R10               ; COMPUTE EVENT FLAG NUMBER
                                0260   418 ;
                                0260   419 ; ****** SYSTEM SERVICE CALL WHICH IS THE SUBJECT OF THIS TEST CASE ******
                                0260   420 ;
                                0260   421         $CLREF_S EFN=R10                ; CLEAR EVENT FLAG
          00000000'8F   50   D1    0269   422         CMPL    R0,#SS$_WASSET          ; CODE RECEIVED = CODE EXPECTED ?
                         60   13  0270   423         BEQLU   40$                     ; YES -- CONTINUE
     00000000'EF   00000000'8F   D0  0272   424         MOVL    #SS$_WASSET,EXPV        ; LOAD UP EXPECTED AND ...
          00000000'EF   50   D0    027D   425         MOVL    R0,RECV                 ; ... RECEIVED VALUES, THEN EXIT
                                0284   426         ERR_EXIT LONG,<INCORRECT STATUS CODE RETURNED FROM CLREF>
                                02D2   427 40$:
                                02D2   428         $READEF_S EFN=R8, STATE=CLUSTER ; READ CURRENT CLUSTER
                                02E1   429         SS_CHECK WASCLR                 ; ... AND CHECK ITS STATUS
                         5B   D4  030F   430         CLRL    R11                     ; CLEAR REGISTERS TO ALLOW ...
                         57   D4  0311   431         CLRL    R7                      ; ... BYTE OPERATIONS ON THEM
               5B   59   01   81  0313   432         ADDB3   #1,R9,R11               ; COMPUTE NUMBER OF 0-BITS TO COMPARE
     00000000'EF   00000000'EF   D0  0317   433         MOVL    ONES,EXPV               ; ESTAB EXPECTED VALUE FOR ...
00000000'EF   5B   00   00   F0  0322   434         INSV    #0,#0,R11,EXPV          ; ... POSSIBLE ERR_EXIT
     00000000'EF   00000008'EF   D0  032B   435         MOVL    CLUSTER,RECV            ; ESTAB RECEIVED VALUE AS WELL
00   00000008'EF   5B   00   EC  0336   436         CMPV    #0,R11,CLUSTER,#0       ; ARE ALL EXPECTED EVENT FLAGS CLEAR ?
                         4D   13  033F   437         BEQL    50$                     ; YES -- GO LOOK AT SET FLAGS
                                0341   438         ERR_EXIT LONG,<EVENT FLAG(S) IN CLUSTER SHOULD BE CLEAR>
                                038E   439                                         ; NO -- GENERATE ERROR & EXIT
                                038E   440 50$:
                    59   1F   91  038E   441         CMPB    #31,R9                  ; IS CURRENT EFN HIGHEST IN CLUSTER ?
                         64   13  0391   442         BEQL    60$                     ; YES -- THEN CLUSTER IS ALL ZERO BITS
               57   1F   59   83  0393   443         SUBB3   R9,#31,R7               ; NO -- COMPUTE NO. OF 1-BITS TO COMPARE
     00000008'EF   57   5B   EC  0397   444         CMPV    R11,R7,CLUSTER,ONES     ; ARE ALL EV FLAGS NOT YET CLRED STILL SET ?
                00000000'EF         039F
                         51   13  03A4   445         BEQL    60$                     ; YES -- GO LOOK AT NEXT EVENT FLAG
```

```
                        03A6   446          ERR_EXIT LONG,<EVENT FLAG(S) IN CLUSTER SHOULD NOT BE CLEAR>
                        03F7   447                                                ; NO -- GENERATE ERROR & EXIT
                        03F7   448 60$:
FE5F 59   01   1F   9D  03F7   449          ACBB    #31,#1,R9,30$                 ; INCR TO NEXT EFN IN THIS CLUSTER & LOOP
                    05  03FD   450          RSB                                   ; RETURN TO CALLER
```

```
                        .SBTTL  VFY_CLEANUP
                ;++
                ; FUNCTIONAL DESCRIPTION:
                ;
                ;               VFY_CLEANUP EXECUTES SYSTEM SERVICES TO UNDO THE
                ; EFFECT OF THOSE ISSUED IN THE VERIFY SUBROUTINE. VFY_CLEANUP MUST
                ; ASSUME THAT VERIFY MAY NOT HAVE EXECUTED IN ITS ENTIRETY (IF AN
                ; ERROR IS FOUND). ALSO, VFY_CLEANUP MAY ISSUE SS_CHECK OR ERR_EXIT
                ; ONLY AFTER PERFORMING ALL OF ITS CLEANUP OPERATIONS; THIS IS REQUIRED
                ; IN THE EVENT THAT VFY_CLEANUP IS CALLED DURING ERROR PROCESSING,
                ; WHEN PERFORMING THE REQUIRED CLEANUP IS MORE IMPORTANT THAN
                ; POSSIBLY DISCOVERING A SECOND ERROR.
                ;
                ; CALLING SEQUENCE:
                ;
                ;       BSBW VFY_CLEANUP
                ;
                ; INPUT PARAMETERS:
                ;
                ;       NONE
                ;
                ; IMPLICIT INPUTS:
                ;
                ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
                ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
                ;       FOR X = 1,2,3,4,5 :
                ;               CONDX_E - ADDRESS OF TABLE OF DATA VALUES FOR CONDX
                ;                 TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
                ;                 ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
                ;                 FOR CONDX_E.
                ;
                ; OUTPUT PARAMETERS:
                ;
                ;       NONE
                ;
                ; IMPLICIT OUTPUTS:
                ;
                ;       NONE
                ;
                ; COMPLETION CODES:
                ;
                ;       EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
                ;
                ; SIDE EFFECTS:
                ;
                ;       SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
                ;       (VIA RSB) IF ERROR ENCOUNTERED.
                ;
                ;--


                VFY_CLEANUP::
05              RSB                                     ; RETURN TO CALLER
                .END
```

```
$$$$                    = 000003B0 R      04
$$$CHARS                = 0000002C
$$$CHARS1               = 00000019
$$$CHARS2               = 00000019
$$$CHARS3               = 00000012
$$$CHARS4               = 00000012
$$$CHARS5               = 00000000
$$$COND_A               = 00000003
$$$STRINGS              = 00000001
$$$STRINGS2             = 00000005
$$T1                    = 00000000
$$T2                    = 00000004
BYTE                    = 00000001   G
CFLAG                     ********     X    04
CHMRTN                    ********     X    04
CHM_CONT                  ********     X    04
CLUSTER                   00000008 R        03
COMP_SC                   ********     X    04
CONDT                     000000B7 RG       04
COND1_C                 = 00000000
COND1_CLEANUP             000000B8 RG       04
COND1_E                   00000086 R        03
COND1_H                   0000001B RG       03
COND1_T                   0000000C R        03
COND1_TAB                 0000001C R        03
COND2                     000000B9 RG       04
COND2_C                 = 00000014
COND2_CLEANUP             000000BA RG       04
COND2_H                   0000008A RG       03
COND2_T                   0000008A R        03
COND2_TAB                 0000008A R        03
COND3                     000000BB RG       04
COND3_C                 = 00000014
COND3_CLEANUP             000000BC RG       04
COND3_H                   0000008B RG       03
COND3_T                   0000008B R        03
COND3_TAB                 0000008B R        03
COND4                     000000BD RG       04
COND4_C                 = 00000014
COND4_CLEANUP             000000BE RG       04
COND4_H                   0000008C RG       03
COND4_T                   0000008C R        03
COND4_TAB                 0000008C R        03
COND5                     000000BF RG       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$GL_PHD                ********     X    04
DESC                    = 00000010   G
EFLAG                     ********     X    04
EXPV                      ********     X    04
FAO_DESC                  ********     X    04
FAO_LEN                   ********     X    04
FORM_CONDS                000000C1 RG       04
FORM_CONDSX               000001AA R        04
LONG                    = 00000004   G
MOD_MSG_CODE              ********     X    04
MOD_MSG_PRINT             ********     X    04
MSG1_INP_CTL              00000019 R        02
MSG3_ERR_CTL              00000039 RG       02
MSG_A                     ********     X    04
MSG_B                     ********     X    04
MSG_CTXT                  ********     X    04
NOTARG                  = 00000000   G
NULL                    = 00000014   G
ONES                      ********     X    04
OUTPUT_MSG                ********     X    04
PCV                       ********     X    04
PHD$Q_PRIVMSK           = 00000000
PRIVMASK                  00000000 R        03
PRIV_ARGS               = 00000002
PROCESS_ERR               ********     X    04
QUAD                    = 00000008   G
RECV                      ********     X    04
REST_REGS                 ********     X    04
SAVE_REGS                 ********     X    04
SS$_NORMAL                ********     X    04
SS$_WASCLR                ********     X    04
SS$_WASSET                ********     X    04
SUCCESS                   ********     X    04
SYS$ASCEFC                ********    GX    04
SYS$CLREF                 ********    GX    04
SYS$CMKRNL                ********    GX    04
SYS$FAO                   ********     X    04
SYS$READEF                ********    GX    04
SYS$SETEF                 ********    GX    04
SYS$SETPRN                ********    GX    04
SYS$SETPRV                ********    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                000000B3 RG       04
TM_SETUP                  00000000 RG       04
VERIFY                    000001AB RG       04
VFY_CLEANUP               000003FE 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
ABS	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
SATSSS54	000003FF (1023.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.08	00:00:00.34
Command processing	110	00:00:00.69	00:00:01.66
Pass 1	231	00:00:05.73	00:00:11.03
Symbol table sort	0	00:00:00.43	00:00:00.65
Pass 2	106	00:00:01.55	00:00:02.43
Symbol table output	14	00:00:00.08	00:00:00.09
Psect synopsis output	1	00:00:00.03	00:00:00.05
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	493	00:00:08.59	00:00:16.25

The working set limit was 1200 pages.
28528 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.
506 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
_255DUA28:[SHRLIB]UETP.MLB;1	8
_255DUA28:[SYS.OBJ]LIB.MLB;1	1
_255DUA28:[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$:SATSSS54/OBJ=OBJ$:SATSSS54 MSRC$:SATSSS54/UPDATE=(ENH$:SATSSS54)+EXECML$/LIB+SHRLIB$:UETP/LIB
```

SATSSS56
LIS

SATSSS60
LIS

SATSSS71
LIS

SATSSS54
LIS

SATSSS73
LIS

SATSSS70
LIS

SATSSS72
LIS

SATSSS55
LIS

SATSSS53
LIS

SATSSS61
LIS