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
```

```
EX

Mo
--
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
SY
LB
LI
```

```
**FILE**ID**KEYCHR                                              E  5
```

```
KK      KK  EEEEEEEEEE  YY      YY    CCCCCCCC  HH      HH  RRRRRRRR
KK      KK  EEEEEEEEEE  YY      YY    CCCCCCCC  HH      HH  RRRRRRRR
KK      KK  EE          YY      YY  CC          HH      HH  RR      RR
KK      KK  EE          YY      YY  CC          HH      HH  RR      RR
KK    KK    EE            YY  YY    CC          HH      HH  RR      RR
KK    KK    EE            YY  YY    CC          HH      HH  RR      RR
KKKKKK      EEEEEEEE        YY      CC          HHHHHHHHHH  RRRRRRRR
KKKKKK      EEEEEEEE        YY      CC          HHHHHHHHHH  RRRRRRRR
KK    KK    EE              YY      CC          HH      HH  RR  RR
KK    KK    EE              YY      CC          HH      HH  RR  RR
KK      KK  EE              YY      CC          HH      HH  RR    RR          ....
KK      KK  EE              YY      CC          HH      HH  RR    RR          ....
KK      KK  EEEEEEEEEE      YY        CCCCCCCC  HH      HH  RR      RR        ....
KK      KK  EEEEEEEEEE      YY        CCCCCCCC  HH      HH  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
```

EDT
V04

S
R
E
L
L
M
C

```
0 %TITLE 'EDT$KEYCHR - get next command character'
0 MODULE EDT$KEYCHR (                                        ! Get next command character
0                 IDENT = 'V04-000'                                   ! File: KEYCHR.BLI Edit: JBS1022
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 !       Get next command character.
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 EDT$$NXT_CMDCH  from module KEYTRAN.
1 ! 1-002 - Regularize headers.  JBS 06-Mar-1981
1 ! 1-003 - Add a check for repeat counts allowed or not.  STS 26-Aug-1981
1 ! 1-004 - Fixed problem with norepeat so it doesn't insert number.
1 !               STS 27-Aug-1981
1 ! 1-005 - Add return value for end of journal file.  JBS 02-Oct-1981
1 ! 1-006 - Don't pass values of keypad except arrow keys. STS 15-Apr-1982
1 ! 1-007 - Accept a flag indicating validity of repeat counts. STS 16-Jun-1982
1 ! 1-008 - Remove reference to TI_STARTECHO.  SMB 22-Jun-1982
1 ! 1-009 - Change numeric test.  JBS 19-Jul-1982
1 ! 1-010 - Don't ring bell if quiet set. STS 09-Aug-1982
1 ! 1-011 - New implementation of defined keys.  JBS 13-Aug-1982
1 ! 1-012 - Don't suppress all keys in NOKEYPAD mode.  JBS 16-Aug-1982
```

```
1 ! 1-013 - Fix up norepeat case.  JBS 16-Aug-1982
1 ! 1-014 - Allow for 8-bit keyboards.  JBS 17-Aug-1982
1 ! 1-015 - Add SS3 for 8-bit keyboards.  JBS 20-Aug-1982
1 ! 1-016 - Erase the message line after the first keystroke, if it has
1 !          a message on it.  JBS 06-Oct-1982
1 ! 1-017 - Output the format buffer just before waiting for input, in case
1 !          there is anything in it.  JBS 07-Oct-1982
1 ! 1-018 - Don't clear the message line until an entire key sequence has been read.
1 !          JBS 09-Oct-1982
1 ! 1-019 - Output the format buffer in another case of waiting for input.  JBS 09-Oct-1982
1 ! 1-020 - Change the call to EDT$$TST_KEYDEF.  JBS 14-Dec-1982
1 ! 1-021 - Complete the implementation of 8-bit keyboards.  JBS 20-Jan-1983
1 ! 1-022 - Add a conditional for VT220 su--ort.  JBS 11-Feb-1983
1 !--
1
```

```
;      0073   1 %SBTTL 'Declarations'
;      0074   1 !
;      0075   1 ! TABLE OF CONTENTS:
;      0076   1 !
;      0077   1
;      0078   1 REQUIRE 'EDTSRC:TRAROUNAM';
;      0517   1
;      0518   1 FORWARD ROUTINE
;      0519   1     EDT$$NXT_CMDCH;
;      0520   1
;      0521   1 !
;      0522   1 ! INCLUDE FILES:
;      0523   1 !
;      0524   1
;      0525   1 REQUIRE 'EDTSRC:EDTREQ';
;      0660   1
;      0661   1 LIBRARY 'EDTSRC:KEYPADDEF';
;      0662   1
;      0663   1 LIBRARY 'EDTSRC:SUPPORTS';
;      0664   1
;      0665   1 !
;      0666   1 ! MACROS:
;      0667   1 !
;      0668   1 !       NONE
;      0669   1 !
;      0670   1 ! EQUATED SYMBOLS:
;      0671   1 !
;      0672   1 !       NONE
;      0673   1 !
;      0674   1 ! OWN STORAGE:
;      0675   1 !
;      0676   1 !       NONE
;      0677   1 !
;      0678   1 ! EXTERNAL REFERENCES:
;      0679   1 !
;      0680   1 !       In the routine
```

```
1 %SBTTL 'EDT$$NXT_CMDCH  - get next command character'
1
1 GLOBAL ROUTINE EDT$$NXT_CMDCH (                        ! Get next command character
1     C,                                                 ! Place to store the character
1     REPEAT                                             ! Accept repeat counts
1     ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Get the next command character.  Keypad keys are converted to their
1 !       numeric equivalent and the functions of GOLD are handled here.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !  C                     The address of a fullword to receive the character.
1 !
1 !  REPEAT                Flag indicating whether to accept repeat counts.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       EDT$$T_CMD_BUF
1 !       EDT$$G_KPAD
1 !       EDT$$A_CMD_BUF
1 !       EDT$$G_RPT
1 !       EDT$$A_CMD_END
1 !       EDT$$G_MSGFLG
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       EDT$$A_CMD_BUF
1 !
1 ! ROUTINE VALUE:
1 !
1 !       0 = control U typed, no command.
1 !       1 = a command key was typed.
1 !       2 = end of journal file.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2    BEGIN
2
2    EXTERNAL ROUTINE
2       EDT$$SC_REVID,                                   ! Turn on reverse video
2       EDT$$PUT_CMDCH : NOVALUE,                        ! Put a character in the command buffer
2       EDT$$TRN_KPADK,                                  ! Read an escape sequence
2       EDT$$TI_INPCH,
2       EDT$$TI_DELK,
2       EDT$$TI_ECHOCH,
2       EDT$$ERA_MSGLN,                                  ! Erase the message lines
2       EDT$$RING_BELL,                                  ! Ring the bell on the terminal
2       EDT$$TST_KEYDEF,                                 ! Test a key to see if it is defined as a particular string
2       EDT$$OUT_FMTBUF;                                 ! Output the format buffer, if it is non-empty
```

```
:  168           0738  2
:  169           0739  2      EXTERNAL
:  170           0740  2          EDT$$G_MSGFLG,                          ! 1 = there is text on the message line
:  171           0741  2          EDT$$G_QUIET,                           ! quiet flag
:  172           0742  2          EDT$$T_CMD_BUF,                         ! Command buffer
:  173           0743  2          EDT$$A_CMD_BUF,                         ! Pointer to next char in command buffer
:  174           0744  2          EDT$$G_KPAD,                            ! in keypad mode?
:  175           0745  2          EDT$$G_RPT,                             ! Flag for repeat counts
:  176           0746  2          EDT$$A_CMD_END                          ! Pointer to end of info in command buffer
:  177           0747  2
:  178         L 0748  2 %IF SUPPORT_VT220
:  179           0749  2 %THEN
:  180           0750  2
:  181           0751  2          EDT$$B_CHAR_INFO : BLOCKVECTOR [256, 1, BYTE]   ! Information about characters
:  182           0752  2 %FI
:  183           0753  2
:  184           0754  2      ;
:  185           0755  2
:  186           0756  2      LOCAL
:  187           0757  2          SAVE_POINT,                             ! Starting EDT$$A_CMD_BUF .
:  188           0758  2          MY_C;
:  189           0759  2
:  190           0760  2      SAVE_POINT = .EDT$$A_CMD_BUF;
:  191           0761  2 !+
:  192           0762  2 ! Make sure the user sees anything which might be in the format buffer.
:  193           0763  2 !-
:  194           0764  2      EDT$$OUT_FMTBUF ();
:  195           0765  2 !+
:  196           0766  2 ! Get a character.
:  197           0767  2 !-
:  198           0768  2
:  199           0769  2      IF (EDT$$TI_INPCH (MY_C) EQL 0) THEN RETURN (2);
:  200           0770  2
:  201           0771  2 !+
:  202           0772  2 ! If the character is an escape, CSI or SS3, then look for a keypad sequence.
:  203           0773  2 !-
:  204           0774  2
:  205         L 0775  2 %IF SUPPORT_VT220
:  206           0776  2 %THEN
:  207           0777  2
:  208           0778  3      IF ((.MY_C EQL ASC_K_ESC) OR (.MY_C EQL ASC_K_CSI) OR (.MY_C EQL ASC_K_SS3))
:  209           0779  2      THEN
:  210         U 0780  2 %ELSE
:  211         U 0781  2
:  212         U 0782  2          IF (.MY_C EQL ASC_K_ESC)
:  213         U 0783  2          THEN
:  214           0784  2 %FI
:  215           0785  2
:  216           0786  3              BEGIN
:  217           0787  3 !+
:  218           0788  3 ! Translate keypad character.
:  219           0789  3 !-
:  220           0790  3
:  221           0791  3              IF (EDT$$TRN_KPADK (MY_C) EQL 0) THEN RETURN (2);
:  222           0792  3
:  223           0793  3 !+
:  224           0794  3 ! If there is any text on the message line, er.se it, since the user
```

```
   3 ! has now had an opportunity to read it.
   3 !-
   3
   3                    IF (.EDT$$G_MSGFLG NEQ 0) THEN EDT$$ERA_MSGLN ();
   3
   4                    IF ( NOT .EDT$$G_KPAD)
   3                    THEN
   4                        BEGIN
   4
   5                        IF ((.MY_C EQL K_UP) OR (.MY_C EQL K_DOWN) OR (.MY_C EQL K_RIGHT) OR (.MY_C EQL K_LEFT))
   4                        THEN
   4                            .C = .MY_C
   4                        ELSE
   4                            .C = K_PF1;
   4
   4                        RETURN (1);
   3                        END;
   3
   2                    END;
   2
   2 !+
   2 ! If the key is defined as GOLD, handle it here.
   2 !-
   2
   2        WHILE EDT$$TST_KEYDEF (.MY_C, UPLIT (BYTE ('GOLD')), 4, 0) DO
   3            BEGIN
   3 !+
   3 ! Look at the next character.  It should be either a digit, a sign
   3 ! or a letter.
   3 !-
   3            EDT$$OUT_FMTBUF ();
   3
   3            IF (EDT$$TI_INPCH (MY_C) EQL 0) THEN RETURN (2);
   3
   3            EDT$$SC_REVID ();
   3
L  3 %IF SUPPORT_VT220
   3 %THEN
   3
   4            IF (((.EDT$$B_CHAR_INFO [.MY_C, 0, 0, 8, 0] EQL %X'F0') OR (.MY_C EQL %C'-')) AND .REPEAT)
   3            THEN
U  3 %ELSE
U  3
U  3                IF (((((.MY_C GEQ %C'0') AND (.MY_C LEQ %C'9'))) OR (.MY_C EQL %C'-')) AND .REPEAT)
U  3                THEN
   3 %FI
   3
   4                    BEGIN
   4 !+
   4 ! Start of a repeat count.  If this was not the first key pressed
   4 ! then re-start the count by clearing the buffer back to the
   4 ! point where we started.
   4 !-
   4
   5                    IF (.EDT$$G_RPT EQL 0)
   4                    THEN
   5                        BEGIN
```

```
  5
  5                     IF ( NOT .EDT$$G_QUIET) THEN EDT$$RING_BELL ();
  5
  5                     EDT$$OUT_FMTBUF ();
  5
  5                     IF (EDT$$TI_INPCH (MY_C) EQL 0) THEN RETURN (2);
  5
L 5 %IF SUPPORT_VT220
  5 %THEN
  5
  6                     IF ((.MY_C EQL ASC_K_ESC) OR (.MY_C EQL ASC_K_CSI) OR (.MY_C EQL ASC_K_SS3))
  5                     THEN
U 5 %ELSE
U 5
U 5                         IF (.MY_C EQL ASC_K_ESC)
U 5                         THEN
  5 %FI
  5
  6                             BEGIN
  6
  6                             IF (EDT$$TRN_KPADK (MY_C) EQL 0) THEN RETURN (2);
  6
  5                             END;
  5
  5                     END
  4                 ELSE
  5                     BEGIN
  5
  6                     IF CH$PTR_NEQ (.EDT$$A_CMD_BUF, .SAVE_POINT)
  5                     THEN
  6                         BEGIN
  6                         EDT$$A_CMD_BUF = .SAVE_POINT;
  6                         EDT$$ERA_MSGLN ();
  5                         END;
  5
  5 !+
  5 ! Now continue reading and echoing characters until a non-digit is found.
  5 !-
  5
  5                     DO
  6                         BEGIN
  6                         EDT$$TI_ECHOCH (.MY_C);
  6                         EDT$$PUT_CMDCH (.MY_C, 0);
  6                         EDT$$OUT_FMTBUF ();
  6
  6                         IF (EDT$$TI_INPCH (MY_C) EQL 0) THEN RETURN (2);
  6
  6 !+
  6 ! Look for delete and CTRL/U
  6 !-
  6
  6                         WHILE (.MY_C EQL ASC_K_DEL) DO
  7                             BEGIN
  7
  8                             IF CH$PTR_NEQ (.EDT$$A_CMD_BUF, .SAVE_POINT)
  7                             THEN
  8                                 BEGIN
```

```
:  339          0909  8                                 EDT$$A_CMD_BUF = CH$PLUS (.EDT$$A_CMD_BUF, -1);
:  340          0910  8                                 EDT$$TI_DECK (CH$RCHAR (.EDT$$A_CMD_BUF));
:  341          0911  7                                 END;
:  342          0912  7
:  343          0913  7                             EDT$$OUT_FMTBUF ();
:  344          0914  7
:  345          0915  7                             IF (EDT$$TI_INPCH (MY_C) EQL 0) THEN RETURN (2);
:  346          0916  7
:  347          0917  6                             END;
:  348          0918  6
:  349          0919  7                         IF (.MY_C EQL ASC_K_CTRL_U)
:  350          0920  6                         THEN
:  351          0921  7                             BEGIN
:  352          0922  7                             EDT$$ERA_MSGLN ();
:  353          0923  7                             EDT$$A_CMD_END = EDT$$T_CMD_BUF;
:  354          0924  7                             RETURN (0);
:  355          0925  6                             END;
:  356          0926  6
:  357          0927  6                         END
:  358          0928  5                     UNTIL
:  359          0929  5
:  360        L 0930  5 %IF SUPPORT_VT220
:  361          0931  5 %THEN
:  362          0932  6                         (.EDT$$B_CHAR_INFO [.MY_C, 0, 0, 8, 0] NEQ %X'F0')
:  363        U 0933  6 %ELSE
:  364        U 0934  6                         ((.MY_C LSS %C'0') OR (.MY_C GTR %C'9'))
:  365          0935  6 %FI
:  366          0936  6
:  367          0937  5                     ;
:  368          0938  5 !+
:  369          0939  5 ! If the repeat sequence was ended by an escape, CSI or SS3 then get the key.
:  370          0940  5 !-
:  371          0941  5
:  372        L 0942  5 %IF SUPPORT_VT220
:  373          0943  5 %THEN
:  374          0944  5
:  375          0945  6                     IF ((.MY_C EQL ASC_K_ESC) OR (.MY_C EQL ASC_K_CSI) OR (.MY_C EQL ASC_K_SS3))
:  376          0946  5                     THEN
:  377        U 0947  5 %ELSE
:  378        U 0948  5
:  379        U 0949  5                         IF (.MY_C EQL ASC_K_ESC)
:  380        U 0950  5                         THEN
:  381          0951  5 %FI
:  382          0952  5
:  383          0953  6                             BEGIN
:  384          0954  6
:  385          0955  6                             IF (EDT$$TRN_KPADK (MY_C) EQL 0) THEN RETURN (2);
:  386          0956  6
:  387          0957  5                             END;
:  388          0958  5
:  389          0959  5                     END
:  390          0960  5
:  391          0961  4                 END
:  392          0962  3             ELSE
:  393          0963  3
:  394        L 0964  3 %IF SUPPORT_VT220
:  395          0965  3 %THEN
```

```
;  396           0966  3
;  397           0967  4                     IF ((.MY_C EQL ASC_K_ESC) OR (.MY_C EQL ASC_K_CSI) OR (.MY_C EQL ASC_K_SS3))
;  398           0968  3                     THEN
;  399         U 0969  3 %ELSE
;  400         U 0970  3
;  401         U 0971  3                         IF (.MY_C EQL ASC_K_ESC)
;  402         U 0972  3                         THEN
;  403           0973  3 %FI
;  404           0974  3
;  405           0975  3 !+
;  406           0976  3 ! Here if we had gold followed by an escape, CSI or SS3.
;  407           0977  3 ! Translate the sequence and use the golded function of the key.
;  408           0978  3 !-
;  409           0979  4                             BEGIN
;  410           0980  4
;  411           0981  4                             IF (EDT$$TRN_KPADK (MY_C) EQL 0) THEN RETURN (2);
;  412           0982  4
;  413           0983  4                             MY_C = .MY_C + K_GOLD_BASE;
;  414           0984  4                             END
;  415           0985  3                         ELSE
;  416           0986  4                             BEGIN
;  417           0987  4 !+
;  418           0988  4 ! Here if we had gold followed by a character from the main keyboard.
;  419           0989  4 !-
;  420           0990  4
;  421         L 0991  4 %IF SUPPORT_VT220
;  422           0992  4 %THEN
;  423           0993  4
;  424           0994  4                             IF .EDT$$B_CHAR_INFO [.MY_C, 0, 0, 1, 0]        ! Lower case
;  425           0995  4                             THEN
;  426         U 0996  4 %ELSE
;  427         U 0997  4
;  428         U 0998  4                                 IF ((.MY_C GEQ %C'a') AND (.MY_C LEQ %C'z'))
;  429         U 0999  4                                 THEN
;  430           1000  4 %FI
;  431           1001  4
;  432           1002  4                                     MY_C = .MY_C - %C'a' + %C'A';   ! Force to upper
;  433           1003  4
;  434           1004  4                             MY_C = .MY_C + K_GOLD_BASE;
;  435           1005  3                             END;
;  436           1006  3
;  437           1007  2                 END;
;  438           1008  2
;  439           1009  2 !+
;  440           1010  2 ! Return the coded character.
;  441           1011  2 !-
;  442           1012  2     .C = .MY_C;
;  443           1013  2     RETURN (T);
;  444           1014  1     END;                                            ! of routine EDT$$NXT_CMDCH


                                                                               .TITLE  EDT$KEYCHR EDT$KEYCHR - get next command charac
;                                                                                                         ter
                                                                               .IDENT  \V04-000\

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

```
                         44  4C  4F  47  00000 P.AAA:  .ASCII  \GOLD\                                          ;

                                                       .EXTRN  EDT$$SC_REVID, EDT$$PUT_CMDCH
                                                       .EXTRN  EDT$$TRN_KPADK, EDT$$TI_INPCH
                                                       .EXTRN  EDT$$TI_DELK, EDT$$TI_ECHOCH
                                                       .EXTRN  EDT$$ERA_MSGLN, EDT$$RING_BELL
                                                       .EXTRN  EDT$$TST_KEYDEF
                                                       .EXTRN  EDT$$OUT_FMTBUF
                                                       .EXTRN  EDT$$G_MSGFLG, EDT$$G_QUIET
                                                       .EXTRN  EDT$$T_CMD_BUF, EDT$$A_CMD_BUF
                                                       .EXTRN  EDT$$G_KPAD, EDT$$G_RPT
                                                       .EXTRN  EDT$$A_CMD_END, EDT$$B_CHAR_INFO

                                   07FC 00000          .ENTRY  EDT$$NXT_CMDCH, Save R2,R3,R4,R5,R6,R7,R8,-  ; 0683
                                                               R9,R10                                       ;
                   5A 00000000G  00  9E 00002          MOVAB   EDT$$B_CHAR_INFO, R10                        ;
                   59 00000000G  00  9E 00009          MOVAB   EDT$$ERA_MSGLN, R9                           ;
                   58 00000000G  00  9E 00010          MOVAB   EDT$$TRN_KPADK, R8                           ;
                   57 00000000G  00  9E 00017          MOVAB   EDT$$TI_INPCH, R7                            ;
                   56 00000000G  00  9E 0001E          MOVAB   EDT$$OUT_FMTBUF, R6                          ;
                   55 00000000G  00  9E 00025          MOVAB   EDT$$A_CMD_BUF, R5                           ;
                   5E            04  C2 0002C          SUBL2   #4, SP                                       ;
                   54            65  D0 0002F          MOVL    EDT$$A_CMD_BUF, SAVE_POINT                   ; 0760
                   66            00  FB 00032          CALLS   #0, EDT$$OUT_FMTBUF                          ; 0764
                                 5E  DD 00035          PUSHL   SP                                           ; 0769
                   67            01  FB 00037          CALLS   #1, EDT$$TI_INPCH                            ;
                                 50  D5 0003A          TSTL    R0                                           ;
                                 1E  13 0003C          BEQL    2$                                           ;
                   1B            6E  D1 0003E          CMPL    MY_C, #27                                    ; 0778
                                 12  13 00041          BEQL    1$                                           ;
        0000009B   8F            6E  D1 00043          CMPL    MY_C, #155                                   ;
                                 09  13 0004A          BEQL    1$                                           ;
        0000008F   8F            6E  D1 0004C          CMPL    MY_C, #143                                   ;
                                 51  12 00053          BNEQ    7$                                           ;
                                 5E  DD 00055 1$:      PUSHL   SP                                           ; 0791
                   68            01  FB 00057          CALLS   #1, EDT$$TRN_KPADK                           ;
                                 50  D5 0005A          TSTL    R0                                           ;
                                 69  13 0005C 2$:      BEQL    9$                                           ;
                      00000000G  00  D5 0005E          TSTL    EDT$$G_MSGFLG                                ; 0798
                                 03  13 00064          BEQL    3$                                           ;
                   69            00  FB 00066          CALLS   #0, EDT$$ERA_MSGLN                           ;
                   36 00000000G  00  E8 00069 3$:      BLBS    EDT$$G_KPAD, 7$                              ; 0800
                   50            6E  D0 00070          MOVL    MY_C, R0                                     ; 0804
        00000138   8F            50  D1 00073          CMPL    R0, #312                                     ;
                                 1B  13 0007A          BEQL    4$                                           ;
        00000139   8F            50  D1 0007C          CMPL    R0, #313                                     ;
                                 12  13 00083          BEQL    4$                                           ;
        0000013A   8F            50  D1 00085          CMPL    R0, #314                                     ;
                                 09  13 0008C          BEQL    4$                                           ;
        0000013B   8F            50  D1 0008E          CMPL    R0, #315                                     ;
                                 06  12 00095          BNEQ    5$                                           ;
              04   BC            50  D0 00097 4$:      MOVL    R0, @C                                       ; 0806
                                 06  11 0009B          BRB     6$                                           ;
              04   BC      0140  8F  3C 0009D 5$:      MOVZWL  #320, @C                                     ; 0808
                               0146  31 000A3 6$:      BRW     31$                                          ; 0810
                   7E            04  7D 000A6 7$:      MOVQ    #4, -(SP)                                    ; 0819
                          FF4F   CF  9F 000A9          PUSHAB  P.AAA                                        ;
```

```
                                                                        C  6
EDT$KEYCHR        EDT$KEYCHR - get next command character          16-Sep-1984 00:41:34    VAX-11 Bliss-32 V4.0-742              Page  11      EDT
V04-000           EDT$$NXT_CMDCH  - get next command character     14-Sep-1984 12:23:20    DISK$VMSMASTER:[EDT.SRC]KEYCHR.BLI;1     (3)     V04

                               0C    AE  DD 000AD         PUSHL    MY_C                                    ;              ;
               00000000G  00         04  FB 000B0         CALLS    #4, EDT$$TST_KEYDEF                     ;              ;
                          03         50  E8 000B7         BLBS     R0, 8$                                  ;              ;
                                   012B  31 000BA         BRW      30$                                     ;              ;
                          66         00  FB 000BD  8$:    CALLS    #0, EDT$$OUT_FMTBUF                     ; 0825
                                     5E  DD 000C0         PUSHL    SP                                      ; 0827
                          67         01  FB 000C2         CALLS    #1, EDT$$TI_INPCH                       ;
                                     50  D5 000C5         TSTL     R0                                      ;
                                     62  13 000C7  9$:    BEQL     15$                                     ;
               00000000G  00         00  FB 000C9         CALLS    #0, EDT$$SC_REVID                       ; 0829         ;
                          52         6E  D0 000D0         MOVL     MY_C, R2                                ; 0834         ;
       53                 52         5A  C1 000D3         ADDL3    R10, R2, R3                             ;              ;
                   F0     8F         63  91 000D7         CMPB     (R3), #240                              ;              ;
                                     08  13 000DB         BEQL     11$                                     ;              ;
                          2D         52  D1 000DD         CMPL     R2, #45                                 ;
                                     03  13 000E0         BEQL     11$                                     ;
                                   00CE  31 000E2  10$:   BRW      25$                                     ;
                          F9     08  AC  E9 000E5  11$:   BLBC     REPEAT, 10$                             ;
                           00000000G 00  D5 000E9         TSTL     EDT$$G_RPT                              ; 0849         ;
                                     3C  12 000EF         BNEQ     16$                                     ;              ;
                          07 00000000G 00  E8 000F1       BLBS     EDT$$G_QUIET, 12$                       ; 0853         ;
               00000000G  00         00  FB 000F8         CALLS    #0, EDT$$RING_BELL                      ;              ;
                          66         00  FB 000FF  12$:   CALLS    #0, EDT$$OUT_FMTBUF                     ; 0855         ;
                                     5E  DD 00102         PUSHL    SP                                      ; 0857         ;
                          67         01  FB 00104         CALLS    #1, EDT$$TI_INPCH                       ;              ;
                                     50  D5 00107         TSTL     R0                                      ;              ;
                                     6F  13 00109         BEQL     21$                                     ;
                          1B         6E  D1 0010B         CMPL     MY_C, #27                               ; 0862
                                     12  13 0010E         BEQL     14$                                     ;
               0000009B   8F         6E  D1 00110         CMPL     MY_C, #155                              ;
                                     09  13 00117         BEQL     14$                                     ;
               0000008F   8F         6E  D1 00119         CMPL     MY_C, #143                              ;
                                     84  12 00120  13$:   BNEQ     7$                                      ;
                                     5E  DD 00122  14$:   PUSHL    SP                                      ; 0872         ;
                          68         01  FB 00124         CALLS    #1, EDT$$TRN_KPADK                      ;
                                     50  D5 00127         TSTL     R0                                      ;              ;
                                     F5  12 00129         BNEQ     13$                                     ;
                                     4D  11 0012B  15$:   BRB      21$                                     ;              ; S
                          54         65  D1 0012D  16$:   CMPL     EDT$$A_CMD_BUF, SAVE_POINT              ; 0880         ; R
                                     06  13 00130         BEQL     17$                                     ;              ; E
                          65         54  D0 00132         MOVL     SAVE_POINT, EDT$$A_CMD_BUF              ; 0883         ; L
                          69         00  FB 00135         CALLS    #0, EDT$$ERA_MSGLN                      ; 0884         ; L
                          53         6E  D0 00138  17$:   MOVL     MY_C, R3                                ; 0893         ; M
                                     53  DD 0013B  18$:   PUSHL    R3                                      ;              ; C
               00000000G  00         01  FB 0013D         CALLS    #1, EDT$$TI_ECHOCH                      ;
                                     7E  D4 00144         CLRL     -(SP)                                   ; 0894
                                     53  DD 00146         PUSHL    R3                                      ;
               00000000G  00         02  FB 00148         CALLS    #2, EDT$$PUT_CMDCH                      ;
                                     1D  11 0014F         BRB      20$                                     ; 0895
               0000007F   8F         6E  D1 00151  19$:   CMPL     MY_C, #127                              ; 0903
                                     22  12 00158         BNEQ     22$                                     ;
                          54         65  D1 0015A         CMPL     EDT$$A_CMD_BUF, SAVE_POINT              ; 0906
                                     0F  13 0015D         BEQL     20$                                     ;
                                     65  D7 0015F         DECL     EDT$$A_CMD_BUF                          ; 0909
                          50         65  D0 00161         MOVL     EDT$$A_CMD_BUF, R0                      ; 0910
                          7E         60  9A 00164         MOVZBL   (R0), -(SP)                             ;
               00000000G  00         01  FB 00167         CALLS    #1, EDT$$TI_DELK                        ;
```

```
                 66          00  FB 0016E 20$:    CALLS    #0, EDT$$OUT_FMTBUF                          ; 0913
                             5E  DD 00171         PUSHL    SP                                           ; 0915
                 67          01  FB 00173         CALLS    #1, EDT$$TI_INPCH                            ;
                             50  D5 00176         TSTL     R0                                           ;
                             D7  12 00178         BNEQ     19$                                          ;
                             57  11 0017A 21$:    BRB      27$                                          ;
                 53          6E  D0 0017C 22$:    MOVL     MY_C, R3                                     ; 0919
                 15          53  D1 0017F         CMPL     R3, #21                                      ;
                             10  12 00182         BNEQ     23$                                          ;
                 69          00  FB 00184         CALLS    #0, EDT$$ERA_MSGLN                           ; 0922
         00000000G  00 00000000G  00  9E 00187    MOVAB    EDT$$T_CMD_BUF, EDT$$A_CMD_END               ; 0923
                             5C  11 00192         BRB      32$                                          ; 0924
            F0   8F        6A43  91 00194 23$:    CMPB     EDT$$B_CHAR_INFO[R3], #240                   ; 0932
                             A0  13 00199         BEQL     18$                                          ;
                 1B          53  D1 0019B         CMPL     R3, #27                                      ; 0945
                             82  13 0019E 24$:    BEQL     14$                                          ;
         0000009B   8F       53  D1 001A0         CMPL     R3, #155                                     ;
                             F5  13 001A7         BEQL     24$                                          ;
         0000008F   8F       53  D1 001A9         CMPL     R3, #143                                     ;
                           FF6D  31 001B0         BRW      13$                                          ;
                 1B          52  D1 001B3 25$:    CMPL     R2, #27                                      ; 0967
                             12  13 001B6         BEQL     26$                                          ;
         0000009B   8F       52  D1 001B8         CMPL     R2, #155                                     ;
                             09  13 001BF         BEQL     26$                                          ;
         0000008F   8F       52  D1 001C1         CMPL     R2, #143                                     ;
                             0D  12 001C8         BNEQ     28$                                          ;
                             5E  DD 001CA 26$:    PUSHL    SP                                           ; 0981
                 68          01  FB 001CC         CALLS    #1, EDT$$TRN_KPADK                           ;
                             50  D5 001CF         TSTL     R0                                           ;
                             0B  12 001D1         BNEQ     29$                                          ;
                 50          02  D0 001D3 27$:    MOVL     #2, R0                                       ;
                                 04 001D6         RET                                                   ;
                 04          63  E9 001D7 28$:    BLBC     (R3), 29$                                    ; 0994
                 6E     E0   A2  9E 001DA         MOVAB    -32(R2), MY_C                                ; 1002
                 6E 000001F4 8F  C0 001DE 29$:    ADDL2    #500, MY_C                                   ; 0983
                           FEBE  31 001E5         BRW      7$                                           ; 0819
            04   BC          6E  D0 001E8 30$:    MOVL     MY_C, @C                                     ; 1012
                 50          01  D0 001EC 31$:    MOVL     #1, R0                                       ; 1013
                                 04 001EF         RET                                                   ;
                             50  D4 001F0 32$:    CLRL     R0                                           ; 1014
                                 04 001F2         RET                                                   ;

; Routine Size:  499 bytes,    Routine Base:  _EDT$CODE + 0004


;  445          1015  1
;  446          1016  1 !<BLF/PAGE>
;  448         1017  1 END                                              ! of module EDT$$KEYCHR
;  449         1018  1
;  450         1019  0 ELUDOM



;                                PSECT SUMMARY
;
;       Name                         Bytes                          Attributes
;
;  _EDT$CODE                               503  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         6        17           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$:KEYCHR/OBJ=OBJ$:KEYCHR MSRC$:KEYCHR.BLI/UPDATE=(ENH$:KEYCHR)

; Size:          499 code + 4 data bytes
; Run Time:         00:23.9
; Elapsed Time:     00:28.3
; Lines/CPU Min:     2558
; Lexemes/CPU-Min:  7810
; Memory Used:  158 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