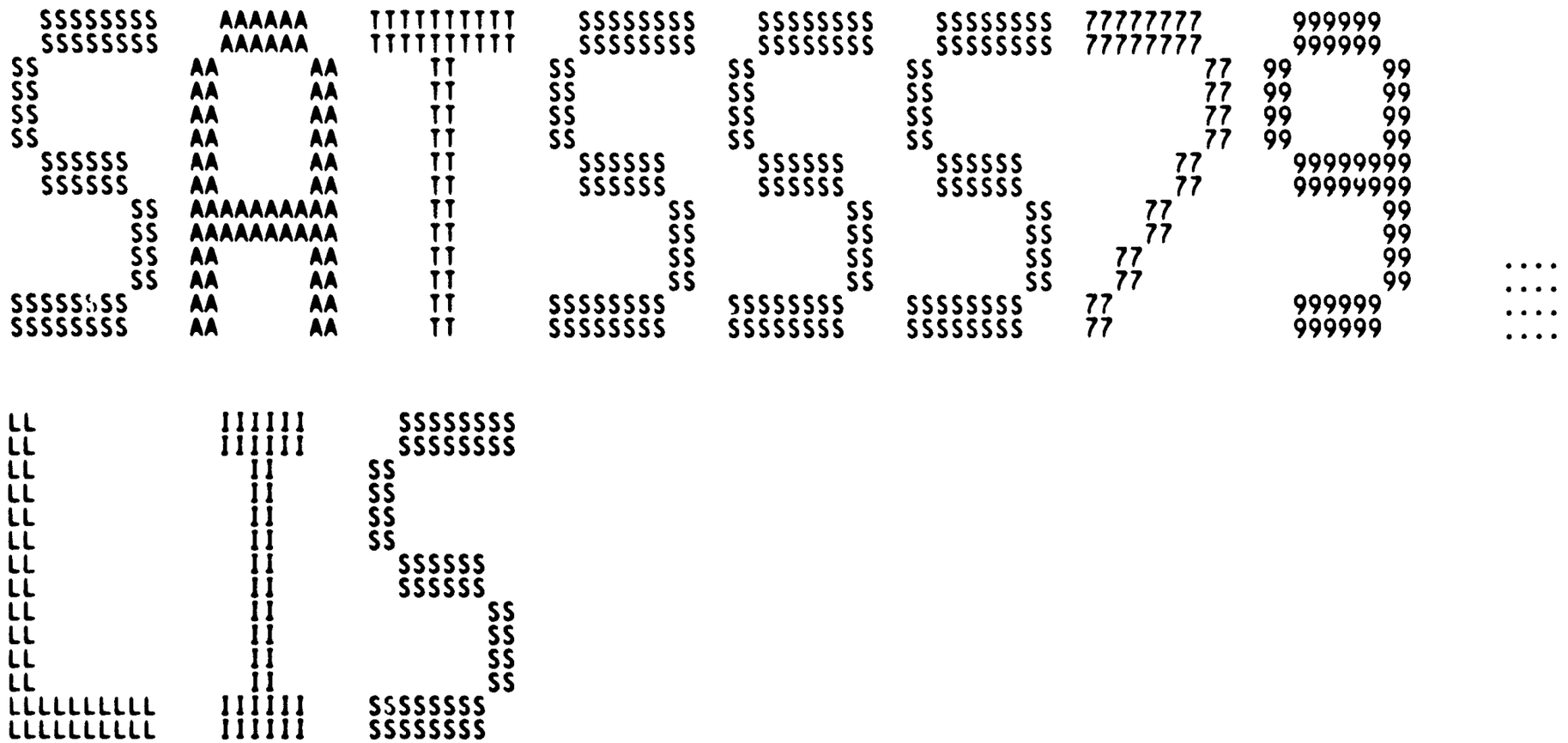
```
UUU          UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP        SSSSSSSSSSS  YYY          YYY
UUU          UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP        SSSSSSSSSSS  YYY          YYY
UUU          UUU  EEEEEEEEEEEEEEE  TTTTTTTTTTTTTTT  PPPPPPPPPPPP        SSSSSSSSSSS  YYY    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


SSSSSSSS AAAAAA TTTTTTTTTT SSSSSSSS SSSSSSSS SSSSSSSS 77777777 999999
SSSSSSSS AAAAAA TTTTTTTTTT SSSSSSSS SSSSSSSS SSSSSSSS 77777777 999999
SS AA AA TT SS SS SS 77 99 99
SS AA AA TT SS SS SS 77 99 99
SS AA AA TT SS SS SS 77 99 99
SS AA AA TT SS SS SS 77 99 99
SSSSSS AA AA TT SSSSSS SSSSSS SSSSSS 77 99999999
SSSSSS AA AA TT SSSSSS SSSSSS SSSSSS 77 99999999
SS AAAAAAAAAA TT SS SS SS 77 99
SS AAAAAAAAAA TT SS SS SS 77 99
SS AA AA TT SS SS SS 77 99
SS AA AA TT SS SS SS 77 99
SSSSSSSS AA AA TT SSSSSSSS SSSSSSSS SSSSSSSS 77 999999
SSSSSSSS AA AA TT SSSSSSSS SSSSSSSS SSSSSSSS 77 999999
LL IIIIII SSSSSSSS
LL IIIIII SSSSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LL II SSSSSS
LL II SSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

(1) 54 DECLARATIONS
(1) 87 CONDITION TABLES
(1) 118 TM_SETUP, TM_CLEANUP
(1) 181 CONDITION SUBROUTINES - SETUP AND CLEANUP
(1) 258 FORM_CONDS
(1) 351 VERIFY
(1) 521 VFY_CLEANUP

```
0000     1          .TITLE  SATSSS79 SATS SYST SERV TESTS  $LCK/ULKPAG (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 SATSSS79 TO TEST SUCCESSFUL
0000    36  ; OPERATION OF THE $LCK/ULKPAG SYSTEM SERVICES. EACH SERVICE IS INVOKED
0000    37  ; UNDER VARIOUS INPUT CONDITIONS WITH VARYING INPUT PARAMETERS. ONLY
0000    38  ; SUCCESSFUL STATUS CODES ARE EXPECTED IN THIS TEST MODULE.  CORRECT
0000    39  ; OPERATION OF EACH SERVICE FOR EACH OF ITS ISSUANCES IS VERIFIED BY
0000    40  ; CHECKING FOR AN SS$_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: APR, 1978
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          $PSLDEF                         ; PROCESSOR STATUS LONGWORD DEFINITIONS
0000     61 ;
0000     62 ; MACROS:
0000     63 ;
0000     64 ;
0000     65 ; EQUATED SYMBOLS:
0000     66 ;
0000     67 ;
0000     68 ; OWN STORAGE:
0000     69 ;
```

```
00000000     71          .PSECT  RODATA,RD,NOWRT,NOEXE,LONG
    0000     72 TEST_MOD_NAME:: STRING  C,<SATSSS79>     ; TEST MODULE NAME
    0009     73 TEST_MOD_NAME_D: STRING I,<SATSSS79>     ; TEST MODULE NAME DESCRIPTOR
    0019     74 MSG1_INP_CTL:   STRING  I,<  SSLUP!4ZW: CONDITIONS:>
    0039     75                                          ; FAO CTL STRING FOR MSG1 IN SUCCOMMON.MAR
    0039     76 MSG3_ERR_CTL::  STRING  I,< *SSLUP!4ZW: !AS>
    0051     77                                          ; FAO CTL STRING FOR MSG3 IN SUCCOMMON.MAR
```

```
                00000000     79          .PSECT  RWDATA,RD,WRT,NOEXE,LONG
00000008  0000  80 PRIVMASK:               .BLKQ   1               ; ADDR OF PRIVILEGE MASK (IN PHD)
0000000C  0008  81 PAGCNT_ERG:             .BLKL   1               ; PAGCNT ARGUMENT FOR EXPREG
00000010  000C  82 REGION_ERG:             .BLKL   1               ; REGION ARGUMENT FOR EXPREG
00000014  0010  83 ACMODE_ERG:             .BLKL   1               ; ACMODE ARGUMENT FOR EXPREG
0000001C  0014  84 RETADR:                 .BLKQ   1               ; RETADR ARG FOR SUBJ $LCKPAG & $ULKPAG
0000001D  001C  85 LOCKMODE:               .BLKB   1               ; SAVE AREA FOR MODE OF LCKPAG/ULKPAG ISSUER
```

```
                        001D     87          .SBTTL  CONDITION TABLES
                        001D     88 ;
                        001D     89 ;        ***** CONDITION TABLES FOR LCKPAG/ULKPAG SYSTEM SERVICES *****
                        001D     90 ;
                        001D     91          COND    1,QUAD,<INADR>,-
                        001D     92                    <CURRENT IMAGE PAGE>,-
                        001D     93                    <PROGRAM REGION -- SINGLE PAGE>,-
                        001D     94                    <PROGRAM REGION -- MULTIPLE PAGES>,-
                        001D     95                    <CONTROL REGION -- SINGLE PAGE>,-
                        001D     96                    <CONTROL REGION -- MULTIPLE PAGES>,-
                        001D     97
                00000000'00000000'  00C9     98                      .LONG       TM_SETUP,TM_SETUP
00000000'00000000'00000000'00000000'  00D1     99                      .LONG       0[8]
00000000'00000000'00000000'00000000'  00E1
                        00F1    100 ;
                        00F1    101          COND    2,NOTARG,<PAGE OWNER/LOCKER ACCESS MODES>,-
                        00F1    102                    <KERNEL/KERNEL>,-
                        00F1    103                    <USER/KERNEL>,-
                        00F1    104                    <USER/USER>,-
                        00F1    105
                  00 00  0141    106                      .BYTE       PSL$C_KERNEL,PSL$C_KERNEL
                  00 03  0143    107                      .BYTE       PSL$C_USER,PSL$C_KERNEL
                  03 03  0145    108                      .BYTE       PSL$C_USER,PSL$C_USER
                        0147    109 ;
                        0147    110          COND    3,NULL
                        0148    111
                        0148    112          COND    4,NULL
                        0149    113
                        0149    114          COND    5,NULL
                        014A    115
                    00000000    116          .PSECT  SATSSS79,RD,WRT,EXE
```

```
                        0000   118          .SBTTL  TM_SETUP, TM_CLEANUP
                        0000   119 ;++
                        0000   120 ; FUNCTIONAL DESCRIPTION:
                        0000   121 ;
                        0000   122 ;                TM_SETUP AND TM_CLEANUP ARE CALLED TO PERFORM
                        0000   123 ; REQUIRED HOUSEKEEPING AT THE BEGINNING AND END, RESPECTIVELY, OF
                        0000   124 ; TEST MODULE EXECUTION.
                        0000   125 ;
                        0000   126 ; CALLING SEQUENCE:
                        0000   127 ;
                        0000   128 ;       BSBW TM_SETUP   BSBW TM_CLEANUP
                        0000   129 ;
                        0000   130 ; INPUT PARAMETERS:
                        0000   131 ;
                        0000   132 ;       NONE
                        0000   133 ;
                        0000   134 ; IMPLICIT INPUTS:
                        0000   135 ;
                        0000   136 ;       NONE
                        0000   137 ;
                        0000   138 ; OUTPUT PARAMETERS:
                        0000   139 ;
                        0000   140 ;       NONE
                        0000   141 ;
                        0000   142 ; IMPLICIT OUTPUTS:
                        0000   143 ;
                        0000   144 ;       TM_SETUP:  COND TABLE INDEX REGISTERS (R2,3,4,5,6) CLEARED;
                        0000   145 ;                  ALL PRIVILEGES ACQUIRED.
                        0000   146 ;
                        0000   147 ; COMPLETION CODES:
                        0000   148 ;
                        0000   149 ;       EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
                        0000   150 ;
                        0000   151 ; SIDE EFFECTS:
                        0000   152 ;
                        0000   153 ;       SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
                        0000   154 ;       (VIA RSB) IF ERROR ENCOUNTERED.
                        0000   155 ;
                        0000   156 ;--
                        0000   157
                        0000   158
                        0000   159
                        0000   160 TM_SETUP::
                  52    D4    0000   161          CLRL    R2                      ; INITIALIZE
                  53    D4    0002   162          CLRL    R3                      ; .. CONDITION
                  54    D4    0004   163          CLRL    R4                      ; .... TABLE
                  55    D4    0006   164          CLRL    R5                      ; ...... INDEX
                  56    D4    0008   165          CLRL    R6                      ; ........ REGISTERS
              FFF3'   30    000A   166          BSBW    MOD_MSG_PRINT           ; PRINT TEST MODULE BEGIN MSG
00000000'EF   00000000'EF   DE    000D   167          MOVAL   TEST_MOD_SUCC,TMD_ADDR  ; ASSUME END MSG WILL SHOW SUCCESS
   03   00   00000000'8F   F0    0018   168          INSV    #SUCCESS,#0,#3,MOD_MSG_CODE ; ADJUST STATUS CODE FOR SUCCESS
              00000000'EF         0020
                        0025   169          MODE    TO,S$,KRNL              ; KERNEL MODE TO ACCESS PHD
        59    00000000'9F   D0    0048   170          MOVL    @#CTL$GL_PHD,R9         ; GET PROCESS HEADER ADDRESS
        00000000'EF   69    DE    004F   171          MOVAL   PHD$Q_PRIVMSK(R9),PRIVMASK ; GET PRIV MASK ADDRESS
                        0056   172          MODE    FROM,S$ ; BACK TO USER MODE
                        0057   173          PRIV    ADD,ALL                 ; GET ALL PRIVILEGES
```

```
                     0077   174          $SETPRN_S TEST_MOD_NAME_D       ; SET PROCESS NAME
                     0084   175          SS_CHECK  NORMAL                ; CHECK STATUS CODE RETURNED FROM SETPRN
               05    00B2   176          RSB                             ; RETURN TO MAIN ROUTINE
                     00B3   177 TM_CLEANUP::
          FF4A'      30    00B3   178          BSBW    MOD_MSG_PRINT    ; PRINT TEST MODULE END MSG
               05    00B6   179          RSB                             ; RETURN TO MAIN ROUTINE
```

```
                     00B7   181          .SBTTL  CONDITION SUBROUTINES - SETUP AND CLEANUP
                     00B7   182 ;++
                     00B7   183 ; FUNCTIONAL DESCRIPTION:
                     00B7   184 ;
                     00B7   185 ;                 CONDX AND CONDX_CLEANUP ARE SUBROUTINES WHICH ARE EXECUTED
                     00B7   186 ; BEFORE AND AFTER THE VERIFY SUBROUTINE, RESPECTIVELY, WHENEVER A NEW
                     00B7   187 ; CONDITION X VALUE IS SELECTED (SEE FUNCTIONAL DESCRIPTION OF SUCCOMMON
                     00B7   188 ; ROUTINE IN SUCCOMMON.MAR). ANY SETUP FUNCTION PARTICULAR TO THE
                     00B7   189 ; CONDITION X TABLE IS INCLUDED IN THE CONDX SUBROUTINE AND CLEANED
                     00B7   190 ; UP, IF NECESSARY, IN THE CONDX_CLEANUP SUBROUTINE. THIS INCLUDES,
                     00B7   191 ; ESPECIALLY, CODE TO DETECT CONFLICTS AMONG CURRENT ENTRIES IN TWO
                     00B7   192 ; OR MORE CONDITION TABLES. IF A CONFLICT IS DETECTED, A NON-ZERO
                     00B7   193 ; VALUE IS STORED INTO CONFLICT, WHICH CAUSES THE CALLING ROUTINE
                     00B7   194 ; (SUCCOMMON) TO SKIP THE CURRENT ENTRY IN THE CONDITION X TABLE.
                     00B7   195 ;
                     00B7   196 ; CALLING SEQUENCE:
                     00B7   197 ;
                     00B7   198 ;       BSBW CONDX   BSBW CONDX_CLEANUP
                     00B7   199 ;         WHERE X = 1,2,3,4,5
                     00B7   200 ;
                     00B7   201 ; INPUT PARAMETERS:
                     00B7   202 ;
                     00B7   203 ;       CONFLICT = 0
                     00B7   204 ;
                     00B7   205 ; IMPLICIT INPUTS:
                     00B7   206 ;
                     00B7   207 ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
                     00B7   208 ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
                     00B7   209 ;
                     00B7   210 ; OUTPUT PARAMETERS:
                     00B7   211 ;
                     00B7   212 ;       CONFLICT SET TO NON-ZERO IF COND TABLE CONFLICT DETECTED.
                     00B7   213 ;
                     00B7   214 ; IMPLICIT OUTPUTS:
                     00B7   215 ;
                     00B7   216 ;       R2,3,4,5,6 PRESERVED
                     00B7   217 ;
                     00B7   218 ; COMPLETION CODES:
                     00B7   219 ;
                     00B7   220 ;       NONE
                     00B7   221 ;
                     00B7   222 ; SIDE EFFECTS:
                     00B7   223 ;
                     00B7   224 ;       NONE
                     00B7   225 ;
                     00B7   226 ;--
                     00B7   227
                     00B7   228
                     00B7   229
                     00B7   230 COND1::
          05         00B7   231          RSB                             ; RETURN TO MAIN ROUTINE
                     00B8   232 COND1_CLEANUP::
          05         00B8   233          RSB                             ; RETURN TO MAIN ROUTINE
                     00B9   234 COND2::
     52   D5         00B9   235          TSTL    R2                      ; FIRST COND 1 ELEMENT ??
     0F   12         00BB   236          BNEQ    COND2X                  ; NO -- NO CONFLICT
     53   D5         00BD   237          TSTL    R3                      ; YES -- FIRST COND 2 ELEMENT ALSO ??
```

```
                     0B  12  00BF   238          BNEQ    COND2X                  ; NO -- NO CONFLICT
00000000'EF  00000000'EF 90  00C1   239          MOVB    ONES,CONFLICT           ; YES -- INDICATE CONFLICT BECAUSE ...
                         00CC   240                                              ; ... CURRENT IMAGE PAGE NOT OWNED BY KRNL
                         00CC   241 COND2X:
                         05  00CC   242          RSB                             ; RETURN TO MAIN ROUTINE
                         00CD   243 COND2_CLEANUP::
                         05  00CD   244          RSB                             ; RETURN TO MAIN ROUTINE
                         00CE   245 COND3::
                         05  00CE   246          RSB                             ; RETURN TO MAIN ROUTINE
                         00CF   247 COND3_CLEANUP::
                         05  00CF   248          RSB                             ; RETURN TO MAIN ROUTINE
                         00D0   249 COND4::
                         05  00D0   250          RSB                             ; RETURN TO MAIN ROUTINE
                         00D1   251 COND4_CLEANUP::
                         05  00D1   252          RSB                             ; RETURN TO MAIN ROUTINE
                         00D2   253 COND5::
                         05  00D2   254          RSB                             ; RETURN TO MAIN ROUTINE
                         00D3   255 COND5_CLEANUP::
                         05  00D3   256          RSB                             ; RETURN TO MAIN ROUTINE
```

```
                                                                L 5
00D4   258          .SBTTL  FORM_CONDS
00D4   259 ;++
00D4   260 ; FUNCTIONAL DESCRIPTION:
00D4   261 ;
00D4   262 ;                FORM_CONDS FORMATS AND PRINTS INFORMATION ABOUT
00D4   263 ;  THE CURRENT ELEMENT IN EACH OF THE CONDITION TABLES.
00D4   264 ;
00D4   265 ; CALLING SEQUENCE:
00D4   266 ;
00D4   267 ;        BSBW FORM_CONDS
00D4   268 ;
00D4   269 ; INPUT PARAMETERS:
00D4   270 ;
00D4   271 ;        NONE
00D4   272 ;
00D4   273 ; IMPLICIT INPUTS:
00D4   274 ;
00D4   275 ;        R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
00D4   276 ;          FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
00D4   277 ;        FOR X = 1,2,3,4,5 :
00D4   278 ;                CONDX_T - TITLE TEXT FOR CONDX TABLE
00D4   279 ;                CONDX_TAB - ELEMENT TEXT FOR CONDX TABLE
00D4   280 ;                CONDX_C - CONTEXT OF THE CONDX TABLE
00D4   281 ;                CONDX_E - DATA ELEMENTS OF THE CONDX TABLE
00D4   282 ;
00D4   283 ; OUTPUT PARAMETERS:
00D4   284 ;
00D4   285 ;        NONE
00D4   286 ;
00D4   287 ; IMPLICIT OUTPUTS:
00D4   288 ;
00D4   289 ;        NONE
00D4   290 ;
00D4   291 ; COMPLETION CODES:
00D4   292 ;
00D4   293 ;        NONE
00D4   294 ;
00D4   295 ; SIDE EFFECTS:
00D4   296 ;
00D4   297 ;        NONE
00D4   298 ;
00D4   299 ;--
00D4   300
00D4   301
00D4   302
00D4   303 FORM_CONDS::
00D4   304          $FAO_S  MSG1_INP_CTL,FAO_LEN,FAO_DESC,TESTNUM
00F3   305                                                  ; FORMAT CONDITIONS HEADER MSG
   FF0A'   30  00F3   306          BSBW    OUTPUT_MSG      ; ... AND PRINT IT
 14   08   91  00F6   307          CMPB    #COND1_C,#NULL  ; IS CONDITION 1 NULL ?
      03   12  00F9   308          BNEQU   10$             ; NO -- CONTINUE
    00CB   31  00FB   309          BRW     FORM_CONDSX     ; YES -- SUBROUTINE IS FINISHED
                00FE   310 10$:
00000000'EF  0000001D'EF  DE  00FE   311          MOVAL   COND1_T,MSG_A   ; SAVE ADDRESS OF CONDITION 1 TITLE FOR FAO
00000000'EF  00000024'EF42  D0  0109   312          MOVL    COND1_TAB[R2],MSG_B ; SAVE ADDR OF COND 1 CURR TEXT ELT FOR FAO
      00000000'EF   08  90  0115   313          MOVB    #COND1_C,MSG_CTXT   ; SAVE CONDITION 1 CONTEXT FOR FAO
                011C   314          MOV_VAL COND1_C,COND1_E[R2],MSG_DATA1 ; GIVE COND 1 DATA VALUE TO FAO
```

```
                                                M 5
                     FED5'   30   0128   315          BSBW    WRITE_MSG2              ; FORMAT AND WRITE CONDITION 1 MSG
                  14   00   91   012B   316          CMPB    #COND2_C,#NULL          ; IS CONDITION 2 NULL ?
                       03   12   012E   317          BNEQU   20$                     ; NO -- CONTINUE
                    0096   31   0130   318          BRW     FORM_CONDSX             ; YES -- SUBROUTINE IS FINISHED
                           0133   319 20$:
  00000000'EF   000000F1'EF   DE   0133   320          MOVAL   COND2_T,MSG_A           ; SAVE ADDRESS OF CONDITION 2 TITLE FOR FAO
00000000'EF   00000111'EF43   D0   013E   321          MOVL    COND2_TAB[R3],MSG_B     ; SAVE ADDR OF COND 2 CURR TEXT ELT FOR FAO
          00000000'EF   00   90   014A   322          MOVB    #COND2_C,MSG_CTXT       ; SAVE CONDITION 2 CONTEXT FOR FAO
                           0151   323          MOV_VAL COND2_C,COND2_E[R3],MSG_DATA1 ; GIVE COND 2 DATA VALUE TO FAO
                     FEAC'   30   0151   324          BSBW    WRITE_MSG2              ; FORMAT AND WRITE CONDITION 2 MSG
                  14   14   91   0154   325          CMPB    #COND3_C,#NULL          ; IS CONDITION 3 NULL ?
                       03   12   0157   326          BNEQU   30$                     ; NO -- CONTINUE
                    006D   31   0159   327          BRW     FORM_CONDSX             ; YES -- SUBROUTINE IS FINISHED
                           015C   328 30$:
  00000000'EF   00000147'EF   DE   015C   329          MOVAL   COND3_T,MSG_A           ; SAVE ADDRESS OF CONDITION 3 TITLE FOR FAO
00000000'EF   00000147'EF44   D0   0167   330          MOVL    COND3_TAB[R4],MSG_B     ; SAVE ADDR OF COND 3 CURR TEXT ELT FOR FAO
          00000000'EF   14   90   0173   331          MOVB    #COND3_C,MSG_CTXT       ; SAVE CONDITION 3 CONTEXT FOR FAO
                           017A   332          MOV_VAL COND3_C,COND3_E[R4],MSG_DATA1 ; GIVE COND 3 DATA VALUE TO FAO
                     FE83'   30   017A   333          BSBW    WRITE_MSG2              ; FORMAT AND WRITE CONDITION 3 MSG
                  14   14   91   017D   334          CMPB    #COND4_C,#NULL          ; IS CONDITION 4 NULL ?
                       47   13   0180   335          BEQLU   FORM_CONDSX             ; YES -- SUBROUTINE IS FINISHED
  00000000'EF   00000148'EF   DE   0182   336          MOVAL   COND4_T,MSG_A           ; SAVE ADDRESS OF CONDITION 4 TITLE FOR FAO
00000000'EF   00000148'EF45   D0   018D   337          MOVL    COND4_TAB[R5],MSG_B     ; SAVE ADDR OF COND 4 CURR TEXT ELT FOR FAO
          00000000'EF   14   90   0199   338          MOVB    #COND4_C,MSG_CTXT       ; SAVE CONDITION 4 CONTEXT FOR FAO
                           01A0   339          MOV_VAL COND4_C,COND4_E[R5],MSG_DATA1 ; GIVE COND 4 DATA VALUE TO FAO
                     FE5D'   30   01A0   340          BSBW    WRITE_MSG2              ; FORMAT AND WRITE CONDITION 4 MSG
                  14   14   91   01A3   341          CMPB    #COND5_C,#NULL          ; IS CONDITION 5 NULL ?
                       21   13   01A6   342          BEQLU   FORM_CONDSX             ; YES -- SUBROUTINE IS FINISHED
  00000000'EF   00000149'EF   DE   01A8   343          MOVAL   COND5_T,MSG_A           ; SAVE ADDRESS OF CONDITION 5 TITLE FOR FAO
00000000'EF   00000149'EF46   D0   01B3   344          MOVL    COND5_TAB[R6],MSG_B     ; SAVE ADDR OF COND 5 CURR TEXT ELT FOR FAO
          00000000'EF   14   90   01BF   345          MOVB    #COND5_C,MSG_CTXT       ; SAVE CONDITION 5 CONTEXT FOR FAO
                           01C6   346          MOV_VAL COND5_C,COND5_E[R6],MSG_DATA1 ; GIVE COND 5 DATA VALUE TO FAO
                     FE37'   30   01C6   347          BSBW    WRITE_MSG2              ; FORMAT AND WRITE CONDITION 5 MSG
                           01C9   348 FORM_CONDSX:
                       05   01C9   349          RSB                             ; RETURN TO CALLER
```

```
01CA    351          .SBTTL  VERIFY
01CA    352 ;++
01CA    353 ; FUNCTIONAL DESCRIPTION:
01CA    354 ;
01CA    355 ;               VERIFY IS CALLED ONCE FOR EACH COMBINATION OF CONDITION
01CA    356 ; TABLE VALUES (AS DETERMINED BY THE INDEX REGISTERS R2,3,4,5,6 FOR
01CA    357 ; COND TABLES 1,2,3,4,5, RESPECTIVELY). VERIFY ESTABLISHES THE CONDITIONS
01CA    358 ; SPECIFIED BY THE COND TABLES AND ISSUES THE SUBJECT SYSTEM SERVICES
01CA    359 ; ($LCKPAG & $ULKPAG). THEN, THE SUCCESSFUL OPERATION OF THE SERVICES IS
01CA    360 ; VERIFIED BY EXAMINING THE STATUS CODES RETURNED, THE VALUES FOR RETURN ARGS
01CA    361 ; AND THE FUNCTIONALITY PERFORMED. THE EXAMINATIONS TAKE THE FORM OF
01CA    362 ; COMPARISONS AGAINST EXPECTED VALUES. ANY FAILING COMPARISON CAUSES AN
01CA    363 ; ERR_EXIT MACRO TO BE EXECUTED (EITHER DIRECTLY, OR INDIRECTLY,
01CA    364 ; THROUGH THE SS_CHECK MACRO); ERR_EXIT SETS EFLAG TO NON-ZERO,
01CA    365 ; PRINTS ERROR MESSAGES AND CAUSES AN IMMEDIATE RSB TO CALLER.
01CA    366 ; WHEN ERR_EXIT IS EXECUTED, FURTHER CALLS TO VERIFY ARE SUPPRESSED,
01CA    367 ; AND, AFTER EXECUTING CLEANUP SUBROUTINES, THE IMAGE EXITS.
01CA    368 ;
01CA    369 ; CALLING SEQUENCE:
01CA    370 ;
01CA    371 ;       BSBW VERIFY
01CA    372 ;
01CA    373 ; INPUT PARAMETERS:
01CA    374 ;
01CA    375 ;       NONE
01CA    376 ;
01CA    377 ; IMPLICIT INPUTS:
01CA    378 ;
01CA    379 ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
01CA    380 ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
01CA    381 ;       FOR X = 1,2,3,4,5 :
01CA    382 ;               CONDX_E - ADDRESS OF TABLE OF DATA VALUES FOR CONDX
01CA    383 ;                 TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
01CA    384 ;                 ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
01CA    385 ;                 FOR CONDX_E.
01CA    386 ;
01CA    387 ; OUTPUT PARAMETERS:
01CA    388 ;
01CA    389 ;       NONE
01CA    390 ;
01CA    391 ; IMPLICIT OUTPUTS:
01CA    392 ;
01CA    393 ;       VERIFY HAS NO OUTPUT. SINCE ITS PURPOSE IS TO TEST FOR ERRORS,
01CA    394 ;       IT MERELY RETURNS TO CALLER NORMALLY AFTER THE TESTS, PROVIDING
01CA    395 ;       ALL WERE SUCCESSFUL; IF AN ERROR IS DISCOVERED, RETURN IS VIA
01CA    396 ;       AN ERR_EXIT OR SS_CHECK MACRO, BOTH OF WHICH DOCUMENT DETECTED
01CA    397 ;       ERRORS.
01CA    398 ;
01CA    399 ; COMPLETION CODES:
01CA    400 ;
01CA    401 ;       EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
01CA    402 ;
01CA    403 ; SIDE EFFECTS:
01CA    404 ;
01CA    405 ;       SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
01CA    406 ;       (VIA RSB) IF ERROR ENCOUNTERED.
01CA    407 ;
```

```
                                01CA   408 ;--
                                01CA   409
                                01CA   410
                                01CA   411
                                01CA   412 VERIFY::
          00000000'EF    95     01CA   413          TSTB    CFLAG                   ; SHOULD CONDITIONS BE PRINTED ?
                  03     13     01D0   414          BEQL    5$                      ; NO -- CONTINUE
                FEFF     30     01D2   415          BSBW    FORM_CONDS              ; YES -- FMT & PRINT ALL CONDS FOR THIS T.C.
                                01D5   416 5$:
    57    00000141'EF43    B0     01D5   417          MOVW    COND2_E[R3],R7          ; GET CURRENT ACMODE SETTINGS
      00000010'EF    57    9A     01DD   418          MOVZBL  R7,ACMODE_ERG           ; ISOLATE PAGE OWNER ACMODE FOR $EXPREG
       57    57    F8 8F    78     01E4   419          ASHL    #-8,R7,R7               ; SHIFT LOCKER'S MODE TO LOW-ORDER BYTE
      0000001C'EF    57    90     01E9   420          MOVB    R7,LOCKMODE             ; ... AND REMEMBER LOCKER'S MODE
                                01F0   421 ;
                                01F0   422 ; SET UP INADR ARGUMENT IN COND 1 TABLE BY ISSUING $EXPREG
                                01F0   423 ;
                    52     D5  01F0   424          TSTL    R2                      ; FIRST COND 1 ELEMENT ??
                    03     12  01F2   425          BNEQ    10$                     ; NO -- CONTINUE
                  00A3     31  01F4   426          BRW     40$                     ; YES -- INADR ALREADY SET UP
                                01F7   427 10$:
       000000C9'EF42     7C  01F7   428          CLRQ    INADR[R2]               ; CLEAR $EXPREG RETURN AREA
         0000000C'EF     D4  01FE   429          CLRL    REGION_ERG              ; ASSUME PROGRAM REGION
      00000008'EF    01    9A  0204   430          MOVZBL  #1,PAGCNT_ERG           ; ... AND SINGLE PAGE COUNT
                02    52    D1  020B   431          CMPL    R2,#2                   ; IS IT PROGRAM REGION ??
                      07    15  020E   432          BLEQ    15$                     ; YES -- GO CHECK PAGE COUNT
      0000000C'EF    01    9A  0210   433          MOVZBL  #1,REGION_ERG           ; NO -- MUST BE CONTROL REGION
                                0217   434 15$:
                01    52    D1  0217   435          CMPL    R2,#1                   ; IS IT SINGLE PAGE ??
                      0C    13  021A   436          BEQL    20$                     ; YES -- GO ESTABLISH MODE
                03    52    D1  021C   437          CMPL    R2,#3                   ; COULD STILL BE SINGLE PAGE
                      07    13  021F   438          BEQL    20$                     ; IT IS -- GO ESTABLISH MODE
      00000008'EF    0A    9A  0221   439          MOVZBL  #10,PAGCNT_ERG          ; MULTIPLE PAGES
                                0228   440 20$:
                                0228   441          MODE    TO,30$,KRNL             ; GET KERNEL MODE, SO ACMODE ARG WORKS
                                024B   442          $EXPREG_S REGION=REGION_ERG, PAGCNT=PAGCNT_ERG, -
                                024B   443                    ACMODE=ACMODE_ERG, RETADR=INADR[R2]
                                026B   444          MODE    FROM,30$                ; BACK TO USER MODE
                                026C   445          SS_CHECK NORMAL                 ; CHECK FOR NORMAL STATUS RETURN
                                029A   446 40$:
                                029A   447 ;
                                029A   448 ; ****** SYSTEM SERVICE CALLS WHICH ARE THE SUBJECT OF THIS TEST CASE ******
                                029A   449 ;
          00    0000001C'EF    91     029A   450          CMPB    LOCKMODE,#PSL$C_KERNEL  ; IS LOCKER'S MODE KERNEL ??
                          3C    12     02A1   451          BNEQ    50$                     ; NO -- SIMPLY ISSUE IT IN USER
                                02A3   452          MODE    TO,60$,KRNL             ; YES -- ISSUE SERVICE IN KERNEL MODE
                                02C6   453          $LCKPAG_S INADR=INADR[R2], RETADR=RETADR
                                02DC   454          MODE    FROM,60$                ; BACK TO USER MODE
                          16    11     02DD   455          BRB     70$                     ; GO CHECK STATUS RETURN VALUE
                                02DF   456 50$:
                                02DF   457          $LCKPAG_S INADR=INADR[R2], RETADR=RETADR
                                02F5   458 70$:
          00000000'8F    50    D1     02F5   459          CMPL    R0,#SS$_WASCLR          ; CODE RECEIVED = CODE EXPECTED ?
                         61    13     02FC   460          BEQLU   80$                     ; YES -- CONTINUE
00000000'EF    00000000'8F    D0     02FE   461          MOVL    #SS$_WASCLR,EXPV        ; NO -- LOAD UP EXPECTED AND ...
          00000000'EF    50    D0     0309   462          MOVL    R0,RECV                 ; ... RECEIVED VALUES, THEN EXIT
                                0310   463          ERR_EXIT LONG,<INCORRECT STATUS CODE RETURNED FROM LCKPAG>
                                035F   464 80$:
```

```
                                035F    465 ;
                                035F    466 ; COMPARE INADR AGAINST RETADR (EXCEPT LOW-ORDER 9 BITS) AND,
                                035F    467 ; IF OK, GO ISSUE $ULKPAG.
                                035F    468 ;
        57   000000C9'EF42    7D  035F    469          MOVQ    INADR[R2],R7            ; GET INADR PAIR INTO REGS
         57    57    17   09   EF  0367    470          EXTZV   #9,#23,R7,R7            ; SHIFT RIGHT 9 BITS
 57  00000014'EF    17   09   ED  036C    471          CMPZV   #9,#23,RETADR,R7        ; INADR = RETADR (FIRST OF PAIR) ??
                         13   12  0375    472          BNEQU   90$                     ; NO -- ERROR
         58    58    17   09   EF  0377    473          EXTZV   #9,#23,R8,R8            ; SHIFT RIGHT 9 BITS
 58  00000018'EF    17   09   ED  037C    474          CMPZV   #9,#23,RETADR+4,R8      ; INADR = RETADR (2ND OF PAIR) ??
                         03   12  0385    475          BNEQU   90$                     ; NO -- ERROR
                   0068   31   0387    476          BRW     100$                    ; YES -- SKIP ERROR PROCESSING
                                038A    477 90$:
00000000'EF   000000C9'EF42    7D  038A    478          MOVQ    INADR[R2],EXPV          ; LOAD UP EXPECTED AND ...
  00000000'EF   00000014'EF    7D  0396    479          MOVQ    RETADR,RECV             ; ... RECEIVED VALUES, THEN EXIT
                                03A1    480          ERR_EXIT QUAD,<UNEXPECTED VALUE FROM LCKPAG FOR>,-
                                03A1    481                       < RETADR PAIR>
                                03F2    482 100$:
                                03F2    483 ;
                                03F2    484 ; SET UP TO ISSUE $ULKPAG
                                03F2    485 ;
        00   0000001C'EF    91  03F2    486          CMPB    LOCKMODE,#PSL$C_KERNEL  ; IS LOCKER'S MODE KERNEL ??
                         3C   12  03F9    487          BNEQ    120$                    ; NO -- SIMPLY ISSUE IT IN USER
                                03FB    488          MODE    TO,110$,KRNL            ; YES -- ISSUE SERVICE IN KERNEL MODE
                                041E    489          $ULKPAG_S INADR=INADR[R2], RETADR=RETADR
                                0434    490          MODE    FROM,110$               ; BACK TO USER MODE
                         16   11  0435    491          BRB     130$                    ; GO CHECK STATUS RETURN VALUE
                                0437    492 120$:
                                0437    493          $ULKPAG_S INADR=INADR[R2], RETADR=RETADR
                                044D    494 130$:
        00000000'8F    50   D1  044D    495          CMPL    R0,#SS$_WASSET          ; CODE RECEIVED = CODE EXPECTED ?
                         61   13  0454    496          BEQLU   140$                    ; YES -- CONTINUE
 00000000'EF   00000000'8F    D0  0456    497          MOVL    #SS$_WASSET,EXPV        ; NO -- LOAD UP EXPECTED AND ...
         00000000'EF    50   D0  0461    498          MOVL    R0,RECV                 ; ... RECEIVED VALUES, THEN EXIT
                                0468    499          ERR_EXIT LONG,<INCORRECT STATUS CODE RETURNED FROM ULKPAG>
                                04B7    500 140$:
                                04B7    501 ;
                                04B7    502 ; COMPARE INADR AGAINST RETADR (EXCEPT LOW-ORDER 9 BITS) AND,
                                04B7    503 ; IF OK, VERIFY ROUTINE IS COMPLETE, GO DO ANOTHER TEST CASE.
                                04B7    504 ;
        57   000000C9'EF42    7D  04B7    505          MOVQ    INADR[R2],R7            ; GET INADR PAIR INTO REGS
         57    57    17   09   EF  04BF    506          EXTZV   #9,#23,R7,R7            ; SHIFT RIGHT 9 BITS
 57  00000014'EF    17   09   ED  04C4    507          CMPZV   #9,#23,RETADR,R7        ; INADR = RETADR (FIRST OF PAIR) ??
                         13   12  04CD    508          BNEQU   150$                    ; NO -- ERROR
         58    58    17   09   EF  04CF    509          EXTZV   #9,#23,R8,R8            ; SHIFT RIGHT 9 BITS
 58  00000018'EF    17   09   ED  04D4    510          CMPZV   #9,#23,RETADR+4,R8      ; INADR = RETADR (2ND OF PAIR) ??
                         03   12  04DD    511          BNEQU   150$                    ; NO -- ERROR
                   0068   31   04DF    512          BRW     VERIFYX                 ; YES -- SKIP ERROR PROCESSING
                                04E2    513 150$:
00000000'EF   000000C9'EF42    7D  04E2    514          MOVQ    INADR[R2],EXPV          ; LOAD UP EXPECTED AND ...
  00000000'EF   00000014'EF    7D  04EE    515          MOVQ    RETADR,RECV             ; ... RECEIVED VALUES, THEN EXIT
                                04F9    516          ERR_EXIT QUAD,<UNEXPECTED VALUE FROM ULKPAG FOR>,-
                                04F9    517                       < RETADR PAIR>
                                054A    518 VERIFYX:
                         05     054A    519          RSB                             ; RETURN TO CALLER
```

```
                     054B    521          .SBTTL  VFY_CLEANUP
                     054B    522 ;++
                     054B    523 ; FUNCTIONAL DESCRIPTION:
                     054B    524 ;
                     054B    525 ;               VFY_CLEANUP EXECUTES SYSTEM SERVICES TO UNDO THE
                     054B    526 ; EFFECT OF THOSE ISSUED IN THE VERIFY SUBROUTINE. VFY_CLEANUP MUST
                     054B    527 ; ASSUME THAT VERIFY MAY NOT HAVE EXECUTED IN ITS ENTIRETY (IF AN
                     054B    528 ; ERROR IS FOUND). ALSO, VFY_CLEANUP MAY ISSUE SS_CHECK OR ERR_EXI
                     054B    529 ; ONLY AFTER PERFORMING ALL OF ITS CLEANUP OPERATIONS; THIS IS REQUIRED
                     054B    530 ; IN THE EVENT THAT VFY_CLEANUP IS CALLED DURING ERROR PROCESSING,
                     054B    531 ; WHEN PERFORMING THE REQUIRED CLEANUP IS MORE IMPORTANT THAN
                     054B    532 ; POSSIBLY DISCOVERING A SECOND ERROR.
                     054B    533 ;
                     054B    534 ; CALLING SEQUENCE:
                     054B    535 ;
                     054B    536 ;       BSBW VFY_CLEANUP
                     054B    537 ;
                     054B    538 ; INPUT PARAMETERS:
                     054B    539 ;
                     054B    540 ;       NONE
                     054B    541 ;
                     054B    542 ; IMPLICIT INPUTS:
                     054B    543 ;
                     054B    544 ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
                     054B    545 ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
                     054B    546 ;       FOR X = 1,2,3,4,5 :
                     054B    547 ;               CONDX_E - ADDRESS OF TABLE OF DATA VALUES FOR CONDX
                     054B    548 ;                 TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
                     054B    549 ;                 ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
                     054B    550 ;                 FOR CONDX_E.
                     054B    551 ;
                     054B    552 ; OUTPUT PARAMETERS:
                     054B    553 ;
                     054B    554 ;       NONE
                     054B    555 ;
                     054B    556 ; IMPLICIT OUTPUTS:
                     054B    557 ;
                     054B    558 ;       NONE
                     054B    559 ;
                     054B    560 ; COMPLETION CODES:
                     054B    561 ;
                     054B    562 ;       EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
                     054B    563 ;
                     054B    564 ; SIDE EFFECTS:
                     054B    565 ;
                     054B    566 ;       SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
                     054B    567 ;       (VIA RSB) IF ERROR ENCOUNTERED.
                     054B    568 ;
                     054B    569 ;--
                     054B    570
                     054B    571
                     054B    572
                     054B    573 VFY_CLEANUP::
   52    D5          054B    574          TSTL    R2                      ; FIRST COND 1 ELEMENT ??
   03    12          054D    575          BNEQ    10$                     ; NO -- CONTINUE
 0077    31          054F    576          BRW     VFY_CLEANUPX            ; YES -- NO PAGES TO DELETE
                     0552    577 10$:
```

```
            57  000000C9'EF42   7D  0552   578          MOVQ    INADR[R2],R7            ; AVOID INDEX MODE, USE REGS
                           57   D5  0555   579          TSTL    R7                      ; DID $EXPREG GET SOME SPACE ABOVE ??
                           03   12  0557   580          BNEQU   20$                     ; YES -- CONTINUE
                         0068   31  0559   581          BRW     VFY_CLEANUPX            ; NO -- NO PAGES TO DELETE
                                    0561   582 20$:
                                    0561   583          MODE TO,30$,KRNL                ; KRNL MODE TO DELETE ACQUIRED PAGES
                                    0584   584          $DELTVA_S INADR=INADR[R2], -
                                    0584   585                    ACMODE=ACMODE_ERG     ; ....
                                    0599   586          MODE FROM,30$                   ; BACK TO USER MODE
                                    059B   587          SS_CHECK NORMAL                 ; CHECK RETURN STATUS VALUE
                                    05C9   588 VFY_CLEANUPX:
                                 05 05C9   589          RSB                             ; RETURN TO CALLER
                                    05CA   590          .END
```

```
$$$$                    = 00000503 R      04
$$$CHARS                = 0000002C
$$$CHARS1               = 0000000D
$$$CHARS2               = 0000000B
$$$CHARS3               = 00000009
$$$CHARS4               = 00000000
$$$CHARS5               = 00000000
$$$COND_A               = 00000002
$$$STRINGS              = 00000001
$$$STRINGS2             = 00000005
$$T1                    = 00000000
$$T2                    = 00000004
ACMODE_ERG                00000010 R      03
BYTE                    = 00000001   G
CFLAG                     ********     X  04
CHMRTN                    ********     X  04
CHM_CONT                  ********     X  04
COMP_SC                   ********     X  04
COND1                     000000B7 RG     04
COND1_C                 = 00000008
COND1_CLEANUP             000000B8 RG     04
COND1_E                   000000C9 R      03
COND1_H                   00000023 RG     03
COND1_T                   0000001D R      03
COND1_TAB                 00000024 R      03
COND2                     000000B9 RG     04
COND2X                    000000CC R      04
COND2_C                 = 00000000
COND2_CLEANUP             000000CD RG     04
COND2_E                   00000141 R      03
COND2_H                   00000110 RG     03
COND2_T                   000000F1 R      03
COND2_TAB                 00000111 R      03
COND3                     000000CE RG     04
COND3_C                 = 00000014
COND3_CLEANUP             000000CF RG     04
COND3_H                   00000147 RG     03
COND3_T                   00000147 R      03
COND3_TAB                 00000147 R      03
COND4                     000000D0 RG     04
COND4_C                 = 00000014
COND4_CLEANUP             000000D1 RG     04
COND4_H                   00000148 RG     03
COND4_T                   00000148 R      03
COND4_TAB                 00000148 R      03
COND5                     000000D2 RG     04
COND5_C                 = 00000014
COND5_CLEANUP             000000D3 RG     04
COND5_H                   00000149 RG     03
COND5_T                   00000149 R      03
COND5_TAB                 00000149 R      03
CONFLICT                  ********     X  04
CTL$GL_PHD                ********     X  04
DESC                    = 00000010   G
EFLAG                     ********     X  04
EXPV                      ********     X  04
FAO_DESC                  ********     X  04
FAO_LEN                   ********     X  04
FORM_CONDS                000000D4 RG     04
FORM_CONDSX               000001C9 R      04
INADR                     000000C9 R      03
LOCKMODE                  0000001C R      03
LONG                    = 00000004   G
MOD_MSG_CODE              ********     X  04
MOD_MSG_PRINT             ********     X  04
MSGT_INP_CTL              00000019 R      02
MSG3_ERR_CTL              00000039 RG     02
MSG_A                     ********     X  04
MSG_B                     ********     X  04
MSG_CNT                   ********     X  04
MSG_DATA1                 ********     X  04
NOTARG                  = 00000000   G
NULL                    = 00000014   G
ONES                      ********     X  04
OUTPUT_MSG                ********     X  04
PAGCNT_ERG                00000008 R      03
PCV                       ********     X  04
PHD$Q_PRIVMSK           = 00000000
PRIVMASK                  00000000 R      03
PRIV_ARGS               = 00000002
PROCESS_ERR               ********     X  04
PSL$C_KERNEL            = 00000000
PSL$C_USER              = 00000003
QUAD                    = 00000008   G
RECV                      ********     X  04
REGION_ERG                0000000C R      03
REST_REGS                 ********     X  04
RETADR                    00000014 R      03
SAVE_REGS                 ********     X  04
SS$_NORMAL                ********     X  04
SS$_WASCLR                ********     X  04
SS$_WASSET                ********     X  04
SUCCESS                   ********     X  04
SYS$CMKRNL                ********    GX  04
SYS$DELTVA                ********    GX  04
SYS$EXPREG                ********    GX  04
SYS$FAO                   ********     X  04
SYS$LCKPAG                ********    GX  04
SYS$SETPRN                ********    GX  04
SYS$SETPRV                ********    GX  04
SYS$ULKPAG                ********    GX  04
TESTNUM                   ********     X  04
TEST_MOD_NAME             00000000 RG     02
TEST_MOD_NAME_D           00000009 R      02
TEST_MOD_SUCC             ********     X  04
TMD_ADDR                  ********     X  04
TM_CLEANUP                000000B3 RG     04
TM_SETUP                  00000000 RG     04
VERIFY                    000001CA RG     04
VERIFYX                   0000054A R      04
VFY_CLEANUP               0000054B RG     04
VFY_CLEANUPX              000005C9 R      04
WORD                    = 00000002   G
WRITE_MSG2                ********     X  04
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
RODATA	00000051 (81.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC LONG
RWDATA	0000014A (330.)	03 (3.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
SATSSS79	000005CA (1482.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	38	00:00:00.06	00:00:00.35
Command processing	148	00:00:00.62	00:00:02.35
Pass 1	249	00:00:06.92	00:00:15.19
Symbol table sort	0	00:00:00.54	00:00:00.69
Pass 2	120	00:00:01.90	00:00:03.20
Symbol table output	14	00:00:00.09	00:00:00.09
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	574	00:00:10.16	00:00:21.90

The working set limit was 1500 pages.
37087 bytes (73 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 345 non-local and 47 local symbols.
590 source lines were read in Pass 1, producing 24 object records in Pass 2.
37 pages of virtual memory were used to define 28 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	15
TOTALS (all libraries)	25

671 GETS were required to define 25 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:SATSSS79/OBJ=OBJ$:SATSSS79 MSRC$:SATSSS79/UPDATE=(ENH$:SATSSS79)+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