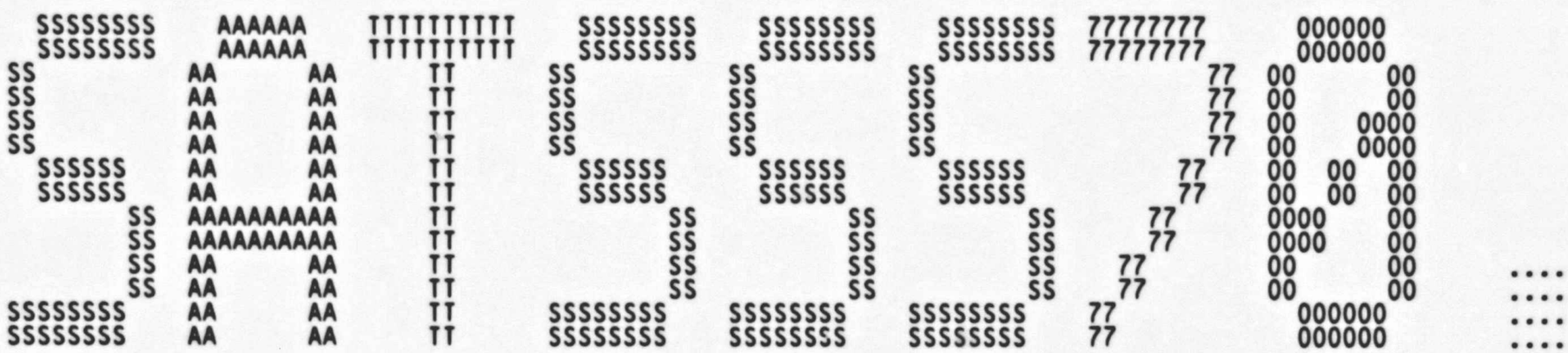
```
UUU         UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP       SSSSSSSSSSS  YYY         YYY
UUU         UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP       SSSSSSSSSSS  YYY         YYY
UUU         UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP       SSSSSSSSSSS  YYY         YYY
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  EEEEEEEEEEE            TTT        PPPPPPPPPPPP      SSSSSSSSS             YYY
UUU         UUU  EEEEEEEEEEE            TTT        PPPPPPPPPPPP      SSSSSSSSS             YYY
UUU         UUU  EEEEEEEEEEE            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
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
UUUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE      TTT        PPP                 SSSSSSSSSSS         YYY
```

FILEID**SATSSS70



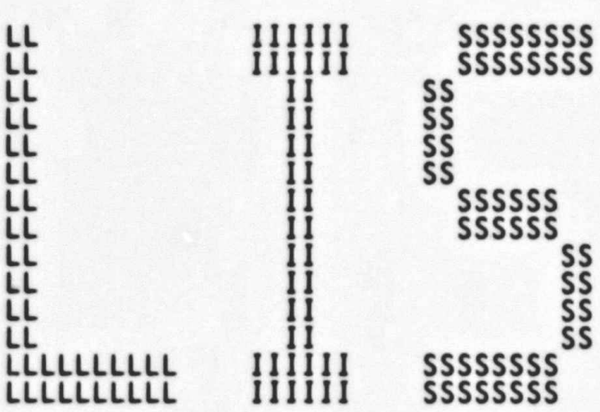


Table of contents

```
0000     1          .TITLE  SATSS70 SATS SYSTEM SERVICE TESTS  $EXPREG (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 SATSSS70 TO TEST SUCCESSFUL
0000    36  ; OPERATION OF THE $EXPREG SYSTEM SERVICE. THE SERVICE IS INVOKED
0000    37  ; UNDER VARIOUS INPUT CONDITIONS WITH VARYING INPUT PARAMETERS. ONLY
0000    38  ; SUCCESSFUL STATUS CODES ARE EXPECTED IN THIS TEST MODULE.  CORRECT
0000    39  ; OPERATION OF THE SERVICE FOR EACH OF ITS ISSUANCES IS VERIFIED BY
0000    40  ; CHECKING FOR AN SS$_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: JUN, 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          $JPIDEF                         ; $GETJPI ITEM-CODE SYMBOLS
0000     61          $PSLDEF                         ; PROCESSOR STATUS LONGWORD DEFINITIONS
0000     62 ;
0000     63 ; MACROS:
0000     64 ;
0000     65 ;
0000     66 ; EQUATED SYMBOLS:
0000     67 ;
0000     68 ;
0000     69 ; OWN STORAGE:
0000     70 ;
```

```
                        72          .PSECT  RODATA,RD,NOWRT,NOEXE,LONG
                        73 TEST_MOD_NAME:: STRING  C,<SATSSS70>     ; TEST MODULE NAME
                        74 TEST_MOD_NAME_D: STRING I,<SATSSS70>     ; TEST MODULE NAME DESCRIPTOR
                        75 MSG1_INP_CTL:   STRING  I,<  SSERG!4ZW: CONDITIONS:>
                        76                                          ; FAO CTL STRING FOR MSG1 IN SUCCOMMON.MAR
                        77 MSG3_ERR_CTL::  STRING  I,< *SSERG!4ZW: !AS>
                        78                                          ; FAO CTL STRING FOR MSG3 IN SUCCOMMON.MAR
00000200                79 PAGESIZE:       .LONG   512              ; PAGE SIZE IN BYTES
                        80 JPITEMS:                                 ; $GETJPI ITEM LIST
    0004                81                 .WORD   4                ; LEN OF RETURN BUFFER FOR 1ST ITEM
    0404                82                 .WORD   JPI$_FREPOVA     ; REQUEST ITEM 1
00000009'               83                 .ADDRESS INITIALADR      ; BUFFER FOR ITEM 1
00000000                84                 .LONG   0                ; DON'T NEED LENGTH RETURN
    0004                85                 .WORD   4                ; LEN OF RETURN BUFFER FOR 2ND ITEM
    0405                86                 .WORD   JPI$_FREP1VA     ; REQUEST ITEM 2
0000000D'               87                 .ADDRESS INITIALADR+4    ; BUFFER FOR ITEM 2
00000000                88                 .LONG   0                ; DON'T NEED LENGTH RETURN
00000000                89                 .LONG   0                ; END OF $GETJPI ITEM LIST
```

```
                00000000     91          .PSECT  RWDATA,RD,WRT,NOEXE,LONG
00000008  0000  00000008     92 PRIVMASK:        .BLKQ   1               ; ADDR OF PRIVILEGE MASK (IN PHD)
00000009  0008  00000009     93 NZERR:           .BLKB   1               ; INDICATOR FOR NON-ZERO ERROR
00000011  0009  00000011     94 INITIALADR:      .BLKQ   1               ; HOLD AREA FOR FREE REGION BOUNDARIES
00000019  0011  00000019     95 RETADR:          .BLKQ   1               ; LONGWORD PAIR FOR SUBJECT EXPREG
00000021  0019  00000021     96 INADR_DVA:       .BLKQ   1               ; LONGWORD PAIR FOR DELTVA
```

```
                                                D 12
                     0021    98          .SBTTL  CONDITION TABLES
                     0021    99 ;
                     0021   100 ;        ***** CONDITION TABLES FOR EXPREG SYSTEM SERVICE *****
                     0021   101 ;
                     0021   102          COND    1,LONG,<REGION>,-
                     0021   103                    <PROGRAM>,-
                     0021   104                    <CONTROL>,-
                     0021   105
 00000000            0041   106                      .LONG       0       ; PROGRAM
 00000001            0045   107                      .LONG       1       ; CONTROL
                     0049   108 ;
                     0049   109          COND    2,LONG,<PAGCNT>,-
                     0049   110                    <SMALL COUNT>,-
                     0049   111                    <MEDIUM COUNT>,-
                     0049   112                    <LARGE COUNT>,-
                     0049   113
 00000001            0082   114                      .LONG       1
 00000005            0086   115                      .LONG       5
 000003E8            008A   116                      .LONG       1000
                     008E   117 ;
                     008E   118          COND    3,LONG,<ACMODE>,-
                     008E   119                    <KERNEL>,-
                     008E   120                    <EXEC>,-
                     008E   121                    <SUPER>,-
                     008E   122                    <USER>,-
                     008E   123
 00000000            00BD   124                      .LONG       PSL$C_KERNEL
 00000001            00C1   125                      .LONG       PSL$C_EXEC
 00000002            00C5   126                      .LONG       PSL$C_SUPER
 00000003            00C9   127                      .LONG       PSL$C_USER
                     00CD   128 ;
                     00CD   129          COND    4,NULL
                     00CE   130
                     00CE   131          COND    5,NULL
                     00CF   132
 00000000            0000   133          .PSECT  SATSSS70,RD,WRT,EXE
```

```
                                  0000        .SBTTL  TM_SETUP, TM_CLEANUP
                                  0000  ;++
                                  0000  ; FUNCTIONAL DESCRIPTION:
                                  0000  ;
                                  0000  ;               TM_SETUP AND TM_CLEANUP ARE CALLED TO PERFORM
                                  0000  ; REQUIRED HOUSEKEEPING AT THE BEGINNING AND END, RESPECTIVELY, OF
                                  0000  ; TEST MODULE EXECUTION.
                                  0000  ;
                                  0000  ; CALLING SEQUENCE:
                                  0000  ;
                                  0000  ;       BSBW TM_SETUP   BSBW TM_CLEANUP
                                  0000  ;
                                  0000  ; INPUT PARAMETERS:
                                  0000  ;
                                  0000  ;       NONE
                                  0000  ;
                                  0000  ; IMPLICIT INPUTS:
                                  0000  ;
                                  0000  ;       NONE
                                  0000  ;
                                  0000  ; OUTPUT PARAMETERS:
                                  0000  ;
                                  0000  ;       NONE
                                  0000  ;
                                  0000  ; IMPLICIT OUTPUTS:
                                  0000  ;
                                  0000  ;       TM_SETUP:  COND TABLE INDEX REGISTERS (R2,3,4,5,6) CLEARED;
                                  0000  ;                  ALL PRIVILEGES ACQUIRED.
                                  0000  ;
                                  0000  ; COMPLETION CODES:
                                  0000  ;
                                  0000  ;       EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
                                  0000  ;
                                  0000  ; SIDE EFFECTS:
                                  0000  ;
                                  0000  ;       SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
                                  0000  ;       (VIA RSB) IF ERROR ENCOUNTERED.
                                  0000  ;
                                  0000  ;--
                                  0000
                                  0000
                                  0000
                                  0000  TM_SETUP::
                         52  D4   0000        CLRL    R2                      ; INITIALIZE
                         53  D4   0002        CLRL    R3                      ; .. CONDITION
                         54  D4   0004        CLRL    R4                      ; .... TABLE
                         55  D4   0006        CLRL    R5                      ; ...... INDEX
                         56  D4   0008        CLRL    R6                      ; ........ REGISTERS
                      FFF3'  30   000A        BSBW    MOD_MSG_PRINT           ; PRINT TEST MODULE BEGIN MSG
00000000'EF   00000000'EF    DE   000D        MOVAL   TEST_MOD_SUCC,TMD_ADDR  ; ASSUME END MSG WILL SHOW SUCCESS
     03    00 00000000'8F    F0   0018        INSV    #SUCCESS,#0,#3,MOD_MSG_CODE ; ADJUST STATUS CODE FOR SUCCESS
              00000000'EF         0020
                                  0025        MODE    TO,5$,KRNL              ; KERNEL MODE TO ACCESS PHD
         59   00000000'9F    D0   0048        MOVL    @#CTL$GL_PHD,R9         ; GET PROCESS HEADER ADDRESS
      00000000'EF        69  DE   004F        MOVAL   PHD$Q_PRIVMSK(R9),PRIVMASK ; GET PRIV MASK ADDRESS
                                  0056        MODE    FROM,5$ ; BACK TO USER MODE
                                  0057        PRIV    ADD,ALL                 ; GET ALL PRIVILEGES
```

```
                        0077   191          $SETPRN_S TEST_MOD_NAME_D       ; SET PROCESS NAME
                        0084   192          SS_CHECK  NORMAL                ; CHECK STATUS CODE RETURNED FROM SETPRN
                        00B2   193 ;
                        00B2   194 ; THE FOLLOWING $GETJPI ESTABLISHES INITIALADR AND INITIALADR+4
                        00B2   195 ; AS THE ADDRESSES OF THE FIRST FREE PAGES IN THE PROGRAM AND
                        00B2   196 ; CONTROL REGIONS.
                        00B2   197 ;
                        00B2   198          $GETJPI_S ITMLST=JPITEMS        ; GET PROG AND CTL REGION BOUNDARIES
                        00C9   199          SS_CHECK NORMAL                 ; CHECK FOR NORMAL RETURN
0000000D'EF 00000051'EF C0      00F7   200          ADDL2   PAGESIZE,INITIALADR+4   ; ADJUST CONTROL REGION BOUNDARY ...
            0000000D'EF D7      0102   201          DECL    INITIALADR+4            ; ... TO FIRST FREE BYTE
                        05      0108   202          RSB                             ; RETURN TO MAIN ROUTINE
                        0109   203 TM_CLEANUP::
                    FEF4' 30    0109   204          BSBW    MOD_MSG_PRINT           ; PRINT TEST MODULE END MSG
                        05      010C   205          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
```

```
     0111   264 COND3::
05   0111   265          RSB                         ; RETURN TO MAIN ROUTINE
     0112   266 COND3_CLEANUP::
05   0112   267          RSB                         ; RETURN TO MAIN ROUTINE
     0113   268 COND4::
05   0113   269          RSB                         ; RETURN TO MAIN ROUTINE
     0114   270 COND4_CLEANUP::
05   0114   271          RSB                         ; RETURN TO MAIN ROUTINE
     0115   272 COND5::
05   0115   273          RSB                         ; RETURN TO MAIN ROUTINE
     0116   274 COND5_CLEANUP::
05   0116   275          RSB                         ; RETURN TO MAIN ROUTINE
```

```
                     0117   277          .SBTTL  FORM_CONDS
                     0117   278 ;++
                     0117   279 ; FUNCTIONAL DESCRIPTION:
                     0117   280 ;
                     0117   281 ;               FORM_CONDS FORMATS AND PRINTS INFORMATION ABOUT
                     0117   282 ;  THE CURRENT ELEMENT IN EACH OF THE CONDITION TABLES.
                     0117   283 ;
                     0117   284 ; CALLING SEQUENCE:
                     0117   285 ;
                     0117   286 ;       BSBW FORM_CONDS
                     0117   287 ;
                     0117   288 ; INPUT PARAMETERS:
                     0117   289 ;
                     0117   290 ;       NONE
                     0117   291 ;
                     0117   292 ; IMPLICIT INPUTS:
                     0117   293 ;
                     0117   294 ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
                     0117   295 ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
                     0117   296 ;       FOR X = 1,2,3,4,5 :
                     0117   297 ;               CONDX_T - TITLE TEXT FOR CONDX TABLE
                     0117   298 ;               CONDX_TAB - ELEMENT TEXT FOR CONDX TABLE
                     0117   299 ;               CONDX_C - CONTEXT OF THE CONDX TABLE
                     0117   300 ;               CONDX_E - DATA ELEMENTS OF THE CONDX TABLE
                     0117   301 ;
                     0117   302 ; OUTPUT PARAMETERS:
                     0117   303 ;
                     0117   304 ;       NONE
                     0117   305 ;
                     0117   306 ; IMPLICIT OUTPUTS:
                     0117   307 ;
                     0117   308 ;       NONE
                     0117   309 ;
                     0117   310 ; COMPLETION CODES:
                     0117   311 ;
                     0117   312 ;       NONE
                     0117   313 ;
                     0117   314 ; SIDE EFFECTS:
                     0117   315 ;
                     0117   316 ;       NONE
                     0117   317 ;
                     0117   318 ;--
                     0117   319
                     0117   320
                     0117   321
                     0117   322 FORM_CONDS::
                     0117   323          $FAO_S  MSG1_INP_CTL,FAO_LEN,FAO_DESC,TESTNUM
                     0136   324                                          ; FORMAT CONDITIONS HEADER MSG
                FEC7' 30     0136   325          BSBW    OUTPUT_MSG      ; ... AND PRINT IT
              14  04 91     0139   326          CMPB    #COND1_C,#NULL  ; IS CONDITION 1 NULL ?
                  03 12     013C   327          BNEQU   10$             ; NO -- CONTINUE
                00E3 31     013E   328          BRW     FORM_CONDSX     ; YES -- SUBROUTINE IS FINISHED
                     0141   329 10$:
  00000000'EF   00000021'EF DE     0141   330          MOVAL   COND1_T,MSG_A           ; SAVE ADDRESS OF CONDITION 1 TITLE FOR FAO
00000000'EF   00000029'EF42 D0     014C   331          MOVL    COND1_TAB[R2],MSG_B     ; SAVE ADDR OF COND 1 CURR TEXT ELT FOR FAO
        00000000'EF   04 90     0158   332          MOVB    #COND1_C,MSG_CTXT       ; SAVE CONDITION 1 CONTEXT FOR FAO
                     015F   333          MOV_VAL COND1_C,COND1_E[R2],MSG_DATA1 ; GIVE COND 1 DATA VALUE TO FAO
```

```
                       FE92'  30  016B           BSBW    WRITE_MSG2                 ; FORMAT AND WRITE CONDITION 1 MSG
                  14     04   91  016E           CMPB    #COND2_C,#NULL             ; IS CONDITION 2 NULL ?
                         03   12  0171           BNEQU   20$                        ; NO -- CONTINUE
                       00AE   31  0173           BRW     FORM_CONDSX                ; YES -- SUBROUTINE IS FINISHED
                                  0176       20$:
    00000000'EF     00000049'EF   DE  0176           MOVAL   COND2_T,MSG_A              ; SAVE ADDRESS OF CONDITION 2 TITLE FOR FAO
  00000000'EF     00000051'EF43   D0  0181           MOVL    COND2_TAB[R3],MSG_B        ; SAVE ADDR OF COND 2 CURR TEXT ELT FOR FAO
            00000000'EF      04   90  018D           MOVB    #COND2_C,MSG_CTXT          ; SAVE CONDITION 2 CONTEXT FOR FAO
                                  0194           MOV_VAL COND2_C,COND2_E[R3],MSG_DATA1 ; GIVE COND 2 DATA VALUE TO FAO
                       FE5D'  30  01A0           BSBW    WRITE_MSG2                 ; FORMAT AND WRITE CONDITION 2 MSG
                  14     04   91  01A3           CMPB    #COND3_C,#NULL             ; IS CONDITION 3 NULL ?
                         03   12  01A6           BNEQU   30$                        ; NO -- CONTINUE
                       0079   31  01A8           BRW     FORM_CONDSX                ; YES -- SUBROUTINE IS FINISHED
                                  01AB       30$:
    00000000'EF     0000008E'EF   DE  01AB           MOVAL   COND3_T,MSG_A              ; SAVE ADDRESS OF CONDITION 3 TITLE FOR FAO
  00000000'EF     00000096'EF44   D0  01B6           MOVL    COND3_TAB[R4],MSG_B        ; SAVE ADDR OF COND 3 CURR TEXT ELT FOR FAO
            00000000'EF      04   90  01C2           MOVB    #COND3_C,MSG_CTXT          ; SAVE CONDITION 3 CONTEXT FOR FAO
                                  01C9           MOV_VAL COND3_C,COND3_E[R4],MSG_DATA1 ; GIVE COND 3 DATA VALUE TO FAO
                       FE28'  30  01D5           BSBW    WRITE_MSG2                 ; FORMAT AND WRITE CONDITION 3 MSG
                  14     14   91  01D8           CMPB    #COND4_C,#NULL             ; IS CONDITION 4 NULL ?
                         47   13  01DB           BEQLU   FORM_CONDSX                ; YES -- SUBROUTINE IS FINISHED
    00000000'EF     000000CD'EF   DE  01DD           MOVAL   COND4_T,MSG_A              ; SAVE ADDRESS OF CONDITION 4 TITLE FOR FAO
  00000000'EF     000000CD'EF45   D0  01E8           MOVL    COND4_TAB[R5],MSG_B        ; SAVE ADDR OF COND 4 CURR TEXT ELT FOR FAO
            00000000'EF      14   90  01F4           MOVB    #COND4_C,MSG_CTXT          ; SAVE CONDITION 4 CONTEXT FOR FAO
                                  01FB           MOV_VAL COND4_C,COND4_E[R5],MSG_DATA1 ; GIVE COND 4 DATA VALUE TO FAO
                       FE02'  30  01FB           BSBW    WRITE_MSG2                 ; FORMAT AND WRITE CONDITION 4 MSG
                  14     14   91  01FE           CMPB    #COND5_C,#NULL             ; IS CONDITION 5 NULL ?
                         21   13  0201           BEQLU   FORM_CONDSX                ; YES -- SUBROUTINE IS FINISHED
    00000000'EF     000000CE'EF   DE  0203           MOVAL   COND5_T,MSG_A              ; SAVE ADDRESS OF CONDITION 5 TITLE FOR FAO
  00000000'EF     000000CE'EF46   D0  020E           MOVL    COND5_TAB[R6],MSG_B        ; SAVE ADDR OF COND 5 CURR TEXT ELT FOR FAO
            00000000'EF      14   90  021A           MOVB    #COND5_C,MSG_CTXT          ; SAVE CONDITION 5 CONTEXT FOR FAO
                                  0221           MOV_VAL COND5_C,COND5_E[R6],MSG_DATA1 ; GIVE COND 5 DATA VALUE TO FAO
                       FDDC'  30  0221           BSBW    WRITE_MSG2                 ; FORMAT AND WRITE CONDITION 5 MSG
                                  0224       FORM_CONDSX:
                              05  0224           RSB                                ; RETURN TO CALLER
```

```
0225    370          .SBTTL  VERIFY
0225    371 ;++
0225    372 ; FUNCTIONAL DESCRIPTION:
0225    373 ;
0225    374 ;               VERIFY IS CALLED ONCE FOR EACH COMBINATION OF CONDITION
0225    375 ; TABLE VALUES (AS DETERMINED BY THE INDEX REGISTERS R2,3,4,5,6 FOR
0225    376 ; COND TABLES 1,2,3,4,5, RESPECTIVELY). VERIFY ESTABLISHES THE CONDITIONS
0225    377 ; SPECIFIED BY THE COND TABLES AND ISSUES THE SUBJECT SYSTEM SERVICE
0225    378 ; ($EXPREG). THEN, THE SUCCESSFUL OPERATION OF THE SERVICE IS VERIFIED
0225    379 ; BY EXAMINING THE STATUS CODE RETURNED, THE VALUES FOR RETURN ARGUMENTS
0225    380 ; AND THE FUNCTIONALITY PERFORMED. THE EXAMINATIONS TAKE THE FORM OF
0225    381 ; COMPARISONS AGAINST EXPECTED VALUES. ANY FAILING COMPARISON CAUSES AN
0225    382 ; ERR_EXIT MACRO TO BE EXECUTED (EITHER DIRECTLY, OR INDIRECTLY,
0225    383 ; THROUGH THE SS_CHECK MACRO); ERR_EXIT SETS EFLAG TO NON-ZERO,
0225    384 ; PRINTS ERROR MESSAGES AND CAUSES AN IMMEDIATE RSB TO CALLER.
0225    385 ; WHEN ERR_EXIT IS EXECUTED, FURTHER CALLS TO VERIFY ARE SUPPRESSED,
0225    386 ; AND, AFTER EXECUTING CLEANUP SUBROUTINES, THE IMAGE EXITS.
0225    387 ;
0225    388 ; CALLING SEQUENCE:
0225    389 ;
0225    390 ;       BSBW VERIFY
0225    391 ;
0225    392 ; INPUT PARAMETERS:
0225    393 ;
0225    394 ;       NONE
0225    395 ;
0225    396 ; IMPLICIT INPUTS:
0225    397 ;
0225    398 ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
0225    399 ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
0225    400 ;       FOR X = 1,2,3,4,5 :
0225    401 ;               CONDX_E - ADDRESS OF TABLE OF DATA VALUES FOR CONDX
0225    402 ;                 TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
0225    403 ;                 ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
0225    404 ;                 FOR CONDX_E.
0225    405 ;
0225    406 ; OUTPUT PARAMETERS:
0225    407 ;
0225    408 ;       NONE
0225    409 ;
0225    410 ; IMPLICIT OUTPUTS:
0225    411 ;
0225    412 ;       VERIFY HAS NO OUTPUT. SINCE ITS PURPOSE IS TO TEST FOR ERRORS,
0225    413 ;       IT MERELY RETURNS TO CALLER NORMALLY AFTER THE TESTS, PROVIDING
0225    414 ;       ALL WERE SUCCESSFUL; IF AN ERROR IS DISCOVERED, RETURN IS VIA
0225    415 ;       AN ERR_EXIT OR SS_CHECK MACRO, BOTH OF WHICH DOCUMENT DETECTED
0225    416 ;       ERRORS.
0225    417 ;
0225    418 ; COMPLETION CODES:
0225    419 ;
0225    420 ;       EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
0225    421 ;
0225    422 ; SIDE EFFECTS:
0225    423 ;
0225    424 ;       SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
0225    425 ;       (VIA RSB) IF ERROR ENCOUNTERED.
0225    426 ;
```

```
                        0225   427 ;--
                        0225   428
                        0225   429
                        0225   430
                        0225   431 VERIFY::
00000000'EF   95        0225   432          TSTB    CFLAG                   ; SHOULD CONDITIONS BE PRINTED ?
          03   13       022B   433          BEQL    5$                      ; NO -- CONTINUE
        FEE7   30       022D   434          BSBW    FORM_CONDS              ; YES -- FMT & PRINT ALL CONDS FOR THIS T.C.
                        0230   435 5$:
5A 00000051'EF   D0     0230   436          MOVL    PAGESIZE,R10            ; GET PAGE SIZE INTO R10 FOR LATER USE
   58    FF 8F   98     0237   437          CVTBL   #-1,R8                  ; ... AND A -1 INTO R8
00000041'EF42   D5      023B   438          TSTL    REGION[R2]              ; IS IT PROGRAM REGION ?
          06   13       0242   439          BEQL    7$                      ; YES -- LEAVE REGS ALONE
       5A   5A   CE     0244   440          MNEGL   R10,R10                 ; GET NEGATIVE PAGE SIZE
       58   58   CE     0247   441          MNEGL   R8,R8                   ; ... AND MAKE R8 = +1
                        024A   442 7$:
                        024A   443          MODE    TO,20$,KRNL             ; TO KERNEL FOR EXPREG
                        026D   444 ;
                        026D   445 ; ****** SYSTEM SERVICE CALL WHICH IS THE SUBJECT OF THIS TEST CASE ******
                        026D   446 ;
                        026D   447          $EXPREG_S PAGCNT[R3],RETADR,ACMODE[R4],REGION[R2]
                        028F   448          MODE    FROM,20$                ; BACK TO USER
00000000'8F   50        0290   449          CMPL    R0,#SS$_NORMAL          ; CODE RECEIVED = CODE EXPECTED ?
          03   12       0297   450          BNEQU   30$                     ; NO -- GO PROCESS ERROR
        0061   31       0299   451          BRW     55$                     ; YES -- MORE VERIFYING
                        029C   452 30$:
00000000'EF   00000000'8F   D0  029C   453          MOVL    #SS$_NORMAL,EXPV        ; LOAD UP EXPECTED AND
00000000'EF   50        D0     02A7   454          MOVL    R0,RECV                 ; ... RECEIVED VALUES, THEN EXIT
                        02AE   455          ERR_EXIT LONG,<INCORRECT STATUS CODE RETURNED FROM EXPREG>
                        02FD   456 55$:
59 00000009'EF42   D0   02FD   457          MOVL    INITIALADR[R2],R9       ; GET START OF FREE AREA
  00000011'EF   59   D1 0305   458          CMPL    R9,RETADR               ; DID EXPREG RETURN EXPECTED START ADDR ?
          56   13       030C   459          BEQLU   60$                     ; YES -- GO ON
00000000'EF   59   D0   030E   460          MOVL    R9,EXPV                 ; NO -- LOAD UP EXPECTED AND
00000000'EF   00000011'EF   D0 0315   461          MOVL    RETADR,RECV             ; ... RECEIVED VALUES, THEN EXIT
                        0320   462          ERR_EXIT LONG,<INCORRECT STARTING PAGE ADDRESS>
                        0364   463 60$:                                  ; NOW CHECK ENDING ADDRESS
```

```
                                    0434      75$:
                           69  95   0434           TSTB    (R9)                      ; FIRST BYTE OF PAGE = ZERO, AS PROMISED ?
                           09  12   0436           BNEQ    80$                       ; NO -- GO INDICATE NON-ZERO ERROR
           69    00000000'EF   90   0438           MOVB    ONES,(R9)                 ; DO A STORE -- NO ACCVIO EXPECTED
                           1A  11   043F           BRB     90$                       ; GO LOOK AT NEXT PAGE
                                    0441      80$:
   00000008'EF   00000000'EF   90   0441           MOVB    ONES,NZERR                ; INDICATE NON-ZERO ERROR FOUND
                 00000000'EF   94   044C           CLRB    EXPV                      ; LOAD UP EXPECTED AND
            00000000'EF    69  90   0452           MOVB    (R9),RECV                 ; ... RECEIVED VALUES, THEN EXIT
                           0A  11   0459           BRB     100$                      ; GO PROCESS ERROR
                                    045B      90$:
    FFCF 59    5A  00000015'EF  F1   045B           ACBL    RETADR+4,R10,R9,75$       ; INCR (OR DECR) TO NEXT PAGE & LOOP
                                    0465      100$:
          03   000000BD'EF44   D1   0465           CMPL    ACMODE[R4],#PSL$C_USER    ; USER MODE ?
                           01  13   046D           BEQLU   110$                      ; YES -- DON'T CHANGE MODE
                                    046F           MODE    FROM,63$                  ; CHANGE MODE BACK TO USER
                                    0470      110$:
                 00000008'EF   95   0470           TSTB    NZERR                     ; WAS A NON-ZERO ERROR ENCOUNTERED ?
                           4D  13   0476           BEQL    VERIFYX                   ; NO -- ALL FINISHED
                                    0478           ERR_EXIT BYTE,<A PAGE IN THE EXPANSION AREA IS NON-ZERO>
                                    04C5      VERIFYX:
                               05   04C5           RSB                               ; RETURN TO CALLER
```

```
                        04C6    507          .SBTTL  VFY_CLEANUP
                        04C6    508 ;++
                        04C6    509 ; FUNCTIONAL DESCRIPTION:
                        04C6    510 ;
                        04C6    511 ;               VFY_CLEANUP EXECUTES SYSTEM SERVICES TO UNDO THE
                        04C6    512 ; EFFECT OF THOSE ISSUED IN THE VERIFY SUBROUTINE. VFY_CLEANUP MUST
                        04C6    513 ; ASSUME THAT VERIFY MAY NOT HAVE EXECUTED IN ITS ENTIRETY (IF AN
                        04C6    514 ; ERROR IS FOUND). ALSO, VFY_CLEANUP MAY ISSUE SS_CHECK OR ERR_EXIT
                        04C6    515 ; ONLY AFTER PERFORMING ALL OF ITS CLEANUP OPERATIONS; THIS IS REQUIRED
                        04C6    516 ; IN THE EVENT THAT VFY_CLEANUP IS CALLED DURING ERROR PROCESSING,
                        04C6    517 ; WHEN PERFORMING THE REQUIRED CLEANUP IS MORE IMPORTANT THAN
                        04C6    518 ; POSSIBLY DISCOVERING A SECOND ERROR.
                        04C6    519 ;
                        04C6    520 ; CALLING SEQUENCE:
                        04C6    521 ;
                        04C6    522 ;       BSBW VFY_CLEANUP
                        04C6    523 ;
                        04C6    524 ; INPUT PARAMETERS:
                        04C6    525 ;
                        04C6    526 ;       NONE
                        04C6    527 ;
                        04C6    528 ; IMPLICIT INPUTS:
                        04C6    529 ;
                        04C6    530 ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
                        04C6    531 ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
                        04C6    532 ;       FOR X = 1,2,3,4,5 :
                        04C6    533 ;               CONDX_E - ADDRESS OF TABLE OF DATA VALUES FOR CONDX
                        04C6    534 ;                 TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
                        04C6    535 ;                 ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
                        04C6    536 ;                 FOR CONDX_E.
                        04C6    537 ;
                        04C6    538 ; OUTPUT PARAMETERS:
                        04C6    539 ;
                        04C6    540 ;       NONE
                        04C6    541 ;
                        04C6    542 ; IMPLICIT OUTPUTS:
                        04C6    543 ;
                        04C6    544 ;       NONE
                        04C6    545 ;
                        04C6    546 ; COMPLETION CODES:
                        04C6    547 ;
                        04C6    548 ;       EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
                        04C6    549 ;
                        04C6    550 ; SIDE EFFECTS:
                        04C6    551 ;
                        04C6    552 ;       SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
                        04C6    553 ;       (VIA RSB) IF ERROR ENCOUNTERED.
                        04C6    554 ;
                        04C6    555 ;--
                        04C6    556
                        04C6    557
                        04C6    558
                        04C6    559 VFY_CLEANUP::
00000019'EF   00000011'EF   7D  04C6    560          MOVQ    RETADR,INADR_DVA        ; PAGE RANGE TO DELETE
                        04D1    561          MODE    TO,10$,KRNL             ; INTO KERNEL FOR DELTVA
                        04F4    562          $DELTVA_S INADR=INADR_DVA, -
                        04F4    563                    ACMODE=ACMODE[R4]     ; GET RID OF ACQUIRED SPACE
```

```
                050A   564          MODE    FROM,10$                ; BACK TO USER MODE
                050B   565          SS_CHECK NORMAL                 ; CHECK FOR NORMAL RETURN FROM DELTVA
05              0539   566          RSB                             ; RETURN TO CALLER
                053A   567          .END
```

```
$$$$                    = 00000482 R     04
$$$CHARS                = 00000028
$$$CHARS1               = 00000006
$$$CHARS2               = 00000004
$$$CHARS3               = 00000005
$$$CHARS4               = 00000004
$$$CHARS5               = 00000000
$$$COND_A               = 00000003
$$$STRINGS              = 00000001
$$$STRINGS2             = 00000005
$$T1                    = 00000000
$$T2                    = 00000004
ACMODE                    000000BD R     03
BYTE                    = 00000001   G
CFLAG                     ********     X   04
CHMRTN                    ********     X   04
CHM_CONT                  ********     X   04
COMP_SC                   ********     X   04
CONDT                     0000010D RG    04
COND1_C                 = 00000004
COND1_CLEANUP             0000010E RG    04
COND1_E                   00000041 R     03
COND1_H                   00000028 RG    03
COND1_T                   00000021 R     03
COND1_TAB                 00000029 R     03
COND2                     0000010F RG    04
COND2_C                 = 00000004
COND2_CLEANUP             00000110 RG    04
COND2_E                   00000082 R     03
COND2_H                   00000050 RG    03
COND2_T                   00000049 R     03
COND2_TAB                 00000051 R     03
COND3                     00000111 RG    04
COND3_C                 = 00000004
COND3_CLEANUP             00000112 RG    04
COND3_E                   000000BD R     03
COND3_H                   00000095 RG    03
COND3_T                   0000008E R     03
COND3_TAB                 00000096 R     03
COND4                     00000113 RG    04
COND4_C                 = 00000014
COND4_CLEANUP             00000114 RG    04
COND4_H                   000000CD RG    03
COND4_T                   000000CD R     03
COND4_TAB                 000000CD R     03
COND5                     00000115 RG    04
COND5_C                 = 00000014
COND5_CLEANUP             00000116 RG    04
COND5_H                   000000CE RG    03
COND5_T                   000000CE R     03
COND5_TAB                 000000CE R     03
CTL$GL_PHD                ********     X   04
DESC                    = 00000010   G
EFLAG                     ********     X   04
EXPV                      ********     X   04
FAO_DESC                  ********     X   04
FAO_LEN                   ********     X   04
FORM_CONDS                00000117 RG    04
FORM_CONDSX               00000224 R     04
INADR_DVA                 00000019 R     03
INITIALADR                00000009 R     03
JPI$_FREPOVA            = 00000404
JPI$_FREP1VA            = 00000405
JPITEMS                   00000055 R     02
LONG                    = 00000004   G
MOD_MSG_CODE              ********     X   04
MOD_MSG_PRINT             ********     X   04
MSGT_INP_CTL              00000019 R     02
MSG3_ERR_CTL              00000039 RG    02
MSG_A                     ********     X   04
MSG_B                     ********     X   04
MSG_CTXT                  ********     X   04
MSG_DATA1                 ********     X   04
NOTARG                  = 00000000   G
NULL                    = 00000014   G
NZERR                     00000008 R     03
ONES                      ********     X   04
OUTPUT_MSG                ********     X   04
PAGCNT                    00000082 R     03
PAGESIZE                  00000051 R     02
PCV                       ********     X   04
PHD$Q_PRIVMSK           = 00000000
PRIVMASK                  00000000 R     03
PRIV_ARGS               = 00000002
PROCESS_ERR               ********     X   04
PSL$C_EXEC              = 00000001
PSL$C_KERNEL            = 00000000
PSL$C_SUPER             = 00000002
PSL$C_USER              = 00000003
QUAD                    = 00000008   G
RECV                      ********     X   04
REGION                    00000041 R     03
REST_REGS                 ********     X   04
RETADR                    00000011 R     03
SAVE_REGS                 ********     X   04
SS$_NORMAL                ********     X   04
SUCCESS                   ********     X   04
SYS$CMEXEC                ********    GX   04
SYS$CMKRNL                ********    GX   04
SYS$DELTVA                ********    GX   04
SYS$EXPREG                ********    GX   04
SYS$FAO                   ********     X   04
SYS$GETJPI                ********    GX   04
SYS$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                00000109 RG    04
TM_SETUP                  00000000 RG    04
VERIFY                    00000225 RG    04
VERIFYX                   000004C5 R     04
```

```
VFY_CLEANUP                     000004C6 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	00000071 (113.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC LONG
RWDATA	000000CF (207.)	03 (3.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
SATSS70	0000053A (1338.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	36	00:00:00.09	00:00:00.34
Command processing	135	00:00:00.61	00:00:02.78
Pass 1	276	00:00:08.14	00:00:19.00
Symbol table sort	0	00:00:00.68	00:00:01.52
Pass 2	121	00:00:02.01	00:00:04.56
Symbol table output	14	00:00:00.10	00:00:00.20
Psect synopsis output	2	00:00:00.02	00:00:00.26
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	586	00:00:11.65	00:00:28.66

The working set limit was 1500 pages.
42361 bytes (83 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 453 non-local and 44 local symbols.
567 source lines were read in Pass 1, producing 24 object records in Pass 2.
39 pages of virtual memory were used to define 30 macros.

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

Macro library name	Macros defined
_255DUA28:[SHRLIB]UETP.MLB;1	9
_255DUA28:[SYS.OBJ]LIB.MLB;1	1
_255DUA28:[SYSLIB]STARLET.MLB;2	17
TOTALS (all libraries)	27

801 GETS were required to define 27 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:SATSS70/OBJ=OBJ$:SATSS70 MSRC$:SATSS70/UPDATE=(ENH$:SATSS70)+EXECML$/LIB+SHRLIB$:UETP/LIB
```

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SATSSS56
LIS

SATSSS60
LIS

SATSSS71
LIS

SATSSS54
LIS

SATSSS73
LIS

SATSSS70
LIS

SATSSS72
LIS

SATSSS55
LIS

SATSSS53
LIS

SATSSS61
LIS