```
EEEEEEEEEEEEEEE  DDDDDDDDDDDD     TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD     TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD     TTTTTTTTTTTTTTT
EEE              DDD        DDD         TTT
EEE              DDD        DDD         TTT
EEE              DDD        DDD         TTT
EEE              DDD        DDD         TTT
EEE              DDD        DDD         TTT
EEE              DDD        DDD         TTT
EEEEEEEEEEEE     DDD        DDD         TTT
EEEEEEEEEEEE     DDD        DDD         TTT
EEEEEEEEEEEE     DDD        DDD         TTT
EEE              DDD        DDD         TTT
EEE              DDD        DDD         TTT
EEE              DDD        DDD         TTT
EEE              DDD        DDD         TTT
EEE              DDD        DDD         TTT
EEE              DDD        DDD         TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD           TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD           TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD           TTT
```

```
EX

Mod
---
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
SYS
LBR
LIB
```

```
                                    M 4
**FILE**ID**SYSVAX





  SSSSSSSS  YY      YY    SSSSSSSS  VV      VV    AAAAAA    XX      XX
  SSSSSSSS  YY      YY    SSSSSSSS  VV      VV    AAAAAA    XX      XX
SS          YY      YY  SS          VV      VV  AA      AA  XX      XX
SS          YY      YY  SS          VV      VV  AA      AA  XX      XX
SS            YY  YY    SS          VV      VV  AA      AA    XX  XX
SS            YY  YY    SS          VV      VV  AA      AA    XX  XX
  SSSSSS        YY        SSSSSS    VV      VV  AA      AA      XX
  SSSSSS        YY        SSSSSS    VV      VV  AA      AA      XX
        SS      YY              SS  VV      VV  AAAAAAAAAA    XX  XX
        SS      YY              SS  VV      VV  AAAAAAAAAA    XX  XX
        SS      YY              SS    VV  VV    AA      AA  XX      XX      ....
        SS      YY              SS    VV  VV    AA      AA  XX      XX      ....
SSSSSSSS        YY      SSSSSSSS        VV      AA      AA  XX      XX      ....
SSSSSSSS        YY      SSSSSSSS        VV      AA      AA  XX      XX      ....

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
```

```
0001 0 %TITLE 'EDT$SYSVAX - VAX/VMS system specific storage'
0002 0 MODULE EDT$SYSVAX (                                  ! VAX/VMS system specific storage
0003 0          IDENT = 'V04-000'                           ! File: SYSVAX.B32 Edit: JBS2034
0004 0          ) =
0005 1 BEGIN
0006 1 !
0007 1 !****************************************************************************
0008 1 !*                                                                          *
0009 1 !*   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                *
0010 1 !*   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
0011 1 !*   ALL RIGHTS RESERVED.                                                   *
0012 1 !*                                                                          *
0013 1 !*   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
0014 1 !*   ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
0015 1 !*   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
0016 1 !*   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
0017 1 !*   OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
0018 1 !*   TRANSFERRED.                                                           *
0019 1 !*                                                                          *
0020 1 !*   THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
0021 1 !*   AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
0022 1 !*   CORPORATION.                                                           *
0023 1 !*                                                                          *
0024 1 !*   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
0025 1 !*   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
0026 1 !*                                                                          *
0027 1 !*                                                                          *
0028 1 !****************************************************************************
0029 1 !
0030 1
0031 1 !++
0032 1 ! FACILITY:     EDT -- The DEC Standard Editor
0033 1 !
0034 1 ! ABSTRACT:
0035 1 !
0036 1 !       This module contains system specific code for the VAX/VMS
0037 1 !       environment.
0038 1 !
0039 1 ! ENVIRONMENT:  VAX/VMS only
0040 1 !
0041 1 ! AUTHOR: Bob Kushlis, CREATION DATE: March 22, 1979
0042 1 !
0043 1 ! MODIFIED BY:
0044 1 !
0045 1 ! Bob Kushlis, 10-JUL-1979
0046 1 !       Convert the case of the file names.
0047 1 ! John Sauter, 19-Dec-1980, 02
0048 1 !       Add tracing.
0049 1 ! 2-003 - Regularize the headers.  JBS 19-Feb-1981
0050 1 ! 2-004 - Allocate an event flag for the "working" message, and make it
0051 1 !          cancel only its own timers.  JBS 19-Feb-1981
0052 1 ! 2-005 - Fix module header and certain symbols.  JBS 30-Mar-1981
0053 1 ! 2-006 - Stop the "working" message only if it is running.  JBS 02-Apr-1981
0054 1 ! 2-007 - Implement the virtual deallocation routine. TMV 6-Aug-81
0055 1 ! 2-008 - EDT$$ALO_HEAP  should return 1 if successful, 0 if not.
0056 1 !          JBS 07-Aug-1981
0057 1 ! 2-009 - Remove calls to LIB$SHOW_VM.  JBS 21-Aug-1981
```

```
1 ! 2-010 - Add date/time routine. STS 02-Sep-1981
1 ! 2-011 - Add deallocation of text string area. STS 06-Oct-1981
1 ! 2-012 - Always do deallocation of text and entity string areas. STS 06-Nov-1981
1 ! 2-013 - Add global for SET/SHOW HELP command.  SMB 16-Dec-1981
1 ! 2-014 - Revise timer AST logic.  JBS 13-Jan-1982
1 ! 2-015 - Change 32-bit line# to 48_bit.  SMB 16-Jan-1982
1 ! 2-016 - Move line number declarations to DATA.BLI.  SMB 29-Jan-1982
1 ! 2-017 - Take out extra space in date when day is single digit. STS 02-Feb-1982
1 ! 2-018 - Fix a race condition in timer AST logic.  JBS 10-Feb-1982
1 ! 2-019 - Take out call to sy$exit. STS 19-Feb-1982
1 ! 2-020 - Add edt$$z_wf_desc to deallocation list. STS 09-Mar-1982
1 ! 2-021 - Define the default startup file names.  JBS 18-Mar-1982
1 ! 2-022 - Correct the length of EDTINI.  JBS 08-Apr-1982
1 ! 2-023 - Change the HELP file default name.  SMB 10-May-1982
1 ! 2-024 - Put the default startup file on SYS$LIBRARY.  JBS 08-Jun-1982
1 ! 2-025 - Erase the working message line in STOP_WKINGMSG.  SMB 28-Jun-1982
1 ! 2-026 - New implementation of defined keys.  JBS 12-Aug-1982
1 ! 2-027 - Change the command file name.  JBS 23-Aug-1982
1 ! 2-028 - Change the command file name again.  JBS 17-Sep-1982
1 ! 2-029 - Change EDT$$FMT_LIT to EDT$$FMT_STR.  JBS 05-Oct-1982
1 ! 2-030 - Remove deallocation of edt$$z_wf_desc. STS 11-Nov-1982
1 ! 2-031 - Add a hack to debug insufficient memory problems.  JBS 15-Nov-1982
1 ! 2-032 - Add a call to deassign terminal channel. STS 21-Dec-1982
1 ! 2-033 - Deassign the terminal channel before halting trace, since the
1 !            terminal deassign may output a keypad setting.  JBS 26-Apr-1983
1 ! 2-034 - Improve the appearance of the listing.  JBS 17-Jun-1983
1 !--
1
```

```
     87         0086    1 %SBTTL 'Declarations'
     88         0087    1 !
     89         0088    1 ! TABLE OF CONTENTS:
     90         0089    1 !
     91         0090    1
     92         0091    1 REQUIRE 'EDTSRC:TRAROUNAM';
     93         0530    1
     94         0531    1 FORWARD ROUTINE
     95         0532    1     EDT$$INTER_ERR : NOVALUE,
     96         0533    1     EDT$$SYS_EXI : NOVALUE,
     97         0534    1     EDT$$GET_DATE : NOVALUE,
     98         0535    1     EDT$$ALO_HEAP,
     99         0536    1     EDT$$DEA_HEAP : NOVALUE,
    100         0537    1     EDT$$DEA_ALLHEAP : NOVALUE,
    101         0538    1     WORKAST : NOVALUE,
    102         0539    1     EDT$$START_WKINGMSG : NOVALUE,
    103         0540    1     EDT$$STOP_WKINGMSG : NOVALUE,
    104         0541    1     EDT$$MSG_TOSTR : NOVALUE;
    105         0542    1
    106         0543    1 !
    107         0544    1 ! INCLUDE FILES:
    108         0545    1 !
    109         0546    1
    110         0547    1 REQUIRE 'EDTSRC:SYSSYM';
    111         0577    1
    112         0578    1 REQUIRE 'EDTSRC:EDTREQ';
    113         0713    1
    114         0714    1 LIBRARY 'EDTSRC:KEYPADDEF';
    115         0715    1
    116         0716    1 REQUIRE 'TRACESEL';
    117         0747    1
    118         0748    1 REQUIRE 'EDTSRC:TRACEMAC';
    119         0975    1
    120         0976    1 !
    121         0977    1 ! MACROS:
    122         0978    1 !
    123         0979    1 !       NONE
    124         0980    1 !
    125         0981    1 ! EQUATED SYMBOLS:
    126         0982    1 !
    127         0983    1 !       NONE
    128         0984    1 !
    129         0985    1 ! OWN STORAGE:
    130         0986    1 !
    131         0987    1
    132         0988    1 GLOBAL
    133         0989    1     EDT$$T_HDEF_NAM : BLOCK [14, BYTE] INITIAL (BYTE (13, 'SYS$HELP:.HLB')),
    134         0990    1     EDT$$T_HDEF_FILE : BLOCK [8, BYTE] INITIAL (BYTE (7, 'EDTHELP')),
    135         0991    1     EDT$$T_HELP_NAM : BLOCK [NAM$C_MAXRSS, BYTE] INITIAL (BYTE ('EDTHELP')),
    136         0992    1     EDT$$G_HELP_NAMLEN : INITIAL (7),
    137         0993    1     EDT$$G_HELP_SET : INITIAL (0),
    138         0994    1     EDT$$Z_LBR_INDEX,                           ! LBR Control index for HELP
    139         0995    1     EDT$$T_CMD_NAM_DEF1 : BLOCK [7, BYTE] INITIAL (BYTE (6, 'EDTSYS')), ! Command file name
    140         0996    1     EDT$$T_CMD_NAM_DEF2 : BLOCK [17, BYTE] INITIAL (BYTE (16, 'SYS$LIBRARY:.EDT')),
    141         0997    1                                                 ! Command file default name
    142         0998    1     EDT$$T_CMD_NAM_DEF3 : BLOCK [7, BYTE] INITIAL (BYTE (6, 'EDTINI')), ! Alternate command file name
    143         0999    1     EDT$$T_CMD_NAM_DEF4 : BLOCK [5, BYTE] INITIAL (BYTE (4, '.EDT'));   ! Alternate command file default nam
```

```

OWN
    MESSAGE : VECTOR [12, BYTE] INITIAL (BYTE ('Bug check   '));

OWN
    DEL_TIME : VECTOR [2] INITIAL (-5000000, -1),
    WORKING_EFN,
    WORK_MESSAGE_RUNNING : VOLATILE INITIAL (0);

OWN
    MEM_USE : INITIAL (0),                         ! Currently allocated memory amount
    MEM_LIMIT : INITIAL (1000000000);              ! Limit on amount of memory to allocate

!
! EXTERNAL REFERENCES:
!

EXTERNAL ROUTINE
    EDT$$TI_WRSTR,
    EDT$$OUT_FMTBUF,
    EDT$$SC_POSCSIF,
    EDT$$SC_ERATOEOL,
    EDT$$TI_WRLN : NOVALUE,
    EDT$$FMT_STR : NOVALUE,
    LIB$GET_VM,
    LIB$FREE_VM,
    SYS$EXIT,
    LIB$DATE_TIME,
    LIB$GET_EF,
    LIB$FREE_EF;

!+
! Define the RABs to be used by EDT
!-

GLOBAL
    EDT$$Z_SYS_PRIRAB : $RAB_DECL,
    EDT$$Z_SYS_JOURAB : $RAB_DECL,
    EDT$$Z_SYS_CMDRAB : $RAB_DECL,
    EDT$$Z_SYS_ALTRAB : $RAB_DECL;

EXTERNAL
    EDT$$A_FMT_WRRUT,                              ! Output format routine
    EDT$$G_MESSAGE_LINE,                           ! Command/message line
    EDT$$G_SECOND : VOLATILE,                      ! Set to 1 once a second for WORKING message
    EDT$$G_WORKCOUNT;                              ! Counter to support WORKING message

```

```
;  1 %SBTTL 'EDT$$INTER_ERR  - internal error'
;  1
;  1 GLOBAL ROUTINE EDT$$INTER_ERR                                  ! Internal error
;  1     : NOVALUE =
;  1
;  1 !++
;  1 ! FUNCTIONAL DESCRIPTION:
;  1 !
;  1 !       If an internal error is detected in EDT, come here to
;  1 !       print a cryptic message and bail out.
;  1 !
;  1 ! FORMAL PARAMETERS:
;  1 !
;  1 !       NONE
;  1 !
;  1 ! IMPLICIT INPUTS:
;  1 !
;  1 !       NONE
;  1 !
;  1 ! IMPLICIT OUTPUTS:
;  1 !
;  1 !       NONE
;  1 !
;  1 ! ROUTINE VALUE:
;  1 !
;  1 !       NONE
;  1 !
;  1 ! SIDE EFFECTS:
;  1 !
;  1 !       Never returns to its caller.
;  1 !
;  1 !--
;  1
;  2     BEGIN
;  2     MESSAGES ((INTERERR));
;  2     SIGNAL_STOP (EDT$_INTERERR);
;  1     END;                                                  ! of routine EDT$$INTER_ERR


                                                               .TITLE  EDT$SYSVAX EDT$SYSVAX - VAX/VMS system specific
;                                                                                  storage
                                                               .IDENT  \V04-000\

                                                               .PSECT  _EDT$DATA,NOEXE,  PIC,2

                                            0D  00000 EDT$$T_HDEF_NAM::
                                                               .BYTE   13
      42  4C  48  2E  3A  50  4C  45  48  24  53  59  53  00001        .ASCII  \SYS$HELP:.HLB\                         ;
                                                0000E          .BLKB   2
                                            07  00010 EDT$$T_HDEF_FILE::
                                                               .BYTE   7                                               ;
                        50  4C  45  48  54  44  45  00011      .ASCII  \EDTHELP\                                       ;
                        50  4C  45  48  54  44  45  00018 EDT$$T_HELP_NAM::
                                                               .ASCII  \EDTHELP\                                       ;
                                                0001F          .BLKB   248
                                                00117          .BLKB   1
                                      00000007  00118 EDT$$G_HELP_NAMLEN::
```

```
                                                                 .LONG   7                                      ;
                                        00000000  0011C EDT$$G_HELP_SET::
                                                                 .LONG   0                                      ;
                                                  00120 EDT$$Z_LBR_INDEX::
                                                                 .BLKB   4
                                              06  00124 EDT$$T_CMD_NAM_DEF1::
                                                                 .BYTE   6                                      ;
                               53 59 53 54 44 45  00125          .ASCII  \EDTSYS\                               ;
                                                  0012B          .BLKB   1
                                              10  0012C EDT$$T_CMD_NAM_DEF2::
                                                                 .BYTE   16                                     ;
44 45 2E 3A 59 52 41 52 42 49 4C 24 53 59 53      0012D          .ASCII  \SYS$LIBRARY:.EDT\                     ;
                                              54  0013C                                                         ;
                                                  0013D          .BLKB   3
                                              06  00140 EDT$$T_CMD_NAM_DEF3::
                                                                 .BYTE   6                                      ;
                               49 4E 49 54 44 45  00141          .ASCII  \EDTINI\                               ;
                                                  00147          .BLKB   1
                                              04  00148 EDT$$T_CMD_NAM_DEF4::
                                                                 .BYTE   4                                      ;
                                     54 44 45 2E  00149          .ASCII  \.EDT\                                 ;
                                                  0014D          .BLKB   3
         20 20 20 6B 63 65 68 63 20 67 75 42      00150 MESSAGE: .ASCII  \Bug check   \                          ;
                              FFFFFFFF  FFB3B4C0  0015C DEL_TIME:
                                                                 .LONG   -5000000, -1                           ;
                                                  00164 WORKING_EFN:
                                                                 .BLKB   4
                                        00000000  00168 WORK_MESSAGE_RUNNING:
                                                                 .LONG   0                                      ;
                                        00000000  0016C MEM_USE: .LONG   0                                      ;
                                        3B9ACA00  00170 MEM_LIMIT:
                                                                 .LONG   1000000000                             ;
                                                  00174 EDT$$Z_SYS_PRIRAB::
                                                                 .BLKB   68
                                                  001B8 EDT$$Z_SYS_JOURAB::
                                                                 .BLKB   68
                                                  001FC EDT$$Z_SYS_CMDRAB::
                                                                 .BLKB   68
                                                  00240 EDT$$Z_SYS_ALTRAB::
                                                                 .BLKB   68

                                                                 .EXTRN  EDT$$TI_WRSTR, EDT$$OUT_FMTBUF
                                                                 .EXTRN  EDT$$SC_POSCSIF
                                                                 .EXTRN  EDT$$SC_ERATOEOL
                                                                 .EXTRN  EDT$$TI_WRLN, EDT$$FMT_STR
                                                                 .EXTRN  LIB$GET_VM, LIB$FREE_VM
                                                                 .EXTRN  SYS$EXIT, LIB$DATE_TIME
                                                                 .EXTRN  LIB$GET_EF, LIB$FREE_EF
                                                                 .EXTRN  EDT$$A_FMT_WRRUT
                                                                 .EXTRN  EDT$$G_MESSAGE_LINE
                                                                 .EXTRN  EDT$$G_SECOND, EDT$$G_WORKCOUNT
                                                                 .EXTRN  EDT$_INTERERR

                                                                 .PSECT  _EDT$CODE,NOWRT,  SHR,  PIC,2

                                            0000  00000          .ENTRY  EDT$$INTER_ERR, Save nothing           ; 1049
                              00000000G   8F  DD  00002          PUSHL   #EDT$_INTERERR                         ; 1082
```

```
                         00000000G  00              01  FB 00008         CALLS    #1, LIB$STOP                                      ;
                                                        04 0000F         RET                                                        ; 1083

; Routine Size:  16 bytes,    Routine Base:  _EDT$CODE + 0000

;  229           1084  1
```

```
  1 %SBTTL 'EDT$$SYS_EXI  - exit back to the operating system'
  1
  1 GLOBAL ROUTINE EDT$$SYS_EXI (                          ! Exit back to the operating system
  1     STATUS                                             ! Exit status code
  1     ) : NOVALUE =
  1
  1 !++
  1 ! FUNCTIONAL DESCRIPTION:
  1 !
  1 !       Final clean-up
  1 !
  1 ! FORMAL PARAMETERS:
  1 !
  1 !       STATUS          Exit status code.  1 = normal.
  1 !
  1 ! IMPLICIT INPUTS:
  1 !
  1 !       NONE
  1 !
  1 ! IMPLICIT OUTPUTS:
  1 !
  1 !       NONE
  1 !
  1 ! ROUTINE VALUE:
  1 !
  1 !       NONE
  1 !
  1 ! SIDE EFFECTS:
  1 !
  1 !       Deallocates all heap memory
  1 !
  1 !--
  1
  2     BEGIN
  2
  2     EXTERNAL ROUTINE
  2         EDT$TI_DEAS;
  2
  2     MESSAGES ((EDITORABO));
  2     EDT$$DEA_ALLHEAP ();                               ! Deallocate all heap storage
  2     EDT$$TI_DEAS ();                                   ! Deassign the terminal channel
  2
L 2 %IF EDT$$TR_ACT
U 2 %THEN
U 2     BEGIN
U 2
U 2     LOCAL
U 2         TRACE_STATUS;
U 2
U 2     EXTERNAL ROUTINE
U 2         EDT$$TR_CLS : ADDRESSING_MODE (GENERAL);
U 2
U 2     EXTERNAL
U 2         EDT$$L_TR_INFLG;
U 2
U 2     $$TRACE (EDT$$TR_EXI, EDT$$TR_SEXI, 0, 0);
U 2     TRACE_STATUS = EDT$$TR_CLS (EDT$$L_TR_INFLG);
```

```
;   288          U 1142  2
;   289          U 1143  2        IF ( NOT .TRACE_STATUS) THEN SIGNAL_STOP (.TRACE_STATUS);
;   290          U 1144  2
;   291          U 1145  2        END;
;   292            1146  2  %FI
;   293            1147  2
;   294            1148  2        IF ( NOT .STATUS) THEN SIGNAL_STOP (EDT$_EDITORABO);
;   295            1149  2
;   296            1150  1        END;                                          ! of routine EDT$$SYS_EXI


                                                                                .EXTRN  EDT$$TI_DEAS, EDT$_EDITORABO

                                                   0000 00000                   .ENTRY  EDT$$SYS_EXI, Save nothing                              ; 1087
                  0000V  CF                  00  FB 00002                       CALLS   #0, EDT$$DEA_ALLHEAP                                    ; 1124
              00000000G  00                  00  FB 00007                       CALLS   #0, EDT$$TI_DEAS                                        ; 1125
                         0D          04      AC  E8 0000E                       BLBS    STATUS, 1$                                              ; 1148
                              00000000G      8F  DD 00012                       PUSHL   #EDT$_EDITORABO                                         ;
              00000000G  00                  01  FB 00018                       CALLS   #1, LIB$STOP                                            ;
                                                 04 0001F 1$:                   RET                                                             ; 1150

; Routine Size:  32 bytes,    Routine Base:  _EDT$CODE + 0010


;   297            1151  1
```

```
1 %SBTTL 'EDT$$GET_DATE  - return the date as an ASCII string'
1
1 GLOBAL ROUTINE EDT$$GET_DATE (                          ! Return the date as an ASCII string
1     LEN,                                                ! Length of the buffer to return the date in
1     BUFFER                                              ! Address of the buffer
1     ) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !      Return the date and time as an ASCII string.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !  LEN                    Length of the buffer in which the date is returned
1 !
1 !  BUFFER                 Address of that buffer.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !      NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !      NONE
1 !
1 ! ROUTINE VALUE:
1 !
1 !      NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !      NONE
1 !
1 !--
1
2     BEGIN
2
2     LOCAL
2         DATE_DESC : BLOCK [8, BYTE],
2         DATE_TIME_STATUS;
2
2     MAP
2         BUFFER : REF VECTOR [, BYTE];
2
2 !+
2 ! Set up the descriptor for the LIB$ routine
2 !-
2     DATE_DESC [DSC$W_LENGTH] = 24;
2     DATE_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
2     DATE_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
2     DATE_DESC [DSC$A_POINTER] = BUFFER [1];
2 !+
2 ! Now call the routine to get the date and time as string
2 !-
2     DATE_TIME_STATUS = LIB$DATE_TIME (DATE_DESC);
2 !+
;  356          1209  2 !_Make sure we got a good status from the library routine else stop
;  357          1210  2 !-
;  358          1211  2
;  359          1212  2     IF ( NOT .DATE_TIME_STATUS) THEN SIGNAL_STOP (.DATE_TIME_STATUS);
;  360          1213  2
;  361          1214  2     BUFFER [0] = %C' ';                               ! begin with a space
;  362          1215  2     BUFFER [21] = %C' ';                              ! and end with a space
;  363          1216  2
;  364          1217  3     IF (.BUFFER [1] EQL %C' ')
;  365          1218  2     THEN
;  366          1219  3         BEGIN
;  367          1220  3         CH$MOVE (20, BUFFER [2], BUFFER [1]);     ! shift left one space
;  368          1221  3         .LEN = 21;
;  369          1222  3         END
;  370          1223  2     ELSE
;  371          1224  2         .LEN = 22;
;  372          1225  2
;  373          1226  1     END;                                          ! of routine EDT$$GET_DATE



                                              003C 00000          .ENTRY   EDT$$GET_DATE, Save R2,R3,R4,R5                    ; 1154
                          5E              04   C2 00002           SUBL2    #4, SP                                             ;
                             010E0018     8F   DD 00005           PUSHL    #17694744                                          ; 1200
                          52       08     AC   D0 0000B           MOVL     BUFFER, R2                                         ; 1203
                   04     AE       01     A2   9E 0000F           MOVAB    1(R2), DATE_DESC+4                                 ;
                                          5E   DD 00014           PUSHL    SP                                                 ; 1207
             00000000G    00              01   FB 00016           CALLS    #1, LIB$DATE_TIME                                  ;
                          09              50   E8 0001D           BLBS     DATE_TIME_STATUS, 1$                               ; 1212
                                          50   DD 00020           PUSHL    DATE_TIME_STATUS                                   ;
             00000000G    00              01   FB 00022           CALLS    #1, LIB$STOP                                       ;
                          62              20   90 00029 1$:       MOVB     #32, (R2)                                          ; 1214
                   15     A2              20   90 0002C           MOVB     #32, 21(R2)                                        ; 1215
                          20       01     A2   91 00030           CMPB     1(R2), #32                                         ; 1217
                                          0B   12 00034           BNEQ     2$                                                 ;
      01     A2    02     A2              14   28 00036           MOVC3    #20, 2(R2), 1(R2)                                  ; 1220
                   04     BC              15   D0 0003C           MOVL     #21, @LEN                                          ; 1221
                                               04 00040           RET                                                         ; 1217
                   04     BC              16   D0 00041 2$:       MOVL     #22, @LEN                                          ; 1224
                                               04 00045           RET                                                         ; 1226

; Routine Size:  70 bytes,    Routine Base:  _EDT$CODE + 0030


;  374          1227  1
;  376      1228  1 %SBTTL 'EDT$$ALO_HEAP  - Allocate memory'
;  377      1229  1
;  378      1230  1 GLOBAL ROUTINE EDT$$ALO_HEAP (                         ! Allocate memory
;  379      1231  1     SIZE,                                              ! Number of bytes to allocate
;  380      1232  1     ADDRESS                                            ! Address of allocated space
;  381      1233  1     ) =
;  382      1234  1
;  383      1235  1 !++
;  384      1236  1 ! FUNCTIONAL DESCRIPTION:
;  385      1237  1 !
;  386      1238  1 !       Allocate memory.
;  387      1239  1 !
;  388      1240  1 ! FORMAL PARAMETERS:
;  389      1241  1 !
;  390      1242  1 !  SIZE                 The number of bytes to allocate
;  391      1243  1 !
;  392      1244  1 !  ADDRESS              Place to store address of allocated space
;  393      1245  1 !
;  394      1246  1 ! IMPLICIT INPUTS:
;  395      1247  1 !
;  396      1248  1 !       NONE
;  397      1249  1 !
;  398      1250  1 ! IMPLICIT OUTPUTS:
;  399      1251  1 !
;  400      1252  1 !       NONE
;  401      1253  1 !
;  402      1254  1 ! ROUTINE VALUE:
;  403      1255  1 !
;  404      1256  1 !       1 = memory successfully allocated, 0 = out of memory.
;  405      1257  1 !
;  406      1258  1 ! SIDE EFFECTS:
;  407      1259  1 !
;  408      1260  1 !       NONE
;  409      1261  1 !
;  410      1262  1 !--
;  411      1263  1
;  412      1264  2     BEGIN
;  413      1265  2
;  414      1266  2     LOCAL
;  415      1267  2         GET_VM_STATUS;
;  416      1268  2
;  417      1269  2     IF ((.MEM_USE + ..SIZE) GTR .MEM_LIMIT) THEN RETURN (0);
;  418      1270  2
;  419      1271  2     GET_VM_STATUS = LIB$GET_VM (.SIZE, .ADDRESS);
;  420      1272  2
;  421      1273  2     IF ( NOT .GET_VM_STATUS) THEN RETURN (0);
;  422      1274  2
;  423      1275  2     MEM_USE = .MEM_USE + ..SIZE;
;  424      1276  2     RETURN (1);
;  425      1277  1     END;                                               ! of routine EDT$$ALO_HEAP



                                       0004 00000          .ENTRY  EDT$$ALO_HEAP, Save R2                          ; 1230
                         52 00000000'  EF   9E 00002          MOVAB   MEM_USE, R2                                     ;
```

```
              50        62      04    BC  C1 00009           ADDL3   @SIZE, MEM_USE, R0                        ; 1269
                   04   A2            50  D1 0000E           CMPL    R0, MEM_LIMIT                             ;
                        16            14  00012              BGTR    1$                                        ;
                        7E      04    AC  7D 00014           MOVQ    SIZE, -(SP)                               ; 1271
              00000000G 00            02  FB 00018           CALLS   #2, LIB$GET_VM                            ;
                        08            50  E9 0001F           BLBC    GET_VM_STATUS, 1$                         ; 1273
                        62      04    BC  C0 00022           ADDL2   @SIZE, MEM_USE                            ; 1275
                        50            01  D0 00026           MOVL    #1, R0                                    ; 1276
                                          04 00029           RET                                               ;
                                      50  D4 0002A 1$:       CLRL    R0                                        ; 1277
                                          04 0002C           RET                                               ;

; Routine Size:  45 bytes,    Routine Base:  _EDT$CODE + 0076


;  426          1278  1
;  428        1279  1 %SBTTL 'EDT$$DEA_HEAP  - Deallocate memory'
;  429        1280  1
;  430        1281  1 GLOBAL ROUTINE EDT$$DEA_HEAP (                           ! Deallocate memory
;  431        1282  1     SIZE,                                                ! Number of bytes to deallocate
;  432        1283  1     ADDRESS                                              ! Address of deallocated space
;  433        1284  1     ) : NOVALUE =
;  434        1285  1
;  435        1286  1 !++
;  436        1287  1 ! FUNCTIONAL DESCRIPTION:
;  437        1288  1 !
;  438        1289  1 !       Deallocate memory.
;  439        1290  1 !
;  440        1291  1 ! FORMAL PARAMETERS:
;  441        1292  1 !
;  442        1293  1 !  SIZE                The number of bytes to deallocate
;  443        1294  1 !
;  444        1295  1 !  ADDRESS             Place to store address of deallocated space
;  445        1296  1 !
;  446        1297  1 ! IMPLICIT INPUTS:
;  447        1298  1 !
;  448        1299  1 !       NONE
;  449        1300  1 !
;  450        1301  1 ! IMPLICIT OUTPUTS:
;  451        1302  1 !
;  452        1303  1 !       NONE
;  453        1304  1 !
;  454        1305  1 ! ROUTINE VALUE:
;  455        1306  1 !
;  456        1307  1 !       NONE
;  457        1308  1 !
;  458        1309  1 ! SIDE EFFECTS:
;  459        1310  1 !
;  460        1311  1 !       Signals on error.
;  461        1312  1 !
;  462        1313  1 !--
;  463        1314  1
;  464        1315  2     BEGIN
;  465        1316  2
;  466        1317  2     LOCAL
;  467        1318  2         FREE_VM_STATUS;
;  468        1319  2
;  469        1320  2     FREE_VM_STATUS = LIB$FREE_VM (.SIZE, .ADDRESS);
;  470        1321  2
;  471        1322  2     IF ( NOT .FREE_VM_STATUS) THEN SIGNAL_STOP (.FREE_VM_STATUS);
;  472        1323  2
;  473        1324  2     MEM_USE = .MEM_USE - ..SIZE;
;  474        1325  2     ASSERT (.MEM_USE GEQ 0);
;  475        1326  1     END;                                                 ! of routine EDT$$DEA_HEAP



                                      0000 00000        .ENTRY  EDT$$DEA_HEAP, Save nothing                        ; 1281
                    7E        04    AC  7D 00002        MOVQ    SIZE, -(SP)                                         ; 1320
          00000000G 00              02  FB 00006        CALLS   #2, LIB$FREE_VM
                    09              50  E8 0000D        BLBS    FREE_VM_STATUS, 1$                                  ; 1322
```

```
                                          50  DD 00010          PUSHL   FREE_VM_STATUS                          ;
                     00000000G  00        01  FB 00012          CALLS   #1, LIB$STOP                            ;
                     00000000'  EF    04  BC  C2 00019  1$:     SUBL2   @SIZE, MEM_USE                          ; 1324
                                          07  18 00021          BGEQ    2$                                      ; 1325
                     00000000G  00        00  FB 00023          CALLS   #0, EDT$$INTER_ERR                      ;
                                              04 0002A  2$:     RET                                             ; 1326

; Routine Size:  43 bytes,    Routine Base:  _EDT$CODE + 00A3


;  476          1327  1
```

```
1 %SBTTL 'EDT$$DEA_ALLHEAP  - Deallocate all memory'
1
1 GLOBAL ROUTINE EDT$$DEA_ALLHEAP                              ! Deallocate all memory
1     : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Deallocate all memory allocated by calls to LIB$GET_VM .
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       EDT$$A_FST_AVLN
1 !       EDT$$A_FST_SCRPTR
1 !       EDT$$A_BUF_LST
1 !       EDT$$A_TRN_TBL
1 !       EDT$$A_US_ENT
1 !       EDT$$A_US_TXT
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       EDT$$A_FST_AVLN
1 !       EDT$$A_FST_SCRPTR
1 !       EDT$$A_BUF_LST
1 !
1 ! ROUTINE VALUE:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Signals on error.
1 !
1 !--
1
2   BEGIN
2
2   EXTERNAL ROUTINE
2       STR$FREE1_DX,
2       EDT$$CAN_KDEF;                                     ! Cancel a key definition
2
2   EXTERNAL
2       EDT$$A_FST_AVLN,
2       EDT$$A_FST_SCRPTR,
2       EDT$$A_BUF_LST,
2       EDT$$A_TRN_TBL : VECTOR,
2       EDT$$A_US_ENT : VECTOR,
2       EDT$$A_US_TXT : VECTOR;
2
2   LOCAL
2       NEW_PTR : REF SCREEN_LINE,
2       NEW_BUF : REF TBCB_BLOCK,
2       LEN,
2           GET_VM_STATUS;
2
2   !+
2   ! Deallocate all buffer headers
2   !-
2       NEW_BUF = .EDT$$A_BUF_LST;
2
2       WHILE (.NEW_BUF NEQA 0) DO
3           BEGIN
3           LEN = .NEW_BUF [TBCB_NAME_LEN] + TBCB_SIZE;
3           EDT$$A_BUF_LST = .NEW_BUF [TBCB_NEXT_BUF];
3           EDT$$DEA_HEAP (LEN, NEW_BUF);
3           NEW_BUF = .EDT$$A_BUF_LST;
2           END;
2
2   !+
2   ! Deallocate memory used for screen data structure.
2   !-
2       NEW_PTR = .EDT$$A_FST_SCRPTR;
2
2       WHILE (.NEW_PTR NEQA 0) DO
3           BEGIN
3           EDT$$A_FST_SCRPTR = .NEW_PTR [SCR_NXT_LINE];
3           EDT$$DEA_HEAP (%REF (SCR_SIZE), NEW_PTR);
3           NEW_PTR = .EDT$$A_FST_SCRPTR;
2           END;
2
2       NEW_PTR = .EDT$$A_FST_AVLN;
2
2       WHILE (.NEW_PTR NEQA 0) DO
3           BEGIN
3           EDT$$A_FST_AVLN = .NEW_PTR [SCR_NXT_LINE];
3           EDT$$DEA_HEAP (%REF (SCR_SIZE), NEW_PTR);
3           NEW_PTR = .EDT$$A_FST_AVLN;
2           END;
2
2   !+
2   ! Deallocate virtual storage allocated for entities
2   !-
2
2       INCR ENT_NUM FROM 0 TO 3 DO
3           BEGIN
3           LEN = CH$RCHAR (.EDT$$A_US_ENT [.ENT_NUM]);
3           EDT$$DEA_HEAP (%REF (.LEN + 1), EDT$$A_US_ENT [.ENT_NUM]);
2           END;
2
2       INCR TEXT_NUM FROM 0 TO 1 DO
3           BEGIN
3           LEN = CH$RCHAR (.EDT$$A_US_TXT [.TEXT_NUM]);
3           EDT$$DEA_HEAP (%REF (.LEN + 1), EDT$$A_US_TXT [.TEXT_NUM]);
2           END;
2
2   !+
2   ! Deallocate virtual storage reserved for the key definitions
2   !-
2
2       INCR TBL_PTR FROM 0 TO K_KPAD_HASHSIZ - 1 DO
```

```
;  592     1442 3          BEGIN
;  593     1443 3
;  594     1444 3          WHILE (.EDT$$A_TRN_TBL [.TBL_PTR] NEQA 0) DO
;  595     1445 4              BEGIN
;  596     1446 4
;  597     1447 4              LOCAL
;  598     1448 4                  KEY_PTR : REF BLOCK [, BYTE] FIELD (KEY_DEF_FIELD);
;  599     1449 4
;  600     1450 4              KEY_PTR = .EDT$$A_TRN_TBL [.TBL_PTR];
;  601     1451 4              EDT$$CAN_KDEF (.KEY_PTR [KEY_DEF_KEY]);
;  602     1452 3              END;
;  603     1453 3
;  604     1454 2          END;
;  605     1455 2
;  606     1456 1      END;                                            ! of routine EDT$$DEA_ALLHEAP


                                                        .EXTRN   STR$FREE1_DX, EDT$$CAN_KDEF
                                                        .EXTRN   EDT$$A_FST_AVLN
                                                        .EXTRN   EDT$$A_FST_SCRPTR
                                                        .EXTRN   EDT$$A_BUF_LST, EDT$$A_TRN_TBL
                                                        .EXTRN   EDT$$A_US_ENT, EDT$$A_US_TXT

                              007C 00000                .ENTRY   EDT$$DEA_ALLHEAP, Save R2,R3,R4,R5,R6          ; 1330
           56 00000000G  00     9E 00002                MOVAB    EDT$$A_BUF_LST, R6                             ;
           55 00000000G  00     9E 00009                MOVAB    EDT$$A_FST_AVLN, R5                            ;
           54 00000000G  00     9E 00010                MOVAB    EDT$$A_FST_SCRPTR, R4                          ;
           53        BB  AF     9E 00017                MOVAB    EDT$$DEA_HEAP, R3                              ;
           5E            10     C2 0001B                SUBL2    #16, SP                                        ;
      04   AE            66     D0 0001E 1$:            MOVL     EDT$$A_BUF_LST, NEW_BUF                        ; 1390
           50        04  AE     D0 00022                MOVL     NEW_BUF, R0                                    ; 1392
                         18     13 00026                BEQL     2$                                             ;
      08   AE        2C  A0     9A 00028                MOVZBL   44(R0), LEN                                    ; 1394
      08   AE            2D     C0 0002D                ADDL2    #45, LEN                                       ;
           66        26  A0     D0 00031                MOVL     38(R0), EDT$$A_BUF_LST                         ; 1395
                     04  AE     9F 00035                PUSHAB   NEW_BUF                                        ; 1396
                     0C  AE     9F 00038                PUSHAB   LEN                                            ;
           63            02     FB 0003B                CALLS    #2, EDT$$DEA_HEAP                              ;
                         DE     11 0003E                BRB      1$                                             ; 1397
      0C   AE            64     D0 00040 2$:            MOVL     EDT$$A_FST_SCRPTR, NEW_PTR                     ; 1403
           50        0C  AE     D0 00044                MOVL     NEW_PTR, R0                                    ; 1405
                         13     13 00048                BEQL     3$                                             ;
           64        04  A0     D0 0004A                MOVL     4(R0), EDT$$A_FST_SCRPTR                       ; 1407
                     0C  AE     9F 0004E                PUSHAB   NEW_PTR                                        ; 1408
      04   AE            0E     D0 00051                MOVL     #14, 4(SP)                                     ;
                     04  AE     9F 00055                PUSHAB   4(SP)                                          ;
           63            02     FB 00058                CALLS    #2, EDT$$DEA_HEAP                              ;
                         E3     11 0005B                BRB      2$                                             ; 1409
      0C   AE            65     D0 0005D 3$:            MOVL     EDT$$A_FST_AVLN, NEW_PTR                       ; 1412
           50        0C  AE     D0 00061                MOVL     NEW_PTR, R0                                    ; 1414
                         13     13 00065                BEQL     4$                                             ;
           65        04  A0     D0 00067                MOVL     4(R0), EDT$$A_FST_AVLN                         ; 1416
                     0C  AE     9F 0006B                PUSHAB   NEW_PTR                                        ; 1417
      04   AE            0E     D0 0006E                MOVL     #14, 4(SP)                                     ;
                     04  AE     9F 00072                PUSHAB   4(SP)                                          ;
           63            02     FB 00075                CALLS    #2, EDT$$DEA_HEAP                              ;
```

```
                                 E3  11 00078       BRB     3$                                  ; 1418
                                 52  D4 0007A 4$:   CLRL    ENT_NUM                             ; 1425
                  50 00000000G0042  DE 0007C 5$:   MOVAL   EDT$$A_US_ENT[ENT_NUM], R0          ; 1427
             08   AE        00  B0  9A 00084       MOVZBL  @0(R0), LEN
                                 50  DD 00089       PUSHL   R0                                  ; 1428
   04   AE   0C   AE            01  C1 0008B       ADDL3   #1, LEN, 4(SP)
                          04    AE  9F 00091       PUSHAB  4(SP)
                  63            02  FB 00094       CALLS   #2, EDT$$DEA_HEAP
        E1        52            03  F3 00097       AOBLEQ  #3, ENT_NUM, 5$                     ; 1425
                                 52  D4 0009B       CLRL    TEXT_NUM                            ; 1431
                  50 00000000G0042  DE 0009D 6$:   MOVAL   EDT$$A_US_TXT[TEXT_NUM], R0         ; 1433
             08   AE        00  B0  9A 000A5       MOVZBL  @0(R0), LEN
                                 50  DD 000AA       PUSHL   R0                                  ; 1434
   04   AE   0C   AE            01  C1 000AC       ADDL3   #1, LEN, 4(SP)
                          04    AE  9F 000B2       PUSHAB  4(SP)
                  63            02  FB 000B5       CALLS   #2, EDT$$DEA_HEAP
        E1        52            01  F3 000B8       AOBLEQ  #1, TEXT_NUM, 6$                    ; 1431
                                 52  D4 000BC       CLRL    TBL_PTR                             ; 1441
                  50 00000000G0042  D0 000BE 7$:   MOVL    EDT$$A_TRN_TBL[TBL_PTR], R0         ; 1444
                                 0D  13 000C6       BEQL    8$
                  7E        04  A0  3C 000C8       MOVZWL  4(KEY_PTR), -(SP)                   ; 1451
          00000000G   00        01  FB 000CC       CALLS   #1, EDT$$CAN_KDEF
                                 E9  11 000D3       BRB     7$                                  ; 1444
        E1        52 000000C6   8F  F3 000D5 8$:   AOBLEQ  #198, TBL_PTR, 7$                   ; 1441
                                 04 000DD       RET                                         ; 1456

; Routine Size:  222 bytes,    Routine Base:  _EDT$CODE + 00CE

;  607          1457  1
;  609          1458  1 %SBTTL 'WORKAST - take a timer AST for the WORKING message'
;  610          1459  1 ROUTINE WORKAST                                  ! Take a timer AST for the WORKING message
;  611          1460  1     : NOVALUE =
;  612          1461  1
;  613          1462  1 !++
;  614          1463  1 ! FUNCTIONAL DESCRIPTION:
;  615          1464  1 !
;  616          1465  1 !       Take a timer AST for the WORKING message.
;  617          1466  1 !
;  618          1467  1 ! FORMAL PARAMETERS:
;  619          1468  1 !
;  620          1469  1 !       NONE
;  621          1470  1 !
;  622          1471  1 ! IMPLICIT INPUTS:
;  623          1472  1 !
;  624          1473  1 !       WORK_MESSAGE_RUNNING
;  625          1474  1 !
;  626          1475  1 ! IMPLICIT OUTPUTS:
;  627          1476  1 !
;  628          1477  1 !       EDT$$G_SECOND
;  629          1478  1 !
;  630          1479  1 ! ROUTINE VALUE:
;  631          1480  1 !
;  632          1481  1 !       NONE
;  633          1482  1 !
;  634          1483  1 ! SIZE EFFECTS:
;  635          1484  1 !
;  636          1485  1 !       Arranges to print the WORKING message on the screen.
;  637          1486  1 !
;  638          1487  1 !--
;  639          1488  1
;  640          1489  2     BEGIN
;  641          1490  2
;  642          1491  2     IF .WORK_MESSAGE_RUNNING
;  643          1492  2     THEN
;  644          1493  3         BEGIN
;  645          1494  3         EDT$$G_SECOND = 1;
;  646          1495  3         $SETIMR (DAYTIM = DEL_TIME, ASTADR = WORKAST, REQIDT = EDT$$G_WORKCOUNT);
;  647          1496  2         END;
;  648          1497  2
;  649          1498  1     END;                                         ! of routine WORKAST


                                                            .EXTRN  SYS$SETIMR

                                         0000 00000 WORKAST:.WORD   Save nothing                              ; 1459
                   1F 00000000'   EF   E9 00002           BLBC    WORK_MESSAGE_RUNNING, 1$                  ; 1491
         00000000G  00            01   D0 00009           MOVL    #1, EDT$$G_SECOND                         ; 1494
                      00000000G   00   9F 00010           PUSHAB  EDT$$G_WORKCOUNT                          ; 1495
                             E7   AF   9F 00016           PUSHAB  WORKAST                                   ;
                      00000000'   EF   9F 00019           PUSHAB  DEL_TIME                                  ;
                                  7E   D4 0001F           CLRL    -(SP)                                     ;
         00000000G  00            04   FB 00021           CALLS   #4, SYS$SETIMR                            ;
                                       04 00028 1$:       RET                                               ; 1498

; Routine Size:  41 bytes,    Routine Base:  _EDT$CODE + 01AC
```

```
   1 %SBTTL 'EDT$$START_WKINGMSG  - initiate the "working" timer'
   1
   1 GLOBAL ROUTINE EDT$$START_WKINGMSG                    ! Initiate the "working" timer
   1     : NOVALUE =
   1
   1 !++
   1 ! FUNCTIONAL DESCRIPTION:
   1 !
   1 !       Start the timer which will cause the "working" message
   1 !       to print occasionally until it is cancelled.
   1 !
   1 ! FORMAL PARAMETERS:
   1 !
   1 !       NONE
   1 !
   1 ! IMPLICIT INPUTS:
   1 !
   1 !       DEL_TIME
   1 !       WORKAST
   1 !       WORK_MESSAGE_RUNNING
   1 !
   1 ! IMPLICIT OUTPUTS:
   1 !
   1 !       EDT$$G_WORKCOUNT
   1 !       WORKING_EFN
   1 !       WORK_MESSAGE_RUNNING
   1 !
   1 ! ROUTINE VALUE:
   1 !
   1 !       NONE
   1 !
   1 ! SIDE EFFECTS:
   1 !
   1 !       Allocates an event flag.
   1 !       Signals any errors.
   1 !
   1 !--
   1
   2     BEGIN
   2
   2     LOCAL
   2         GETEF_STATUS,
   2         SETIMR_STATUS;
   2
   2 !+
   2 ! If the "working" message is already running, don't start it again.
   2 !-
   2
   2     IF .WORK_MESSAGE_RUNNING THEN RETURN;
   2
   2     GETEF_STATUS = LIB$GET_EF (WORKING_EFN);
   2
   2     IF ( NOT .GETEF_STATUS) THEN SIGNAL_STOP (.GETEF_STATUS);
   2
P  2     SETIMR_STATUS = $SETIMR (EFN = .WORKING_EFN, DAYTIM = DEL_TIME, ASTADR = WORKAST,
   2         REQIDT = EDT$$G_WORKCOUNT);
   2
```

```
;  708        1556  2        IF ( NOT .SETIMR_STATUS) THEN SIGNAL_STOP (.SETIMR_STATUS);
;  709        1557  2
;  710        1558  2        EDT$$G_WORKCOUNT = 0;
;  711        1559  2        WORK_MESSAGE_RUNNING = 1;
;  712        1560  1        END;                                               ! of routine EDT$$START_WKINGMSG



                                               001C 00000          .ENTRY  EDT$$START_WKINGMSG, Save R2,R3,R4           ; 1501
                        54 00000000G     00  9E 00002          MOVAB   EDT$$G_WORKCOUNT, R4                          ;
                        53 00000000G     00  9E 00009          MOVAB   LIB$STOP, R3                                  ;
                        52 00000000'     EF  9E 00010          MOVAB   WORK_MESSAGE_RUNNING, R2                      ;
                        31               62  E8 00017          BLBS    WORK_MESSAGE_RUNNING, 3$                      ; 1547
                                    FC   A2  9F 0001A          PUSHAB  WORKING_EFN                                   ; 1549
              00000000G  00              01  FB 0001D          CALLS   #1, LIB$GET_EF                                ;
                         05              50  E8 00024          BLBS    GETEF_STATUS, 1$                              ; 1551
                                         50  DD 00027          PUSHL   GETEF_STATUS                                  ;
                         63              01  FB 00029          CALLS   #1, LIB$STOP                                  ;
                                         54  DD 0002C  1$:     PUSHL   R4                                            ; 1554
                                    A6   AF  9F 0002E          PUSHAB  WORKAST                                       ;
                                    F4   A2  9F 00031          PUSHAB  DEL_TIME                                      ;
                                    FC   A2  DD 00034          PUSHL   WORKING_EFN                                   ;
              00000000G  00              04  FB 00037          CALLS   #4, SYS$SETIMR                                ;
                         05              50  E8 0003E          BLBS    SETIMR_STATUS, 2$                             ; 1556
                                         50  DD 00041          PUSHL   SETIMR_STATUS                                 ;
                         63              01  FB 00043          CALLS   #1, LIB$STOP                                  ;
                                         64  D4 00046  2$:     CLRL    EDT$$G_WORKCOUNT                              ; 1558
                         62              01  D0 00048          MOVL    #1, WORK_MESSAGE_RUNNING                      ; 1559
                                             04 0004B  3$:     RET                                                   ; 1560

; Routine Size:  76 bytes,    Routine Base:  _EDT$CODE + 01D5


;  713        1561  1
```

```
1 %SBTTL 'EDT$$STOP_WKINGMSG  - cancel the "working" timer'
1
1 GLOBAL ROUTINE EDT$$STOP_WKINGMSG                ! Cancel the "working" timer
1     : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Cancel the "working" timer.  The "working" message will not print
1 !       until it is initiated again.  Also, erase the working message.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       WORKING_EFN
1 !       EDT$$G_WORKCOUNT
1 !       WORK_MESSAGE_RUNNING
1 !       EDT$$G_MESSAGE_LINE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       WORK_MESSAGE_RUNNING
1 !
1 ! ROUTINE VALUE:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Deallocates an event flag.
1 !       Repositions the cursor to beginning of message line
1 !
1 !--
1
2     BEGIN
2
2     LOCAL
2         FORMAT_ROUTINE,
2         FREEEF_STATUS,
2         CANTIM_STATUS;
2
2 !+
2 ! If the "working" message is not running, do nothing.
2 !-
2
2     IF ( NOT .WORK_MESSAGE_RUNNING) THEN RETURN;
2
2     WORK_MESSAGE_RUNNING = 0;
2     CANTIM_STATUS = $CANTIM (REQIDT = EDT$$G_WORKCOUNT);
2
2     IF ( NOT .CANTIM_STATUS) THEN SIGNAL_STOP (.CANTIM_STATUS);
2
2     FREEEF_STATUS = LIB$FREE_EF (WORKING_EFN);
2
```

```
;  772        1619 2        IF ( NOT .FREEEF_STATUS) THEN SIGNAL_STOP (.FREEEF_STATUS);
;  773        1620 2
;  774        1621 2    !+
;  775        1622 2    ! Erase the working message when it is stopped if not already done
;  776        1623 2    !-
;  777        1624 2        FORMAT_ROUTINE = .EDT$$A_FMT_WRRUT;
;  778        1625 2        EDT$$A_FMT_WRRUT = EDT$$TI_WRSTR;
;  779        1626 2
;  780        1627 3        IF (.EDT$$G_WORKCOUNT)
;  781        1628 2        THEN
;  782        1629 3            BEGIN
;  783        1630 3            EDT$$SC_POSCSIF (.EDT$$G_MESSAGE_LINE + 1, 0);
;  784        1631 3            EDT$$SC_ERATOEOL ();
;  785        1632 3            EDT$$OUT_FMTBUF ();
;  786        1633 2            END;
;  787        1634 2
;  788        1635 2    !+
;  789        1636 2    ! If "working" was printed then reposition the cursor to the left-most
;  790        1637 2    ! position of the prompt.
;  791        1638 2    !-
;  792        1639 2
;  793        1640 3        IF (.EDT$$G_WORKCOUNT NEQ 0)
;  794        1641 2        THEN
;  795        1642 3            BEGIN
;  796        1643 3            EDT$$SC_POSCSIF (.EDT$$G_MESSAGE_LINE + 1, 0);
;  797        1644 3            EDT$$OUT_FMTBUF ();
;  798        1645 2            END;
;  799        1646 2
;  800        1647 2        EDT$$A_FMT_WRRUT = .FORMAT_ROUTINE;
;  801        1648 2        EDT$$G_SECOND = 0;
;  802        1649 1        END;                                                  ! of routine EDT$$STOP_WKINGMSG


                                                                  .EXTRN  SYS$CANTIM

                                            03FC 00000            .ENTRY  EDT$$STOP_WKINGMSG, Save R2,R3,R4,R5,R6,R7,-; 1564
                                                                          R8,R9                                        :
                        59 00000000G  00  9E 00002                MOVAB   EDT$$OUT_FMTBUF, R9                          :
                        58 00000000G  00  9E 00009                MOVAB   EDT$$SC_POSCSIF, R8                          :
                        57 00000000G  00  9E 00010                MOVAB   EDT$$G_MESSAGE_LINE, R7                      :
                        56 00000000G  00  9E 00017                MOVAB   LIB$STOP, R6                                 :
                        55 00000000'  EF  9E 0001E                MOVAB   WORK_MESSAGE_RUNNING, R5                     :
                        54 00000000G  00  9E 00025                MOVAB   EDT$$A_FMT_WRRUT, R4                         :
                        53 00000000G  00  9E 0002C                MOVAB   EDT$$G_WORKCOUNT, R3                         :
                        60                65  E9 00033                BLBC    WORK_MESSAGE_RUNNING, 5$                     ; 1610
                                          65  D4 00036                CLRL    WORK_MESSAGE_RUNNING                         ; 1612
                                          7E  D4 00038                CLRL    -(SP)                                        ; 1613
                                          53  DD 0003A                PUSHL   R3                                           :
             00000000G  00                02  FB 0003C                CALLS   #2, SYS$CANTIM                               :
                        05                50  E8 00043                BLBS    CANTIM_STATUS, 1$                            ; 1615
                                          50  DD 00046                PUSHL   CANTIM_STATUS                                :
                        66                01  FB 00048                CALLS   #1, LIB$STOP                                 :
                                   FC     A5  9F 0004B 1$:            PUSHAB  WORKING_EFN                                  ; 1617
             00000000G  00                01  FB 0004E                CALLS   #1, LIB$FREE_EF                              :
                        05                50  E8 00055                BLBS    FREEEF_STATUS, 2$                            ; 1619
                                          50  DD 00058                PUSHL   FREEEF_STATUS                                :
```

```
                   66           01 FB 0005A        CALLS   #1, LIB$STOP                                    ;
                   52           64 D0 0005D 2$:    MOVL    EDT$$A_FMT_WRRUT, FORMAT_ROUTINE                 ; 1624
                   64 00000000G 00 9E 00060        MOVAB   EDT$$TI_WRSTR, EDT$$A_FMT_WRRUT                  ; 1625
                   13           63 E9 00067        BLBC    EDT$$G_WORKCOUNT, 3$                             ; 1627
                                7E D4 0006A        CLRL    -(SP)                                            ; 1630
         7E        67           01 C1 0006C        ADDL3   #1, EDT$$G_MESSAGE_LINE, -(SP)                   ;
                   68           02 FB 00070        CALLS   #2, EDT$$SC_POSCSIF                              ;
            00000000G           00 FB 00073        CALLS   #0, EDT$$SC_ERATOEOL                             ; 1631
                   69           00 FB 0007A        CALLS   #0, EDT$$OUT_FMTBUF                              ; 1632
                                63 D5 0007D 3$:    TSTL    EDT$$G_WORKCOUNT                                 ; 1640
                                0C 13 0007F        BEQL    4$                                               ;
                                7E D4 00081        CLRL    -(SP)                                            ; 1643
         7E        67           01 C1 00083        ADDL3   #1, EDT$$G_MESSAGE_LINE, -(SP)                   ;
                   68           02 FB 00087        CALLS   #2, EDT$$SC_POSCSIF                              ;
                   69           00 FB 0008A        CALLS   #0, EDT$$OUT_FMTBUF                              ; 1644
                   64           52 D0 0008D 4$:    MOVL    FORMAT_ROUTINE, EDT$$A_FMT_WRRUT                 ; 1647
                      00000000G 00 D4 00090        CLRL    EDT$$G_SECOND                                    ; 1648
                                   04 00096 5$:    RET                                                      ; 1649

; Routine Size:  151 bytes,    Routine Base:  _EDT$CODE + 0221


;  803          1650  1
;  805        1651  1 %SBTTL 'EDT$$MSG_TOSTR  - print a system message'
;  806        1652  1
;  807        1653  1 GLOBAL ROUTINE EDT$$MSG_TOSTR (                       ! Print a system message
;  808        1654  1     MESS_NUM                                          ! message number
;  809        1655  1     ) : NOVALUE =
;  810        1656  1
;  811        1657  1 !++
;  812        1658  1 ! FUNCTIONAL DESCRIPTION:
;  813        1659  1 !
;  814        1660  1 !       Print a system message, given its message number.
;  815        1661  1 !
;  816        1662  1 ! FORMAL PARAMETERS:
;  817        1663  1 !
;  818        1664  1 !  MESS_NUM              The number of the message to print
;  819        1665  1 !
;  820        1666  1 ! IMPLICIT INPUTS:
;  821        1667  1 !
;  822        1668  1 !       NONE
;  823        1669  1 !
;  824        1670  1 ! IMPLICIT OUTPUTS:
;  825        1671  1 !
;  826        1672  1 !       NONE
;  827        1673  1 !
;  828        1674  1 ! ROUTINE VALUE:
;  829        1675  1 !
;  830        1676  1 !       NONE
;  831        1677  1 !
;  832        1678  1 ! SIDE EFFECTS:
;  833        1679  1 !
;  834        1680  1 !       Prints a message on the terminal.
;  835        1681  1 !
;  836        1682  1 !--
;  837        1683  1
;  838        1684  2     BEGIN
;  839        1685  2
;  840        1686  2     LOCAL
;  841        1687  2         MSGBUF : BLOCK [CH$ALLOCATION (80)],
;  842        1688  2         MSGDESC : VECTOR [2],
;  843        1689  2         MSGLEN;
;  844        1690  2
;  845        1691  2     MSGDESC [0] = 80;
;  846        1692  2     MSGDESC [1] = MSGBUF;
;  847        1693  2     $GETMSG (MSGID = .MESS_NUM, MSGLEN = MSGLEN, BUFADR = MSGDESC, FLAGS = 1);
;  848        1694  2     EDT$$FMT_STR (MSGBUF, .MSGLEN<0, 16>);
;  849        1695  1     END;                                              ! of routine EDT$$MSG_TOSTR


                                                                  .EXTRN   SYS$GETMSG

                                         0000 00000               .ENTRY   EDT$$MSG_TOSTR, Save nothing                      ; 1653
                    5E        A4    AE    9E 00002               MOVAB    -92(SP), SP                                        ;
             04     AE        50    8F    9A 00006               MOVZBL   #80, MSGDESC                                       ; 1691
             08     AE        0C    AE    9E 0000B               MOVAB    MSGBUF, MSGDESC+4                                  ; 1692
                    7E              01    7D 00010               MOVQ     #1, -(SP)                                          ; 1693
                              0C    AE    9F 00013               PUSHAB   MSGDESC                                            ;
                              0C    AE    9F 00016               PUSHAB   MSGLEN                                             ;
```

```
                                        04   AC  DD 00019          PUSHL    MESS_NUM                              ;
                     00000000G  00           05  FB 0001C          CALLS    #5, SYS$GETMSG                        ;
                                7E           6E  3C 00023          MOVZWL   MSGLEN, -(SP)                         ; 1694
                                        10   AE  9F 00026          PUSHAB   MSGBUF                                ;
                     00000000G  00           02  FB 00029          CALLS    #2, EDT$$FMT_STR                      ;
                                                 04 00030          RET                                            ; 1695

; Routine Size:  49 bytes,    Routine Base:  _EDT$CODE + 02B8


;  850          1696  1
;  851          1697  1 !<BLF/PAGE>
;  853        1698  1 END                                                   ! of module EDT$SYSVAX
;  854        1699  1
;  855        1700  0 ELUDOM




                                                                     .EXTRN  LIB$STOP

;                                 PSECT SUMMARY
;
;       Name                   Bytes                      Attributes
;
;  _EDT$DATA                       644  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;  _EDT$CODE                       745  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)



;                            Library Statistics
;
;                                        -------- Symbols --------        Pages      Processing
;       File                            Total    Loaded   Percent         Mapped     Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1      9776       14         0           581       00:02.6
;  _$255$DUA28:[EDT.SRC]EDT.L32;1          377       37         9            40       00:00.8
;  _$255$DUA28:[EDT.SRC]PSECTS.L32;1         2        1        50             7       00:00.1
;  _$255$DUA28:[EDT.SRC]KEYPADDEF.L32;1     34        6        17             7       00:00.2




;                                 COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS$:SYSVAX/OBJ=OBJ$:SYSVAX MSRC$:SYSVAX.B32/UPDATE=(ENH$:SYSVAX)

; Size:           745 code + 644 data bytes
; Run Time:          00:43.4
; Elapsed Time:      00:58.6
; Lines/CPU Min:      2351
; Lexemes/CPU-Min:    8231
; Memory Used:   153 pages
; Compilation Complete
```

0140 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SCRUPDATM
LIS

STARTEDT
LIS

TISAVE
LIS

SETCOMMAN
LIS

TIAUTO
LIS

TIDELETE
LIS

TILINE
LIS

TRACEMAC
LIS

UFBUFFER
LIS

SCRZAPSIN
LIS

UEXACASE
LIS

SCRUPDATN
LIS

TITYPAHD
LIS

TSTKEYDEF
LIS

TICLRAUD
LIS

SUPPORTS
LIS

TICHAR
LIS

TIECHO
LIS

TRAROUNAM
LIS

TRANSLATE
LIS

UFSTRING
LIS

SYSVAX
LIS

SCRWID
LIS

TRACELIT
LIS

UCVTCASE
LIS