```
EEEEEEEEEEEEEEE  DDDDDDDDDDD     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  DDDDDDDDDDD          TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD         TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD         TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD         TTT
```

```
FFFFFFFFFF     IIIIII     NN      NN  DDDDDDDD    KK      KK  EEEEEEEEEE  YY      YY
FFFFFFFFFF     IIIIII     NN      NN  DDDDDDDD    KK      KK  EEEEEEEEEE  YY      YY
FF               II       NN      NN  DD      DD  KK      KK  EE          YY      YY
FF               II       NN      NN  DD      DD  KK      KK  EE          YY      YY
FF               II       NNNN    NN  DD      DD  KK    KK    EE            YY  YY
FF               II       NNNN    NN  DD      DD  KK    KK    EE            YY  YY
FFFFFFFF         II       NN  NN  NN  DD      DD  KKKKKK      EEEEEEEE        YY
FFFFFFFF         II       NN  NN  NN  DD      DD  KKKKKK      EEEEEEEE        YY
FF               II       NN    NNNN  DD      DD  KK    KK    EE              YY
FF               II       NN    NNNN  DD      DD  KK    KK    EE              YY
FF               II       NN      NN  DD      DD  KK      KK  EE              YY          ....
FF               II       NN      NN  DD      DD  KK      KK  EE              YY          ....
FF             IIIIII     NN      NN  DDDDDDDD    KK      KK  EEEEEEEEEE      YY          ....
FF             IIIIII     NN      NN  DDDDDDDD    KK      KK  EEEEEEEEEE      YY          ....


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

```
;    1          0001  0 %TITLE 'EDT$FINDKEY - find a key''s definition'
;    2          0002  0 MODULE EDT$FINDKEY (                          ! Find a key's definition
;    3          0003  0                     IDENT = 'V04-000'                ! File: FINDKEY.BLI Edit: JBS1002
;    4          0004  0                     ) =
;    5          0005  1 BEGIN
;    6          0006  1 !
;    7          0007  1 !**********************************************************************
;    8          0008  1 !*                                                                    *
;    9          0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                           *
;   10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.            *
;   11          0011  1 !*  ALL RIGHTS RESERVED.                                             *
;   12          0012  1 !*                                                                    *
;   13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
;   14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;   15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
;   16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
;   17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
;   18          0018  1 !*  TRANSFERRED.                                                       *
;   19          0019  1 !*                                                                    *
;   20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
;   21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
;   22          0022  1 !*  CORPORATION.                                                       *
;   23          0023  1 !*                                                                    *
;   24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
;   25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.            *
;   26          0026  1 !*                                                                    *
;   27          0027  1 !*                                                                    *
;   28          0028  1 !**********************************************************************
;   29          0029  1 !
;   30          0030  1
;   31          0031  1 !++
;   32          0032  1 ! FACILITY:     EDT -- The DEC Standard Editor
;   33          0033  1 !
;   34          0034  1 ! ABSTRACT:
;   35          0035  1 !
;   36          0036  1 !       Find a key's definition.
;   37          0037  1 !
;   38          0038  1 ! ENVIRONMENT:  Runs at any access mode - AST reentrant                    ;
;   39          0039  1 !
;   40          0040  1 ! AUTHOR: John Sauter, CREATION DATE: August 13, 1982
;   41          0041  1 !                                                                         ;
;   42          0042  1 ! MODIFIED BY:                                                            ;
;   43          0043  1 !
;   44          0044  1 ! 1-001 - Original.  JBS 13-Aug-1982
;   45          0045  1 ! 1-002 - Don't store into the argument list.  JBS 10-Nov-1982
;   46          0046  1 !--
;   47          0047  1
```

```
;    49          0048  1 %SBTTL 'Declarations'
;    50          0049  1 !
;    51          0050  1 ! TABLE OF CONTENTS:
;    52          0051  1 !
;    53          0052  1
;    54          0053  1 REQUIRE 'EDTSRC:TRAROUNAM';
;    55          0492  1
;    56          0493  1 FORWARD ROUTINE
;    57          0494  1     EDT$$FIND_KEY;                  ! Find a key's definition
;    58          0495  1
;    59          0496  1 !
;    60          0497  1 ! INCLUDE FILES:
;    61          0498  1 !
;    62          0499  1
;    63          0500  1 REQUIRE 'EDTSRC:EDTREQ';
;    64          0635  1
;    65          0636  1 LIBRARY 'EDTSRC:KEYPADDEF';
;    66          0637  1
;    67          0638  1 !
;    68          0639  1 ! MACROS:
;    69          0640  1 !
;    70          0641  1 !       NONE
;    71          0642  1 !
;    72          0643  1 ! EQUATED SYMBOLS:
;    73          0644  1 !
;    74          0645  1 !       NONE
;    75          0646  1 !
;    76          0647  1 ! OWN STORAGE:
;    77          0648  1 !
;    78          0649  1 !       NONE
;    79          0650  1 !
;    80          0651  1 ! EXTERNAL REFERENCES:
;    81          0652  1 !
;    82          0653  1 !       In the routine
```

```
;    84          0654  1 %SBTTL 'EDT$$FIND_KEY - find a key''s definition'
;    85          0655  1
;    86          0656  1 GLOBAL ROUTINE EDT$$FIND_KEY (                    ! Find a key's definition
;    87          0657  1     KEY,                                          ! Key number
;    88          0658  1     KEY_PTR                                       ! Pointer to key definition
;    89          0659  1     ) =
;    90          0660  1
;    91          0661  1 !++
;    92          0662  1 ! FUNCTIONAL DESCRIPTION:
;    93          0663  1 !
;    94          0664  1 !       Find the definition of a key in the key definition table.  Return that
;    95          0665  1 !       definition or an indication that the key is not defined.
;    96          0666  1 !
;    97          0667  1 ! FORMAL PARAMETERS:
;    98          0668  1 !
;    99          0669  1 !  KEY                  The number of the key whose definition is to be found
;   100          0670  1 !
;   101          0671  1 !  KEY_PTR              Pointer to the key's definition
;   102          0672  1 !
;   103          0673  1 ! IMPLICIT INPUTS:
;   104          0674  1 !
;   105          0675  1 !       EDT$$A_TRN_TBL
;   106          0676  1 !
;   107          0677  1 ! IMPLICIT OUTPUTS:
;   108          0678  1 !
;   109          0679  1 !       NONE
;   110          0680  1 !
;   111          0681  1 ! ROUTINE VALUE:
;   112          0682  1 !
;   113          0683  1 !       1 = key found, KEY_PTR points to its definition
;   114          0684  1 !       0 = no definition for this key
;   115          0685  1 !
;   116          0686  1 ! SIDE EFFECTS:
;   117          0687  1 !
;   118          0688  1 !       NONE
;   119          0689  1 !
;   120          0690  1 !--
;   121          0691  1
;   122          0692  2     BEGIN
;   123          0693  2
;   124          0694  2     MAP
;   125          0695  2         KEY_PTR : REF BLOCK [, BYTE] FIELD (KEY_DEF_FIELD);
;   126          0696  2
;   127          0697  2     EXTERNAL
;   128          0698  2         EDT$$A_TRN_TBL : VECTOR [K_KPAD_HASHSIZ]:       ! Table of pointers to key definitions
;   129          0699  2
;   130          0700  2     LOCAL
;   131          0701  2         L_KEY,
;   132          0702  2         L_KEY_PTR : REF BLOCK [, BYTE] FIELD (KEY_DEF_FIELD),
;   133          0703  2         KEY_FOUND;
;   134          0704  2
;   135          0705  2     L_KEY = .KEY;
;   136          0706  2     ASSERT ((.L_KEY LSS K_KEY_MAX) AND (.L_KEY GEQ 0));
;   137          0707  2     L_KEY_PTR = EDT$$A_TRN_TBL [.L_KEY MOD K_KPAD_HASHSIZ];
;   138          0708  2     KEY_FOUND = 0;
;   139          0709  2
;   140          0710  2     WHILE ( NOT .KEY_FOUND) DO
```

```
;  141          0711  3          BEGIN
;  142          0712  3          L_KEY_PTR = .L_KEY_PTR [KEY_DEF_NEXT];
;  143          0713  3
;  144          0714  3          IF (.L_KEY_PTR EQLA 0) THEN RETURN (0);
;  145          0715  3
;  146          0716  3          IF (.L_KEY_PTR [KEY_DEF_KEY] EQL .L_KEY) THEN KEY_FOUND = 1;
;  147          0717  3
;  148          0718  3          ASSERT ((.L_KEY MOD K_KPAD_HASHSIZ) EQL (.L_KEY_PTR [KEY_DEF_KEY] MOD K_KPAD_HASHSIZ));
;  149          0719  2          END;
;  150          0720  2
;  151          0721  2      .KEY_PTR = .L_KEY_PTR;
;  152          0722  2      RETURN (1);
;  153          0723  1      END;                                    ! of routine EDT$$FIND_KEY

;

                                                                     .TITLE  EDT$FINDKEY EDT$FINDKEY - find a key's definiti
                                                                                     on
                                                                     .IDENT  \V04-000\

                                                                     .EXTRN  EDT$$A_TRN_TBL, EDT$$INTER_ERR

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

                                                 007C 00000          .ENTRY  EDT$$FIND_KEY, Save R2,R3,R4,R5,R6          ; 0656
                          56 00000000G   00       9E 00002          MOVAB   EDT$$INTER_ERR, R6
                          55       04    AC       D0 00009          MOVL    KEY, L_KEY                                  ; 0705
                 000003E7   8F                    55   D1 0000D     CMPL    L_KEY, #999                                 ; 0706
                                                  04   18 00014     BGEQ    1$
                                                  55   D5 00016     TSTL    L_KEY
                                                  03   18 00018     BGEQ    2$
                                 66               00   FB 0001A 1$: CALLS   #0, EDT$$INTER_ERR
                    7E            00             55   01   7A 0001D 2$: EMUL    #1, L_KEY, #0, -(SP)                        ; 0707
                    53            53             8E 000000C7   8F   7B 00022     EDIV    #199, (SP)+, R3, R3
                                                 52 00000000G00043   DE 0002B     MOVAL   EDT$$A_TRN_TBL[R3], L_KEY_PTR
                                                  54   D4 00033     CLRL    KEY_FOUND                                   ; 0708
                                 2C              54   E8 00035 3$: BLBS    KEY_FOUND, 5$                               ; 0710
                                 52              62   D0 00038     MOVL    (L_KEY_PTR), L_KEY_PTR                      ; 0712
                                                  2F   13 0003B     BEQL    6$                                          ; 0714
                    55      04    A2             00   ED 0003D     CMPZV   #0, #16, 4(L_KEY_PTR), L_KEY                ; 0716
                                                  03   12 00043     BNEQ    4$
                                 54               01   D0 00045     MOVL    #1, KEY_FOUND
                                 50       04      A2   3C 00048 4$: MOVZWL  4(L_KEY_PTR), R0                           ; 0718
                    7E            00             50   01   7A 0004C     EMUL    #1, R0, #0, -(SP)
                    50            50             8E 000000C7   8F   7B 00051     EDIV    #199, (SP)+, R0, R0
                                                  50   53   D1 0005A     CMPL    R3, R0
                                                  D6   13 0005D     BEQL    3$
                                 66               00   FB 0005F     CALLS   #0, EDT$$INTER_ERR
                                                  D1   11 00062     BRB     3$                                          ; 0710
                          08     BC               52   D0 00064 5$: MOVL    L_KEY_PTR, @KEY_PTR                         ; 0721
                                 50               01   D0 00068     MOVL    #T, R0                                      ; 0722
                                                       04 0006B     RET
                                                  50   D4 0006C 6$: CLRL    R0                                          ; 0723
                                                       04 0006E     RET

; Routine Size:  111 bytes,    Routine Base:  _EDT$CODE + 0000
```

```
;  154          0724  1
;  155          0725  1 !<BLF/PAGE>
```

```
;  157          0726  1 END                                        ! of module EDT$FINDKEY
;  158          0727  1
;  159          0728  0 ELUDOM
```

```
;                                  PSECT SUMMARY
;
;      Name                      Bytes                           Attributes
;
;  _EDT$CODE                        111  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
```

```
;                                Library Statistics
;
;                                          -------- Symbols --------      Pages       Processing
;      File                                Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[EDT.SRC]EDT.L32;1           377         1         0          40        00:00.2
;  _$255$DUA28:[EDT.SRC]PSECTS.L32;1          2         1        50           7        00:00.1
;  _$255$DUA28:[EDT.SRC]KEYPADDEF.L32;1      34         7        20           7        00:00.1
```

```
;                                    COMMAND QUALIFIERS

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

; Size:         111 code + 0 data bytes
; Run Time:       00:11.6
; Elapsed Time:   00:31.1
; Lines/CPU Min:   3768
; Lexemes/CPU-Min: 11792
; Memory Used:  75 pages
; Compilation Complete
```

EXTEND
LIS

FDEC
LIS

FILL
LIS

FINDPARA
LIS

FORLF
LIS

EDT
LIS

EXEC
LIS

EXECNOO
LIS

FILEIO
LIS

EDTVECTOR
LIS

FINDKEY
LIS

FCOLINC
LIS

FINAL
LIS

FIOHNDLR
LIS

DEFKEY
LIS

ERRMSG
LIS

FCHAR
LIS