C 10

```
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
```

```
EXE

Moc
---
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
SYS
LBI
LII
```

```
**FILE**ID**KEYFMTSTR
```

EDT
V04

```
KK      KK  EEEEEEEEEE  YY      YY  FFFFFFFFFF  MM      MM  TTTTTTTTTT    SSSSSSSS  TTTTTTTTTT  RRRRRRRR
KK      KK  EEEEEEEEEE  YY      YY  FFFFFFFFFF  MM      MM  TTTTTTTTTT    SSSSSSSS  TTTTTTTTTT  RRRRRRRR
KK      KK  EE          YY      YY  FF          MMMM  MMMM      TT      SS              TT      RR      RR
KK      KK  EE          YY      YY  FF          MMMM  MMMM      TT      SS              TT      RR      RR
KK    KK    EE            YY  YY    FF          MM  MM  MM      TT      SS              TT      RR      RR
KK    KK    EE            YY  YY    FF          MM  MM  MM      TT      SS              TT      RR      RR
KKKKKK      EEEEEEEE        YY      FFFFFFFF    MM      MM      TT        SSSSSS        TT      RRRRRRRR
KKKKKK      EEEEEEEE        YY      FFFFFFFF    MM      MM      TT        SSSSSS        TT      RRRRRRRR
KK    KK    EE              YY      FF          MM      MM      TT              SS      TT      RR  RR
KK    KK    EE              YY      FF          MM      MM      TT              SS      TT      RR  RR
KK      KK  EE              YY      FF          MM      MM      TT              SS      TT      RR    RR      ....
KK      KK  EE              YY      FF          MM      MM      TT              SS      TT      RR    RR      ....
KK      KK  EEEEEEEEEE      YY      FF          MM      MM      TT      SSSSSSSS        TT      RR      RR    ....
KK      KK  EEEEEEEEEE      YY      FF          MM      MM      TT      SSSSSSSS        TT      RR      RR    ....

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
```

```
0 %TITLE 'EDT$KEYFMTSTR - translate a key string'
0 MODULE EDT$KEYFMTSTR (                              ! Translate a key string
0                IDENT = 'V04-000'                          ! File: KEYFMTSTR.BLI Edit: JBS1026
0                ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:     EDT -- The DEC Standard Editor
1 !
1 ! ABSTRACT:
1 !
1 !       Translate a key string.
1 !
1 ! ENVIRONMENT:  Runs at any access mode - AST reentrant
1 !
1 ! AUTHOR: Bob Kushlis, CREATION DATE: April 7, 1979
1 !
1 ! MODIFIED BY:
1 !
1 ! 1-001 - Original.  DJS 24-Feb-1981.  This module was created by
1 !       extracting routine COM_STRING from module KEYTRAN.
1 !
1 ! 1-002.  DJS 26-Feb-1981.  Handling of prompt following ? in a key
1 !       definition modified to allow double quotes as well as single
1 !       quotes to enclose a prompt string.
1 ! 1-003 - Regularize headers.  JBS 09-Mar-1981
1 ! 1-004 - Build in T_ADV, T_BACK and T_RESET.  JBS 31-Mar-1981
1 ! 1-005 - Add return value for end of journal file.  JBS 02-Oct-1981
1 ! 1-006 - Add an additional terminator when prompt is ?*. STS 21-Oct-1981
1 ! 1-007 - If terminator of search string begins with adv or back then insert
1 !       this and throw away rest of key definition.   STS 16-Nov-1981
1 ! 1-008 - Revise the call to turn on autorepeat.  JBS 30-Jan-1982
1 ! 1-009 - Remove an unused external reference.  JBS 09-Jun-1982
```

```
1 ! 1-010 - Modify message output. SMB 30-Jun-1982
1 ! 1-011 - Add resetting search string and set direction forward to RESET.
1 !           STS 02-Jul-1982
1 ! 1-012 - Only erase the last line on a reset.  SMB 15-Jul-1982
1 ! 1-013 - Change the string match call.  JBS 19-Jul-1982
1 ! 1-014 - Catch errors in prompt string. STS 03-Aug-1982
1 ! 1-015 - Position cursor after error message. STS 10-Aug-1982
1 ! 1-016 - New implementation of defined keys.  JBS 13-Aug-1982
1 ! 1-017 - Fix a bug in recognizing ADV and BACK.  JBS 16-Aug-1982
1 ! 1-018 - Make RESET not reset the search string.  JBS for STS 26-Aug-1982
1 ! 1-019 - New screen update logic.  JBS 13-Sep-1982
1 ! 1-020 - Remove external declaration of EDT$$FMT_LIT, not used.  JBS 05-Oct-1982
1 ! 1-021 - Change the cursor positioning call, to obsolete a module.  JBS 05-Oct-1982
1 ! 1-022 - Use partial matching for ADV and BACK.  JBS 14-Dec-1982
1 ! 1-023 - Insert the whole text of a terminator to a prompt, if it starts with
1 !          ADV or BACK.  This is for compatibility with EDT version 2.  JBS 15-Dec-1982
1 ! 1-024 - Don't position cursor if no position recorded, and treat a cancelled command the same
1 !          as an undefined key.  JBS 17-Jan-1983
1 ! 1-025 - Put WPS support under a conditional.  JBS 10-Feb-1983
1 ! 1-026 - Allow unprompted input, for compatibility with EDT version 2.  JBS 01-Apr-1983
1 !--
1
```

```
;   81          0080   1 %SBTTL 'Declarations'
;   82          0081   1 !
;   83          0082   1 ! TABLE OF CONTENTS:
;   84          0083   1 !
;   85          0084   1
;   86          0085   1 REQUIRE 'EDTSRC:TRAROUNAM';
;   87          0524   1
;   88          0525   1 FORWARD ROUTINE
;   89          0526   1     EDT$$TRN_KSTR;
;   90          0527   1
;   91          0528   1 !
;   92          0529   1 ! INCLUDE FILES:
;   93          0530   1 !
;   94          0531   1
;   95          0532   1 REQUIRE 'EDTSRC:EDTREQ';
;   96          0667   1
;   97          0668   1 LIBRARY 'EDTSRC:KEYPADDEF';
;   98          0669   1
;   99          0670   1 LIBRARY 'EDTSRC:SUPPORTS';
;  100          0671   1
;  101          0672   1 !
;  102          0673   1 ! MACROS:
;  103          0674   1 !
;  104          0675   1 !        NONE
;  105          0676   1 !
;  106          0677   1 ! EQUATED SYMBOLS:
;  107          0678   1 !
;  108          0679   1 !        NONE
;  109          0680   1 !
;  110          0681   1 ! OWN STORAGE:
;  111          0682   1 !
;  112          0683   1 !        NONE
;  113          0684   1 !
;  114          0685   1 ! EXTERNAL REFERENCES:
;  115          0686   1 !
;  116          0687   1 !        In the routine
```

```
1 %SBTTL 'EDT$$TRN_KSTR  - translate a key string'
1
1 GLOBAL ROUTINE EDT$$TRN_KSTR (                       ! Translate a key string
1     K,                                               ! Key number
1     EXPAND,                                          ! 1 = do prompting
1     ECHO                                             ! 1 = echo characters
1     ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine processes the translation string of a key placing the
1 !       characters in the buffer.  If a '?' is seen, and the EXPAND parameter
1 !       is 1, then the user is prompted for input which replaces the '?' and
1 !       the prompt string in the buffer.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !  K                       The number of the key to translate.
1 !
1 !  EXPAND                  Flag indicating ? should be expanded.
1 !
1 !  ECHO                    Flag indicating that the characters should be echoed.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       EDT$$G_MESSAGE_LINE
1 !       EDT$$G_CS_LNO
1 !       EDT$$G_CUR_COL
1 !       EDT$$A_SEL_BUF
1 !       EDT$$T_CMD_BUF
1 !       EDT$$A_CUR_BUF
1 !       EDT$$G_TI_TYP
1 !       EDT$$G_TIN_ECHOPOS
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       EDT$$A_CMD_BUF
1 !       EDT$$A_CMD_END
1 !       EDT$$G_TIN_ECHOPOS
1 !       EDT$$G_TIN_ECHOFLG
1 !
1 ! ROUTINE VALUE:
1 !
1 !       0 = definition did not end with a period
1 !       1 = key was null or translation was successful and definition ended with period
1 !           or translation was cancelled
1 !       2 = end of journal file
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
```

```
;  175            0745  2       EXTERNAL ROUTINE
;  176            0746  2           EDT$$SC_ERATOEOL,
;  177            0747  2           EDT$$MSG_BELL,
;  178            0748  2           EDT$$OUT_MSG,
;  179            0749  2           EDT$$SC_REVID,
;  180            0750  2           EDT$$SC_NONREVID,
;  181            0751  2           EDT$$OUT_FMTBUF,
;  182            0752  2           EDT$$KPAD_INP,                       . Auxiliary call to EDT$$TI_RDCMDLN
;  183            0753  2           EDT$$PUT_CMDCH : NOVALUE,            ! Put a character in the command buffer
;  184            0754  2           EDT$$SC_POSCSIF : NOVALUE,           ! Position the cursor
;  185            0755  2           EDT$$TI_ENBLAUTREP,                  ! Enable or disable autorepeat
;  186            0756  2           EDT$$TI_INPCH,
;  187            0757  2           EDT$$TI_DELK,
;  188            0758  2           EDT$$TST_KEYDEF,                     ! See if a key is defined as a particular string
;  189            0759  2           EDT$$FIND_KEY;                       ! Find the definition of a key
;  190            0760  2
;  191            0761  2       EXTERNAL
;  192            0762  2           EDT$$G_MESSAGE_LINE,                 ! Message line is this + 1
;  193            0763  2           EDT$$G_CS_LNO,                       ! Line on which the cursor is positioned
;  194            0764  2           EDT$$G_CUR_COL,                      ! Column on which the cursor is positioned.
;  195            0765  2           EDT$$A_SEL_BUF,                      ! Buffer in which the mark resides
;  196            0766  2           EDT$$T_CMD_BUF,                      ! Command buffer
;  197            0767  2           EDT$$A_CMD_BUF,                      ! Pointer to next char in command buffer
;  198            0768  2           EDT$$A_CMD_END,                      ! Pointer to end of info in command buffer
;  199            0769  2           EDT$$A_CUR_BUF,                      ! Pointer to current text buffer TBCB
;  200            0770  2           EDT$$G_TI_TYP,                       ! terminal type
;  201            0771  2           EDT$$G_TIN_ECHOFLG,                  ! Was a character echoed?
;  202            0772  2           EDT$$G_TIN_ECHOPOS,                  ! The position on the echo line.
;  203            0773  2           EDT$$G_DIR_MOD,                      ! the current direction
;  204            0774  2
;  205          L 0775  2 %IF SUPPORT_WPS
;  206            0776  2 %THEN
;  207            0777  2           EDT$$G_DFLT_VERB,                    ! the current default mode
;  208            0778  2 %FI
;  209            0779  2
;  210            0780  2           EDT$$G_SEA_STRLEN;                   ! length of the current search string
;  211            0781  2
;  212            0782  2       LOCAL
;  213            0783  2           C,
;  214            0784  2           TERM,
;  215            0785  2           ALT_TERM,
;  216            0786  2           TRAN_POINT,
;  217            0787  2           TRAN_END,
;  218            0788  2           KEY_PTR : REF BLOCK [, BYTE] FIELD (KEY_DEF_FIELD);
;  219            0789  2
;  220            0790  2 !+
;  221            0791  2 ! Check for a key that is null.
;  222            0792  2 !-
;  223            0793  2
;  224            0794  3       IF ( NOT EDT$$FIND_KEY (.K, KEY_PTR))
;  225            0795  2       THEN
;  226            0796  3           BEGIN
;  227            0797  3           EDT$$A_CMD_END = CH$PTR (.EDT$$A_CMD_BUF);
;  228            0798  3           RETURN (1);
;  229            0799  2           END;
;  230            0800  2
;  231            0801  2 !+
  2 ! Look for the string RESET, which is handled specially.
  2 !-
  2
  2     IF EDT$$TST_KEYDEF (.K, UPLIT (BYTE ('RESET')), 5, 0)
  2     THEN
  3         BEGIN
  3 !+
  3 ! Reset was seen.  Flush the buffer, and undo the select range.
  3 !-
  3         EDT$$A_CMD_BUF = CH$PTR (EDT$$T_CMD_BUF);
  3         EDT$$A_CMD_END = CH$PTR (EDT$$T_CMD_BUF);
  3
  4         IF (.ECHO)
  3         THEN
  4             BEGIN
  4             EDT$$SC_POSCSIF (.EDT$$G_MESSAGE_LINE + 1, 0);      ! Erase key definition
  4             EDT$$SC_ERATOEOL ();
  4             EDT$$OUT_FMTBUF ();
  3             END;
  3
L 3 %IF SUPPORT_WPS
  3 %THEN
  3         EDT$$G_DFLT_VERB = VERB_K_MOVE;         ! reset mode to move
  3 %FI
  3
  3         EDT$$G_DIR_MOD = DIR_FORWARD;           ! reset direction to forward
  3         EDT$$A_SEL_BUF = 0;                     ! There is no more select buffer
  3         RETURN (1);
  2         END;
  2
  2 !+
  2 ! Get a pointer to the end of the translation string.
  2 !-
  2     TRAN_POINT = KEY_PTR [KEY_DEF_TEXT];
  2     TRAN_END = CH$PTR (.TRAN_POINT, .KEY_PTR [KEY_DEF_LEN]);
  2     TERM = 0;
  2 !+
  2 ! Loop through the string.
  2 !-
  2
  2     WHILE CH$PTR_LSS (.TRAN_POINT, .TRAN_END) DO
  2
  3         IF ((CH$RCHAR (.TRAN_POINT) EQL %C'?') AND .EXPAND)
  2         THEN
  2 !+
  2 ! Character is ? and expand is on.
  2 !-
  3             BEGIN
  3
  3             LOCAL
  3                 QUOTE,                          ! Character following ?
  3                 P_ADDR,                         ! Address of prompt.
  3                 STATUS,                         ! for status of kpad_inp routine
  3                 P_LEN;                          ! Length of prompt.
  3
  3             TRAN_POINT = CH$PLUS (.TRAN_POINT, 1);
  3 !+
3 ! Look for a second ?, in which case it is literal.
3 !-
3
4                 IF (CH$RCHAR (.TRAN_POINT) EQL %C'?')
3                 THEN
3                     EDT$$PUT_CMDCH (CH$RCHAR_A (TRAN_POINT), .ECHO)
3                 ELSE
3 !+
3 ! Expand the ?.
3 !-
4                     BEGIN
4                     P_LEN = 0;
4 !+
4 ! See if it's the special prompt which takes a CR.
4 !-
4
5                     IF (CH$RCHAR (.TRAN_POINT) EQL %C'*')
4                     THEN
5                         BEGIN
5                         TRAN_POINT = CH$PLUS (.TRAN_POINT, 1);       ! point to the next char
5                         ALT_TERM = ASC_K_CR;          ! set up an alternative terminator of CR
5                         END
4                     ELSE
4                         ALT_TERM = ASC_K_ESC;         ! otherwise we just stop at escapes
4
4 !+
4 ! Look for a quote mark following to start the prompt.
4 !-
4                     QUOTE = CH$RCHAR (.TRAN_POINT);
4
5                     IF ((.QUOTE EQL %C'''') OR (.QUOTE EQL %C'"'))
4                     THEN
5                         BEGIN
5 !+
5 ! Get the entire prompt string.
5 !-
5                         TRAN_POINT = CH$PLUS (.TRAN_POINT, 1);
5                         P_ADDR = .TRAN_POINT;
5
5                         WHILE (CH$PTR_GTR (.TRAN_END, .TRAN_POINT) AND (CH$RCHAR_A (TRAN_POINT) NEQ .QUOTE)) DO
5                             P_LEN = .P_LEN + 1;
5
4                         END;
4
4 !+
4 ! Position the cursor and put out the prompt.
4 !-
4                     EDT$$OUT_MSG (.EDT$$G_MESSAGE_LINE, 0, .P_ADDR, .P_LEN);
4                     EDT$$TI_ENBLAUTREP (1);
4                     EDT$$G_TIN_ECHOPOS = .EDT$$G_TIN_ECHOPOS + .P_LEN;
4                     EDT$$G_TIN_ECHOFLG = 1;
4 !+
4 ! Now, read up to the next escape character.
4 !-
4
4                     IF (EDT$$TI_INPCH (C) EQL 0) THEN RETURN (2);
4
```

```
4               STATUS = EDT$$KPAD_INP (.C, .ALT_TERM, TERM);
4
4               IF (.STATUS EQL 2) THEN RETURN (2);
4
5               IF (.STATUS EQL 0)
4               THEN
4 !+
4 ! Command was canceled, flush the buffer and return 1.
4 !-
5                  BEGIN
5                  EDT$$SC_NONREVID ();
5
5                  IF (.EDT$$G_CS_LNO GEQ 0) THEN EDT$$SC_POSCSIF (.EDT$$G_CS_LNO, .EDT$$G_CUR_COL);
5
5                  EDT$$A_CMD_BUF = CH$PTR (EDT$$T_CMD_BUF);
5                  EDT$$A_CMD_END = CH$PTR (EDT$$T_CMD_BUF);
5                  RETURN (1);
4                  END;
4
4 !+
4 ! Do a positioning sequence to left column so that we know
4 ! the command has been seen.
4 !-
4               EDT$$SC_POSCSIF (.EDT$$G_MESSAGE_LINE, 0);
4               EDT$$OUT_FMTBUF ();
4               END
4
3          END
2       ELSE
2          EDT$$PUT_CMDCH (CH$RCHAR_A (TRAN_POINT), .ECHO);
2
2 !+
2 ! If the translation ended with a period, then remove it from the
2 ! buffer and return a one, otherwise, return a 0.
2 !-
2
3    IF (CH$RCHAR (CH$PTR (.EDT$$A_CMD_BUF, -1)) EQL %C'.')
2    THEN
3       BEGIN
3       EDT$$A_CMD_BUF = CH$PTR (.EDT$$A_CMD_BUF, -1);
3
3       IF .ECHO THEN EDT$$TI_DELK (CH$RCHAR (.EDT$$A_CMD_BUF));
3
3 !+
3 ! If the terminator of input was advance or backup, then throw it into the buffer
3 ! so we can implement the KED feature of hitting advance or backup after a search
3 ! string.  For compatibility with EDT version 2, any key whose definition starts
3 ! with ADV is considered an advance key, and any key whose definition starts with
3 ! BACK is considered a backup key.  The entire text of the definition is inserted.
3 !-
3
4       IF (EDT$$TST_KEYDEF (.TERM, UPLIT (BYTE ('ADV')), 3, 1) OR      !
4          EDT$$TST_KEYDEF (.TERM, UPLIT (BYTE ('BACK')), 4, 1))
3       THEN
3          EDT$$TRN_KSTR (.TERM, .EXPAND, .ECHO);
3
3       EDT$$A_CMD_END = CH$PTR (.EDT$$A_CMD_BUF);
```

```
;  403          0973  3          RETURN (1);
;  404          0974  2          END;
;  405          0975  2
;  406          0976  2       RETURN (0);
;  407          0977  1       END;                                                of routine EDT$$TRN_KSTR


                                                                                  .TITLE  EDT$KEYFMTSTR EDT$KEYFMTSTR - translate a key s
;                                                                                                            tring
                                                                                  .IDENT  \V04-000\

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

                                         54  45  53  45  52  00000 P.AAA:  .ASCII  \RESET\                                  ;
                                                             00005         .BLKB   3
                                                 56  44  41  00008 P.AAB:  .ASCII  \ADV\                                    ;
                                                             0000B         .BLKB   1
                                             4B  43  41  42  0000C P.AAC:  .ASCII  \BACK\                                   ;

                                                                           .EXTRN  EDT$$SC_ERATOEOL
                                                                           .EXTRN  EDT$$MSG_BELL, EDT$$OUT_MSG
                                                                           .EXTRN  EDT$$SC_REVID, EDT$$SC_NONREVID
                                                                           .EXTRN  EDT$$OUT_FMTBUF
                                                                           .EXTRN  EDT$$KPAD_INP, EDT$$PUT_CMDCH
                                                                           .EXTRN  EDT$$SC_POSCSIF
                                                                           .EXTRN  EDT$$TI_ENBLAUTREP
                                                                           .EXTRN  EDT$$TI_INPCH, EDT$$TI_DELK
                                                                           .EXTRN  EDT$$TST_KEYDEF
                                                                           .EXTRN  EDT$$FIND_KEY, EDT$$G_MESSAGE_LINE
                                                                           .EXTRN  EDT$$G_CS_LNO, EDT$$G_CUR_COL
                                                                           .EXTRN  EDT$$A_SEL_BUF, EDT$$T_CMD_BUF
                                                                           .EXTRN  EDT$$A_CMD_BUF, EDT$$A_CMD_END
                                                                           .EXTRN  EDT$$A_CUR_BUF, EDT$$G_TI_TYP
                                                                           .EXTRN  EDT$$G_TIN_ECHOFLG
                                                                           .EXTRN  EDT$$G_TIN_ECHOPOS
                                                                           .EXTRN  EDT$$G_DIR_MOD, EDT$$G_DFLT_VERB
                                                                           .EXTRN  EDT$$G_SEA_STRLEN

                                                       0FFC  00000         .ENTRY  EDT$$TRN_KSTR, Save R2,R3,R4,R5,R6,R7,R8,-   ; 0690
                                                                                   R9,R10,R11                                   ;
                                 5B 00000000G  00  9E  00002         MOVAB   EDT$$G_MESSAGE_LINE, R11                       ;
                                 5A 00000000G  00  9E  00009         MOVAB   EDT$$A_CMD_END, R10                            ;
                                 59 00000000G  00  9E  00010         MOVAB   EDT$$TST_KEYDEF, R9                            ;
                                 58 00000000G  00  9E  00017         MOVAB   EDT$$T_CMD_BUF, R8                             ;
                                 57 00000000G  00  9E  0001E         MOVAB   EDT$$A_CMD_BUF, R7                             ;
                                 5E            0C  C2  00025         SUBL2   #12, SP                                        ;
                                               5E  DD  00028         PUSHL   SP                                             ; 0794
                                         04    AC  DD  0002A         PUSHL   K                                              ;
                      00000000G  00            02  FB  0002D         CALLS   #2, EDT$$FIND_KEY                              ;
                                 03            50  E8  00034         BLBS    R0, 1$                                         ;
                                           018D    31  00037         BRW     22$                                            ;
                                 7E            05  7D  0003A 1$:     MOVQ    #5, -(SP)                                      ; 0805
                                         B0    AF  9F  0003D         PUSHAB  P.AAA                                          ;
                                         04    AC  DD  00040         PUSHL   K                                              ;
                                 69            04  FB  00043         CALLS   #4, EDT$$TST_KEYDEF                            ;
                                 3B            50  E9  00046         BLBC    R0, 3$                                         ;
                                 67            68  9E  00049         MOVAB   EDT$$T_CMD_BUF, EDT$$A_CMD_BUF                 ; 0811
```

```
                                                            N  8
EDT$KEYFMTSTR    EDT$KEYFMTSTR - translate a key string     16-Sep-1984 00:43:41    VAX-11 Bliss-32 V4.0-742           Page  10
V04-000          EDT$$TRN_KSTR  - translate a key string    14-Sep-1984 12:23:22    [EDT.SRC]KEYFMTSTR.BLI;1                 (3)

                      6A              68  9E 0004C            MOVAB   EDT$$T_CMD_BUF, EDT$$A_CMD_END         ; 0812
                      1B          0C  AC  E9 0004F            BLBC    ECHO, 2$                               ; 0814
                                      7E  D4 00053            CLRL    -(SP)                                  ; 0817
          7E          6B              01  C1 00055            ADDL3   #1, EDT$$G_MESSAGE_LINE, -(SP)         ;
             00000000G  00            02  FB 00059            CALLS   #2, EDT$$SC_POSCSIF                    ;
             00000000G  00            00  FB 00060            CALLS   #0, EDT$$SC_ERATOEOL                   ; 0818
             00000000G  00            00  FB 00067            CALLS   #0, EDT$$OUT_FMTBUF                    ; 0819
                         00000000G    00  D4 0006E  2$:       CLRL    EDT$$G_DFLT_VERB                       ; 0824
             00000000G  00            01  D0 00074            MOVL    #1, EDT$$G_DIR_MOD                     ; 0827
                         00000000G    00  D4 0007B            CLRL    EDT$$A_SEL_BUF                         ; 0828
                                    0146  31 00081            BRW     23$                                    ; 0829
                      50              6E  D0 00084  3$:       MOVL    KEY_PTR, R0                            ; 0835
                      52          07  A0  9E 00087            MOVAB   7(R0), TRAN_POINT                      ;
                      55          06  A0  9A 0008B            MOVZBL  6(R0), TRAN_END                        ; 0836
                      55              52  C0 0008F            ADDL2   TRAN_POINT, TRAN_END                   ;
                                  08  AE  D4 00092            CLRL    TERM                                   ; 0837
                      55              52  D1 00095  4$:       CMPL    TRAN_POINT, TRAN_END                   ; 0842
                                      03  1F 00098            BLSSU   5$                                     ;
                                    00E0  31 0009A            BRW     19$                                    ;
                      3F              62  91 0009D  5$:       CMPB    (TRAN_POINT), #63                      ; 0844
                                      03  13 000A0            BEQL    7$                                     ;
                                    00C8  31 000A2  6$:       BRW     17$                                    ;
                      F9          08  AC  E9 000A5  7$:       BLBC    EXPAND, 6$                             ;
                                      52  D6 000A9            INCL    TRAN_POINT                             ; 0857
                      3F              62  91 000AB            CMPB    (TRAN_POINT), #63                      ; 0862
                                      F2  13 000AE            BEQL    6$                                     ;
                                      53  D4 000B0            CLRL    P_LEN                                  ; 0870
                      2A              62  91 000B2            CMPB    (TRAN_POINT), #42                      ; 0875
                                      07  12 000B5            BNEQ    8$                                     ;
                                      52  D6 000B7            INCL    TRAN_POINT                             ; 0878
                      56              0D  D0 000B9            MOVL    #13, ALT_TERM                          ; 0879
                                      03  11 000BC            BRB     9$                                     ; 0875
                      56              1B  D0 000BE  8$:       MOVL    #27, ALT_TERM                          ; 0882
                      50              62  9A 000C1  9$:       MOVZBL  (TRAN_POINT), QUOTE                    ; 0887
                      27              50  D1 000C4            CMPL    QUOTE, #39                             ; 0889
                                      05  13 000C7            BEQL    10$                                    ;
                      22              50  D1 000C9            CMPL    QUOTE, #34                             ;
                                      16  12 000CC            BNEQ    12$                                    ;
                                      52  D6 000CE  10$:      INCL    TRAN_POINT                             ; 0895
                      51              52  D0 000D0            MOVL    TRAN_POINT, P_ADDR                     ; 0896
                      52              55  D1 000D3  11$:      CMPL    TRAN_END, TRAN_POINT                   ; 0898
                                      0C  1B 000D6            BLEQU   12$                                    ;
                      54              82  9A 000D8            MOVZBL  (TRAN_POINT)+, R4                      ;
                      50              54  D1 000DB            CMPL    R4, QUOTE                              ;
                                      04  13 000DE            BEQL    12$                                    ;
                                      53  D6 000E0            INCL    P_LEN                                  ; 0899
                                      EF  11 000E2            BRB     11$                                    ;
                                      0A  BB 000E4  12$:      PUSHR   #^M<R1,R3>                             ; 0906
                                      7E  D4 000E6            CLRL    -(SP)                                  ;
                                      6B  DD 000E8            PUSHL   EDT$$G_MESSAGE_LINE                    ;
             00000000G  00            04  FB 000EA            CALLS   #4, EDT$$OUT_MSG                       ;
                                      01  DD 000F1            PUSHL   #1                                     ; 0907
             00000000G  00            01  FB 000F3            CALLS   #1, EDT$$TI_ENBLAUTREP                 ;
             00000000G  00            53  C0 000FA            ADDL2   P_LEN, EDT$$G_TIN_ECHOPOS              ; 0908
             00000000G  00            01  D0 00101            MOVL    #1, EDT$$G_TIN_ECHOFLG                 ; 0909
                                  04  AE  9F 00108            PUSHAB  C                                      ; 0914
             00000000G  00            01  FB 0010B            CALLS   #1, EDT$$TI_INPCH                      ;
```

```
                                                    B  9
EDT$KEYFMTSTR   EDT$KEYFMTSTR - translate a key string        16-Sep-1984 00:43:41    VAX-11 Bliss-32 V4.0-742          Page  11
V04-000         EDT$$TRN_KSTR  - translate a key string       14-Sep-1984 12:23:22    [EDT.SRC]KEYFMTSTR.BLI;1                 (3)

                                    50  D5 00112          TSTL    R0                                       :
                                    14  13 00114          BEQL    13$                                      :
                          08        AE  9F 00116          PUSHAB  TERM                                     : 0916
                                    56  DD 00119          PUSHL   ALT_TERM                                 :
                          0C        AE  DD 0011B          PUSHL   C                                        :
           00000000G  00            03  FB 0011E          CALLS   #3, EDT$$KPAD_INP                        :
                      02            50  D1 00125          CMPL    STATUS, #2                               : 0918
                                    04  12 00128          BNEQ    14$                                      :
                      50            02  D0 0012A 13$:     MOVL    #2, R0                                   :
                                        04 0012D          RET                                              :
                                    50  D5 0012E 14$:     TSTL    STATUS                                   : 0920
                                    27  12 00130          BNEQ    16$                                      :
           00000000G  00            00  FB 00132          CALLS   #0, EDT$$SC_NONREVID                     : 0926
                      50 00000000G  00  D0 00139          MOVL    EDT$$G_CS_LNO, R0                        : 0928
                                    0F  19 00140          BLSS    15$                                      :
                         00000000G  00  DD 00142          PUSHL   EDT$$G_CUR_COL                           :
                                    50  DD 00148          PUSHL   R0                                       :
           00000000G  00            02  FB 0014A          CALLS   #2, EDT$$SC_POSCSIF                      :
                      67            68  9E 00151 15$:     MOVAB   EDT$$T_CMD_BUF, EDT$$A_CMD_BUF           : 0930
                      6A            68  9E 00154          MOVAB   EDT$$T_CMD_BUF, EDT$$A_CMD_END           : 0931
                                    71  11 00157          BRB     23$                                      : 0932
                                    7E  D4 00159 16$:     CLRL    -(SP)                                    : 0939
                                    6D  DD 0015B          PUSHL   EDT$$G_MESSAGE_LINE                      :
           00000000G  00            02  FB 0015D          CALLS   #2, EDT$$SC_POSCSIF                      :
           00000000G  00            00  FB 00164          CALLS   #0, EDT$$OUT_FMTBUF                      : 0940
                                    0D  11 0016B          BRB     18$                                      : 0862
                          0C        AC  DD 0016D 17$:     PUSHL   ECHO                                     : 0945
                      7E            82  9A 00170          MOVZBL  (TRAN_POINT)+, -(SP)                     :
           00000000G  00            02  FB 00173          CALLS   #2, EDT$$PUT_CMDCH                       :
                                  FF18  31 0017A 18$:     BRW     4$                                       : 0844
                      50            67  D0 0017D 19$:     MOVL    EDT$$A_CMD_BUF, R0                       : 0952
                      2E  FF        A0  91 00180          CMPB    -1(R0), #46                              :
                                    48  12 00184          BNEQ    24$                                      :
                                    67  D7 00186          DECL    EDT$$A_CMD_BUF                           : 0955
                      0D  0C        AC  E9 00188          BLBC    ECHO, 20$                                : 0957
                      50            67  D0 0018C          MOVL    EDT$$A_CMD_BUF, R0                       :
                      7E            60  9A 0018F          MOVZBL  (R0), -(SP)                              :
           00000000G  00            01  FB 00192          CALLS   #1, EDT$$TI_DELK                         :
                                    01  DD 00199 20$:     PUSHL   #1                                       : 0967
                                    03  DD 0019B          PUSHL   #3                                       :
                        FE57        CF  9F 0019D          PUSHAB  P.AAB                                    :
                          14        AE  DD 001A1          PUSHL   TERM                                     :
                      69            04  FB 001A4          CALLS   #4, EDT$$TST_KEYDEF                      :
                      11            50  E8 001A7          BLBS    R0, 21$                                  :
                                    01  DD 001AA          PUSHL   #1                                       : 0968
                                    04  DD 001AC          PUSHL   #4                                       :
                        FE4A        CF  9F 001AE          PUSHAB  P.AAC                                    :
                          14        AE  DD 001B2          PUSHL   TERM                                     :
                      69            04  FB 001B5          CALLS   #4, EDT$$TST_KEYDEF                      :
                      0C            50  E9 001B8          BLBC    R0, 22$                                  :
                      7E  08        AC  7D 001BB 21$:     MOVQ    EXPAND, -(SP)                            : 0970
                          10        AE  DD 001BF          PUSHL   TERM                                     :
                FE39  CF            03  FB 001C2          CALLS   #3, EDT$$TRN_KSTR                        :
                      6A            67  D0 001C7 22$:     MOVL    EDT$$A_CMD_BUF, EDT$$A_CMD_END           : 0972
                      50            01  D0 001CA 23$:     MOVL    #1, R0                                   : 0973
                                        04 001CD          RET                                              :
                                    50  D4 001CE 24$:     CLRL    R0                                       : 0976
```

```
                                               04 001D0        RET                                      ; 0977

; Routine Size:  465 bytes,    Routine Base:  _EDT$CODE + 0010

;  408          0978  1
;  409          0979  1 !<BLF/PAGE>
```

```
: 411        0980  1 END                                             ! of module EDT$KEYFMTSTR
: 412        0981  1
: 413        0982  0 ELUDOM



:                                  PSECT SUMMARY
:
:      Name                    Bytes                     Attributes
:
:   _EDT$CODE                        481  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         6         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         5        14            7        00:00.1
:   _$255$DUA28:[EDT.SRC]SUPPORTS.L32;1            2         1        50            5        00:00.1



:                                   COMMAND QUALIFIERS

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

; Size:          465 code + 16 data bytes
; Run Time:         00:23.6
; Elapsed Time:     00:28.0
; Lines/CPU Min:     2497
; Lexemes/CPU-Min:  7571
; Memory Used:  160 pages
; Compilation Complete
```

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

KEYPADDEF
LIS

LDIVISION
LIS

KEYDEFKEY
LIS

KEYFMTSTR
LIS

LDEFK
LIS

LFILL
LIS

KEYCHR
LIS

KEYIMMINP
LIS

LDELETE
LIS

KEYCOM
LIS

KEYPUTCHR
LIS

LCLEAR
LIS

LFCOUNT
LIS

KEYPAD
LIS

LDEFM
LIS

LFLNO
LIS

KEYTRNCHR
LIS