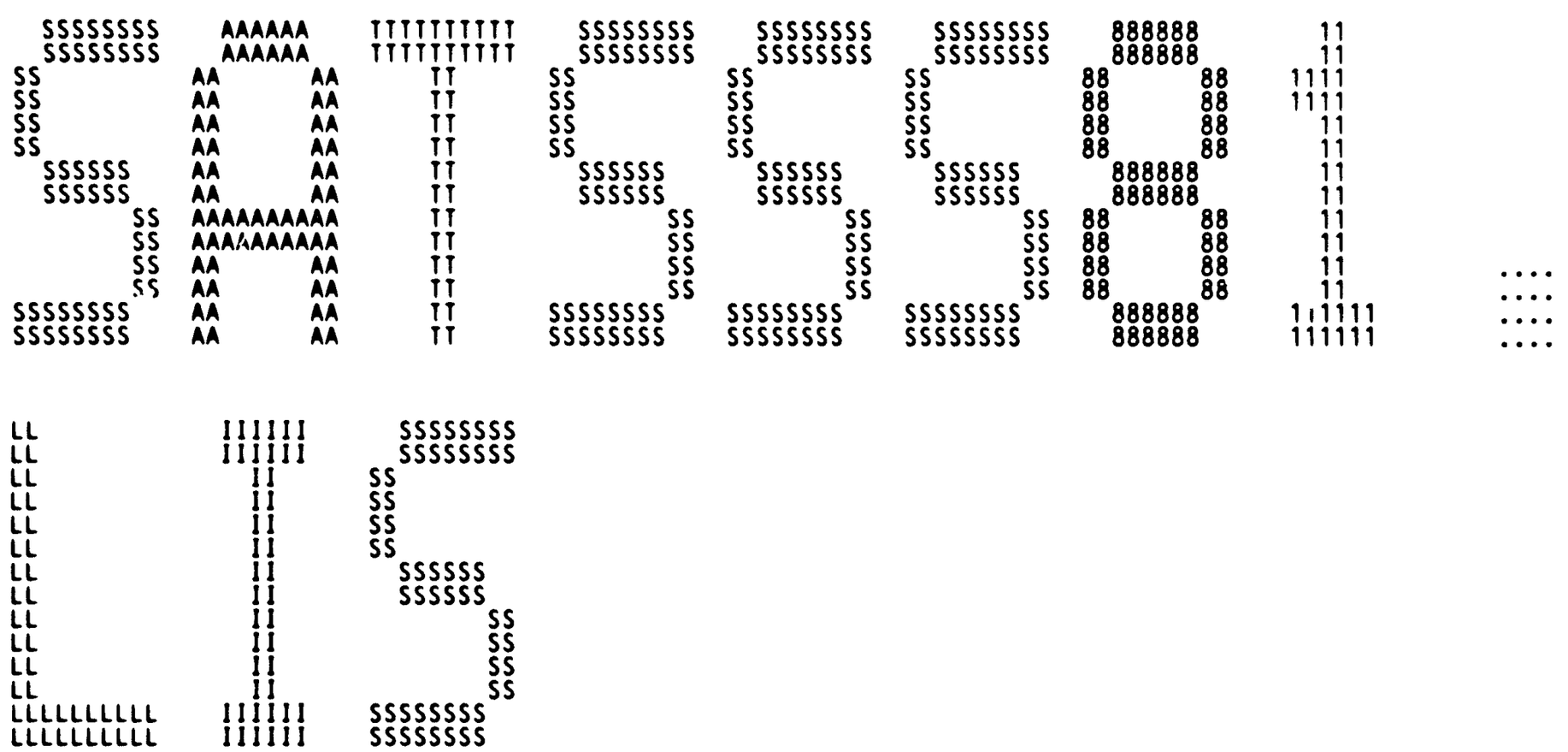
```
UUU          UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP        SSSSSSSSSSS  YYY          YYY
UUU          UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP        SSSSSSSSSSS  YYY          YYY
UUU          UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP        SSSSSSSSSSS  YYY          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
```

FILEID**SATSSS81



```
0000     1          .TITLE  SATSSS81 SATS SYSTEM SERVICE TESTS  $ADJWSL (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 SATSSS81 TO TEST SUCCESSFUL
0000    36 ; OPERATION OF THE $ADJWSL SYSTEM SERVICE. THE SERVICE IS INVOKED
0000    37 ; UNDER VARIOUS INPUT CONDITIONS WITH VARYING INPUT PARAMETERS. ONLY
0000    38 ; SUCCESSFUL STATUS CODES ARE EXPECTED IN THIS TEST MODULE.  CORRECT
0000    39 ; OPERATION OF THE SERVICE FOR EACH OF ITS ISSUANCES IS VERIFIED BY
0000    40 ; CHECKING FOR AN SS$_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: AUG, 1977
0000    47 ;
0000    48 ; MODIFIED BY:
0000    49 ;
0000    50 ;       V03-001 KDM0002         Kathleen D. Morse       28-Jun-1982
0000    51 ;               Added $SSDEF.
0000    52 ;
0000    53 ; 01    -
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          $SSDEF                          ; SYSTEM STATUS CODE DEFINITIONS
0000     63 ;
0000     64 ; MACROS:
0000     65 ;
0000     66 ;
0000     67 ; EQUATED SYMBOLS:
0000     68 ;
0000     69 ;
0000     70 ; OWN STORAGE:
0000     71 ;
```

```
                00000000     73          .PSECT  RODATA,RD,NOWRT,NOEXE,LONG
                    0000     74 TEST_MOD_NAME:: STRING  C,<SATSSS81>     ; TEST MODULE NAME
                    0009     75 TEST_MOD_NAME_D: STRING I,<SATSSS81>     ; TEST MODULE NAME DESCRIPTOR
                    0019     76 MSG1_INP_CTL:   STRING  I,<  SSAWS!4ZW: CONDITIONS:>
                    0039     77                                          ; FAO CTL STRING FOR MSG1 IN SUCCOMMON.MAR
                    0039     78 MSG3_ERR_CTL::  STRING  I,< *SSAWS!4ZW: !AS>
                    0051     79                                          ; FAO CTL STRING FOR MSG3 IN SUCCOMMON.MAR
01E84800            0051     80 MAXPAGCNT:      .LONG   32000000         ; A PAGCNT VALUE GREATER THAN MAX POSSIBLE
FE17B800            0055     81 MINPAGCNT:      .LONG   -32000000        ; A PAGCNT VALUE LESS THAN MINIMUM POSSIBLE
```

```
                                                D 8
                                00000000     83          .PSECT  RWDATA,RD,WRT,NOEXE,LONG
00000008  0000                  00000000     84 PRIVMASK:        .BLKQ   1               ; ADDR OF PRIVILEGE MASK (IN PHD)
0000000C  0008                  00000008     85 PAGCNT:          .BLKL   1               ; PAGCNT ARGUMENT FOR ADJWSL
00000010  000C                  0000000C     86 WSETLM:          .BLKL   1               ; WSETLM ARGUMENT FOR ADJWSL
00000014  0010                  00000010     87 WSETENTR:        .BLKL   1               ; WORKING SET VALUE AT ENTRY TO TEST MODULE
00000018  0014                  00000014     88 WSETENTR_P1:     .BLKL   1               ; W.S. ENTRY VALUE PLUS 1
0000001C  0018                  00000018     89 WSETULIM:        .BLKL   1               ; W.S. UPPER LIMIT DEFINED BY SYSTEM MGR
00000020  001C                  0000001C     90 WSETLLIM:        .BLKL   1               ; W.S. LOWER LIMIT DEFINED BY SYSTEM MGR
00000024  0020                  00000020     91 WSETLM_VFY:      .BLKL   1               ; W.S. VALUE USED FOR VERIFICATION
```

```
                        0024     93          .SBTTL  CONDITION TABLES
                        0024     94 ;
                        0024     95 ;       ***** CONDITION TABLES FOR ADJWSL SYSTEM SERVICE *****
                        0024     96 ;
                        0024     97          COND    1,NOTARG,<NEW WORKING SET VALUE>,-
                        0024     98                    <1 GREATER THAN VALUE AT ENTRY>,-
                        0024     99                    <UPPER LIMIT>,-
                        0024    100                    <1 GREATER THAN UPPER LIMIT>,-
                        0024    101                    <LOWER LIMIT>,-
                        0024    102                    <1 LESS THAN LOWER LIMIT>,-
                        0024    103
00000014'               00B8    104                      .ADDRESS    WSETENTR_P1
00000018'               00BC    105                      .ADDRESS    WSETULIM
00000018'               00C0    106                      .ADDRESS    WSETULIM
0000001C'               00C4    107                      .ADDRESS    WSETLLIM
0000001C'               00C8    108                      .ADDRESS    WSETLLIM
                        00CC    109 ;
                        00CC    110          COND    2,NULL
                        00CD    111
                        00CD    112          COND    3,NULL
                        00CE    113
                        00CE    114          COND    4,NULL
                        00CF    115
                        00CF    116          COND    5,NULL
                        00D0    117
                    00000000    118          .PSECT  SATSSS81,RD,WRT,EXE
```

```
                                        0000   120          .SBTTL  TM_SETUP, TM_CLEANUP
                                        0000   121 ;++
                                        0000   122 ; FUNCTIONAL DESCRIPTION:
                                        0000   123 ;
                                        0000   124 ;                   TM_SETUP AND TM_CLEANUP ARE CALLED TO PERFORM
                                        0000   125 ; REQUIRED HOUSEKEEPING AT THE BEGINNING AND END, RESPECTIVELY, OF
                                        0000   126 ; TEST MODULE EXECUTION.
                                        0000   127 ;
                                        0000   128 ; CALLING SEQUENCE:
                                        0000   129 ;
                                        0000   130 ;       BSBW TM_SETUP   BSBW TM_CLEANUP
                                        0000   131 ;
                                        0000   132 ; INPUT PARAMETERS:
                                        0000   133 ;
                                        0000   134 ;       NONE
                                        0000   135 ;
                                        0000   136 ; IMPLICIT INPUTS:
                                        0000   137 ;
                                        0000   138 ;       NONE
                                        0000   139 ;
                                        0000   140 ; OUTPUT PARAMETERS:
                                        0000   141 ;
                                        0000   142 ;       NONE
                                        0000   143 ;
                                        0000   144 ; IMPLICIT OUTPUTS:
                                        0000   145 ;
                                        0000   146 ;       TM_SETUP:  COND TABLE INDEX REGISTERS (R2,3,4,5,6) CLEARED;
                                        0000   147 ;                  ALL PRIVILEGES ACQUIRED.
                                        0000   148 ;
                                        0000   149 ; COMPLETION CODES:
                                        0000   150 ;
                                        0000   151 ;       EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
                                        0000   152 ;
                                        0000   153 ; SIDE EFFECTS:
                                        0000   154 ;
                                        0000   155 ;       SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
                                        0000   156 ;       (VIA RSB) IF ERROR ENCOUNTERED.
                                        0000   157 ;
                                        0000   158 ;--
                                        0000   159
                                        0000   160
                                        0000   161
                                        0000   162 TM_SETUP::
                              52    D4  0000   163          CLRL    R2                      ; INITIALIZE
                              53    D4  0002   164          CLRL    R3                      ; .. CONDITION
                              54    D4  0004   165          CLRL    R4                      ; .... TABLE
                              55    D4  0006   166          CLRL    R5                      ; ...... INDEX
                              56    D4  0008   167          CLRL    R6                      ; ........ REGISTERS
                         FFF3'      30  000A   168          BSBW    MOD_MSG_PRINT           ; PRINT TEST MODULE BEGIN MSG
00000000'EF    00000000'EF    DE  000D   169          MOVAL   TEST_MOD_SUCC,TMD_ADDR  ; ASSUME END MSG WILL SHOW SUCCESS
    03    00   00000000'8F    F0  0018   170          INSV    #SUCCESS,#0,#3,MOD_MSG_CODE ; ADJUST STATUS CODE FOR SUCCESS
               00000000'EF        0020
                                  0025   171          MODE    TO,S$,KRNL              ; KERNEL MODE TO ACCESS PHD
          59   00000000'9F    D0  0048   172          MOVL    @#CTL$GL_PHD,R9         ; GET PROCESS HEADER ADDRESS
          00000000'EF   69    DE  004F   173          MOVAL   PHD$Q_PRIVMSK(R9),PRIVMASK ; GET PRIV MASK ADDRESS
                                  0056   174          MODE    FROM,S$ ; BACK TO USER MODE
                                  0057   175          PRIV    ADD,ALL                 ; GET ALL PRIVILEGES
```

```
                     0077   176          $SETPRN_S TEST_MOD_NAME_D        ; SET PROCESS NAME
                     0084   177          SS_CHECK  NORMAL                 ; CHECK STATUS CODE RETURNED FROM SETPRN
                     00AE   178          $ADJWSL_S #0,WSETENTR            ; GET WORKING SET VALUE AT ENTRY
                     00BD   179          SS_CHECK NORMAL                  ; CHECK NORMAL RETURN
00000014'EF  00000010'EF  01  C1  00E7   180          ADDL3   #1,WSETENTR,WSETENTR_P1 ; REMEMBER WSETENTR PLUS 1
                     00F3   181          $ADJWSL_S MAXPAGCNT,WSETULIM     ; GET UPPER LIMIT FOR WORKING SET
                     0106   182          SS_CHECK NORMAL                  ; CHECK NORMAL RETURN
                     0130   183          $ADJWSL_S MINPAGCNT,WSETLLIM     ; GET LOWER LIMIT FOR WORKING SET
                     0143   184          SS_CHECK NORMAL                  ; CHECK NORMAL RETURN
     00000010'EF  0000001C'EF  C3  016D   185          SUBL3   WSETLLIM,WSETENTR,PAGCNT ; GET INCR FROM ENTRY TO LLIM
                  00000008'EF      0178
                     017D   186          $ADJWSL_S PAGCNT                 ; ... AND GET BACK TO ENTRY VALUE
                     018C   187          SS_CHECK NORMAL                  ; CHECK NORMAL RETURN
                  05 01B6   188          RSB                              ; RETURN TO MAIN ROUTINE
                     01B7   189 TM_CLEANUP::
                     01B7   190          $ADJWSL_S #0,WSETLM              ; GET CURRENT W.S. VALUE
     00000010'EF  0000000C'EF  C3  01C6   191          SUBL3   WSETLM,WSETENTR,PAGCNT  ; COMPUTE DISTANCE TO ENTRY VALUE
                  00000008'EF      01D1
                     01D6   192          $ADJWSL_S PAGCNT                 ; ... AND GO BACK THERE
                        FE18'  30  01E5   193          BSBW    MOD_MSG_PRINT    ; PRINT TEST MODULE END MSG
                  05 01E8   194          RSB                              ; RETURN TO MAIN ROUTINE
```

```
                        01E9    196          .SBTTL  CONDITION SUBROUTINES - SETUP AND CLEANUP
                        01E9    197 ;++
                        01E9    198 ; FUNCTIONAL DESCRIPTION:
                        01E9    199 ;
                        01E9    200 ;               CONDX AND CONDX_CLEANUP ARE SUBROUTINES WHICH ARE EXECUTED
                        01E9    201 ; BEFORE AND AFTER THE VERIFY SUBROUTINE, RESPECTIVELY, WHENEVER A NEW
                        01E9    202 ; CONDITION X VALUE IS SELECTED (SEE FUNCTIONAL DESCRIPTION OF SUCCOMMON
                        01E9    203 ; ROUTINE IN SUCCOMMON.MAR). ANY SETUP FUNCTION PARTICULAR TO THE
                        01E9    204 ; CONDITION X TABLE IS INCLUDED IN THE CONDX SUBROUTINE AND CLEANED
                        01E9    205 ; UP, IF NECESSARY, IN THE CONDX_CLEANUP SUBROUTINE. THIS INCLUDES,
                        01E9    206 ; ESPECIALLY, CODE TO DETECT CONFLICTS AMONG CURRENT ENTRIES IN TWO
                        01E9    207 ; OR MORE CONDITION TABLES. IF A CONFLICT IS DETECTED, A NON-ZERO
                        01E9    208 ; VALUE IS STORED INTO CONFLICT, WHICH CAUSES THE CALLING ROUTINE
                        01E9    209 ; (SUCCOMMON) TO SKIP THE CURRENT ENTRY IN THE CONDITION X TABLE.
                        01E9    210 ;
                        01E9    211 ; CALLING SEQUENCE:
                        01E9    212 ;
                        01E9    213 ;       BSBW CONDX   BSBW CONDX_CLEANUP
                        01E9    214 ;         WHERE X = 1,2,3,4,5
                        01E9    215 ;
                        01E9    216 ; INPUT PARAMETERS:
                        01E9    217 ;
                        01E9    218 ;       CONFLICT = 0
                        01E9    219 ;
                        01E9    220 ; IMPLICIT INPUTS:
                        01E9    221 ;
                        01E9    222 ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
                        01E9    223 ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
                        01E9    224 ;
                        01E9    225 ; OUTPUT PARAMETERS:
                        01E9    226 ;
                        01E9    227 ;       CONFLICT SET TO NON-ZERO IF COND TABLE CONFLICT DETECTED.
                        01E9    228 ;
                        01E9    229 ; IMPLICIT OUTPUTS:
                        01E9    230 ;
                        01E9    231 ;       R2,3,4,5,6 PRESERVED
                        01E9    232 ;
                        01E9    233 ; COMPLETION CODES:
                        01E9    234 ;
                        01E9    235 ;       NONE
                        01E9    236 ;
                        01E9    237 ; SIDE EFFECTS:
                        01E9    238 ;
                        01E9    239 ;       NONE
                        01E9    240 ;
                        01E9    241 ;--
                        01E9    242
                        01E9    243
                        01E9    244
                        01E9    245 COND1::
                05      01E9    246          RSB                                ; RETURN TO MAIN ROUTINE
                        01EA    247 COND1_CLEANUP::
                05      01EA    248          RSB                                ; RETURN TO MAIN ROUTINE
                        01EB    249 COND2::
                05      01EB    250          RSB                                ; RETURN TO MAIN ROUTINE
                        01EC    251 COND2_CLEANUP::
                05      01EC    252          RSB                                ; RETURN TO MAIN ROUTINE
```

```
                01ED    253 COND3::
 05             01ED    254             RSB                             ; RETURN TO MAIN ROUTINE
                01EE    255 COND3_CLEANUP::
 05             01EE    256             RSB                             ; RETURN TO MAIN ROUTINE
                01EF    257 COND4::
 05             01EF    258             RSB                             ; RETURN TO MAIN ROUTINE
                01F0    259 COND4_CLEANUP::
 05             01F0    260             RSB                             ; RETURN TO MAIN ROUTINE
                01F1    261 COND5::
 05             01F1    262             RSB                             ; RETURN TO MAIN ROUTINE
                01F2    263 COND5_CLEANUP::
 05             01F2    264             RSB                             ; RETURN TO MAIN ROUTINE
```

```
                                        .SBTTL  FORM_CONDS
                                ;++
                                ; FUNCTIONAL DESCRIPTION:
                                ;
                                ;               FORM_CONDS FORMATS AND PRINTS INFORMATION ABOUT
                                ;  THE CURRENT ELEMENT IN EACH OF THE CONDITION TABLES.
                                ;
                                ; CALLING SEQUENCE:
                                ;
                                ;       BSBW FORM_CONDS
                                ;
                                ; 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_T - TITLE TEXT FOR CONDX TABLE
                                ;               CONDX_TAB - ELEMENT TEXT FOR CONDX TABLE
                                ;               CONDX_C - CONTEXT OF THE CONDX TABLE
                                ;               CONDX_E - DATA ELEMENTS OF THE CONDX TABLE
                                ;
                                ; OUTPUT PARAMETERS:
                                ;
                                ;       NONE
                                ;
                                ; IMPLICIT OUTPUTS:
                                ;
                                ;       NONE
                                ;
                                ; COMPLETION CODES:
                                ;
                                ;       NONE
                                ;
                                ; SIDE EFFECTS:
                                ;
                                ;       NONE
                                ;
                                ;--
                                
                                
                                FORM_CONDS::
                                        $FAO_S  MSG1_INP_CTL,FAO_LEN,FAO_DESC,TESTNUM
                                                                        ; FORMAT CONDITIONS HEADER MSG
                          FDEB'  30             BSBW    OUTPUT_MSG      ; ... AND PRINT IT
                      14   00   91             CMPB    #COND1_C,#NULL  ; IS CONDITION 1 NULL ?
                           03   12             BNEQU   10$             ; NO -- CONTINUE
                         00BF   31             BRW     FORM_CONDSX     ; YES -- SUBROUTINE IS FINISHED
                                                10$:
  00000000'EF   00000024'EF   DE                MOVAL   COND1_T,MSG_A   ; SAVE ADDRESS OF CONDITION 1 TITLE FOR FAO
00000000'EF   0000003B'EF42   D0                MOVL    COND1_TAB[R2],MSG_B     ; SAVE ADDR OF COND 1 CURR TEXT ELT FOR FAO
            00000000'EF   00  90                MOVB    #COND1_C,MSG_CTXT       ; SAVE CONDITION 1 CONTEXT FOR FAO
                                                MOV_VAL COND1_C,COND1_E[R2],MSG_DATA1 ; GIVE COND 1 DATA VALUE TO FAO
```

```
                                                    K 8
                      FDC2'   30   023B    323          BSBW    WRITE_MSG2                  ; FORMAT AND WRITE CONDITION 1 MSG
                  14    14   91   023E    324          CMPB    #COND2_C,#NULL              ; IS CONDITION 2 NULL ?
                        03   12   0241    325          BNEQU   20$                         ; NO -- CONTINUE
                      0096   31   0243    326          BRW     FORM_CONDSX                 ; YES -- SUBROUTINE IS FINISHED
                             0246    327 20$:
  00000000'EF   000000CC'EF   DE   0246    328          MOVAL   COND2_T,MSG_A               ; SAVE ADDRESS OF CONDITION 2 TITLE FOR FAO
00000000'EF   000000CC'EF43   D0   0251    329          MOVL    COND2_TAB[R3],MSG_B         ; SAVE ADDR OF COND 2 CURR TEXT ELT FOR FAO
          00000000'EF   14    90   025D    330          MOVB    #COND2_C,MSG_CTXT           ; SAVE CONDITION 2 CONTEXT FOR FAO
                             0264    331          MOV_VAL COND2_C,COND2_E[R3],MSG_DATA1 ; GIVE COND 2 DATA VALUE TO FAO
                      FD99'   30   0264    332          BSBW    WRITE_MSG2                  ; FORMAT AND WRITE CONDITION 2 MSG
                  14    14   91   0267    333          CMPB    #COND3_C,#NULL              ; IS CONDITION 3 NULL ?
                        03   12   026A    334          BNEQU   30$                         ; NO -- CONTINUE
                      006D   31   026C    335          BRW     FORM_CONDSX                 ; YES -- SUBROUTINE IS FINISHED
                             026F    336 30$:
  00000000'EF   000000CD'EF   DE   026F    337          MOVAL   COND3_T,MSG_A               ; SAVE ADDRESS OF CONDITION 3 TITLE FOR FAO
00000000'EF   000000CD'EF44   D0   027A    338          MOVL    COND3_TAB[R4],MSG_B         ; SAVE ADDR OF COND 3 CURR TEXT ELT FOR FAO
          00000000'EF   14    90   0286    339          MOVB    #COND3_C,MSG_CTXT           ; SAVE CONDITION 3 CONTEXT FOR FAO
                             028D    340          MOV_VAL COND3_C,COND3_E[R4],MSG_DATA1 ; GIVE COND 3 DATA VALUE TO FAO
                      FD70'   30   028D    341          BSBW    WRITE_MSG2                  ; FORMAT AND WRITE CONDITION 3 MSG
                  14    14   91   0290    342          CMPB    #COND4_C,#NULL              ; IS CONDITION 4 NULL ?
                        47   13   0293    343          BEQLU   FORM_CONDSX                 ; YES -- SUBROUTINE IS FINISHED
  00000000'EF   000000CE'EF   DE   0295    344          MOVAL   COND4_T,MSG_A               ; SAVE ADDRESS OF CONDITION 4 TITLE FOR FAO
00000000'EF   000000CE'EF45   D0   02A0    345          MOVL    COND4_TAB[R5],MSG_B         ; SAVE ADDR OF COND 4 CURR TEXT ELT FOR FAO
          00000000'EF   14    90   02AC    346          MOVB    #COND4_C,MSG_CTXT           ; SAVE CONDITION 4 CONTEXT FOR FAO
                             02B3    347          MOV_VAL COND4_C,COND4_E[R5],MSG_DATA1 ; GIVE COND 4 DATA VALUE TO FAO
                      FD4A'   30   02B3    348          BSBW    WRITE_MSG2                  ; FORMAT AND WRITE CONDITION 4 MSG
                  14    14   91   02B6    349          CMPB    #COND5_C,#NULL              ; IS CONDITION 5 NULL ?
                        21   13   02B9    350          BEQLU   FORM_CONDSX                 ; YES -- SUBROUTINE IS FINISHED
  00000000'EF   000000CF'EF   DE   02BB    351          MOVAL   COND5_T,MSG_A               ; SAVE ADDRESS OF CONDITION 5 TITLE FOR FAO
00000000'EF   000000CF'EF46   D0   02C6    352          MOVL    COND5_TAB[R6],MSG_B         ; SAVE ADDR OF COND 5 CURR TEXT ELT FOR FAO
          00000000'EF   14    90   02D2    353          MOVB    #COND5_C,MSG_CTXT           ; SAVE CONDITION 5 CONTEXT FOR FAO
                             02D9    354          MOV_VAL COND5_C,COND5_E[R6],MSG_DATA1 ; GIVE COND 5 DATA VALUE TO FAO
                      FD24'   30   02D9    355          BSBW    WRITE_MSG2                  ; FORMAT AND WRITE CONDITION 5 MSG
                             02DC    356 FORM_CONDSX:
                        05   02DC    357          RSB                                 ; RETURN TO CALLER
```

```
02DD    359          .SBTTL  VERIFY
02DD    360 ;++
02DD    361 ; FUNCTIONAL DESCRIPTION:
02DD    362 ;
02DD    363 ;                   VERIFY IS CALLED ONCE FOR EACH COMBINATION OF CONDITION
02DD    364 ; TABLE VALUES (AS DETERMINED BY THE INDEX REGISTERS R2,3,4,5,6 FOR
02DD    365 ; COND TABLES 1,2,3,4,5, RESPECTIVELY). VERIFY ESTABLISHES THE CONDITIONS
02DD    366 ; SPECIFIED BY THE COND TABLES AND ISSUES THE SUBJECT SYSTEM SERVICE
02DD    367 ; ($ADJWSL). THEN, THE SUCCESSFUL OPERATION OF THE SERVICE IS VERIFIED
02DD    368 ; BY EXAMINING THE STATUS CODE RETURNED, THE VALUES FOR RETURN ARGUMENTS
02DD    369 ; AND THE FUNCTIONALITY PERFORMED. THE EXAMINATIONS TAKE THE FORM OF
02DD    370 ; COMPARISONS AGAINST EXPECTED VALUES. ANY FAILING COMPARISON CAUSES AN
02DD    371 ; ERR_EXIT MACRO TO BE EXECUTED (EITHER DIRECTLY, OR INDIRECTLY,
02DD    372 ; THROUGH THE SS_CHECK MACRO); ERR_EXIT SETS EFLAG TO NON-ZERO,
02DD    373 ; PRINTS ERROR MESSAGES AND CAUSES AN IMMEDIATE RSB TO CALLER.
02DD    374 ; WHEN ERR_EXIT IS EXECUTED, FURTHER CALLS TO VERIFY ARE SUPPRESSED,
02DD    375 ; AND, AFTER EXECUTING CLEANUP SUBROUTINES, THE IMAGE EXITS.
02DD    376 ;
02DD    377 ; CALLING SEQUENCE:
02DD    378 ;
02DD    379 ;       BSBW VERIFY
02DD    380 ;
02DD    381 ; INPUT PARAMETERS:
02DD    382 ;
02DD    383 ;       NONE
02DD    384 ;
02DD    385 ; IMPLICIT INPUTS:
02DD    386 ;
02DD    387 ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
02DD    388 ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
02DD    389 ;       FOR X = 1,2,3,4,5 :
02DD    390 ;               CONDX_E - ADDRESS OF TABLE OF DATA VALUES FOR CONDX
02DD    391 ;                 TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
02DD    392 ;                 ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
02DD    393 ;                 FOR CONDX_E.
02DD    394 ;
02DD    395 ; OUTPUT PARAMETERS:
02DD    396 ;
02DD    397 ;       NONE
02DD    398 ;
02DD    399 ; IMPLICIT OUTPUTS:
02DD    400 ;
02DD    401 ;       VERIFY HAS NO OUTPUT. SINCE ITS PURPOSE IS TO TEST FOR ERRORS,
02DD    402 ;       IT MERELY RETURNS TO CALLER NORMALLY AFTER THE TESTS, PROVIDING
02DD    403 ;       ALL WERE SUCCESSFUL; IF AN ERROR IS DISCOVERED, RETURN IS VIA
02DD    404 ;       AN ERR_EXIT OR SS_CHECK MACRO, BOTH OF WHICH DOCUMENT DETECTED
02DD    405 ;       ERRORS.
02DD    406 ;
02DD    407 ; COMPLETION CODES:
02DD    408 ;
02DD    409 ;       EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
02DD    410 ;
02DD    411 ; SIDE EFFECTS:
02DD    412 ;
02DD    413 ;       SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
02DD    414 ;       (VIA RSB) IF ERROR ENCOUNTERED.
02DD    415 ;
```

```
                                02DD   416 ;--
                                02DD   417
                                02DD   418
                                02DD   419
                                02DD   420 VERIFY::
                00000000'EF  95 02DD   421         TSTB    CFLAG                   ; SHOULD CONDITIONS BE PRINTED ?
                          03 13 02E3   422         BEQL    5$                      ; NO -- CONTINUE
                        FF0B 30 02E5   423         BSBW    FORM_CONDS              ; YES -- FMT & PRINT ALL CONDS FOR THIS T.C.
                                02E8   424 5$:
                          52 95 02E8   425         TSTB    R2                      ; FIRST CONDITION 1 ELEMENT ?
                          12 12 02EA   426         BNEQ    10$                     ; NO -- LOOK AT NEXT ELEMENT
                                02EC   427         $ADJWSL_S #1,WSETLM             ; YES -- ADJUST TO ENTRY VALUE + 1
                        00AF 31 02FB   428         BRW     50$                     ; ... AND GO DO SOME VERIFICATION
                                02FE   429 10$:
                     01   52 91 02FE   430         CMPB    R2,#1                   ; 2ND CONDITION 1 ELEMENT ?
                          26 12 0301   431         BNEQ    20$                     ; NO -- LOOK AT NEXT ELEMENT
    00000018'EF 00000010'EF C3 0303   432         SUBL3   WSETENTR,WSETULIM,PAGCNT ; GET OFFSET TO UPPER LIMIT
                00000008'EF        030E
                                0313   433         $ADJWSL_S PAGCNT,WSETLM         ; ... AND ADJUST TO IT
                        0084 31 0326   434         BRW     50$                     ; VERIFY THE SUBJECT SERVICE
                                0329   435 20$:
                     02   52 91 0329   436         CMPB    R2,#2                   ; 3RD CONDITION 1 ELEMENT ?
                          2C 12 032C   437         BNEQ    30$                     ; NO -- GO CHECK NEXT ELEMENT
    00000018'EF 00000010'EF C3 032E   438         SUBL3   WSETENTR,WSETULIM,PAGCNT ; FIND OUT HOW FAR AWAY ULIM IS
                00000008'EF        0339
                00000008'EF D6 033E   439         INCL    PAGCNT                  ; GO ONE FURTHER
                                0344   440         $ADJWSL_S PAGCNT,WSETLM         ; ADJUST TO ONE BEYOND UPPER LIMIT
                        0053 31 0357   441         BRW     50$                     ; ... AND VERIFY IT
                                035A   442 30$:
                     03   52 91 035A   443         CMPB    R2,#3                   ; 4TH CONDITION 1 ELEMENT ?
                          25 12 035D   444         BNEQ    40$                     ; NO -- MUST BE THE 5TH
    0000001C'EF 00000010'EF C3 035F   445         SUBL3   WSETENTR,WSETLLIM,PAGCNT ; COMPUTE DISTANCE TO LOWER LIMIT
                00000008'EF        036A
                                036F   446         $ADJWSL_S PAGCNT,WSETLM         ; ... AND ADJUST TO THERE
                          29 11 0382   447         BRB     50$                     ; GO VERIFY SERVICE
                                0384   448 40$:
    0000001C'EF 00000010'EF C3 0384   449         SUBL3   WSETENTR,WSETLLIM,PAGCNT ; FIND DISTANCE TO LOWER LIMIT
                00000008'EF        038F
                00000008'EF D7 0394   450         DECL    PAGCNT                  ; ... AND GO ONE BELOW IT
                                039A   451         $ADJWSL_S PAGCNT,WSETLM         ; ISSUE SUBJECT SERVICE
                                03AD   452 50$:                                 ; VERIFY SUBJECT SERVICE
                     01   50 D1 03AD   453         CMPL    R0,#SS$_NORMAL          ; CODE RETURNED = CODE EXPECTED  ?
                          5D 13 03B0   454         BEQL    60$                     ; YES -- MORE TO VERIFY
          00000000'EF    01 D0 03B2   455         MOVL    #SS$_NORMAL,EXPV        ; NO -- LOAD UP EXPECTED AND
          00000000'EF    50 D0 03B9   456         MOVL    R0,RECV                 ; ... RECEIVED VALUES, THEN EXIT
                                03C0   457         ERR_EXIT LONG,<INCORRECT STATUS CODE RETURNED FROM ADJWSL>
                                040F   458 60$:
        51   000000B8'EF42 D0 040F   459         MOVL    COND1_E[R2],R1          ; GET EXPECTED VALUE SAFELY INTO REGISTER
          61   0000000C'EF D1 0417   460         CMPL    WSETLM,(R1)             ; IS COMPUTED W.S. VALUE = THAT EXPECTED ?
                          03 12 041E   461         BNEQ    65$                     ; NO -- COULD BE AN ERROR
                        007E 31 0420   462         BRW     80$                     ; YES -- ON TO MORE VERIFICATION
                                0423   463 65$:
                          52 95 0423   464         TSTB    R2                      ; FIRST CONDITION 1 ELEMENT ?
                          1A 12 0425   465         BNEQ    70$                     ; NO -- THEN IT REALLY IS AN ERROR
    00000010'EF 0000000C'EF D1 0427   466         CMPL    WSETLM,WSETENTR         ; IS W.S. AT ENTRY VALUE ?
                          0D 12 0432   467         BNEQ    70$                     ; NO -- THAT'S AN ERROR
    00000018'EF 00000010'EF D1 0434   468         CMPL    WSETENTR,WSETULIM       ; IS W.S. AT UPPER LIMIT ?
```

```
                          60   13  043F   469          BEQL    80$                     ; YES -- THEN IT'S OK
                               0441   470 70$:
          00000000'EF     61   D0  0441   471          MOVL    (R1),EXPV               ; LOAD EXPECTED AND
00000000'EF   0000000C'EF     D0  0448   472          MOVL    WSETLM,RECV             ; ... RECEIVED VALUES, THEN EXIT
                               0453   473          ERR_EXIT LONG,<INCORRECT WSETLM VALUE RETURNED BY ADJWSL>
                               04A1   474 80$:
                               04A1   475          $ADJWSL_S #0,WSETLM_VFY         ; GET CURRENT WSETLM TO VERIFY IT
                               04B0   476          SS_CHECK NORMAL                 ; CHECK NORMAL RETURN
00000020'EF   0000000C'EF     D1  04DA   477          CMPL    WSETLM,WSETLM_VFY       ; DID ADJWSL REMEMBER W.S. VALUE ?
                          59   13  04E5   478          BEQL    VERIFYX                 ; YES -- ALL TESTS PASSED
0000C000'EF   0000000C'EF     D0  04E7   479          MOVL    WSETLM,EXPV             ; NO -- LOAD EXPECTED AND
00000000'EF   00000020'EF     D0  04F2   480          MOVL    WSETLM_VFY,RECV         ; ... RECEIVED VALUES, THEN EXIT
                               04FD   481          ERR_EXIT LONG,<WSETLM NOT PRESERVED BY ADJWSL>
                               0540   482 VERIFYX:
                               05  0540   483          RSB                             ; RETURN TO CALLER
```

```
                     0541   485          .SBTTL  VFY_CLEANUP
                     0541   486 ;++
                     0541   487 ; FUNCTIONAL DESCRIPTION:
                     0541   488 ;
                     0541   489 ;               VFY_CLEANUP EXECUTES SYSTEM SERVICES TO UNDO THE
                     0541   490 ; EFFECT OF THOSE ISSUED IN THE VERIFY SUBROUTINE. VFY_CLEANUP MUST
                     0541   491 ; ASSUME THAT VERIFY MAY NOT HAVE EXECUTED IN ITS ENTIRETY (IF AN
                     0541   492 ; ERROR IS FOUND). ALSO, VFY_CLEANUP MAY ISSUE SS_CHECK OR ERR_EXIT
                     0541   493 ; ONLY AFTER PERFORMING ALL OF ITS CLEANUP OPERATIONS; THIS IS REQUIRED
                     0541   494 ; IN THE EVENT THAT VFY_CLEANUP IS CALLED DURING ERROR PROCESSING,
                     0541   495 ; WHEN PERFORMING THE REQUIRED CLEANUP IS MORE IMPORTANT THAN
                     0541   496 ; POSSIBLY DISCOVERING A SECOND ERROR.
                     0541   497 ;
                     0541   498 ; CALLING SEQUENCE:
                     0541   499 ;
                     0541   500 ;       BSBW VFY_CLEANUP
                     0541   501 ;
                     0541   502 ; INPUT PARAMETERS:
                     0541   503 ;
                     0541   504 ;       NONE
                     0541   505 ;
                     0541   506 ; IMPLICIT INPUTS:
                     0541   507 ;
                     0541   508 ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
                     0541   509 ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
                     0541   510 ;       FOR X = 1,2,3,4,5 :
                     0541   511 ;               CONDX_E - ADDRESS OF TABLE OF DATA VALUES FOR CONDX
                     0541   512 ;                 TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
                     0541   513 ;                 ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
                     0541   514 ;                 FOR CONDX_E.
                     0541   515 ;
                     0541   516 ; OUTPUT PARAMETERS:
                     0541   517 ;
                     0541   518 ;       NONE
                     0541   519 ;
                     0541   520 ; IMPLICIT OUTPUTS:
                     0541   521 ;
                     0541   522 ;       NONE
                     0541   523 ;
                     0541   524 ; COMPLETION CODES:
                     0541   525 ;
                     0541   526 ;       EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
                     0541   527 ;
                     0541   528 ; SIDE EFFECTS:
                     0541   529 ;
                     0541   530 ;       SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
                     0541   531 ;       (VIA RSB) IF ERROR ENCOUNTERED.
                     0541   532 ;
                     0541   533 ;--
                     0541   534
                     0541   535
                     0541   536
                     0541   537 VFY_CLEANUP::
                     0541   538          $ADJWSL_S #0,WSETLM                ; GET CURRENT W.S. VALUE
                     0550   539          SS_CHECK NORMAL                    ; CHECK NORMAL RETURN
00000010'EF  0000000C'EF  C3 057A   540          SUBL3   WSETLM,WSETENTR,PAGCNT  ; COMPUTE DISTANCE TO ENTRY VALUE
             00000008'EF     0585
```

```
                058A    541         $ADJWSL_S PAGCNT                ; ... AND GO BACK THERE
                0599    542         SS_CHECK NORMAL                 ; CHECK NORMAL RETURN
          05    05C3    543         RSB                             ; RETURN TO CALLER
                05C4    544         .END
```

```
$$$$                    = 00000507 R      04
$$$CHARS                = 0000001E
$$$CHARS1               = 0000001D
$$$CHARS2               = 0000000B
$$$CHARS3               = 0000001A
$$$CHARS4               = 0000000B
$$$CHARS5               = 00000017
$$$COND_A               = 00000004
$$$STRINGS              = 00000001
$$$STRINGS2             = 00000005
$$T2                    = 00000004
BYTE                    = 00000001  G
CFLAG                     ********    X   04
CHMRTN                    ********    X   04
CHM_CONT                  ********    X   04
COMP_SC                   ********    X   04
CONDT                     000001E9 RG     04
COND1_C                 = 00000000
COND1_CLEANUP             000001EA RG     04
COND1_E                   000000B8 R      03
COND1_H                   0000003A RG     03
COND1_T                   00000024 R      03
COND1_TAB                 0000003B R      03
COND2                     000001EB RG     04
COND2_C                 = 00000014
COND2_CLEANUP             000001EC RG     04
COND2_H                   000000CC RG     03
COND2_T                   000000CC R      03
COND2_TAB                 000000CC R      03
COND3                     000001ED RG     04
COND3_C                 = 00000014
COND3_CLEANUP             000001EE RG     04
COND3_H                   000000CD RG     03
COND3_T                   000000CD R      03
COND3_TAB                 000000CD R      03
COND4                     000001EF RG     04
COND4_C                 = 00000014
COND4_CLEANUP             000001F0 RG     04
COND4_H                   000000CE RG     03
COND4_T                   000000CE R      03
COND4_TAB                 000000CE R      03
COND5                     000001F1 RG     04
COND5_C                 = 00000014
COND5_CLEANUP             000001F2 RG     04
COND5_H                   000000CF RG     03
COND5_T                   000000CF R      03
COND5_TAB                 000000CF R      03
CTL$GL_PHD                ********      X 04
DESC                    = 00000010  G
EFLAG                     ********    X   04
EXPV                      ********    X   04
FAO_DESC                  ********    X   04
FAO_LEN                   ********    X   04
FORM_CONDS                000001F3 RG     04
FORM_CONDSX               000002DC R      04
LONG                    = 00000004  G
MAXPAGCNT                 00000051 R      02
MINPAGCNT                 00000055 R      02
MOD_MSG_CODE              ********    X   04
MOD_MSG_PRINT             ********    X   04
MSG1_INP_CTL              00000019 R      02
MSG3_ERR_CTL              00000039 RG     02
MSG_A                     ********    X   04
MSG_B                     ********    X   04
MSG_CTXT                  ********    X   04
NOTARG                  = 00000000  G
NULL                    = 00000014  G
OUTPUT_MSG                ********    X   04
PAGCNT                    00000008 R      03
PCV                       ********    X   04
PHD$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$_NORMAL              = 00000001
SUCCESS                   ********    X   04
SYS$ADJWSL                ********   GX   04
SYS$CMKRNL                ********   GX   04
SYS$FAO                   ********    X   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                000001B7 RG     04
TM_SETUP                  00000000 RG     04
VERIFY                    000002DD RG     04
VERIFYX                   00000540 R      04
VFY_CLEANUP               00000541 RG     04
WORD                    = 00000002  G
WRITE_MSG2                ********    X   04
WSETENTR                  00000010 R      03
WSETENTR_P1               00000014 R      03
WSETLLIM                  0000001C R      03
WSETLM                    0000000C R      03
WSETLM_VFY                00000020 R      03
WSETULIM                  00000018 R      03
```

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	00000059 (89.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC LONG
RWDATA	000000D0 (208.)	03 (3.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
SATSSS81	000005C4 (1476.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.10	00:00:01.54
Command processing	136	00:00:00.71	00:00:05.37
Pass 1	283	00:00:09.17	00:00:20.16
Symbol table sort	0	00:00:01.16	00:00:02.13
Pass 2	115	00:00:02.10	00:00:03.84
Symbol table output	12	00:00:00.08	00:00:00.33
Psect synopsis output	3	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	586	00:00:13.36	00:00:33.41

The working set limit was 1500 pages.
49896 bytes (98 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 703 non-local and 39 local symbols.
544 source lines were read in Pass 1, producing 23 object records in Pass 2.
32 pages of virtual memory were used to define 23 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	11
TOTALS (all libraries)	20

1003 GETS were required to define 20 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:SATSSS81/OBJ=OBJ$:SATSSS81 MSRC$:SATSSS81/UPDATE=(ENH$:SATSSS81)+EXECML$/LIB+SHRLIB$:UETP/LIB
```

SATSS574
LIS

SATSS578
LIS

SATSS580
LIS

SATSS582
LIS

SATSS590
LIS

SATSUT01
LIS

SATSUT05
LIS

SATSUT04
LIS

SATSUT06
LIS

SATSS581
LIS

SATSS591
LIS

SATSS579
LIS

SATSS583
LIS