```
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
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
```

```
**FILE**ID**KEYDEFKEY                                                    D  7




KK      KK  EEEEEEEEEE  YY      YY  DDDDDDDD    EEEEEEEEEE  FFFFFFFFFF  KK      KK  EEEEEEEEEE  YY      YY
KK      KK  EEEEEEEEEE  YY      YY  DDDDDDDD    EEEEEEEEEE  FFFFFFFFFF  KK      KK  EEEEEEEEEE  YY      YY
KK      KK  EE          YY      YY  DD      DD  EE          FF          KK      KK  EE          YY      YY
KK      KK  EE          YY      YY  DD      DD  EE          FF          KK      KK  EE          YY      YY
KK    KK    EE            YY  YY    DD      DD  EE          FF          KK    KK    EE            YY  YY
KK    KK    EE            YY  YY    DD      DD  EE          FF          KK    KK    EE            YY  YY
KKKKKK      EEEEEEEE        YY      DD      DD  EEEEEEEE    FFFFFFFF    KKKKKK      EEEEEEEE        YY
KKKKKK      EEEEEEEE        YY      DD      DD  EEEEEEEE    FFFFFFFF    KKKKKK      EEEEEEEE        YY
KK    KK    EE              YY      DD      DD  EE          FF          KK    KK    EE              YY
KK    KK    EE              YY      DD      DD  EE          FF          KK    KK    EE              YY
KK      KK  EE              YY      DD      DD  EE          FF          KK      KK  EE              YY          ....
KK      KK  EE              YY      DD      DD  EE          FF          KK      KK  EE              YY          ....
KK      KK  EEEEEEEEEE      YY      DDDDDDDD    EEEEEEEEEE  FF          KK      KK  EEEEEEEEEE      YY          ....
KK      KK  EEEEEEEEEE      YY      DDDDDDDD    EEEEEEEEEE  FF          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
```

```
0001  0 %TITLE 'EDT$KEYDEFKEY - interactive define key'
0002  0 MODULE EDT$KEYDEFKEY (                          ! Interactive define key
0003  0                 IDENT = 'V04-000'                       ! File: KEYDEFKEY.BLI Edit: JBS1017
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 !       Interactive define key
0037  1 !
0038  1 ! ENVIRONMENT:  Runs at any access mode - AST reentrant
0039  1 !
0040  1 ! AUTHOR: Bob Kushlis, CREATION DATE: April 7, 1979
0041  1 !
0042  1 ! MODIFIED BY:
0043  1 !
0044  1 ! 1-001 - Original.  DJS 24-Feb-1981.  This module was created by
0045  1 !       extracting routine EDT$$LN_DEFK  from module KEYTRAN.
0046  1 ! 1-002 - Regularize headers.  JBS 09-Mar-1981
0047  1 ! 1-003 - Use new message codes.  JBS 04-Aug-1981
0048  1 ! 1-004 - Add return values.  JBS 02-Oct-1981
0049  1 ! 1-005 - Revise call to enable autorepeat.  JBS 30-Jan-1982
0050  1 ! 1-006 - Make CR a valid part of the key definition.  SMB 17-Feb-1982
0051  1 ! 1-007 - Make DEFK illegal in NOKEYPAD mode - used only for defining
0052  1 !       keys in KEYPAD mode.  SMB 1-Mar-1982
0053  1 ! 1-008 - Fix routine header.  JBS 22-Apr-1982
0054  1 ! 1-009 - EDT$$DEFK returns a status.  JBS 18-May-1982
0055  1 ! 1-010 - Set a flag if control C actually aborts something.  JBS 24-May-1982
0056  1 ! 1-011 - Return true success from EDT$$NXT_CMDCH. STS 16-Jun-1982
0057  1 ! 1-012 - Pass flag to EDT$$NXT_CMDCH to indicate whether or not we wish
```

```
;    58          0058  1 !            to accept repeat counts. STS 16-Jun-1982
;    59          0059  1 ! 1-013 - Modify message processing.  SMB 22-Jun-1982
;    60          0060  1 ! 1-014 - New implementation of defined keys.  JBS 13-Aug-1982
;    61          0061  1 ! 1-015 - Change cursor positioning call to obsolete a module.  JBS 05-Oct-1982
;    62          0062  1 ! 1-016 - Worry about having no cursor position to return to.  This can happen
;    63          0063  1 !            if this command is typed ahead initially.  JBS 22-Feb-1983
;    64          0064  1 ! 1-017 - Don't allow GOLD repeat count in key definitions.  JBS 09-May-1983
;    65          0065  1 !--
;    66          0066  1
;    68           0067   1 %SBTTL 'Declarations'
;    69           0068   1 !
;    70           0069   1 ! TABLE OF CONTENTS:
;    71           0070   1 !
;    72           0071   1
;    73           0072   1 REQUIRE 'EDTSRC:TRAROUNAM';
;    74           0511   1
;    75           0512   1 FORWARD ROUTINE
;    76           0513   1     EDT$$LN_DEFK;                              ! Interactive "define key"
;    77           0514   1
;    78           0515   1 !
;    79           0516   1 ! INCLUDE FILES:
;    80           0517   1 !
;    81           0518   1
;    82           0519   1 REQUIRE 'EDTSRC:EDTREQ';
;    83           0654   1
;    84           0655   1 LIBRARY 'EDTSRC:KEYPADDEF';
;    85           0656   1
;    86           0657   1 !
;    87           0658   1 ! MACROS:
;    88           0659   1 !
;    89           0660   1 !       NONE
;    90           0661   1 !
;    91           0662   1 ! EQUATED SYMBOLS:
;    92           0663   1 !
;    93           0664   1 !       NONE
;    94           0665   1 !
;    95           0666   1 ! OWN STORAGE:
;    96           0667   1 !
;    97           0668   1 !       NONE
;    98           0669   1 !
;    99           0670   1 ! EXTERNAL REFERENCES:
;   100           0671   1 !
;   101           0672   1 !       In the routine
```

```
1 %SBTTL 'EDT$$LN_DEFK  - interactive define key'
1
1 GLOBAL ROUTINE EDT$$LN_DEFK                          ! Interactive define key
1     =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine is called when the change mode command DEFK is processed
1 !       to process the interactive define key command.  The user is prompted
1 !       for the key to be defined and the defining string.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       EDT$$T_CMD_BUF
1 !       EDT$$G_CS_LNO
1 !       EDT$$G_CUR_COL
1 !       EDT$$G_KPAD
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       EDT$$A_CMD_BUF
1 !       EDT$$A_CMD_END
1 !       EDT$$G_CC_DONE
1 !
1 ! ROUTINE VALUE:
1 !
1 !       1 = ok, 2 = end of journal file, 0 = fail or control C
1 !
1 ! SIDE EFFECTS:
1 !
1 !       May define a key, or undefine a key.
1 !
1 !--
1
2     BEGIN
2
2     EXTERNAL ROUTINE
2         EDT$$STOP_WKINGMSG,
2         EDT$$CHK_CC,
2         EDT$$SC_POSCSIF : NOVALUE,           ! Position the cursor
2         EDT$$ERA_MSGLN,
2         EDT$$MSG_BELL,
2         EDT$$OUT_MSG,
2         EDT$$TST_BADK,
2         EDT$$DEFK,
2         EDT$$CAN_KDEF,
2         EDT$$NXT_CMDCH,                      ! Get the next command character
2         EDT$$TRN_KSTR,                       ! Build the command buffer
2         EDT$$PUT_CMDCH : NOVALUE,            ! Put a character in the command buffer
2         EDT$$SC_NONREVID,
2         EDT$$TI_ENBLAUTREP,                  ! Enable or disable autorepeat
2         EDT$$TI_DELK;
```

```
;  160       0730  2
;  161       0731  2      EXTERNAL
;  162       0732  2          EDT$$G_MSGFLG,                       ! Message flag
;  163       0733  2          EDT$$G_MESSAGE_LINE,                 ! Line on which to print prompts
;  164       0734  2          EDT$$G_KPAD,                         ! Are we in KEY or NOKEY mode
;  165       0735  2          EDT$$T_CMD_BUF,                      ! Command buffer
;  166       0736  2          EDT$$A_CMD_BUF,                      ! Pointer to next char in command buffer
;  167       0737  2          EDT$$A_CMD_END,                      ! Pointer to end of info 'n command buffer
;  168       0738  2          EDT$$G_CS_LNO,                       ! Line on which the cursor is positioned
;  169       0739  2          EDT$$G_CUR_COL,                      ! Column on which the cursor is positioned.
;  170       0740  2          EDT$$G_CC_DONE;                      ! Set to 1 if control C actually aborts something
;  171       0741  2
;  172       0742  2      MESSAGES ((PRSKEYDEF, KEYNOTDEF, NOWENTDEF, BADDEFK));
;  173       0743  2
;  174       0744  2      LOCAL
;  175       0745  2          KEY_ADDR,
;  176       0746  2          KEY : INITIAL (K_KEY_MAX + 1),       ! Initialize to an illegal value
;  177       0747  2          CON_U,
;  178       0748  2          C,
;  179       0749  2          SUCCEED,
;  180       0750  2          STATUS;
;  181       0751  2
;  182       0752  2  !+
;  183       0753  2  ! Since this command uses the bottom two lines for prompting and echoing,
;  184       0754  2  ! we don't want the Working message to continue printing.
;  185       0755  2  !-
;  186       0756  2      EDT$$STOP_WKINGMSG ();
;  187       0757  2  !+
;  188       0758  2  ! The interactive define key facility only works in keypad mode.
;  189       0759  2  !-
;  190       0760  2
;  191       0761  3      IF ( NOT .EDT$$G_KPAD)
;  192       0762  2      THEN
;  193       0763  3          BEGIN
;  194       0764  3          EDT$$MSG_BELL (EDT$_BADDEFK);
;  195       0765  3          EDT$$A_CMD_END = CH$PTR (EDT$$T_CMD_BUF);
;  196       0766  4          RETURN (0)
;  197       0767  2          END;
;  198       0768  2
;  199       0769  2      SUCCEED = 1;
;  200       0770  2  !+
;  201       0771  2  ! Set up the buffer for the definition to go into
;  202       0772  2  !-
;  203       0773  2      EDT$$A_CMD_BUF = CH$PTR (EDT$$T_CMD_BUF);
;  204       0774  2      EDT$$A_CMD_END = CH$PTR (EDT$$T_CMD_BUF, 256);
;  205       0775  2  !+
;  206       0776  2  ! Prompt for the key.
;  207       0777  2  !-
;  208       0778  2      EDT$$OUT_MSG (.EDT$$G_MESSAGE_LINE, EDT$_PRSKEYDEF, 0, 0);
;  209       0779  2      SUCCEED = EDT$$NXT_CMDCH (KEY, 0);           ! zero means no repeat counts allowed
;  210       0780  2
;  211       0781  2      IF (.SUCCEED NEQ 1) THEN RETURN (.SUCCEED);
;  212       0782  2
;  213       0783  2  !+
;  214       0784  2  ! Check for the validity of the key.
;  215       0785  2  !-
;  216       0786  2
;  217          0787  2       IF EDT$$TST_BADK (.KEY)
;  218          0788  2       THEN
;  219          0789  3           BEGIN
;  220          0790  3  !+
;  221          0791  3  ! Bad key.  Put out a message and return.
;  222          0792  3  !-
;  223          0793  3           EDT$$MSG_BELL (EDT$_KEYNOTDEF);
;  224          0794  3           EDT$$A_CMD_END = CH$PTR (EDT$$T_CMD_BUF);
;  225          0795  3           RETURN (0);
;  226          0796  2           END;
;  227          0797  2
;  228          0798  2  !+
;  229          0799  2  ! Now prompt for the definition.
;  230          0800  2  !-
;  231          0801  2       EDT$$OUT_MSG (.EDT$$G_MESSAGE_LINE, EDT$_NOWENTDEF, 0, 0);
;  232          0802  2  !+
;  233          0803  2  ! Loop, reading characters until we see the enter key.  Keypad and function keys
;  234          0804  2  ! and gold/anything are translated, delete and CTRL/U have the usual
;  235          0805  2  ! meanings and all other characters are inserted directly.
;  236          0806  2  !-
;  237          0807  2       CON_U = 0;                                  ! Note whether CONTROL U entered
;  238          0808  2
;  239          0809  2       WHILE 1 DO
;  240          0810  3           BEGIN
;  241          0811  3           SUCCEED = EDT$$NXT_CMDCH (C, 0);        ! No GOLD repeat counts
;  242          0812  3
;  243          0813  3           IF (.SUCCEED NEQ 1) THEN RETURN (2);
;  244          0814  3
;  245          0815  3           IF EDT$$CHK_CC ()
;  246          0816  3           THEN
;  247          0817  4               BEGIN
;  248          0818  4               EDT$$A_CMD_END = EDT$$T_CMD_BUF;
;  249          0819  4               EDT$$G_CC_DONE = 1;
;  250          0820  4               RETURN (0);
;  251          0821  3               END;
;  252          0822  3
;  253          0823  3           SELECTONE .C OF
;  254          0824  3               SET
;  255          0825  3
;  256          0826  3               [K_ENTER] :                         ! The ENTER key, end of definition
;  257          0827  3                   EXITLOOP;
;  258          0828  3
;  259          0829  3               [K_KPAD_BASE to K_KEY_MAX] :
;  260          0830  4                   BEGIN
;  261          0831  4  !+
;  262          0832  4  ! Keypad, function or gold/something key, place the translation in the buffer.
;  263          0833  4  !-
;  264          0834  4                   SUCCEED = EDT$$TRN_KSTR (.C, 0, 1);
;  265          0835  4
;  266          0836  4                   IF (.SUCCEED EQL 2) THEN RETURN (2);
;  267          0837  4
;  268          0838  4                   EDT$$A_CMD_END = CH$PTR (EDT$$T_CMD_BUF, 256);
;  269          0839  3                   END;
;  270          0840  3
;  271          0841  3               [ASC_K_DEL] :
;  272          0842  4                   BEGIN
;  273          0843  4  !+
;  274     0844  4  ! Delete key.  Delete the previous character unless we are at
;  275     0845  4  ! the beginning.
;  276     0846  4  !-
;  277     0847  4
;  278     0848  5                  IF CH$PTR_GTR (.EDT$$A_CMD_BUF, EDT$$T_CMD_BUF)
;  279     0849  4                  THEN
;  280     0850  5                      BEGIN
;  281     0851  5                      EDT$$A_CMD_BUF = CH$PLUS (.EDT$$A_CMD_BUF, -1);
;  282     0852  5                      EDT$$TI_DECK (CH$RCHAR (.EDT$$A_CMD_BUF));
;  283     0853  4                      END;
;  284     0854  4
;  285     0855  3                  END;
;  286     0856  3
;  287     0857  3              [ASC_K_CTRL_U] :
;  288     0858  4                  BEGIN
;  289     0859  4  !+
;  290     0860  4  ! CTRL/U: abort the define key command.
;  291     0861  4  !-
;  292     0862  4                  CON_U = 1;
;  293     0863  4                  EDT$$A_CMD_BUF = CH$PTR (EDT$$T_CMD_BUF);
;  294     0864  4                  EXITLOOP;
;  295     0865  3                  END;
;  296     0866  3
;  297     0867  3              [OTHERWISE] :
;  298     0868  3  !+
;  299     0869  3  ! None of the above: must be a character (other than DEL or CTRL/U) from
;  300     0870  3  ! the main keyboard.  Insert it in the definition.
;  301     0871  3  !-
;  302     0872  3                  EDT$$PUT_CMDCH (.C, 1);
;  303     0873  3              TES;
;  304     0874  3
;  305     0875  3  !+
;  306     0876  3  ! Turn on autorepeat, since seeing a keypad key will have turned
;  307     0877  3  ! it off.
;  308     0878  3  !-
;  309     0879  3          EDT$$TI_ENBLAUTREP (1);
;  310     0880  2          END;
;  311     0881  2
;  312     0882  2  !+
;  313     0883  2  ! Now define the key.
;  314     0884  2  !-
;  315     0885  2
;  316     0886  3      IF CH$PTR_NEQ (.EDT$$A_CMD_BUF, EDT$$T_CMD_BUF)
;  317     0887  2      THEN
;  318     0888  3          BEGIN
;  319     0889  3          STATUS = EDT$$DEFK (.KEY, EDT$$T_CMD_BUF, CH$DIFF (.EDT$$A_CMD_BUF, EDT$$T_CMD_BUF));
;  320     0890  3
;  321     0891  3          IF ( NOT .STATUS) THEN SUCCEED = 0;
;  322     0892  3
;  323     0893  3          END
;  324     0894  2      ELSE
;  325     0895  3          BEGIN
;  326     0896  3  !+
;  327     0897  3  ! Clear the definition of a key defined as the null string,
;  328     0898  3  ! except do nothing if the definition was terminated by a CTRL/U.
;  329     0899  3  !-
;  330     0900  3
;  331      0901  4             IF (.CON_U EQL 0)
;  332      0902  3             THEN
;  333      0903  4                 BEGIN
;  334      0904  4                 STATUS = EDT$$CAN_KDEF (.KEY);
;  335      0905  4
;  336      0906  4                 IF ( NOT .STATUS) THEN SUCCEED = 0;
;  337      0907  4
;  338      0908  3                 END;
;  339      0909  3
;  340      0910  2             END;
;  341      0911  2
;  342      0912  2 !+
;  343      0913  2 ! Reset the buffer pointer, erase the command line and reposition the cursor if necessary
;  344      0914  2 ! before continuing.
;  345      0915  2 !-
;  346      0916  2     EDT$$A_CMD_BUF = EDT$$T_CMD_BUF;
;  347      0917  2     EDT$$A_CMD_END = EDT$$T_CMD_BUF;
;  348      0918  2     EDT$$G_MSGFLG = 1;
;  349      0919  2     EDT$$ERA_MSGLN ();
;  350      0920  2
;  351      0921  2     IF (.EDT$$G_CS_LNO GEQ 0) THEN EDT$$SC_POSCSIF (.EDT$$G_CS_LNO, .EDT$$G_CUR_COL);
;  352      0922  2
;  353      0923  2     EDT$$SC_NONREVID ();
;  354      0924  2 !+
;  355      0925  2 ! If we had an error print an appropriate message.
;  356      0926  2 !-
;  357      0927  2
;  358      0928  2     IF (.SUCCEED EQL 0) THEN EDT$$MSG_BELL (.STATUS);
;  359      0929  2
;  360      0930  2     RETURN (.SUCCEED);
;  361      0931  1     END;                                          ! of routine EDT$$LN_DEFKEY


                                                                   .TITLE  EDT$KEYDEFKEY EDT$KEYDEFKEY - interactive defin
;                                                                                                e key
                                                                   .IDENT  \V04-000\

                                                                   .EXTRN  EDT$$STOP_WKINGMSG
                                                                   .EXTRN  EDT$$CHK_CC, EDT$$SC_POSCSIF
                                                                   .EXTRN  EDT$$ERA_MSGLN, EDT$$MSG_BELL
                                                                   .EXTRN  EDT$$OUT_MSG, EDT$$TST_BADK
                                                                   .EXTRN  EDT$$DEFK, EDT$$CAN_KDEF
                                                                   .EXTRN  EDT$$NXT_CMDCH, EDT$$TRN_KSTR
                                                                   .EXTRN  EDT$$PUT_CMDCH, EDT$$SC_NONREVID
                                                                   .EXTRN  EDT$$TI_ENBLAUTREP
                                                                   .EXTRN  EDT$$TI_DELK, EDT$$G_MSGFLG
                                                                   .EXTRN  EDT$$G_MESSAGE_LINE
                                                                   .EXTRN  EDT$$G_KPAD, EDT$$T_CMD_BUF
                                                                   .EXTRN  EDT$$A_CMD_BUF, EDT$$A_CMD_END
                                                                   .EXTRN  EDT$$G_CS_LNO, EDT$$G_CUR_COL
                                                                   .EXTRN  EDT$$G_CC_DONE, EDT$_PRSKEYDEF
                                                                   .EXTRN  EDT$_KEYNOTDEF, EDT$_NOWENTDEF
                                                                   .EXTRN  EDT$_BADDEFK

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

                                              0FFC 00000           .ENTRY  EDT$$LN_DEFK, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 0675
```

```
                                                              R10,R11                               ;
             5B 00000000G   00   9E 00002         MOVAB    EDT$$MSG_BELL, R11                    ;
             5A 00000000G   00   9E 00009         MOVAB    EDT$$NXT_CMDCH, R10                   ;
             59 00000000G   00   9E 00010         MOVAB    EDT$$OUT_MSG, R9                      ;
             58 00000000G   00   9E 00017         MOVAB    EDT$$G_MESSAGE_LINE, R8               ;
             57 00000000G   00   9E 0001E         MOVAB    EDT$$A_CMD_END, R7                    ;
             56 00000000G   00   9E 00025         MOVAB    EDT$$A_CMD_BUF, R6                    ;
             55 00000000G   00   9E 0002C         MOVAB    EDT$$T_CMD_BUF, R5                    ;
             5E             04   C2 00033         SUBL2    #4, SP                                ;
             7E       03E8  8F   3C 00036         MOVZWL   #1000, KEY                            ; 0712
 00000000G   00             00   FB 0003B         CALLS    #0, EDT$$STOP_WKINGMSG                ; 0756
             08 00000000G   00   E8 00042         BLBS     EDT$$G_KPAD, 1$                       ; 0761
                00000000G   8F   DD 00049         PUSHL    #EDT$_BADDEFK                         ; 0764
                            3D   11 0004F         BRB      3$                                    ;
             53             01   D0 00051 1$:     MOVL     #1, SUCCEED                           ; 0769
             66             65   9E 00054         MOVAB    EDT$$T_CMD_BUF, EDT$$A_CMD_BUF        ; 0773
             67       0100  C5   9E 00057         MOVAB    EDT$$T_CMD_BUF+256, EDT$$A_CMD_END    ; 0774
                            7E   7C 0005C         CLRQ     -(SP)                                 ; 0778
                00000000G   8F   DD 0005E         PUSHL    #EDT$_PRSKEYDEF                       ;
                            68   DD 00064         PUSHL    EDT$$G_MESSAGE_LINE                   ;
             69             04   FB 00066         CALLS    #4, EDT$$OUT_MSG                      ;
                            7E   D4 00069         CLRL     -(SP)                                 ; 0779
                       04   AE   9F 0006B         PUSHAB   KEY                                   ;
             6A             02   FB 0006E         CALLS    #2, EDT$$NXT_CMDCH                    ;
             53             50   D0 00071         MOVL     R0, SUCCEED                           ;
             01             53   D1 00074         CMPL     SUCCEED, #1                           ; 0781
                            03   13 00077         BEQL     2$                                    ;
                          0146   31 00079         BRW      19$                                   ;
                            6E   DD 0007C 2$:     PUSHL    KEY                                   ; 0787
 00000000G   00             01   FB 0007E         CALLS    #1, EDT$$TST_BADK                     ;
             0E             50   E9 00085         BLBC     R0, 4$                                ;
                00000000G   8F   DD 00088         PUSHL    #EDT$_KEYNOTDEF                       ; 0793
             6B             01   FB 0008E 3$:     CALLS    #1, EDT$$MSG_BELL                     ;
             67             65   9E 00091         MOVAB    EDT$$T_CMD_BUF, EDT$$A_CMD_END        ; 0794
                            33   11 00094         BRB      6$                                    ; 0795
                            7E   7C 00096 4$:     CLRQ     -(SP)                                 ; 0801
                00000000G   8F   DD 00098         PUSHL    #EDT$_NOWENTDEF                       ;
                            68   DD 0009E         PUSHL    EDT$$G_MESSAGE_LINE                   ;
             69             04   FB 000A0         CALLS    #4, EDT$$OUT_MSG                      ;
                            54   D4 000A3         CLRL     CON_U                                 ; 0807
                            7E   D4 000A5 5$:     CLRL     -(SP)                                 ; 0811
                       08   AE   9F 000A7         PUSHAB   C                                     ;
             6A             02   FB 000AA         CALLS    #2, EDT$$NXT_CMDCH                    ;
             53             50   D0 000AD         MOVL     R0, SUCCEED                           ;
             01             53   D1 000B0         CMPL     SUCCEED, #1                           ; 0813
                            4B   12 000B3         BNEQ     8$                                    ;
 00000000G   00             00   FB 000B5         CALLS    #0, EDT$$CHK_CC                       ; 0815
             0D             50   E9 000BC         BLBC     R0, 7$                                ;
             67             65   9E 000BF         MOVAB    EDT$$T_CMD_BUF, EDT$$A_CMD_END        ; 0818
 00000000G   00             01   D0 000C2         MOVL     #1, EDT$$G_CC_DONE                    ; 0819
                          00FA   31 000C9 6$:     BRW      20$                                   ; 0820
             52        04   AE   D0 000CC 7$:     MOVL     C, R2                                 ; 0823
 00000141    8F             52   D1 000D0         CMPL     R2, #321                              ; 0826
                            78   13 000D7         BEQL     14$                                   ;
 0000012C    8F             52   D1 000D9         CMPL     R2, #300                              ; 0829
                            29   19 000E0         BLSS     10$                                   ;
 000003E7    8F             52   D1 000E2         CMPL     R2, #999                              ;
```

```
                                     20  14 000E9         BGTR     10$                                  ;
                                     01  DD 000EB         PUSHL    #1                                   ; 0834
                                     7E  D4 000ED         CLRL     -(SP)                                ;
                                     52  DD 000EF         PUSHL    R2                                   ;
           00000000G  00             03  FB 000F1         CALLS    #3, EDT$$TRN_KSTR                    ;
                      53             50  D0 000F8         MOVL     R0, SUCCEED                          ;
                      02             53  D1 000FB         CMPL     SUCCEED, #2                          ; 0836
                                     04  12 000FE         BNEQ     9$                                   ;
                      50             02  D0 00100 8$:     MOVL     #2, R0                               ;
                                         04 00103         RET                                           ;
                      67      0100   C5  9E 00104 9$:     MOVAB    EDT$$T_CMD_BUF+256, EDT$$A_CMD_END   ; 0838
                                     3A  11 00109         BRB      13$                                  ; 0823
           0000007F   8F             52  D1 0010B 10$:    CMPL     R2, #127                             ; 0841
                                     19  12 00112         BNEQ     11$                                  ;
                      50             65  9E 00114         MOVAB    EDT$$T_CMD_BUF, R0                   ; 0848
                      50             66  D1 00117         CMPL     EDT$$A_CMD_BUF, R0                   ;
                                     29  1B 0011A         BLEQU    13$                                  ;
                                     66  D7 0011C         DECL     EDT$$A_CMD_BUF                       ; 0851
                      50             66  D0 0011E         MOVL     EDT$$A_CMD_BUF, R0                   ; 0852
                      7E             60  9A 00121         MOVZBL   (R0), -(SP)                          ;
           00000000G  00             01  FB 00124         CALLS    #1, EDT$$1I_DELK                     ;
                                     18  11 0012B         BRB      13$                                  ; 0823
                      15             52  D1 0012D 11$:    CMPL     R2, #21                              ; 0857
                                     08  12 00130         BNEQ     12$                                  ;
                      54             01  D0 00132         MOVL     #1, CON_U                            ; 0862
                      66             65  9E 00135         MOVAB    EDT$$T_CMD_BUF, EDT$$A_CMD_BUF       ; 0863
                                     17  11 00138         BRB      14$                                  ; 0858
                                     01  DD 0013A 12$:    PUSHL    #1                                   ; 0872
                                     52  DD 0013C         PUSHL    R2                                   ;
           00000000G  00             02  FB 0013E         CALLS    #2, EDT$$PUT_CMDCH                   ;
                                     01  DD 00145 13$:    PUSHL    #1                                   ; 0879
           00000000G  00             01  FB 00147         CALLS    #1, EDT$$TI_ENBLAUTREP               ;
                                   FF54  31 0014E         BRW      5$                                   ; 0809
                      51             66  D0 00151 14$:    MOVL     EDT$$A_CMD_BUF, R1                   ; 0886
                      50             65  9E 00154         MOVAB    EDT$$T_CMD_BUF, R0                   ;
                      50             51  D1 00157         CMPL     R1, R0                               ;
                                     15  13 0015A         BEQL     15$                                  ;
                      50             65  9E 0015C         MOVAB    EDT$$T_CMD_BUF, R0                   ; 0889
      7E              51             50  C3 0015F         SUBL3    R0, R1, -(SP)                        ;
                                     55  DD 00163         PUSHL    R5                                   ;
                                08   AE  DD 00165         PUSHL    KEY                                  ;
           00000000G  00             03  FB 00168         CALLS    #3, EDT$$DEFK                        ;
                                     0D  11 0016F         BRB      16$                                  ;
                                     54  D5 00171 15$:    TSTL     CON_U                                ; 0901
                                     11  12 00173         BNEQ     17$                                  ;
                                     6E  DD 00175         PUSHL    KEY                                  ; 0904
           00000000G  00             01  FB 00177         CALLS    #1, EDT$$CAN_KDEF                    ;
                      52             50  D0 0017E 16$:    MOVL     R0, STATUS                           ;
                      02             52  E8 00181         BLBS     STATUS, 17$                          ; 0906
                                     53  D4 00184         CLRL     SUCCEED                              ;
                      66             65  9E 00186 17$:    MOVAB    EDT$$T_CMD_BUF, EDT$$A_CMD_BUF       ; 0916
                      67             65  9E 00189         MOVAB    EDT$$T_CMD_BUF, EDT$$A_CMD_END       ; 0917
           00000000G  00             01  D0 0018C         MOVL     #1, EDT$$G_MSGFLG                    ; 0918
           00000000G  00             00  FB 00193         CALLS    #0, EDT$$ERA_MSGLN                   ; 0919
                      50 00000000G   00  D0 0019A         MOVL     EDT$$G_CS_LNO, R0                    ; 0921
                                     0F  19 001A1         BLSS     18$                                  ;
                         00000000G   00  DD 001A3         PUSHL    EDT$$G_CUR_COL                       ;
```

```
                                      50  DD 001A9         PUSHL    R0                                        ;
                    00000000G  00     02  FB 001AB         CALLS    #2, EDT$$SC_POSCSIF                       ;
                    00000000G  00     00  FB 001B2  18$:   CALLS    #0, EDT$$SC_NONREVID                      ; 0923
                                      53  D5 001B9         TSTL     SUCCEED                                   ; 0928
                                      05  12 001BB         BNEQ     19$                                       ;
                                      52  DD 001BD         PUSHL    STATUS                                    ;
                               6B     01  FB 001BF         CALLS    #1, EDT$$MSG_BELL                         ;
                               50     53  D0 001C2  19$:   MOVL     SUCCEED, R0                               ; 0930
                                          04 001C5         RET                                                ;
                                      50  D4 001C6  20$:   CLRL     R0                                        ; 0931
                                          04 001C8         RET                                                ;

; Routine Size:  457 bytes,    Routine Base:  _EDT$CODE + 0000


;  362          0932  1
;  363          0933  1 !<BLF/PAGE
;  365          0934  1 END                                                   ! of module EDT$KEYDEFKEY
;  366          0935  1
;  367          0936  0 ELUDOM




;                                         PSECT SUMMARY
;
;       Name                         Bytes                          Attributes
;
;  _EDT$CODE                               457  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          5         1            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          3         8             7          00:00.1




;                                         COMMAND QUALIFIERS

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

; Size:          457 code + 0 data bytes
; Run Time:         00:20.7
; Elapsed Time:     00:26.7
; Lines/CPU Min:     2710
; Lexemes/CPU-Min:  8021
; Memory Used:  139 pages
; Compilation Complete
```

0135 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY