L 12

```
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
   SSSSSSSSSSSS  YYY         YYY     SSSSSSSSSSSS
SSS              YYY         YYY  SSS
SSS              YYY         YYY  SSS
SSS              YYY         YYY  SSS
SSS                 YYY   YYY     SSS
SSS                 YYY   YYY     SSS
SSS                 YYY   YYY     SSS
   SSSSSSSSS           YYY           SSSSSSSSS
   SSSSSSSSS           YYY           SSSSSSSSS
   SSSSSSSSS           YYY           SSSSSSSSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
            SSS        YYY                    SSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
SSSSSSSSSSSS           YYY        SSSSSSSSSSSS
```

```
_$
Ps
--
YZ
Z$
Z$
Z$
Z$
Z$
Z$
Z$
Z$
Z$
Z$
Z$
Z$
```

```
**FILE**iD**SYSCVRTIM                                    F  5




  SSSSSSSS  YY      YY    SSSSSSSS    CCCCCCCC  VV      VV  RRRRRRRR    TTTTTTTTTT    IIIIII    MM      MM
  SSSSSSSS  YY      YY    SSSSSSSS    CCCCCCCC  VV      VV  RRRRRRRR    TTTTTTTTTT    IIIIII    MM      MM
SS          YY      YY  SS          CC          VV      VV  RR      RR      TT          II      MMMM  MMMM
SS          YY      YY  SS          CC          VV      VV  RR      RR      TT          II      MMMM  MMMM
SS            YY  YY    SS          CC          VV      VV  RR      RR      TT          II      MM  MM  MM
SS            YY  YY    SS          CC          VV      VV  RR      RR      TT          II      MM  MM  MM
  SSSSSS        YY        SSSSSS    CC          VV      VV  RRRRRRRR        TT          II      MM      MM
  SSSSSS        YY        SSSSSS    CC          VV      VV  RRRRRRRR        TT          II      MM      MM
        SS      YY              SS  CC          VV      VV  RR  RR          TT          II      MM      MM
        SS      YY              SS  CC          VV      VV  RR  RR          TT          II      MM      MM
        SS      YY              SS  CC            VV  VV    RR    RR        TT          II      MM      MM      ....
        SS      YY              SS  CC            VV  VV    RR    RR        TT          II      MM      MM      ....
SSSSSSSS        YY      SSSSSSSS      CCCCCCCC      VV      RR      RR      TT        IIIIII    MM      MM      ....
SSSSSSSS        YY      SSSSSSSS      CCCCCCCC      VV      RR      RR      TT        IIIIII    MM      MM      ....

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            .TITLE  SYSCVRTIM - SYSTEM SERVICES TO CONVERT TIME
0000            .IDENT  'V04-000'
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000    ; D. N. CUTLER 6-JAN-76
0000    ;
0000    ; SYSTEM SERVICES TO CONVERT TIME
0000    ;
0000    ;       CONVERT BINARY TIME TO ASCII STRING
0000    ;       CONVERT ASCII STRING TO BINARY TIME
0000    ;       CONVERT BINARY TIME TO NUMERIC FORMAT
0000    ;
0000    ;       THE CONVERSION ALGORITHMS USED HEREIN WERE DEVELOPED BY P. CONKLIN,
0000    ;       M. SPIER, AND D. ROSENBERY ON THE PDP-10.
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       V03-001 KDM0086         Kathleen D. Morse       02-Apr-1982
0000    ;               Correctly acquire system time, even in case where
0000    ;               secondary processor is accessing EXE$GQ_SYSTIME while
0000    ;               the primary processor is updating it (11/782 case).
0000    ;
0000    ;       V02-004 ROW37307        Ralph O. Weber          27-Jul-1981
0000    ;               Fix EXE$BINTIM to treat decimal point preceeding hundredths of
0000    ;               a second field as a true decimal point.  IE: to cause 0:0:0.1
0000    ;               to convert to 1 tenth of a second rather than to 1 hundredth
0000    ;               of a second.  Also allow indefinite length fractional value
0000    ;               fields.  Use the thousandths digit to round the hundredths
0000    ;               value, and ignore all digits following the thousandths digit.
0000    ;               The entire field, upto the first trailing blank, is still
0000    ;               processed.  Therefore, non-numeric characters in the
0000    ;               fractional seconds field will still produce an Invalid Time
0000    ;               return code.
0000    ;
```

```
                 0000  ;       V02-003 TCM0001         Trudy C. Matthews       03-Jun-1981
                 0000  ;               Fix CONVERT subroutine in EXE$BINTIM to ignore blanks.  This
                 0000  ;               fix allows trailing blanks after a truncated time field.
                 0000  ;
                 0000  ;
                 0000  ;
                 0000  ; MACRO LIBRARY CALLS
                 0000  ;
                 0000
                 0000          $SSDEF                          ;DEFINE SYSTEM STATUS VALUES
                 0000
                 0000  ;
                 0000  ; LOCAL SYMBOLS
                 0000  ;
                 0000  ; ARGUMENT LIST OFFSET DEFINTIONS FOR CONVERT BINARY TIME TO ASCII STRING
                 0000  ;
                 0000
       00000004  0000  ATIMLEN=4                               ;ADDRESS OF WORD TO STORE LENGTH
       00000008  0000  ATIMBUF=8                               ;ADDRESS OF OUTPUT BUFFER DESCRIPTOR
       0000000C  0000  ATIMADR=12                              ;ADDRESS OF 64-BIT ABSOLUTE OR DELTA TIME
       00000010  0000  ACVTFLG=16                              ;CONVERSION INDICATOR
                 0000
                 0000  ;
                 0000  ; ARGUMENT LIST OFFSET DEFINITIONS FOR CONVERT ASCII STRING TO BINARY TIME
                 0000  ;
                 0000
       00000004  0000  BTIMBUF=4                               ;ADDRESS OF ASCII STRING DESCRIPTOR
       00000008  0000  BTIMADR=8                               ;ADDRESS TO STORE 64-BIT ABSOLUTE OR DELTA T
                 0000
                 0000  ;
                 0000  ; ARGUMENT LIST OFFSET DEFINITIONS FOR CONVERT BINARY TIME TO NUMERIC TIME
                 0000  ;
                 0000
       00000004  0000  NTIMBUF=4                               ;ADDRESS OF 7-WORD BUFFER TO RECEIVE TIME
       00000008  0000  NTIMADR=8                               ;ADDRESS OF 64-BIT ABSOLUTE OR DELTA TIME
                 0000
                 0000  ;
                 0000  ; CONVERSION CONSTANTS
                 0000  ;
                 0000  ; TOTAL DAYS IN A CENTURY
                 0000  ;
                 0000
       00008EAC  0000  CENTURYDAYS=<100*365>+<100/4>-<100/100> ;
                 0000
                 0000  ;
                 0000  ; AVERAGE QUARTER DAYS PER CENTURY
                 0000  ;
                 0000
       00023AB1  0000  QDAYSPCENT=<<<100*365>+<100/4>-<100/100>>*4>+<400/400> ;
                 0000
                 0000  ;
                 0000  ; AVERAGE QUARTER DAYS PER YEAR
                 0000  ;
                 0000
       000005B5  0000  QDAYSPYEAR=<365*4>+1                    ;
                 0000
                 0000  ;
```

```
                  0000   ; TOTAL DAYS IN A QUADRICENTURY
                  0000   ;
                  0000
        00023AB1  0000   QUADRIDAYS=<400*365>+<400/4>-<400/100>+<400/400> ;
                  0000
                  0000   ;
                  0000   ; TOTAL DAYS IN A QUADYEAR
                  0000   ;
                  0000
        000005B5  0000   QUADYEARDAYS=<365*4>+1                          ;
                  0000
                  0000   ;
                  0000   ; OFFSET IN DAYS FROM 1-JAN-1501 TO 17-NOV-1858
                  0000   ;
                  0000
        0001FE98  0000   TIMOFF1=<<1858-1501>*365>+<<1858-1501>/4>-<<1858-1501>/100>+<<1858-1501>/400>+ -;
                  0000          31+28+31+30+31+30+31+31+30+31+17 ;
                  0000
                  0000   ;
                  0000   ; OFFSET IN DAYS FROM 1-JAN-1601 TO 17-NOV-1858
                  0000   ;
                  0000
                  0000   TIMOFF2=<<1858-1601>*365>+<<1858-1601>/4>-<<1858-1601>/100>+<<1858-1601>/400>+ -;
        00016FEC  0000          31+28+31+30+31+30+31+31+30+31+17 ;
                  0000
                  0000   ;
                  0000   ; CHARACTER CODE DEFINITIONS
                  0000   ;
                  0000
        00000020  0000   BLANK=32                                        ;
        0000003A  0000   COLON=58                                        ;
        0000002D  0000   HYPHEN-45                                       ;
        00000039  0000   NINE=57                                         ;
        00000030  0000   ONE=48                                          ;
        0000002E  0000   PERIOD=46                                       ;
                  0000
                  0000   ;
                  0000   ; NUMERIC TIME BUFFER OFFSET DEFINITIONS
                  0000   ;
                  0000
        00000000  0000   YEAR=0                                          ;
        00000002  0000   MONTH=2                                         ;
        00000004  0000   DAY=4                                           ;
        00000006  0000   HOUR=6                                          ;
        00000008  0000   MINUTE=8                                        ;
        0000000A  0000   SECOND=10                                       ;
        0000000C  0000   HUNDREDTH=12                                    ;
                  0000
                  0000   ;
                  0000   ; LOCAL DATA
                  0000   ;
                  0000   ; MONTH, DAY CONVERSION TABLE
                  0000   ;
              00000000         .PSECT  Y$EXEPAGED
                  0000   DATETABLE:                                      ;DATE CONVERSION TABLE
              1F  0000          .BYTE   31                               ;JANUARY
              1D  0001          .BYTE   29                               ;FEBRUARY
```

```
                                  1F  0002          .BYTE   31                      ;MARCH
                                  1E  0003          .BYTE   30                      ;APRIL
                                  1F  0004          .BYTE   31                      ;MAY
                                  1E  0005          .BYTE   30                      ;JUNE
                                  1F  0006          .BYTE   31                      ;JULY
                                  1F  0007          .BYTE   31                      ;AUGUST
                                  1E  0008          .BYTE   30                      ;SEPTEMBER
                                  1F  0009          .BYTE   31                      ;OCTOBER
                                  1E  000A          .BYTE   30                      ;NOVEMBER
                                  1F  000B          .BYTE   31                      ;DECEMBER
                                      000C
                                      000C  ;
                                      000C  ; MONTH CONVERSION TABLE
                                      000C  ;
                                      000C
                                      000C  MONTHTAB:                               ;
                         4E 41 4A 03  000C          .ASCII  <3>/JAN/                ;
                         42 45 46 03  0010          .ASCII  <3>/FEB/                ;
                         52 41 4D 03  0014          .ASCII  <3>/MAR/                ;
                         52 50 41 03  0018          .ASCII  <3>/APR/                ;
                         59 41 4D 03  001C          .ASCII  <3>/MAY/                ;
                         4E 55 4A 03  0020          .ASCII  <3>/JUN/                ;
                         4C 55 4A 03  0024          .ASCII  <3>/JUL/                ;
                         47 55 41 03  0028          .ASCII  <3>/AUG/                ;
                         50 45 53 03  002C          .ASCII  <3>/SEP/                ;
                         54 43 4F 03  0030          .ASCII  <3>/OCT/                ;
                         56 4F 4E 03  0034          .ASCII  <3>/NOV/                ;
                         43 45 44 03  0038          .ASCII  <3>/DEC/                ;
                                      003C
                                      003C  ;
                                      003C  ; HOURS, MINUTES, SECONDS, HUNDREDTHS CONVERSION TABLE
                                      003C  ;
                                      003C
                                      003C  TIMETABLE:                              ;TIME CONVERSION TABLE
                                  64  003C          .BYTE   100                     ;HUNDREDTHS
                                  3C  003D          .BYTE   60                      ;SECONDS
                                  3C  003E          .BYTE   60                      ;MINUTES AND HOURS
                                      003F
                                      003F  ;
                                      003F  ; CONVERSION CONTROL STRINGS
                                      003F  ;
                                      003F
5A 34 21 2D 43 41 21 2D 57 53 32 21  003F  DATE:   .ASCII  /!2SW-!AC-!4ZW /        ;
                               20 57  004B
               20 57 53 34 21  004D  DELTA:  .ASCII  /!4SW /                 ;
32 21 3A 57 5A 32 21 3A 57 5A 32 21  0052  TIME:   .ASCII  /!2ZW:!2ZW:!2ZW.!2ZW/   ;
         57 5A 32 21 2E 57 5A  005E
```

```
                        0065            .SBTTL  CONVERT BINARY TIME TO ASCII STRING
                        0065    ;+
                        0065    ; EXE$ASCTIM - CONVERT BINARY TIME TO ASCII STRING
                        0065    ;
                        0065    ; THIS SERVICE PROVIDES THE CAPABILITY TO CONVERT AN ABSOLUTE OR DELTA
                        0065    ; TIME FROM 64-BIT FORMAT TO AN ASCII STRING.
                        0065    ;
                        0065    ; INPUTS:
                        0065    ;
                        0065    ;       ATIMLEN(AP) = ADDRESS OF WORD TO RECEIVE OUTPUT LENGTH.
                        0065    ;       ATIMBUF(AP) = ADDRESS OF OUTPUT BUFFER DESCRIPTOR.
                        0065    ;       ATIMADR(AP) = ADDRESS OF 64-BIT TIME VALUE. IF ZERO, THEN THE CURRENT
                        0065    ;               SYSTEM TIME IS USED. POSITIVE VALUES ARE INTERPRETED AS
                        0065    ;               ABSOLUTE TIMES AND NEGATIVE VALUES AS DELTA TIMES.
                        0065    ;       ACVTFLG(AP) = CONVERSION INDICATOR.
                        0065    ;               LOW BIT CLEAR INDICATES BOTH DATE AND TIME ARE TO BE CON-
                        0065    ;                       VERTED.
                        0065    ;               LOW BIT SET INDICATES ONLY TIME IS TO BE CONVERTED.
                        0065    ;
                        0065    ; OUTPUTS:
                        0065    ;
                        0065    ;       R0 LOW BIT CLEAR INDICATES FAILURE TO CONVERT TIME TO ASCII.
                        0065    ;
                        0065    ;               R0 = SS$_ACCVIO - 64-BIT TIME VALUE OR OUTPUT BUFFER DESCRIPTOR
                        0065    ;                       CANNOT BE READ BY CALLING ACCESS MODE, OR OUTPUT BUFFER
                        0065    ;                       CANNOT BE WRITTEN BY CALLING ACCESS MODE.
                        0065    ;
                        0065    ;               R0 = SS$_IVTIME - SPECIFIED DELTA TIME IS GREATER THAN 9999
                        0065    ;                       DAYS.
                        0065    ;
                        0065    ;       R0 LOW BIT SET INDICATES SUCCESSFUL COMPLETION.
                        0065    ;
                        0065    ;               R0 = SS$_NORMAL - NORMAL COMPLETION.
                        0065    ;-
                        0065
                        0065    EXE$ASCTIM::                                    ;CONVERT TIME TO ASCII
                  007C  0065            .WORD   ^M<R2,R3,R4,R5,R6>              ;ENTRY MASK
        7E   08 BC  7D  0067            MOVQ    @ATIMBUF(AP),-(SP)              ;SAVE OUTPUT BUFFER DESCRIPTOR
          56    5E  D0  006B            MOVL    SP,R6                           ;SAVE ADDRESS OF OUTPUT BUFFER DESCRIPTOR
                7E  D4  006E            CLRL    -(SP)                           ;CLEAR SPACE FOR LENGTH FROM FAO
          55    5E  D0  0070            MOVL    SP,R5                           ;SAVE ADDRESS OF LENGTH
                52  D4  0073            CLRL    R2                              ;ASSUME ABSOLUTE TIME SPECIFIED
        53   0C AC  D0  0075            MOVL    ATIMADR(AP),R3                  ;GET ADDRESS OF 64-BIT TIME VALUE
                07  13  0079            BEQL    10$                             ;IF EQL NONE SPECIFIED
          50    63  7D  007B            MOVQ    (R3),R0                         ;GET 64-BIT TIME VALUE
                02  18  007E            BGEQ    10$                             ;IF GEQ ABSOLUTE TIME
                52  D6  0080            INCL    R2                              ;INDICATE DELTA TIME
          5E    10  C2  0082    10$:    SUBL    #<<<7*2>+3>/4>*4,SP             ;ALLOCATE NUMERIC TIME BUFFER
          54    5E  D0  0085            MOVL    SP,R4                           ;SAVE ADDRESS OF NUMERIC TIME BUFFER
                        0088            $NUMTIM_S (R4),(R3)                     ;CONVERT TIME TO NUMERIC FORMAT
          6F    50  E9  0093            BLBC    R0,60$                          ;IF LBC CONVERSION FAILURE
                        0096
                        0096    ;
                        0096    ; CONVERT TIME TO ASCII FORMAT
                        0096    ;
                        0096
         3E 10 AC   E8  0096            BLBS    ACVTFLG(AP),40$                 ;IF LBS ONLY TIME IS TO BE CONVERTED
```

```
          12 52   E8  009A          BLBS    R2,20$                      ;IF LBS DELTA TIME SPECIFIED
                      009D
                      009D      ;
                      009D      ; CONVERT DATE
                      009D      ;
                      009D
     52   02 A4   3C  009D          MOVZWL  MONTH(R4),R2                ;GET NUMERIC MONTH VALUE
 52   FF62 CF42   DE  00A1          MOVAL   W^MONTHTAB-4[R2],R2         ;GET `DDRESS OF MONTH COUNTED STRING
        FF94 CF   DF  00A7          PUSHAL  W^DATE                      ;BUILD DESCRIPTOR FOR CONTROL STRING
             0E   DD  00AB          PUSHL   #DELTA-DATE                 ;
             06   11  00AD          BRB     30$                         ;
                      00AF
                      00AF      ;
                      00AF      ; CONVERT DELTA TIME
                      00AF      ;
                      00AF
        FF9A CF   DF  00AF  20$:    PUSHAL  W^DELTA                     ;BUILD CONTROL STRING DESCRIPTOR
             05   DD  00B3          PUSHL   #TIME-DELTA                 ;
        51   5E   D0  00B5  30$:    MOVL    SP,R1                       ;COPY ADDRESS OF CONTROL STRING DESCRIPTOR
                      00B8          $FAO_S  (R1),(R5),(R6),DAY(R4),R2,YEAR(R4) ;CONVERT DELTA TIME OR DATE
          36 50   E9  00CC          BLBC    R0,60$                      ;IF LBC CONVERT FAILURE
        66   65   A2  00CF          SUBW    (R5),(R6)                   ;ANY SPACE LEFT IN TIME BUFFER?
             27   15  00D2          BLEQ    50$                         ;IF LEQ NO
     04 A6   65   C0  00D4          ADDL    (R5),4(R6)                  ;UPDATE TIME BUFFER ADDRESS
                      00D8
                      00D8      ;
                      00D8      ; CONVERT TIME
                      00D8      ;
                      00D8
        FF76 CF   DF  00D8  40$:    PUSHAL  W^TIME                      ;BUILD CONTROL STRING DESCRIPTOR
             13   DD  00DC          PUSHL   #EXE$ASCTIM-TIME            ;
        51   5E   D0  00DE          MOVL    SP,R1                       ;COPY ADDRESS OF CONTROL STRING DESCRIPTOR
                      00E1          $FAO_S  (R1),2(R5),(R6),HOUR(R4),MINUTE(R4),SECOND(R4),HUNDREDTH(R4) ;
     51   04 AC   D0  00FB  50$:    MOVL    ATIMLEN(AP),R1              ;LENGTH ADDRESS SPECIFIED?
             04   13  00FF          BEQL    60$                         ;IF EQL NO
   61    65  85   A1  0101          ADDW3   (R5)+,(R5),(R1)             ;COMPUTE AND RETURN OUTPUT LENGTH
                  04  0105  60$:    RET                                 ;
```

```
                        0106          .SBTTL  CONVERT ASCII STRING TO BINARY TIME
                        0106    ;+
                        0106    ; EXE$BINTIM - CONVERT ASCII STRING TO BINARY TIME
                        0106    ;
                        0106    ; THIS SERVICE PROVIDES THE CAPABILITY TO CONVERT AN ASCII STRING TO A
                        0106    ; 64-BIT ABSOLUTE OR DELTA TIME.
                        0106    ;
                        0106    ; INPUTS:
                        0106    ;
                        0106    ;       BTIMBUF(AP) = ADDRESS OF ASCII STRING DESCRIPTOR.
                        0106    ;       BTIMADR(AP) = ADDRESS TO STORE 64-BIT TIME VALUE.
                        0106    ;
                        0106    ; OUTPUTS:
                        0106    ;
                        0106    ;       R0 LOW BIT CLEAR INDICATES FAILURE TO CONVERT TIME TO ASCII.
                        0106    ;
                        0106    ;               R0 = SS$_IVTIME - ASCII STRING HAS INVALID SYNTAX OR TIME
                        0106    ;                       COMPONENT IS OUT OF RANGE.
                        0106    ;
                        0106    ;       R0 LOW BIT SET INDICATES SUCCESSFUL COMPLETION.
                        0106    ;
                        0106    ;               R0 = SS$_NORMAL - NORMAL COMPLETION.
                        0106    ;-
                        0106
                        0106   EXE$BINTIM::                                    ;CONVERT ASCII STRING TO BINARY TIME
                   01FC 0106            .WORD   ^M<R2,R3,R4,R5,R6,R7,R8> ;ENTRY MASK
          5E    10   C2 0108            SUBL    #<<<7*2>+3>/4>*4,SP     ;ALLOCATE NUMERIC TIME BUFFER
          57    5E   D0 010B            MOVL    SP,R7                   ;SAVE ADDRESS OF NUMERIC TIME BUFFER
     55    04 BC     7D 010E            MOVQ    @BTIMBUF(AP),R5         ;GET ADDRESS AND LENGTH OF ASCII STRING
                58   D4 0112            CLRL    R8                      ;ASSUME DELTA TIME
                55   B7 0114   10$:     DECW    R5                      ;ANY MORE CHARACTERS?
                64   19 0116            BLSS    30$                     ;IF LSS NO
          86    20   91 0118            CMPB    #BLANK,(R6)+            ;SKIP LEADING BLANK?
                F7   13 011B            BEQL    10$                     ;IF EQL YES
                55   B6 011D            INCW    R5                      ;CORRECT NUMBER OF CHARACTERS
     76   55    2D   3A 011F            LOCC    #HYPHEN,R5,-(R6)        ;ABSOLUTE TIME FORMAT?
                57   13 0123            BEQL    30$                     ;IF EQL NO
                58   D6 0125            INCL    R8                      ;INDICATE ABSOLUTE TIME
                        0127            $NUMTIM_S (R7)                  ;CONVERT CURRENT TIME TO NUMERIC FORMAT
                        0132
                        0132    ;
                        0132    ; CONVERT ABSOLUTE TIME
                        0132    ;
                        0132
     54    04 A7     DE 0132            MOVAL   DAY(R7),R4              ;SET ADDRESS TO STORE DAY
              0081   30 0136            BSBW    CONVERT                 ;CONVERT DAY FIELD
                     2D 0139            .BYTE   HYPHEN                  ;EXPECTED TERMINATOR
                55   B5 013A            TSTW    R5                      ;ANY MORE CHARACTERS?
                03   12 013C            BNEQ    11$                     ;BRNCH IF THERE ARE MORE CHARACTERS.
              00DB   31 013E            BRW     CVRTIME                 ;IF NO MORE CHARACTERS, CONVERT TIME.
          86    2D   91 0141   11$:     CMPB    #HYPHEN,(R6)+           ;MONTH FIELD VOID?
                2F   13 0144            BEQL    20$                     ;IF EQL YES
FEBF CF   30    76   03 39 0146         MATCHC  #3,-(R6),#4*12,W^MONTHTAB ;SEARCH FOR MONTH SUBSTRING MATCH
                03   13 014D            BEQL    14$                     ;SKIP ERROR BRANCH IF MATCH FOUND
              0092   31 014F            BRW     IVTIME                  ;IF NEQ NO MATCH FOUND
     52   30    52   C3 0152   14$:     SUBL3   R2,#4*12,R2             ;CALCULATE CHARACTERS TO START OF SUBSTRING
          52    03   D3 0156            BITL    #3,R2                   ;MULTIPLE OF 4?
```

```
                   03   13  0159              BEQL    16$                     ;BRANCH IF MULTIPLE OF 4
                 0086   31  015B              BRW     IVTIME                  ;IF NOT MULTIPLE OF 4, THEN ERROR
      02 A7   52   04   A7  015E     16$:     DIVW3   #4,R2,MONTH(R7)         ;CONVERT TO MONTH AND STORE
              56   03   C0  0163              ADDL    #3,R6                   ;UPDATE ADDRESS OF ASCII STRING
              55   03   A2  0166              SUBW    #3,R5                   ;UPDATE COUNT OF REMAINING CHARACTERS
                   79   19  0169              BLSS    IVTIME                  ;IF LSS INVALID SYNTAX
                   03   14  016B              BGTR    18$                     ;IF GTR CHARACTERS REMAINING
                 00AC   31  016D              BRW     CVRTIME                 ;OTHERWISE END OF STRING
              86   2D   91  0170     18$:     CMPB    #HYPHEN,(R6)+           ;FIELD TERMINATED PROPERLY?
                   6F   12  0173              BNEQ    IVTIME                  ;IF NEQ NO
                   55   B7  0175     20$:     DECW    R5                      ;DECREMENT COUNT OF REMAINING CHARACTERS
              54   67   DE  0177              MOVAL   YEAR(R7),R4             ;SET ADDRESS TO STORE YEAR
                   0A   11  017A              BRB     40$                     ;
                            017C
                            017C       ;
                            017C       ; CONVERT DELTA TIME
                            017C       ;
                            017C
              54   67   DE  017C     30$:     MOVAL   YEAR(R7),R4             ;GET ADDRESS TO STORE YEAR
                   84   D4  017F              CLRL    (R4)+                   ;CLEAR YEAR AND MONTH
                   64   7C  0181              CLRQ    (R4)                    ;CLEAR DAY, HOUR, MINUTE, AND SECOND
                0C A7   B4  0183              CLRW    HUNDREDTH(R7)           ;CLEAR HUNDREDTH
                   62   10  0186     40$:     BSBB    CONVERT                 ;CONVERT RELATIVE DAY OR YEAR FIELD
                   20   0188                  .BYTE   BLANK                   ;EXPECTED TERMINATOR
                   55   B7  0189     50$:     DECW    R5                      ;ANY REMAINING CHARACTERS?
                   03   18  018B              BGEQ    53$                     ;BRANCH IF CHARACTERS REMAINING
                 008C   31  018D              BRW     CVRTIME                 ;ELSE GO PROCESS WHAT WE'VE GOT
              86   20   91  0190     53$:     CMPB    #BLANK,(R6)+            ;NEXT CHARACTER BLANK?
                   F4   13  0193              BEQL    50$                     ;IF EQL YES
                   56   D7  0195              DECL    R6                      ;BACK UP TO NONBLANK CHARACTER
                   55   D6  0197              INCL    R5                      ;ADJUST REMAINING CHARACTER COUNT
                            0199
                            0199       ;
                            0199       ; CONVERT TIME
                            0199       ;
                            0199
          54    06 A7   DE  0199              MOVAL   HOUR(R7),R4             ;SET ADDRESS TO STORE HOUR
                   4B   10  019D              BSBB    CONVERT                 ;CONVERT HOUR FIELD
                        3A  019F              .BYTE   COLON                   ;EXPECTED TERMINATOR
                   48   10  01A0              BSBB    CONVERT                 ;CONVERT MINUTE FIELD
                        3A  01A2              .BYTE   COLON                   ;EXPECTED TERMINATOR
                   45   10  01A3              BSBB    CONVERT                 ;CONVERT SECOND FIELD
                        2E  01A5              .BYTE   PERIOD                  ;EXPECTED TERMINATOR
                            01A6
                            01A6                                              ;Convert Hundredth Field
                            01A6                                              ;This must be done differently because
                            01A6                                              ;this is a fractional value.
                            01A6                                              ;Establish max useable digits,
              53   03   D0  01A6              MOVL    #3, R3                  ;including the rounding digit.
                   64   B4  01A9              CLRW    (R4)                    ;Clear accumulated value.
                   55   B7  01AB     70$:     DECW    R5                      ;Any more characters?
                   2C   19  01AD              BLSS    80$                     ;Branch if no more characters.
              51   86   9A  01AF              MOVZBL  (R6)+, R1               ;Get the next character.
              20   51   91  01B2              CMPB    R1, #BLANK              ;A blank marks the end of the field.
                   24   13  01B5              BEQL    80$                     ;Branch if at end of the field.
              51   30   C2  01B7              SUBL    #ONE, R1                ;Subtract out character bias.
                   28   19  01BA              BLSS    IVTIME                  ;Branch if invalid character.
```

```
         51   09   D1  01BC            CMPL    #NINE-ONE, R1          ;Result value within digit range?
              23   19  01BF            BLSS    IVTIME                 ;Branch if invalid character.
           0B 53   F5  01C1            SOBGTR  R3, 73$                ;Branch if using this digit directly.
              E5   19  01C4            BLSS    70$                    ;Branch if ignoring this digit.
         51   05   D1  01C6            CMPL    #5, R1                 ;Else digit as the rounding digit.
              E0   14  01C9            BGTR    70$                    ;Branch if rounding has no effect.
              64   B6  01CB            INCW    (R4)                   ;If rounding up, do it.
              DC   11  01CD            BRB     70$                    ;Then loop, but for a regular digit,
         64   0A   A4  01CF    73$:    MULW    #10, (R4)              ;multiply partial result by 10.
              10   1D  01D2            BVS     IVTIME                 ;An overflow means an invalid time.
         64   51   A0  01D4            ADDW    R1, (R4)               ;Accumulate fracitonal value.
              0B   1D  01D7            BVS     IVTIME                 ;Overflow means invalid time.
              D0   11  01D9            BRB     70$                    ;Loop till end occurs.
                       01DB
              53   D7  01DB    80$:    DECL    R3                     ;Insure that truncated digits are
              3D   15  01DD            BLEQ    CVRTIME                ;included as zeros in the final
         64   0A   A4  01DF            MULW    #10, (R4)              ;fractional (hundredths) field value.
              F7   11  01E2            BRB     80$                    ;NB: this will always overflow a word
                       01E4                                           ;if the fractional field has a
                       01E4                                           ;resolution greater than thousandths.
                       01E4    ;
                       01E4    ; INVALID SYNTAX OR TIME COMPONENT
                       01E4    ;
                       01E4
   50    0184 8F   3C  01E4    IVTIME: MOVZWL  #SS$_IVTIME,R0         ;SET INVALID TIME
                   04  01E9            RET
                       01EA
                       01EA    ;
                       01EA    ; SUBROUTINE TO CONVERT NUMERIC FIELD TO BINARY
                       01EA    ;
                       01EA
                       01EA    CONVERT:                               ;CONVERT FIELD
              50   D4  01EA            CLRL    R0                     ;CLEAR ACCUMULATED VALUE
              84   B5  01EC    10$:    TSTW    (R4)+                  ;POINT PAST NEXT FIELD
              55   B7  01EE    11$:    DECW    R5                     ;ANY MORE CHARACTERS?
              2A   19  01F0            BLSS    CVRTIME                ;IF LSS NO
         51   86   9A  01F2            MOVZBL  (R6)+,R1               ;GET NEXT CHARACTER
      00 BE   51   91  01F5            CMPB    R1,@(SP)               ;EXPECTED TERMINATOR?
              1E   13  01F9            BEQL    20$                    ;IF EQL YES
         20   51   91  01FB            CMPB    R1,#BLANK              ;BLANK CHARACTER?
              EE   13  01FE            BEQL    11$                    ;IGNORE BLANKS
         51   30   C2  0200            SUBL    #ONE,R1                ;SUBTRACT OUT CHARACTER BIAS
              DF   19  0203            BLSS    IVTIME                 ;IF LSS INVALID CHARACTER
         51   09   D1  0205            CMPL    #NINE-ONE,R1           ;RESULT VALUE WITHIN RANGE?
              DA   19  0208            BLSS    IVTIME                 ;IF LSS INVALID CHARACTER
         50   0A   A4  020A            MULW    #10,R0                 ;MULTIPLY PARTIAL RESULT BY 10
              D5   1D  020D            BVS     IVTIME                 ;IF VS INVALID TIME
         50   51   A0  020F            ADDW    R1,R0                  ;ACCUMULATE VALUE
              D0   1D  0212            BVS     IVTIME                 ;IF VS INVALID TIME VALUE
         74   50   B0  0214            MOVW    R0,-(R4)               ;STORE VALUE
              D3   11  0217            BRB     10$                    ;
              6E   D6  0219    20$:    INCL    (SP)                   ;INCREMENT PAST TERMINATOR
              05       021B            RSB                            ;
                       021C
                       021C    ;
                       021C    ; CHECK CONVERTED DATE AND TIME VALUES
                       021C    ;
```

```
                                      021C   484
                                      021C   485 CVRTIME:                                                 ;
          04 A7   270F 8F        B1   021C   486          CMPW    #9999,DAY(R7)               ;DAY WITHIN UPPER LIMIT?
                       C0        1F   0222   487          BLSSU   IVTIME                      ;IF LSSU NO
               06 A7   18        B1   0224   488          CMPW    #24,HOUR(R7)                ;HOUR WITHIN LIMITS?
                       BA        1B   0228   489          BLEQU   IVTIME                      ;IF LEQU NO
               08 A7   3C        B1   022A   490          CMPW    #60,MINUTE(R7)              ;MINUTE WITHIN LIMITS?
                       B4        1B   022E   491          BLEQU   IVTIME                      ;IF LEQU NO
               0A A7   3C        B1   0230   492          CMPW    #60,SECOND(R7)              ;SECOND WITHIN LIMITS?
                       AE        1B   0234   493          BLEQU   IVTIME                      ;IF LEQU NO
               55   04 A7        3C   0236   494          MOVZWL  DAY(R7),R5                  ;GET DAY VALUE
                    03 58        E8   023A   495          BLBS    R8,5$                       ;IF LBS ABSOLUTE TIME
                     0097        31   023D   496          BRW     40$                         ;
                                      0240   497
                                      0240   498 ;
                                      0240   499 ; CONVERT YEARS TO QUADRICENTURIES, CENTURIES, QUADYEARS, YEARS
                                      0240   500 ;
                                      0240   501
                       A2        13   0240   502 5$:      BEQL    IVTIME                      ;IF EQL INVALID TIME
                  50   67        3C   0242   503          MOVZWL  YEAR(R7),R0                 ;GET YEAR VALUE
             50   F9BF C0        3E   0245   504          MOVAW   -1601(R0),R0                ;CALCULATE YEARS PAST 1601
                       98        19   024A   505          BLSS    IVTIME                      ;IF LSS INVALID TIME
                       51        D4   024C   506          CLRL    R1                          ;CLEAR HIGH PART OF DIVIDEND
  51    50    50   00000190 8F   7B   024E   507          EDIV    #400,R0,R0,R1               ;CALCULATE QUADRICENTURIES
                       52        D4   0257   508          CLRL    R2                          ;CLEAR HIGH PART OF DIVIDEND
  52    51    51   00000064 8F   7B   0259   509          EDIV    #100,R1,R1,R2               ;CALCULATE CENTURIES
                       53        D4   0262   510          CLRL    R3                          ;CLEAR HIGH PART OF DIVIDEND
        53    52    52     04    7B   0264   511          EDIV    #4,R2,R2,R3                 ;CALCULATE QUADYEARS AND YEARS
                                      0269   512
                                      0269   513 ;
                                      0269   514 ; CONVERT QUADRICENTURIES, CENTURIES, QUADYEARS, YEARS TO DAYS
                                      0269   515 ;
                                      0269   516
             53    016D 8F       A4   0269   517          MULW    #365,R3                     ;CALCULATE NUMBER OF DAYS PAST LEAP YEAR
  52    53    52   000005B5 8F   7A   026E   518          EMUL    #QUADYEARDAYS,R2,R3,R2      ;CALCULATE NUMBER OF QUADYEAR DAYS AND SUM
              51   00008EAC 8F   C4   0277   519          MULL    #CENTURYDAYS,R1             ;CALCULATE NUMBER OF CENTURY DAYS
  55    51    50   00023AB1 8F   7A   027E   520          EMUL    #QUADRIDAYS,R0,R1,R5        ;CALCULATE NUMBER OF QUADRIDAYS AND SUM
                       50        D4   0287   521          CLRL    R0                          ;CLEAR INITIAL LOOP INDEX
               56   02 A7        3C   0289   522          MOVZWL  MONTH(R7),R6                ;GET SPECIFIED MONTH VALUE
                  55   52        C0   028D   523 10$:     ADDL    R2,R5                       ;ACCUMULATE TOTAL DAYS
          52    FD6B E40         9A   0290   524          MOVZBL  W^DATETABLE[R0],R2          ;GET NUMBER OF DAYS IN MONTH
                  50   01        D1   0296   525          CMPL    #1,R0                       ;SECOND MONTH OF YEAR?
                       1E        12   0299   526          BNEQ    30$                         ;IF NEQ NO
                  53   67        3C   029B   527          MOVZWL  YEAR(R7),R3                 ;GET SPECIFIED YEAR VALUE
                  53   03        D3   029E   528          BITL    #3,R3                       ;YEAR MULTIPLE OF 4?
                       14        12   02A1   529          BNEQ    20$                         ;IF NEQ NO
                       54        D4   02A3   530          CLRL    R4                          ;CLEAR HIGH PART OF DIVIDEND
  54    53    53   00000064 8F   7B   02A5   531          EDIV    #100,R3,R3,R4               ;CALCULATE CENTURY AND YEAR IN CENTURY
                       54        D5   02AE   532          TSTL    R4                          ;YEAR MULTIPLE OF 100?
                       07        12   02B0   533          BNEQ    30$                         ;IF NEQ NO
                  53   03        D3   02B2   534          BITL    #3,R3                       ;YEAR MULT'PLE OF 400?
                       02        13   02B5   535          BEQL    30$                         ;IF EQL YES
                       52        D7   02B7   536 20$:     DECL    R2                          ;REDUCE NUMBER OF DAYS IN MONTH
            D0 50      56        F2   02B9   537 30$:     AOBLSS  R6,R0,10$                   ;ANY MORE DAYS TO ACCUMULATE?
          50    0184 8F          3C   02BD   538          MOVZWL  #SS$_IVTIME,R0              ;ASSUME INVALID DAY OF MONTH
            51    04 A7          3C   02C2   539          MOVZWL  DAY(R7),R1                  ;GET SPECIFIED DAY
       55   00016FEC 8F          C2   02C6   540          SUBL    #TIMOFF2,R5                 ;SUBTRACT OUT NUMBER OF DAYS TO 17-NOV-1858
```

```
                        55    51   C0  02CD         ADDL    R1,R5                    ;CALCULATE TOTAL NUMBER OF DAYS
                              57   19  02D0         BLSS    60$                      ;IF LSS INVALID TIME
                        52    51   D1  02D2         CMPL    R1,R2                    ;DAY WITHIN LIMITS?
                              52   1A  02D5         BGTRU   60$                      ;IF GTRU NO
                                       02D7
                                       02D7     ;
                                       02D7     ; CONVERT TIME TO TENTHS OF MICROSECONDS
                                       02D7     ;
                                       02D7
                     50    06 A7   3C  02D7 40$:    MOVZWL  HOUR(R7),R0              ;GET HOUR VALUE
                     51    08 A7   3C  02DB         MOVZWL  MINUTE(R7),R1            ;GET MINUTE VALUE
               50   51    50   3C  7A  02DF         EMUL    #60,R0,R1,R0             ;CONVERT HOURS TO MINUTES AND SUM
                     51    0A A7   3C  02E4         MOVZWL  SECOND(R7),R1            ;GET SECOND VALUE
               50   51    50   3C  7A  02E8         EMUL    #60,R0,R1,R0             ;CONVERT MINUTES TO SECONDS AND SUM
                     51    0C A7   3C  02ED         MOVZWL  HUNDREDTH(R7),R1         ;GET HUNDREDTH VALUE
   50    51    50   00000064 8F    7A  02F1         EMUL    #100,R0,R1,R0            ;CONVERT SECONDS TO HUNDREDTHS AND SUM
   50    00    50   000186A0 8F    7A  02FA         EMUL    #100000,R0,#0,R0         ;CONVERT TO TENTHS OF MICROSECONDS
                                       0303
                                       0303     ;
                                       0303     ; CONVERT DAYS TO TENTHS OF MICROSECONDS
                                       0303     ;
                                       0303
   52    00    55   324A9A70 8F    7A  0303         EMUL    #843750000,R5,#0,R2      ;MULTIPLY BY 864000000000/1024
                     52    52 0A   79  030C         ASHQ    #10,R2,R2                ;MULTIPLY BY 1024
                                       0310
                                       0310     ;
                                       0310     ; COMBINE RESULTS AND STORE 64-BIT TIME
                                       0310     ;
                                       0310
                        52    50   C0  0310         ADDL    R0,R2                    ;ADD LOW ORDER PARTS
                        53    51   D8  0313         ADWC    R1,R3                    ;ADD HIGH ORDER PARTS
                        50    01   3C  0316         MOVZWL  #SS$_NORMAL,R0           ;SET NORMAL COMPLETION
                         09   58   E8  0319         BLBS    R8,50$                   ;IF LBS ABSOLUTE TIME
                        53    53   CE  031C         MNEGL   R3,R3                    ;CONVERT TO DELTA TIME
                        52    52   CE  031F         MNEGL   R2,R2                    ;
                        53    00   D9  0322         SBWC    #0,R3                    ;
                     08 BC    52   7D  0325 50$:    MOVQ    R2,@BTIMADR(AP)          ;STORE 64-BIT TIME VALUE
                                   04  0329 60$:    RET                              ;
```

```
                                 032A         .SBTTL  CONVERT BINARY TIME TO NUMERIC TIME
                                 032A   ;+
                                 032A   ; EXE$NUMTIM - CONVERT BINARY TIME TO NUMERIC TIME
                                 032A   ;
                                 032A   ; THIS SERVICE PROVIDES THE CAPABILITY TO CONVERT AN ABSOLUTE OR DELTA TIME
                                 032A   ; FROM 64-BIT FORMAT TO INTEGER DATE AND TIME VALUES.
                                 032A   ;
                                 032A   ; INPUTS:
                                 032A   ;
                                 032A   ;       NTIMBUF(AP) = ADDRESS OF 7-WORD BUFFER TO RECEIVE CONVERTED DATE AND
                                 032A   ;               TIME VALUES.
                                 032A   ;       NTIMADR(AP) = ADDRESS OF 64-BIT TIME VALUE. IF ZERO, THEN THE CURRENT
                                 032A   ;               SYSTEM TIME IS USED. POSITIVE VALUES ARE INTERPRETED AS
                                 032A   ;               ABSOLUTE TIMES AND NEGATIVE VALUES AS DELTA TIMES.
                                 032A   ;
                                 032A   ; OUTPUTS:
                                 032A   ;
                                 032A   ;       R0 LOW BIT CLEAR INDICATES FAILURE TO CONVERT TO NUMERIC TIME.
                                 032A   ;
                                 032A   ;               R0 = SS$_ACCVIO - 64-BIT TIME VALUE CANNOT BE READ BY CALLING
                                 032A   ;                       ACCESS MODE OR TIME BUFFER CANNOT BE WRITTEN BY
                                 032A   ;                       CALLING ACCESS MODE.
                                 032A   ;
                                 032A   ;               R0 = SS$_IVTIME - SPECIFIED DELTA TIME IS GREATER THAN 9999
                                 032A   ;                       DAYS.
                                 032A   ;
                                 032A   ;       R0 LOW BIT SET INDICATES SUCCESSFUL COMPLETION.
                                 032A   ;
                                 032A   ;               R0 = SS$_NORMAL - NORMAL COMPLETION.
                                 032A   ;-
                                 032A
                                 032A   EXE$NUMTIM::                                    ;CONVERT TO NUMERIC TIME
                           00FC  032A           .WORD   ^M<R2,R3,R4,R5,R6,R7>           ;ENTRY MASK
        57    04 AC        D0    032C           MOVL    NTIMBUF(AP),R7                  ;GET ADDRESS OF 7-WORD TIME BUFFER
                                 0330           IFNOWRT #7*2,(R7),10$                   ;CAN TIME BUFFER BE WRITTEN?
         50    01          3C    0336           MOVZWL  #SS$_NORMAL,R0                  ;ASSUME NORMAL COMPLETION
  51  00000000'EF          7D    0339   5$:     MOVQ    EXE$GQ_SYSTIME,R1               ;ASSUME TIME NOT SPECIFIED
  51  00000000'EF          D1    0340           CMPL    EXE$GQ_SYSTIME,R1               ;VERIFY THAT THE VALUE ACQUIRED
                     F0    12    0347           BNEQ    5$                              ; WAS NOT BEING MODIFIED DURING
  52  00000004'EF          D1    0349           CMPL    EXE$GQ_SYSTIME+4,R2             ; THE ACQUISITION. THIS SYNCHS ACCESS BY
                     E7    12    0350           BNEQ    5$                              ; THE SECONDARY IN THE 11/782 SYSTEM.
        53    08 AC        D0    0352           MOVL    NTIMADR(AP),R3                  ;GET ADDRESS OF 64-TIME VALUE
                     1C    13    0356           BEQL    20$                             ;IF EQL NONE SPECIFIED
                                 0358           IFNORD  #8,(R3),10$                     ;CAN 64-BIT TIME VALUE BE READ?
         51    63          7D    035E           MOVQ    (R3),R1                         ;GET 64-BIT TIME VALUE
               11          18    0361           BGEQ    20$                             ;IF GEQ ABSOLUTE TIME
         52    52          CE    0363           MNEGL   R2,R2                           ;NEGATE DELTA TIME VALUE
         51    51          CE    0366           MNEGL   R1,R1                           ;
         52    00          D9    0369           SBWC    #0,R2                           ;
      04 50    00          E4    036C           BBSC    #0,R0,20$                       ;INDICATE DELTA TIME VALUE
         50    0C          3C    0370   10$:    MOVZWL  #SS$_ACCVIO,R0                  ;SET ACCESS VIOLATION
                           04    0373           RET                                     ;
                                 0374
                                 0374   ;
                                 0374   ; R1 AND R2 CONTAIN 64-BIT ABSOLUTE TIME VALUE IN UNITS OF TENTHS OF MICRO-
                                 0374   ; SECONDS. CALCULATE DAYS PAST BASE TIME AND FRACTION OF DAY BY DIVIDING
                                 0374   ; BY 864000000000 WHICH IS THE NUMBER OF TENTHS OF MICROSECONDS IN A DAY.
                                 0374      ; THE DIVISION IS PERFORMED IN THREE STEPS TO INSURE BOTH QUOTIENT AND
                                 0374      ; REMAINDER STAY WITHIN 32 BITS.
                                 0374      ;
                                 0374      ; CALCULATE DAYS BY DIVIDING BY 1024 AND THEN 843750000. QUOTIENT IS DAYS
                                 0374      ; AND REMAINDER IS FRACTION OF DAY.
                                 0374      ;
                                 0374
        54   51   0A    00   EF  0374      20$:    EXTZV   #0,#10,R1,R4             ;SAVE REMAINDER FROM NEXT DIVIDE
             51   51 F6 8F   79  0379              ASHQ    #-10,R1,R1               ;DIVIDE BY 1024
   52   51   51  324A9A70 8F  7B  037E              EDIV    #843750000,R1,R1,R2      ;CALCULATE DAYS AND FRACTION OF DAY
                                 0387
                                 0387      ;
                                 0387      ; R1 CONTAINS DAYS PAST BASE TIME, R2 PLUS R4 CONTAIN FRACTION OF DAY.
                                 0387      ; R2 CONTAINS PART OF FRACTION IN UNITS OF 864000000000/1024 AND
                                 0387      ; R4 CONTAINS REMAINDER IN UNITS OF TENTHS OF MICROSECONDS.
                                 0387      ;
                                 0387      ; CALCULATE FRACTION OF DAY IN HUNDREDTHS OF SECONDS BY DIVIDING BY
                                 0387      ; 100000 WHICH IS THE NUMBER OF TENTHS OF MICROSECONDS IN A HUNDRETH
                                 0387      ; OF A SECOND.
                                 0387      ;
                                 0387
                   53        D4  0387              CLRL    R3                       ;CLEAR HIGH PART OF DIVIDEND
             52    52   0A   79  0389              ASHQ    #10,R2,R2                ;CONVERT BACK TO TENTHS OF MICROSECONDS
                   52   54   C8  038D              BISL    R4,R2                    ;ADD REMAINDER BACK
   52   55   52  000186A0 8F  7B  0390              EDIV    #100000,R2,R5,R2         ;CALCULATE FRACTION OF DAY IN HUNDREDTHS
                                 0399
                                 0399      ;
                                 0399      ; R1 CONTAINS DAYS PAST THE BASE TIME AND R5 CONTAINS THE FRACTION OF DAY
                                 0399      ; IN HUNDREDTHS OF A SECOND.
                                 0399      ;
                                 0399
             7E 50      00   E3  0399              BBCS    #0,R0,70$                ;IF CLR, DELTA TIME SPECIFIED
                                 039D
                                 039D      ;
                                 039D      ; ADD TIME OFFSET SO THAT DAY IS RELATIVE TO 1-JAN-1501.
                                 039D      ;
                                 039D
             51  0001FE98 8F  C0  039D              ADDL    #TIMOFF1,R1              ;ADD TIME OFFSET
                                 03A4
                                 03A4      ;
                                 03A4      ; CALCULATE NUMBER OF QUADRICENTURIES THAT HAVE PAST SINCE 1501.
                                 03A4      ;
                                 03A4
                   52        D4  03A4              CLRL    R2                       ;CLEAR HIGH PART OF DIVIDEND
   52   51   51  00023AB1 8F  7B  03A6              EDIV    #QUADRIDAYS,R1,R1,R2     ;CALCULATE NUMBER OF QUADRICENTURIES
                                 03AF
                                 03AF      ;
                                 03AF      ; R1 CONTAINS THE NUMBER OF QUADRICENTURIES AND R2 CONTAINS THE NUMBER OF
                                 03AF      ; DAYS INTO THE NEXT QUADRICENTURY. CALCULATE THE NUMBER OF CENTURIES BY
                                 03AF      ; CONVERTING TO QUARTER DAYS INTO NEXT QUADRICENTURY AND THEN DIVIDING BY
                                 03AF      ; THE AVERAGE NUMBER OF QUARTER DAYS IN A CENTURY.
                                 03AF      ;
                                 03AF
                   52   04   C4  03AF              MULL    #4,R2                    ;CALCULATE NUMBER OF QUARTER DAYS
                        53   D4  03B2              CLRL    R3                       ;CLEAR HIGH PART OF DIVIDEND
   53   52   52  00023AB1 8F  7B  03B4              EDIV    #QDAYSPCENT,R2,R2,R3     ;CALCULATE NUMBER OF CENTURIES
                                 03BD
```

```
                                        03BD   ;
                                        03BD   ; R2 CONTAINS THE NUMBER OF CENTURIES AND R3 CONTAINS THE NUMBER OF QUARTER
                                        03BD   ; DAYS INTO THE NEXT CENTURY.
                                        03BD   ;
                                        03BD   ; CALCULATE YEARS BY DISCARDING ANY FRACTION OF A DAY, ADDING 3/4'THS OF A
                                        03BD   ; DAY, AND DIVIDING BY THE AVERAGE NUMBER OF DAYS IN A YEAR. THE LEAP DAY
                                        03BD   ; OF EACH FOUR YEAR CYCLE IS FORCED INTO THE FOURTH YEAR.
                                        03BD   ;
                                        03BD
                             54   D4    03BD            CLRL    R4                          ;CLEAR HIGH PART OF DIVIDEND
                        53   03   C8    03BF            BISL    #3,R3                       ;TRUNCATE FRACTION AND ADD 3/4'THS OF DAY
  54   53   53    000005B5 8F     7B    03C2            EDIV    #QDAYSPYEAR,R3,R3,R4        ;CALCULATE NUMBER OF YEARS
                        54   04   C6    03CB            DIVL    #4,R4                       ;CALCULATE NUMBER OF DAYS MINUS ONE
                             54   D6    03CE            INCL    R4                          ;CONVERT TO ACTUAL JULIAN DAY OF YEAR
                                        03D0
                                        03D0   ;
                                        03D0   ; R1 CONTAINS NUMBER OF QUADRICENTURIES.
                                        03D0   ; R2 CONTAINS NUMBER OF CENTURIES.
                                        03D0   ; R3 CONTAINS NUMBER OF YEARS.
                                        03D0   ; R4 CONTAINS JULIAN DAY OF YEAR.
                                        03D0   ;
                                        03D0   ; CALCULATE ACTUAL YEAR.
                                        03D0   ;
                                        03D0
                     51   6241    DE    03D0            MOVAL   (R2)[R1],R1                 ;COMBINE CENTURIES AND QUADRICENTURIES
                        51   32   C4    03D4            MULL    #50,R1                      ;CALCULATE NUMBER OF DOUBLE CENTURIES
               51   05DD C341     3E    03D7            MOVAW   1501(R3)[R1],R1             ;CALCULATE ACTUAL YEAR
                        87   51   B0    03DD            MOVW    R1,(R7)+                    ;STORE YEAR
                                        03E0
                                        03E0   ;
                                        03E0   ; TEST FOR NONLEAP YEAR AND BIAS DAY IF AFTER 28-FEB.
                                        03E0   ;
                                        03E0
                        51   03   D3    03E0            BITL    #3,R1                       ;YEAR MULTIPLE OF 4?
                             14   12    03E3            BNEQ    30$                         ;IF NEQ NO
                             52   D4    03E5            CLRL    R2                          ;CLEAR HIGH PART OF DIVIDEND
  52   51   51    00000064 8F     7B    03E7            EDIV    #100,R1,R1,R2               ;CALCULATE CENTURY AND YEAR IN CENTURY
                             52   D5    03F0            TSTL    R2                          ;YEAR MULTIPLE OF 100?
                             0C   12    03F2            BNEQ    40$                         ;IF NEQ NO
                        51   03   D3    03F4            BITL    #3,R1                       ;YEAR MULTIPLE OF 400?
                             07   13    03F7            BEQL    40$                         ;IF EQL YES
                        54   3B   D1    03F9   30$:     CMPL    #31+28,R4                   ;AFTER 28-FEB?
                             02   18    03FC            BGEQ    40$                         ;IF GEQ NO
                             54   D6    03FE            INCL    R4                          ;ADJUST FOR TABLE BIAS
                        51   01   D0    0400   40$:     MOVL    #1,R1                       ;INITIALIZE MONTH
               52   FBF7 CF41     9A    0403   50$:     MOVZBL  W^DATETABLE-1[R1],R2        ;GET NUMBER OF DAYS IN MONTH
                        54   52   C2    0409            SUBL    R2,R4                       ;SUBTRACT FROM JULIAN DAY
                             04   15    040C            BLEQ    60$                         ;IF LEQ CORRECT MONTH FOUND
                     F1 51   0C   F3    040E            AOBLEQ  #12,R1,50$                  ;LOOP THROUGH ALL MONTHS
                        87   51   B0    0412   60$:     MOVW    R1,(R7)+                    ;STORE MONTH
                  87    54   52   A1    0415            ADDW3   R2,R4,(R7)+                 ;STORE DAY
                             14   11    0419            BRB     80$                         ;
                                        041B
                                        041B   ;
                                        041B   ; DELTA TIME SPECIFIED - STORE RELATIVE DAY
                                        041B   ;
                                        041B
```

```
                 87   D4  041B   751 70$:     CLRL    (R7)+                      ;CLEAR YEAR AND MONTH
            87   51   B0  041D   752          MOVW    R1,(R7)+                   ;STORE DAY
  51   00002710  8F   D1  0420   753          CMPL    #10000,R1                  ;RELATIVE DAY WITHIN LIMITS?
                 06   1A  0427   754          BGTRU   80$                        ;IF GTRU YES
       50   0184 8F   3C  0429   755          MOVZWL  #SS$_IVTIME,R0             ;SET INVALID TIME
                      04  042E   756          RET                                ;
                          042F   757
                          042F   758 ;
                          042F   759 ; R5 CONTAINS FRACTION OF DAY IN HUNDREDTHS OF SECONDS.
                          042F   760 ;
                          042F   761 ; CALCULATE HOUR, MINUTE, SECOND, AND HUNDREDTH OF SECOND.
                          042F   762 ;
                          042F   763
            57   08   C0  042F   764 80$:     ADDL    #8,R7                      ;POINT TWO BYTES PAST END OF BUFFER
                 51   D4  0432   765          CLRL    R1                         ;CLEAR LOOP INDEX
    52    FC03 CF41   9A  0434   766 90$:     MOVZBL  W^TIMETABLE[R1],R2         ;GET NEXT UNIT DIVISOR
                 56   D4  043A   767          CLRL    R6                         ;CLEAR HIGH PART OF DIVIDEND
  56   55   55   52   7B  043C   768          EDIV    R2,R5,R5,R6                ;CALCULATE NEXT PART
            77   56   B0  0441   769          MOVW    R6,-(R7)                   ;STORE NEXT PART
         EC 51   02   F3  0444   770          AOBLEQ  #2,R1,90$                  ;LOOP FOR HUNDREDTHS, SECONDS, AND MINUTES
            77   55   B0  0448   771          MOVW    R5,-(R7)                   ;STORE HOUR
                      04  044B   772          RET                                ;
                          044C   773
                          044C   774          .END
```

```
                                                               J  6
SYSCVRTIM                    - SYSTEM SERVICES TO CONVERT TIME         16-SEP-1984 01:54:51  VAX/VMS Macro V04-00        Page  16
Symbol table                                                            5-SEP-1984 03:50:12  [SYS.SRC]SYSCVRTIM.MAR;1        (1)

$$T2                         = 00000007
ACVTFLG                      = 00000010
ATIMADR                      = 0000000C
ATIMBUF                      = 00000008
ATIMLEN                      = 00000004
BLANK                        = 00000020
BTIMADR                      = 00000008
BTIMBUF                      = 00000004
CENTURYDAYS                  = 00008EAC
COLON                        = 0000003A
CONVERT                        000001EA R      02
CVRTIME                        0000021C R      02
DATE                           0000003F R      02
DATETABLE                      00000000 R      02
DAY                          = 00000004
DELTA                          0000004D R      02
EXE$ASCTIM                     00000065 RG     02
EXE$BINTIM                     00000106 RG     02
EXE$GQ_SYSTIME                 ********   X    02
EXE$NUMTIM                     0000032A RG     02
HOUR                         = 00000006
HUNDREDTH                    = 0000000C
HYPHEN                       = 0000002D
IVTIME                         000001E4 R      02
MINUTE                       = 00000008
MONTH                        = 00000002
MONTHTAB                       0000000C R      02
NINE                         = 00000039
NTIMADR                      = 00000008
NTIMBUF                      = 00000004
ONE                          = 00000030
PERIOD                       = 0000002E
QDAYSPCENT                   = 00023AB1
QDAYSPYEAR                   = 000005B5
QUADRIDAYS                   = 00023AB1
QUADYEARDAYS                 = 000005B5
SECOND                       = 0000000A
SS$_ACCVIO                   = 0000000C
SS$_IVTIME                   = 00000184
SS$_NORMAL                   = 00000001
SYS$FAO                        ********   X    02
SYS$NUMTIM                     ********  GX    02
TIME                           00000052 R      02
TIMETABLE                      0000003C R      02
TIMOFF1                      = 0001FE98
TIMOFF2                      = 00016FEC
YEAR                         = 00000000

                                                  +----------------+
                                                  ! 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
Y$EXEPAGED                    0000044C  ( 1100.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
```

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

Phase                    Page faults    CPU Time       Elapsed Time
-----                    -----------    --------       ------------
Initialization                   29    00:00:00.08    00:00:01.01
Command processing              110    00:00:00.58    00:00:04.22
Pass 1                          232    00:00:05.96    00:00:20.69
Symbol table sort                 0    00:00:00.68    00:00:02.58
Pass 2                          143    00:00:01.89    00:00:05.95
Symbol table output               8    00:00:00.07    00:00:00.32
Psect synopsis output             1    00:00:00.02    00:00:00.02
Cross-reference output            0    00:00:00.00    00:00:00.00
Assembler run totals            525    00:00:09.28    00:00:34.79

The working set limit was 1500 pages.
34724 bytes (68 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 451 non-local and 39 local symbols.
774 source lines were read in Pass 1, producing 15 object records in Pass 2.
14 pages of virtual memory were used to define 12 macros.

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

Macro library name                          Macros defined
------------------                          --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                         2
_$255$DUA28:[SYSLIB]STARLET.MLB;2                      7
TOTALS (all libraries)                                 9

505 GETS were required to define 9 macros.

There were no errors, warnings or information messages.

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

0383 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY