```
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
```

**FILE**ID**SATSSS60

SA
VO

```
  SSSSSSSS    AAAAAA    TTTTTTTTTT    SSSSSSSS    SSSSSSSS    SSSSSSSS    666666      000000
  SSSSSSSS    AAAAAA    TTTTTTTTTT    SSSSSSSS    SSSSSSSS    SSSSSSSS    666666      000000
SS           AA      AA      TT      SS          SS          SS          66          00      00
SS           AA      AA      TT      SS          SS          SS          66          00      00
SS           AA      AA      TT      SS          SS          SS          66          00    0000
SS           AA      AA      TT      SS          SS          SS          66          00    0000
  SSSSSS     AA      AA      TT        SSSSSS      SSSSSS      SSSSSS    66666666    00  00  00
  SSSSSS     AA      AA      TT        SSSSSS      SSSSSS      SSSSSS    66666666    00  00  00
        SS   AAAAAAAAAA      TT              SS          SS          SS  66      66  0000    00
        SS   AAAAAAAAAA      TT              SS          SS          SS  66      66  0000    00
        SS   AA      AA      TT              SS          SS          SS  66      66  00      00      ....
        SS   AA      AA      TT              SS          SS          SS  66      66  00      00      ....
SSSSSSSS     AA      AA      TT      SSSSSSSS    SSSSSSSS    SSSSSSSS      666666      000000      ....
SSSSSSSS     AA      AA      TT      SSSSSSSS    SSSSSSSS    SSSSSSSS      666666      000000      ....

LL             IIIIII      SSSSSSSS
LL             IIIIII      SSSSSSSS
LL               II      SS
LL               II      SS
LL               II      SS
LL               II      SS
LL               II        SSSSSS
LL               II        SSSSSS
LL               II              SS
LL               II              SS
LL               II              SS
LL               II              SS
LLLLLLLLLL     IIIIII    SSSSSSSS
LLLLLLLLLL     IIIIII    SSSSSSSS
```

```
0000     1          .TITLE  SATSSS60 SATS SYST SRV TESTS  TIME SERVS (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 INFOPMATION 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 SATSSS60 TO TEST SUCCESSFUL
0000    36 ; OPERATION OF THE TIME 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 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, 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 ;
                     0000     61 ; MACROS:
                     0000     62 ;
                     0000     63 ;
                     0000     64 ; EQUATED SYMBOLS:
                     0000     65 ;
00000001             0000     66 SHORT_ASC        = 1                     ; INDICATOR FOR SHORT $ASCTIM OUTPUT
00000000             0000     67 LONG_ASC         = 0                     ; INDICATOR FOR LONG $ASCTIM OUTPUT
                     0000     68 ;
                     0000     69 ; OWN STORAGE:
                     0000     70 ;
```

```
                                00000000    72          .PSECT  RODATA,RD,NOWRT,NOEXE,LONG
                                    0000    73 TEST_MOD_NAME:: STRING  C,<SATSSS60>     ; TEST MODULE NAME
                                    0009    74 TEST_MOD_NAME_D: STRING I,<SATSSS60>     ; TEST MODULE NAME DESCRIPTOR
                                    0019    75 MSG1_INP_CTL:   STRING  I,<  SSTIM!4ZW: CONDITIONS:>
                                    0039    76                                          ; FAO CTL STRING FOR MSG1 IN SUCCOMMON.MAR
                                    0039    77 MSG3_ERR_CTL::  STRING  I,< *SSTIM!4ZW: !AS>
                                    0051    78                                          ; FAO CTL STRING FOR MSG3 IN SUCCOMMON.MAR
                                    0051    79 ;
                                    0051    80 ; TABLE OF TIMBUF STRINGS FOR COND1 TABLE
                                    0051    81 ;
20 33 37 39 31 2D 43 45 44 2D 35 32 0051    82 C1_ELT1:        .ASCII  \25-DEC-1973 21:46:00.05\
   35 30 2E 30 30 3A 36 34 3A 31 32 005D
20 31 30 30 32 2D 47 55 41 2D 31 33 0068    83 C1_ELT2:        .ASCII  \31-AUG-2001  01:32:29.94\
34 39 2E 39 32 3A 32 33 3A 31 30 20 0074
3A 38 31 20 20 20 33 35 37 35 20 20 0080    84 C1_ELT3:        .ASCII  \  5753   18:56:01.13 \
         20 33 31 2E 31 30 3A 36 35 008C
                        2D 2D       0095    85 C1_ELT4:        .ASCII  \--\
                                    0097    86 ;
                                    0097    87 STD_TIMADR_BTM:                          ; TABLE OF STANDARD TIMADR ARGUMENT
                                    0097    88                                          ; ... VALUES OUTPUT BY $BINTIM
                                    0097    89 ;
                            A2C67D20 0097    90                 .LONG   ^XA2C67D20       ; LOW-ORDER LONGWORD ...  1ST COND 1 ELEMENT
                            00810CAC 009B    91                 .LONG   ^X00810CAC       ; HIGH-ORDER LONGWORD ... 1ST COND 1 ELEMENT
                            4347A340 009F    92                 .LONG   ^X4347A340       ; LOW-ORDER LONGWORD ...  2ND COND 1 ELEMENT
                            00A01545 00A3    93                 .LONG   ^X00A01545       ; HIGH-ORDER LONGWORD ... 2ND COND 1 ELEMENT
                            2A513360 00A7    94                 .LONG   ^X2A513360       ; LOW-ORDER LONGWORD ...  3RD COND 1 ELEMENT
                            FFEE56A7 00AB    95                 .LONG   ^XFFEE56A7       ; HIGH-ORDER LONGWORD ... 3RD COND 1 ELEMENT
                                     00AF    96 ;
                                     00AF    97 STD_TIMBUF_ATMS:                         ; TABLE OF STANDARD TIMBUF ARGUMENT
                                     00AF    98                                          ; ... VALUES OUTPUT BY $ASCTIM (SHORT)
                                     00AF    99                                          ; NOTE -- WILL SKIP CHECKING HUNDREDTHS
                                     00AF   100                                          ; ... OF A SECOND DUE TO ROUNDING ERROR
                                     00AF   101 ;
                    000000EB'0000000A 00AF   102                 .LONG   10,S_T_A_DATA+12 ; STRING DESCRIPTOR FOR 1ST COND 1 ELEMENT
                    00000101'0000000A 00B7   103                 .LONG   10,S_T_A_DATA+34 ; STRING DESCRIPTOR FOR 2ND COND 1 ELEMENT
                    00000110'0000000A 00BF   104                 .LONG   10,S_T_A_DATA+49 ; STRING DESCRIPTOR FOR 3RD COND 1 ELEMENT
                                      00C7   105 ;
                                      00C7   106 STD_TIMBUF_ATML:                        ; TABLE OF STANDARD TIMBUF ARGUMENT
                                      00C7   107                                         ; ... VALUES OUTPUT BY $ASCTIM (LONG)
                                      00C7   108                                         ; NOTE -- WILL SKIP CHECKING HUNDREDTHS
                                      00C7   109                                         ; ... OF A SECOND DUE TO ROUNDING ERROR
                                      00C7   110 ;
                    000000DF'00000016 00C7   111                 .LONG   22,S_T_A_DATA    ; STRING DESCRIPTOR FOR 1ST COND 1 ELEMENT
                    000000F5'00000016 00CF   112                 .LONG   22,S_T_A_DATA+22 ; STRING DESCRIPTOR FOR 2ND COND 1 ELEMENT
                    0000010B'0000000F 00D7   113                 .LONG   15,S_T_A_DATA+44 ; STRING DESCRIPTOR FOR 3RD COND 1 ELEMENT
                                      00DF   114 ;
                                      00DF   115 S_T_A_DATA:                             ; TABLE OF STRINGS FOR THE STD_TIMBUF_ATM
                                      00DF   116                                         ; ... DESCRIPTORS ABOVE
                                      00DF   117 ;
20 33 37 39 31 2D 43 45 44 2D 35 32 00DF   118                 .ASCII  \25-DEC-1973 21:46:00.0\
   30 2E 30 30 3A 36 34 3A 31 32 00EB
20 31 30 30 32 2D 47 55 41 2D 31 33 00F5   119                 .ASCII  \31-AUG-2001 01:32:29.9\
   39 2E 39 32 3A 32 33 3A 31 30 0101
30 3A 36 35 3A 38 31 20 33 35 37 35 010B   120                 .ASCII  \5753 18:56:01.1\
                     31 2E 31       0117
                                    011A   121 ;
                                    011A   122 STD_TIMBUF_NMT:                          ; TABLE OF ADDRESSES FOR STANDARD TIMBUF
```

```
                                011A   123                             ; ... ARGUMENT VALUES OUTPUT BY $NUMTIM
                                011A   124 ;
                00000126'       011A   125                 .ADDRESS S_T_N_DATA      ; ADDRESS OF BUFFER FOR 1ST COND 1 ELEMENT
                00000134'       011E   126                 .ADDRESS S_T_N_DATA+14   ; ADDRESS OF BUFFER FOR 2ND COND 1 ELEMENT
                00000142'       0122   127                 .ADDRESS S_T_N_DATA+28   ; ADDRESS OF BUFFER FOR 3RD COND 1 ELEMENT
                                0126   128 ;
                                0126   129 S_T_N_DATA:                              ; TABLE OF 7-WORD BUFFERS POINTED TO BY
                                0126   130                             ; ... THE STD_TIMBUF_NMT ADDRESSES ABOVE
                                0126   131 ;
0005 0000 002E 0015 0019 000C 07B5  0126   132                 .WORD   1973,12,25,21,46,0,5
005E 001D 0020 0001 001F 0008 07D1  0134   133                 .WORD   2001,8,31,1,32,29,94
000D 0001 0038 0012 1679 0000 0000  0142   134                 .WORD   0,0,5753,18,56,1,13
```

```
                00000000   136          .PSECT  RWDATA,RD,WRT,NOEXE,LONG
00000008  0000  137 PRIVMASK:          .BLKQ   1               ; ADDR OF PRIVILEGE MASK (IN PHD)
00000010  0008  138 TIMADR_BTM:        .BLKQ   1               ; TIMADR ARGUMENT FOR $BINTIM
          0010  139 TIMBUF_ATM:        STRING  0,24            ; TIMBUF ARGUMENT FOR $ASCTIM
00000032  0030  140 TIMLEN_ATM:        .BLKW   1               ; TIMLEN ARGUMENT FOR $ASCTIM
00000040  0032  141 TIMBUF_NMT:        .BLKW   7               ; TIMBUF ARGUMENT FOR $NUMTIM
```

```
                     0040   143          .SBTTL  CONDITION TABLES
                     0040   144 ;
                     0040   145 ;       ***** CONDITION TABLES FOR TIME SYSTEM SERVICES *****
                     0040   146 ;
                     0040   147          COND    1,DESC,<TIMBUF>,-
                     0040   148                    <DATE IN PAST (INPUT TO $BINTIM)>,-
                     0040   149                    <DATE IN FUTURE (INPUT TO $BINTIM)>,-
                     0040   150                    <DELTA TIME INTERVAL (INPUT TO $BINTIM)>,-
                     0040   151                    <CURRENT TIME (INPUT TO $BINTIM)>,-
                     0040   152                    <CURRENT TIME (FROM $GETTIM)>,-
                     0040   153
           00000017  0101   154                      .LONG       23       ; LENGTH OF STRING
           00000051' 0105   155                      .ADDRESS    C1_ELT1  ; ADDRESS OF STRING
           00000018  0109   156                      .LONG       24       ; LENGTH OF STRING
           00000068' 010D   157                      .ADDRESS    C1_ELT2  ; ADDRESS OF STRING
           00000015  0111   158                      .LONG       21       ; LENGTH OF STRING
           00000080' 0115   159                      .ADDRESS    C1_ELT3  ; ADDRESS OF STRING
           00000002  0119   160                      .LONG       2        ; LENGTH OF STRING
           00000095' 011D   161                      .ADDRESS    C1_ELT4  ; ADDRESS OF STRING
  00000000 00000000  0121   162                      .LONG       0,0      ; NO DESCRIPTOR FOR THIS TABLE ELEMENT
                     0129   163 ;
                     0129   164          COND    2,LONG,<CVTFLG>,-
                     0129   165                    <SHORT FORM OF $ASCTIM OUTPUT>,-
                     0129   166                    <LONG FORM OF $ASCTIM OUTPUT>,-
                     0129   167
           00000001  0172   168                      .LONG       SHORT_ASC ; THE SHORT AND ...
           00000000  0176   169                      .LONG       LONG_ASC  ; ... LONG OF IT
                     017A   170 ;
                     017A   171          COND    3,NULL
                     017B   172
                     017B   173          COND    4,NULL
                     017C   174
                     017C   175          COND    5,NULL
                     017D   176
  00000000           177          .PSECT  SATSSS60,RD,WRT,EXE
```

```
                                0000   179          .SBTTL  TM_SETUP, TM_CLEANUP
                                0000   180 ;++
                                0000   181 ; FUNCTIONAL DESCRIPTION:
                                0000   182 ;
                                0000   183 ;                TM_SETUP AND TM_CLEANUP ARE CALLED TO PERFORM
                                0000   184 ; REQUIRED HOUSEKEEPING AT THE BEGINNING AND END, RESPECTIVELY, OF
                                0000   185 ; TEST MODULE EXECUTION.
                                0000   186 ;
                                0000   187 ; CALLING SEQUENCE:
                                0000   188 ;
                                0000   189 ;       BSBW TM_SETUP   BSBW TM_CLEANUP
                                0000   190 ;
                                0000   191 ; INPUT PARAMETERS:
                                0000   192 ;
                                0000   193 ;       NONE
                                0000   194 ;
                                0000   195 ; IMPLICIT INPUTS:
                                0000   196 ;
                                0000   197 ;       NONE
                                0000   198 ;
                                0000   199 ; OUTPUT PARAMETERS:
                                0000   200 ;
                                0000   201 ;       NONE
                                0000   202 ;
                                0000   203 ; IMPLICIT OUTPUTS:
                                0000   204 ;
                                0000   205 ;       TM_SETUP:  COND TABLE INDEX REGISTERS (R2,3,4,5,6) CLEARED;
                                0000   206 ;                  ALL PRIVILEGES ACQUIRED.
                                0000   207 ;
                                0000   208 ; COMPLETION CODES:
                                0000   209 ;
                                0000   210 ;       EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
                                0000   211 ;
                                0000   212 ; SIDE EFFECTS:
                                0000   213 ;
                                0000   214 ;       SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
                                0000   215 ;       (VIA RSB) IF ERROR ENCOUNTERED.
                                0000   216 ;
                                0000   217 ;--
                                0000   218
                                0000   219
                                0000   220
                                0000   221 TM_SETUP::
                          52    D4    0000   222          CLRL    R2                      ; INITIALIZE
                          53    D4    0002   223          CLRL    R3                      ; .. CONDITION
                          54    D4    0004   224          CLRL    R4                      ; .... TABLE
                          55    D4    0006   225          CLRL    R5                      ; ...... INDEX
                          56    D4    0008   226          CLRL    R6                      ; ........ REGISTERS
                      FFF3'     30    000A   227          BSBW    MOD_MSG_PRINT           ; PRINT TEST MODULE BEGIN MSG
00000000'EF     00000000'EF     DE    000D   228          MOVAL   TEST_MOD_SUCC,TMD_ADDR  ; ASSUME END MSG WILL SHOW SUCCESS
    03    00    00000000'8F     F0    0018   229          INSV    #SUCCESS,#0,#3,MOD_MSG_CODE ; ADJUST STATUS CODE FOR SUCCESS
              00000000'EF           0020
                                    0025   230          MODE    TO,5$,KRNL              ; KERNEL MODE TO ACCESS PHD
          59  00000000'9F     D0    0048   231          MOVL    @#CTL$GL_PHD,R9         ; GET PROCESS HEADER ADDRESS
          00000000'EF   69    DE    004F   232          MOVAL   PHD$Q_PRIVMSK(R9),PRIVMASK ; GET PRIV MASK ADDRESS
                                    0056   233          MODE    FROM,5$ ; BACK TO USER MODE
                                    0057   234          PRIV    ADD,ALL                 ; GET ALL PRIVILEGES
```

```
                     0077   235          $SETPRN_S TEST_MOD_NAME_D       ; SET PROCESS NAME
                     0084   236          SS_CHECK  NORMAL                ; CHECK STATUS CODE RETURNED FROM SETPRN
                  05 00B2   237          RSB                             ; RETURN TO MAIN ROUTINE
                     00B3   238 TM_CLEANUP::
              FF4A'  30 00B3   239          BSBW    MOD_MSG_PRINT           ; PRINT TEST MODULE END MSG
                  05 00B6   240          RSB                             ; RETURN TO MAIN ROUTINE
```

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


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

```
                00BB   299 COND3::
 05             00BB   300          RSB                             ; RETURN TO MAIN ROUTINE
                00BC   301 COND3_CLEANUP::
 05             00BC   302          RSB                             ; RETURN TO MAIN ROUTINE
                00BD   303 COND4::
 05             00BD   304          RSB                             ; RETURN TO MAIN ROUTINE
                00BE   305 COND4_CLEANUP::
 05             00BE   306          RSB                             ; RETURN TO MAIN ROUTINE
                00BF   307 COND5::
 05             00BF   308          RSB                             ; RETURN TO MAIN ROUTINE
                00C0   309 COND5_CLEANUP::
 05             00C0   310          RSB                             ; RETURN TO MAIN ROUTINE
```

```
                                00C1   312          .SBTTL  FORM_CONDS
                                00C1   313 ;++
                                00C1   314 ; FUNCTIONAL DESCRIPTION:
                                00C1   315 ;
                                00C1   316 ;                FORM_CONDS FORMATS AND PRINTS INFORMATION ABOUT
                                00C1   317 ;  THE CURRENT ELEMENT IN EACH OF THE CONDITION TABLES.
                                00C1   318 ;
                                00C1   319 ; CALLING SEQUENCE:
                                00C1   320 ;
                                00C1   321 ;       BSBW FORM_CONDS
                                00C1   322 ;
                                00C1   323 ; INPUT PARAMETERS:
                                00C1   324 ;
                                00C1   325 ;       NONE
                                00C1   326 ;
                                00C1   327 ; IMPLICIT INPUTS:
                                00C1   328 ;
                                00C1   329 ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
                                00C1   330 ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
                                00C1   331 ;       FOR X = 1,2,3,4,5 :
                                00C1   332 ;               CONDX_T - TITLE TEXT FOR CONDX TABLE
                                00C1   333 ;               CONDX_TAB - ELEMENT TEXT FOR CONDX TABLE
                                00C1   334 ;               CONDX_C - CONTEXT OF THE CONDX TABLE
                                00C1   335 ;               CONDX_E - DATA ELEMENTS OF THE CONDX TABLE
                                00C1   336 ;
                                00C1   337 ; OUTPUT PARAMETERS:
                                00C1   338 ;
                                00C1   339 ;       NONE
                                00C1   340 ;
                                00C1   341 ; IMPLICIT OUTPUTS:
                                00C1   342 ;
                                00C1   343 ;       NONE
                                00C1   344 ;
                                00C1   345 ; COMPLETION CODES:
                                00C1   346 ;
                                00C1   347 ;       NONE
                                00C1   348 ;
                                00C1   349 ; SIDE EFFECTS:
                                00C1   350 ;
                                00C1   351 ;       NONE
                                00C1   352 ;
                                00C1   353 ;--
                                00C1   354
                                00C1   355
                                00C1   356
                                00C1   357 FORM_CONDS::
                                00C1   358          $FAO_S  MSG1_INP_CTL,FAO_LEN,FAO_DESC,TESTNUM
                                00E0   359                                          ; FORMAT CONDITIONS HEADER MSG
                    FF1D'   30  00E0   360          BSBW    OUTPUT_MSG              ; ... AND PRINT IT
                 14   10   91  00E3   361          CMPB    #COND1_C,#NULL          ; IS CONDITION 1 NULL ?
                      03   12  00E6   362          BNEQU   10$                     ; NO -- CONTINUE
                    00D7   31  00E8   363          BRW     FORM_CONDSX             ; YES -- SUBROUTINE IS FINISHED
                               00EB   364 10$:
  00000000'EF   00000040'EF   DE  00EB   365          MOVAL   COND1_T,MSG_A           ; SAVE ADDRESS OF CONDITION 1 TITLE FOR FAO
00000000'EF   00000048'EF42   D0  00F6   366          MOVL    COND1_TAB[R2],MSG_B     ; SAVE ADDR OF COND 1 CURR TEXT ELT FOR FAO
          00000000'EF   10    90  0102   367          MOVB    #COND1_C,MSG_CTXT       ; SAVE CONDITION 1 CONTEXT FOR FAO
                                0109   368          MOV_VAL COND1_E[R2],MSG_DATA1 ; GIVE COND 1 DATA VALUE TO FAO
```

```
                     FEE8'  30   0115   369          BSBW    WRITE_MSG2                ; FORMAT AND WRITE CONDITION 1 MSG
                 14    04   91   0118   370          CMPB    #COND2_C,#NULL            ; IS CONDITION 2 NULL ?
                       03   12   011B   371          BNEQU   20$                       ; NO -- CONTINUE
                    00A2    31   011D   372          BRW     FORM_CONDSX               ; YES -- SUBROUTINE IS FINISHED
                            0120   373 20$:
  00000000'EF   00000129'EF   DE   0120   374          MOVAL   COND2_T,MSG_A             ; SAVE ADDRESS OF CONDITION 2 TITLE FOR FAO
00000000'EF   00000131'EF43   D0   012B   375          MOVL    COND2_TAB[R3],MSG_B       ; SAVE ADDR OF COND 2 CURR TEXT ELT FOR FAO
          00000000'EF    04   90   0137   376          MOVB    #COND2_C,MSG_CTXT         ; SAVE CONDITION 2 CONTEXT FOR FAO
                            013E   377          MOV_VAL COND2_C,COND2_E[R3],MSG_DATA1 ; GIVE COND 2 DATA VALUE TO FAO
                     FEB3'  30   014A   378          BSBW    WRITE_MSG2                ; FORMAT AND WRITE CONDITION 2 MSG
                 14    14   91   014D   379          CMPB    #COND3_C,#NULL            ; IS CONDITION 3 NULL ?
                       03   12   0150   380          BNEQU   30$                       ; NO -- CONTINUE
                    006D    31   0152   381          BRW     FORM_CONDSX               ; YES -- SUBROUTINE IS FINISHED
                            0155   382 30$:
  00000000'EF   0000017A'EF   DE   0155   383          MOVAL   COND3_T,MSG_A             ; SAVE ADDRESS OF CONDITION 3 TITLE FOR FAO
00000000'EF   0000017A'EF44   D0   0160   384          MOVL    COND3_TAB[R4],MSG_B       ; SAVE ADDR OF COND 3 CURR TEXT ELT FOR FAO
          00000000'EF    14   90   016C   385          MOVB    #COND3_C,MSG_CTXT         ; SAVE CONDITION 3 CONTEXT FOR FAO
                            0173   386          MOV_VAL COND3_C,COND3_E[R4],MSG_DATA1 ; GIVE COND 3 DATA VALUE TO FAO
                     FE8A'  30   0173   387          BSBW    WRITE_MSG2                ; FORMAT AND WRITE CONDITION 3 MSG
                 14    14   91   0176   388          CMPB    #COND4_C,#NULL            ; IS CONDITION 4 NULL ?
                       47   13   0179   389          BEQLU   FORM_CONDSX               ; YES -- SUBROUTINE IS FINISHED
  00000000'EF   0000017B'EF   DE   017B   390          MOVAL   COND4_T,MSG_A             ; SAVE ADDRESS OF CONDITION 4 TITLE FOR FAO
00000000'EF   0000017B'EF45   D0   0186   391          MOVL    COND4_TAB[R5],MSG_B       ; SAVE ADDR OF COND 4 CURR TEXT ELT FOR FAO
          00000000'EF    14   90   0192   392          MOVB    #COND4_C,MSG_CTXT         ; SAVE CONDITION 4 CONTEXT FOR FAO
                            0199   393          MOV_VAL COND4_C,COND4_E[R5],MSG_DATA1 ; GIVE COND 4 DATA VALUE TO FAO
                     FE64'  30   0199   394          BSBW    WRITE_MSG2                ; FORMAT AND WRITE CONDITION 4 MSG
                 14    14   91   019C   395          CMPB    #COND5_C,#NULL            ; IS CONDITION 5 NULL ?
                       21   13   019F   396          BEQLU   FORM_CONDSX               ; YES -- SUBROUTINE IS FINISHED
  00000000'EF   0000017C'EF   DE   01A1   397          MOVAL   COND5_T,MSG_A             ; SAVE ADDRESS OF CONDITION 5 TITLE FOR FAO
00000000'EF   0000017C'EF46   D0   01AC   398          MOVL    COND5_TAB[R6],MSG_B       ; SAVE ADDR OF COND 5 CURR TEXT ELT FOR FAO
          00000000'EF    14   90   01B8   399          MOVB    #COND5_C,MSG_CTXT         ; SAVE CONDITION 5 CONTEXT FOR FAO
                            01BF   400          MOV_VAL COND5_C,COND5_E[R6],MSG_DATA1 ; GIVE COND 5 DATA VALUE TO FAO
                     FE3E'  30   01BF   401          BSBW    WRITE_MSG2                ; FORMAT AND WRITE CONDITION 5 MSG
                            01C2   402 FORM_CONDSX:
                       05   01C2   403          RSB                               ; RETURN TO CALLER
```

```
01C3    405          .SBTTL  VERIFY
01C3    406 ;++
01C3    407 ; FUNCTIONAL DESCRIPTION:
01C3    408 ;
01C3    409 ;                   VERIFY IS CALLED ONCE FOR EACH COMBINATION OF CONDITION
01C3    410 ; TABLE VALUES (AS DETERMINED BY THE INDEX REGISTERS R2,3,4,5,6 FOR
01C3    411 ; COND TABLES 1,2,3,4,5, RESPECTIVELY). VERIFY ESTABLISHES THE CONDITIONS
01C3    412 ; SPECIFIED BY THE COND TABLES AND ISSUES THE SUBJECT SYSTEM SERVICE.
01C3    413 ;  THEN, THE SUCCESSFUL OPERATION OF THE SERVICE IS VERIFIED
01C3    414 ; BY EXAMINING THE STATUS CODE RETURNED, THE VALUES FOR RETURN ARGUMENTS
01C3    415 ; AND THE FUNCTIONALITY PERFORMED. THE EXAMINATIONS TAKE THE FORM OF
01C3    416 ; COMPARISONS AGAINST EXPECTED VALUES. ANY FAILING COMPARISON CAUSES AN
01C3    417 ; ERR_EXIT MACRO TO BE EXECUTED (EITHER DIRECTLY, OR INDIRECTLY,
01C3    418 ; THROUGH THE SS_CHECK MACRO); ERR_EXIT SETS EFLAG TO NON-ZERO,
01C3    419 ; PRINTS ERROR MESSAGES AND CAUSES AN IMMEDIATE RSB TO CALLER.
01C3    420 ; WHEN ERR_EXIT IS EXECUTED, FURTHER CALLS TO VERIFY ARE SUPPRESSED,
01C3    421 ; AND, AFTER EXECUTING CLEANUP SUBROUTINES, THE IMAGE EXITS.
01C3    422 ;
01C3    423 ; CALLING SEQUENCE:
01C3    424 ;
01C3    425 ;       BSBW VERIFY
01C3    426 ;
01C3    427 ; INPUT PARAMETERS:
01C3    428 ;
01C3    429 ;       NONE
01C3    430 ;
01C3    431 ; IMPLICIT INPUTS:
01C3    432 ;
01C3    433 ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
01C3    434 ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
01C3    435 ;       FOR X = 1,2,3,4,5 :
01C3    436 ;               CONDX_E - ADDRESS OF TABLE OF DATA VALUES FOR CONDX
01C3    437 ;                 TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
01C3    438 ;                 ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
01C3    439 ;                 FOR CONDX_E.
01C3    440 ;
01C3    441 ; OUTPUT PARAMETERS:
01C3    442 ;
01C3    443 ;       NONE
01C3    444 ;
01C3    445 ; IMPLICIT OUTPUTS:
01C3    446 ;
01C3    447 ;       VERIFY HAS NO OUTPUT. SINCE ITS PURPOSE IS TO TEST FOR ERRORS,
01C3    448 ;       IT MERELY RETURNS TO CALLER NORMALLY AFTER THE TESTS, PROVIDING
01C3    449 ;       ALL WERE SUCCESSFUL; IF AN ERROR IS DISCOVERED, RETURN IS VIA
01C3    450 ;       AN ERR_EXIT OR SS_CHECK MACRO, BOTH OF WHICH DOCUMENT DETECTED
01C3    451 ;       ERRORS.
01C3    452 ;
01C3    453 ; COMPLETION CODES:
01C3    454 ;
01C3    455 ;       EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
01C3    456 ;
01C3    457 ; SIDE EFFECTS:
01C3    458 ;
01C3    459 ;       SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
01C3    460 ;       (VIA RSB) IF ERROR ENCOUNTERED.
01C3    461 ;
```

```
                                01C3    462 ;--
                                01C3    463
                                01C3    464
                                01C3    465
                                01C3    466 VERIFY::
            00000000'EF    95   01C3    467          TSTB    CFLAG                   ; SHOULD CONDITIONS BE PRINTED ?
                           03   13   01C9    468          BEQL    5$                      ; NO -- CONTINUE
                      FEF3      30   01CB    469          BSBW    FORM_CONDS              ; YES -- FMT & PRINT ALL CONDS FOR THIS T.C.
                                01CE    470 5$:
                        04 52   D1   01CE    471          CMPL    R2,#4                   ; 5TH COND 1 ELEMENT ?
                           03   13   01D1    472          BEQL    8$                      ; YES -- ISSUE $GETTIM INSTEAD OF $BINTIM
                       007A     31   01D3    473          BRW     15$                     ; NO -- GO ISSUE $BINTIM
                                01D6    474 8$:
                                01D6    475 ;
                                01D6    476 ; ****** SYSTEM SERVICE CALL WHICH IS THE SUBJECT OF THIS TEST CASE ******
                                01D6    477 ;
                                01D6    478          $GETTIM_S TIMADR=TIMADR_BTM
            00000000'8F    50   D1   01E3    479          CMPL    R0,#SS$_NORMAL          ; CODE RECEIVED = CODE EXPECTED ?
                           03   12   01EA    480          BNEQU   12$                     ; NO -- PROCESS ERROR
                       016E     31   01EC    481          BRW     30$                     ; YES -- CONTINUE WITH $ASCTIM
                                01EF    482 12$:
    00000000'EF    00000000'8F    D0   01EF    483          MOVL    #SS$_NORMAL,EXPV        ; NO -- LOAD UP EXPECTED AND ...
            00000000'EF    50   D0   01FA    484          MOVL    R0,RECV                 ; ... RECEIVED VALUES, THEN EXIT
                                0201    485          ERR_EXIT LONG,<INCORRECT STATUS CODE RETURNED FROM GETTIM>
                                0250    486 15$:
                                0250    487 ;
                                0250    488 ; ****** SYSTEM SERVICE CALL WHICH IS THE SUBJECT OF THIS TEST CASE ******
                                0250    489 ;
                                0250    490          $BINTIM_S TIMBUF=TIMBUF[R2], -
                                0250    491                    TIMADR=TIMADR_BTM
            00000000'8F    50   D1   0264    492          CMPL    R0,#SS$_NORMAL          ; CODE RECEIVED = CODE EXPECTED ?
                           61   13   026B    493          BEQLU   20$                     ; YES -- CONTINUE
    00000000'EF    00000000'8F    D0   026D    494          MOVL    #SS$_NORMAL,EXPV        ; NO -- LOAD UP EXPECTED AND ...
            00000000'EF    50   D0   0278    495          MOVL    R0,RECV                 ; ... RECEIVED VALUES, THEN EXIT
                                027F    496          ERR_EXIT LONG,<INCORRECT STATUS CODE RETURNED FROM BINTIM>
                                02CE    497 20$:
                        02 52   D1   02CE    498          CMPL    R2,#2                   ; LOOKING AT CURRENT TIME ?
                           03   15   02D1    499          BLEQ    25$                     ; NO -- COMPARE AGAINST STANDARDS
                       0087     31   02D3    500          BRW     30$                     ; YES -- SKIP COMPARE
                                02D6    501 25$:
        57 00000097'EF42    7E   02D6    502          MOVAQ   STD_TIMADR_BTM[R2],R7   ; AVOID INDEX MODE IN CMPC
                   0000'8F  BB   02DE    503          PUSHR   #CMPC_SAV               ; SAVE SOME REGS USED BY CMPC (R2 AND R3)
     67 00000008'EF    08   29   02E2    504          CMPC    #8,TIMADR_BTM,(R7)      ; DID $BINTIM PRODUCE EXPECTED RESULT ?
                   0000'8F  BA   02EA    505          POPR    #CMPC_SAV               ; RESTORE SOME REGS USED BY CMPC (R2 AND R3)
                           6D   13   02EE    506          BEQLU   30$                     ; YES -- CONTINUE
00000000'EF    00000097'EF42    7D   02F0    507          MOVQ    STD_TIMADR_BTM[R2],EXPV ; NO -- LOAD UP EXPECTED AND
  00000000'EF    00000008'EF    7D   02FC    508          MOVQ    TIMADR_BTM,RECV         ; ... RECEIVED VALUES, THEN EXIT
                                0307    509          ERR_EXIT QUAD,<UNEXPECTED VALUE FOR TIMADR ARGUMENT >, -
                                0307    510                       <FROM $BINTIM>
                                035D    511 30$:
        00 00000172'EF43    D1   035D    512          CMPL    CVTFLG[R3],#LONG_ASC    ; LONG $ASCTIM OUTPUT INDICATED ?
                           11   13   0365    513          BEQL    33$                     ; YES -- GO GET STD TABLE OF LONG VALUES
        59 000000AF'EF42    7E   0367    514          MOVAQ   STD_TIMBUF_ATMS[R2],R9  ; NO -- GET STD TABLE OF SHORT TIMBUF VALUES
                                036F    515                                          ; (R9 USED LATER FOR CMPC INSTR)
          00000010'EF    0B   D0   036F    516          MOVL    #11,TIMBUF_ATM          ; ... AND USE A SHORT BUFFER
                           0F   11   0376    517          BRB     37$                     ; GO ISSUE $ASCTIM
                                0378    518 33$:
```

```
                59   000000C7'EF42    7E  0378    519          MOVAQ   STD_TIMBUF_ATML[R2],R9  ; GET STD TABLE OF LONG TIMBUF VALUES
                  00000010'EF   18    D0  0380    520          MOVL    #24,TIMBUF_ATM          ; ... AND USE A LONG BUFFER
                                      0387    521 37$:
                                      0387    522 ;
                                      0387    523 ; ****** SYSTEM SERVICE CALL WHICH IS THE SUBJECT OF THIS TEST CASE ******
                                      0387    524 ;
                                      0387    525          $ASCTIM_S TIMLEN=TIMLEN_ATM,    -
                                      0387    526                    TIMBUF=TIMBUF_ATM,    -
                                      0387    527                    TIMADR=TIMADR_BTM,    -
                                      0387    528                    CVTFLG=CVTFLG[R3]
                    00000000'8F   50  D1  03A7    529          CMPL    R0,#SS$_NORMAL          ; CODE RECEIVED = CODE EXPECTED ?
                                  61  13  03AE    530          BEQLU   40$                     ; YES -- CONTINUE
        00000000'EF   00000000'8F     D0  03B0    531          MOVL    #SS$_NORMAL,EXPV        ; NO -- LOAD UP EXPECTED AND ...
                    00000000'EF   50  D0  03BB    532          MOVL    R0,RECV                 ; ... RECEIVED VALUES, THEN EXIT
                                      03C2    533          ERR_EXIT LONG,<INCORRECT STATUS CODE RETURNED FROM ASCTIM>
                                      0411    534 40$:
                            02    52  D1  0411    535          CMPL    R2,#2                   ; LOOKING AT CURRENT TIME ?
                                  03  15  0414    536          BLEQ    45$                     ; NO -- COMPARE AGAINST STANDARDS
                               007B   31  0416    537          BRW     50$                     ; YES -- SKIP COMPARE
                                      0419    538 45$:
                              0000'8F BB  0419    539          PUSHR   #CMPC_SAV               ; SAVE R2 & R3 BEFORE CMPC
         00000014'FF    04 B9    69  29  041D    540          CMPC    (R9),@4(R9),@TIMBUF_ATM+4 ; DID $ASCTIM PRODUCE EXPECTED RESULT ?
                              0000'8F BA  0426    541          POPR    #CMPC_SAV               ; RESTORE R2 & R3 AFTER CMPC
                                  68  13  042A    542          BEQLU   50$                     ; YES -- GO ON TO $NUMTIM
                    00000000'EF   69  7D  042C    543          MOVQ    (R9),EXPV               ; NO -- LOAD UP EXPECTED AND
        00000000'EF   00000010'EF   7D  0433    544          MOVQ    TIMBUF_ATM,RECV         ; ... RECEIVED VALUES, THEN EXIT
                                      043E    545          ERR_EXIT DESC,<UNEXPECTED VALUE FOR TIMBUF ARGUMENT >, -
                                      043E    546                       <FROM $ASCTIM>
                                      0494    547 50$:
                                      0494    548 ;
                                      0494    549 ; ****** SYSTEM SERVICE CALL WHICH IS THE SUBJECT OF THIS TEST CASE ******
                                      0494    550 ;
                                      0494    551          $NUMTIM_S TIMBUF=TIMBUF_NMT, -
                                      0494    552                    TIMADR=TIMADR_BTM
                    00000000'8F   50  D1  04A7    553          CMPL    R0,#SS$_NORMAL          ; CODE RECEIVED = CODE EXPECTED ?
                                  61  13  04AE    554          BEQLU   60$                     ; YES -- CONTINUE
        00000000'EF   00000000'8F     D0  04B0    555          MOVL    #SS$_NORMAL,EXPV        ; NO -- LOAD UP EXPECTED AND ...
                    00000000'EF   50  D0  04BB    556          MOVL    R0,RECV                 ; ... RECEIVED VALUES, THEN EXIT
                                      04C2    557          ERR_EXIT LONG,<INCORRECT STATUS CODE RETURNED FROM NUMTIM>
                                      0511    558 60$:
                            02    52  D1  0511    559          CMPL    R2,#2                   ; LOOKING AT CURRENT TIME ?
                                  03  15  0514    560          BLEQ    65$                     ; NO -- COMPARE AGAINST STANDARDS
                               017C   31  0516    561          BRW     VERIFYX                 ; YES -- SKIP COMPARE
                                      0519    562 65$:
             57   0000011A'EF42    D0  0519    563          MOVL    STD_TIMBUF_NMT[R2],R7   ; GET READY FOR ...
               58   00000032'EF    DE  0521    564          MOVAL   TIMBUF_NMT,R8           ; ... SERIES OF COMPARES
                          68   67  D1  0528    565          CMPL    (R7),(R8)               ; DO WORDS 1 & 2 OF TIMBUF MATCH STD ?
                                  72  13  052B    566          BEQLU   70$                     ; YES -- GO LOOK AT MORE OF TIMBUF
                    00000000'EF   67  D0  052D    567          MOVL    (R7),EXPV               ; NO -- LOAD UP EXPECTED AND
                    00000000'EF   68  D0  0534    568          MOVL    (R8),RECV               ; ... RECEIVED VALUES, THEN EXIT
                                      053B    569          E R_EXIT LONG,<UNEXPECTED VALUE FOR TIMBUF ARGUMENT >, -
                                      053B    570                       <(WORDS 1 & 2) FROM $NUMTIM>
                                      059F    571 70$:
                  04 A8   04 A7   D1  059F    572          CMPL    4(R7),4(R8)             ; DO WORDS 3 & 4 OF TIMBUF MATCH STD ?
                                  74  13  05A4    573          BEQLU   80$                     ; YES -- GO LOOK AT MORE OF TIMBUF
                00000000'EF   04 A7   D0  05A6    574          MOVL    4(R7),EXPV              ; NO -- LOAD UP EXPECTED AND
                00000000'EF   04 A8   D0  05AE    575          MOVL    4(R8),RECV              ; ... RECEIVED VALUES, THEN EXIT
```

```
                        05B6   576          ERR_EXIT LONG,<UNEXPECTED VALUE FOR TIMBUF ARGUMENT >, -
                        05B6   577                        <(WORDS 3 & 4) FROM $NUMTIM>
                        061A   578 80$:
          08 A8   08 A7  D1  061A   579          CMPL    8(R7),8(R8)             ; DO WORDS 5 & 6 OF TIMBUF MATCH STD ?
                     74  13  061F   580          BEQLU   VERIFYX                 ; YES -- DON'T LOOK AT WORD 7 ...
                         0621   581                                              ; ... BECAUSE OF ROUNDING ERROR
00000000'EF   08 A7  D0  0621   582          MOVL    8(R7),EXPV              ; NO -- LOAD UP EXPECTED AND
00000000'EF   08 A8  D0  0629   583          MOVL    8(R8),RECV              ; ... RECEIVED VALUES, THEN EXIT
                         0631   584          ERR_EXIT LONG,<UNEXPECTED VALUE FOR TIMBUF ARGUMENT >, -
                         0631   585                        <(WORDS 5 & 6) FROM $NUMTIM>
                         0695   586 VERIFYX:
                     05  0695   587          RSB                             ; RETURN TO CALLER
```

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


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

```
$$$$                    = 0000063B R      04
$$$CHARS                = 0000003F
$$$CHARS1               = 0000001C
$$$CHARS2               = 0000001B
$$$CHARS3               = 00000000
$$$CHARS4               = 00000000
$$$CHARS5               = 00000000
$$$COND_A               = 00000001
$$$STRINGS              = 00000001
$$$STRINGS2             = 00000005
$$T2                    = 00000004
BYTE                    = 00000001  G
C1_ELT1                   00000051 R      02
C1_ELT2                   00000068 R      02
C1_ELT3                   00000080 R      02
C1_ELT4                   00000095 R      02
CFLAG                     ********   X    04
CHMRTN                    ********   X    04
CHM_CONT                  ********   X    04
CMPC_SAV                  ********   X    04
COMP_SC                   ********   X    04
COND1                     000000B7 RG     04
COND1_C                 = 00000010
COND1_CLEANUP             000000B8 RG     04
COND1_E                   00000101 R      03
COND1_H                   00000047 RG     03
COND1_T                   00000040 R      03
COND1_TAB                 00000048 R      03
COND2                     000000B9 RG     04
COND2_C                 = 00000004
COND2_CLEANUP             000000BA RG     04
COND2_E                   00000172 R      03
COND2_H                   00000130 RG     03
COND2_T                   00000129 R      03
COND2_TAB                 00000131 R      03
COND3                     000000BB RG     04
COND3_C                 = 00000014
COND3_CLEANUP             000000BC RG     04
COND3_H                   0000017A RG     03
COND3_T                   0000017A R      03
COND3_TAB                 0000017A R      03
COND4                     000000BD RG     04
COND4_C                 = 00000014
COND4_CLEANUP             000000BE RG     04
COND4_H                   0000017B RG     03
COND4_T                   0000017B R      03
COND4_TAB                 0000017B R      03
COND5                     000000BF RG     04
COND5_C                 = 00000014
COND5_CLEANUP             000000C0 RG     04
COND5_H                   0000017C RG     03
COND5_T                   0000017C R      03
COND5_TAB                 0000017C R      03
CTL$GL_PHD                ********   X    04
CVTFLG                    00000172 R      03
DESC                    = 00000010  G
EFLAG                     ********   X    04
EXPV                      ********   X    04
FAO_DESC                  ********   X    04
FAO_LEN                   ********   X    04
FORM_CONDS                000000C1 RG     04
FORM_CONDSX               000001C2 R      04
LONG                    = 00000004  G
LONG_ASC                = 00000000
MOD_MSG_CODE              ********   X    04
MOD_MSG_PRINT             ********   X    04
MSGT_INP_CTL              00000019 R      02
MSG3_ERR_CTL              00000039 RG     02
MSG_A                     ********   X    04
MSG_B                     ********   X    04
MSG_CTXT                  ********   X    04
MSG_DATA1                 ********   X    04
NOTARG                  = 00000000  G
NULL                    = 00000014  G
OUTPUT_MSG                ********   X    04
PCV                       ********   X    04
PHD$Q_PRIVMSK           = 00000000
PRIVMASK                  00000000 R      03
P_IV_ARGS               = 00000002
P_OCESS_ERR               ********   X    04
QUAD                    = 00000008  G
RECV                      ********   X    04
REST_REGS                 ********   X    04
SAVE_REGS                 ********   X    04
SHORT_ASC               = 00000001
SS$_NORMAL                ********   X    04
STD_TIMADR_BTM            00000097 R      02
STD_TIMBUF_ATML           000000C7 R      02
STD_TIMBUF_ATMS           000000AF R      02
STD_TIMBUF_NMT            0000011A R      02
SUCCESS                   ********   X    04
SYS$ASCTIM                ********  GX    04
SYS$BINTIM                ********  GX    04
SYS$CMKRNL                ********  GX    04
SYS$FAO                   ********   X    04
SYS$GETTIM                ********  GX    04
SYS$NUMTIM                ********  GX    04
SYS$SETPRN                ********  GX    04
SYS$SETPRV                ********  GX    04
S_T_A_DATA                000000DF R      02
S_T_N_DATA                00000126 R      02
TESTNUM                   ********   X    04
TEST_MOD_NAME             00000000 RG     02
TEST_MOD_NAME_D           00000009 R      02
TEST_MOD_SUCC             ********   X    04
TIMADR_BTM                00000008 R      03
TIMBUF                    00000101 R      03
TIMBUF_ATM                00000010 R      03
TIMBUF_NMT                00000032 R      03
TIMLEN_ATM                00000030 R      03
TMD_ADDR                  ********   X    04
TM_CLEANUP                000000B3 RG     04
TM_SETUP                  00000000 RG     04
VERIFY                    000001C3 RG     04
```

```
VERIFYX                          00000695 R      04
VFY_CLEANUP                      00000696 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 | 00000150 ( 336.) | 02 ( 2.) | NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC LONG |
| RWDATA | 0000017D ( 381.) | 03 ( 3.) | NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG |
| SATSSS60 | 00000697 ( 1687.) | 04 ( 4.) | NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE |

## Performance indicators

| Phase | Page faults | CPU Time | Elapsed Time |
|---|---|---|---|
| Initialization | 35 | 00:00:00.08 | 00:00:00.42 |
| Command processing | 133 | 00:00:00.69 | 00:00:01.94 |
| Pass 1 | 248 | 00:00:06.94 | 00:00:14.48 |
| Symbol table sort | 0 | 00:00:00.49 | 00:00:00.49 |
| Pass 2 | 135 | 00:00:01.95 | 00:00:03.33 |
| Symbol table output | 15 | 00:00:00.10 | 00:00:00.10 |
| Psect synopsis output | 2 | 00:00:00.02 | 00:00:00.02 |
| Cross-reference output | 0 | 00:00:00.00 | 00:00:00.00 |
| Assembler run totals | 570 | 00:00:10.27 | 00:00:20.79 |

```
The working set limit was 1500 pages.
37385 bytes (74 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 312 non-local and 39 local symbols.
643 source lines were read in Pass 1, producing 25 object records in Pass 2.
35 pages of virtual memory were used to define 26 macros.
```

## Macro library statistics

| Macro library name | Macros defined |
|---|---|
| _$255$DUA28:[SHRLIB]UETP.MLB;1 | 9 |
| _$255$DUA28:[SYS.OBJ]LIB.MLB;1 | 1 |
| _$255$DUA28:[SYSLIB]STARLET.MLB;2 | 13 |
| TOTALS (all libraries) | 23 |

```
611 GETS were required to define 23 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:SATSSS60/OBJ=OBJ$:SATSSS60 MSRC$:SATSSS60/UPDATE=(ENH$:SATSSS60)+EXECML$/LIB+SHRLIB$:UETP/LIB
```

SATSSS56
LIS

SATSSS60
LIS

SATSSS71
LIS

SATSSS54
LIS

SATSSS73
LIS

SATSSS70
LIS

SATSSS72
LIS

SATSSS55
LIS

SATSSS53
LIS

SATSSS61
LIS