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

FILEID**EXEC

```
EEEEEEEEEE  XX      XX  EEEEEEEEEE    CCCCCCCC
EEEEEEEEEE  XX      XX  EEEEEEEEEE    CCCCCCCC
EE          XX      XX  EE          CC
EE          XX      XX  EE          CC
EE            XX  XX    EE          CC
EE            XX  XX    EE          CC
EEEEEEEE        XX      EEEEEEEE    CC
EEEEEEEE        XX      EEEEEEEE    CC
EE            XX  XX    EE          CC
EE            XX  XX    EE          CC
EE          XX      XX  EE          CC              ....
EE          XX      XX  EE          CC              ....
EEEEEEEEEE  XX      XX  EEEEEEEEEE    CCCCCCCC      ....
EEEEEEEEEE  XX      XX  EEEEEEEEEE    CCCCCCCC      ....

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

```
;  1          0001  0 %TITLE 'EDT$EXEC - enter and exit line and change mode'
;  2          0002  0 MODULE EDT$EXEC (                       . Enter and Exit Line and Change Mode
;  3          0003  0                 IDENT = 'V04-000'                ! File: EXEC.BLI Edit: JBS1024
;  4          0004  0                 ) =
;  5          0005  1 BEGIN
;  6          0006  1 !
;  7          0007  1 !**************************************************************************
;  8          0008  1 !*                                                                        *
;  9          0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                              *
; 10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                *
; 11          0011  1 !*  ALL RIGHTS RESERVED.                                                  *
; 12          0012  1 !*                                                                        *
; 13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
; 14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
; 16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
; 17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
; 18          0018  1 !*  TRANSFERRED.                                                           *
; 19          0019  1 !*                                                                        *
; 20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
; 21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
; 22          0022  1 !*  CORPORATION.                                                           *
; 23          0023  1 !*                                                                        *
; 24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
; 25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
; 26          0026  1 !*                                                                        *
; 27          0027  1 !*                                                                        *
; 28          0028  1 !**************************************************************************
; 29          0029  1 !
; 30          0030  1
; 31          0031  1 !++
; 32          0032  1 ! FACILITY:     EDT -- The DEC Standard Editor
; 33          0033  1 !
; 34          0034  1 ! ABSTRACT:
; 35          0035  1 !
; 36          0036  1 !       This module provides the initialization and termination
; 37          0037  1 !       processing for entering and exiting line and change mode.
; 38          0038  1 !
; 39          0039  1 ! ENVIRONMENT:  Runs at any access mode - AST reentrant
; 40          0040  1 !
; 41          0041  1 ! AUTHOR: Bob Kushlis, CREATION DATE: 6-AUG-1979
; 42          0042  1 !
; 43          0043  1 ! MODIFIED BY:
; 44          0044  1 !
; 45          0045  1 ! 1-001 - Original.  DJS 18-FEB-1981.  This module was created by
; 46          0046  1 !       extracting routine EDT$$EXE_CMD  from module EDTCTRL.
; 47          0047  1 ! 1-002 - Regularized the headers.  JBS 24-Feb-1981
; 48          0048  1 ! 1-003 - Fix the file name and module name, and remove cortrol C handling.
; 49          0049  1 !         JBS 04-Mar-1981
; 50          0050  1 ! 1-004 - Use the ASSERT macro.  JBS 01-Jun-1981
; 51          0051  1 ! 1-005 - Don't change the current mode to line just because the change mode
; 52          0052  1 !         processor exits, since it now exits at the end of the journal file.
; 53          0053  1 !         Rearrange the looping logic to take the new recovery procedure into
; 54          0054  1 !         account; this means this module no longer has EDT$$G_RCOV_MOD as an
; 55          0055  1 !         implicit input.  Also, remove L_LINE and L_CHANGE.  JBS 02-Oct-1981
; 56          0056  1 ! 1-006 - Don't fail to enter change mode if the terminal is unknown or hard copy.
; 57          0057  1 !         JBS 20-Oct-1981
```

```
;    58          0058  1 ! 1-007 - Take the prompt from the global rather than from a constant.  JBS 21-Oct-1981
;    59          0059  1 ! 1-008 - Remove length of prompt string.  JBS 23-Oct-1981
;    60          0060  1 ! 1-009 - Correct an error in a comment.  JBS 17-Nov-1981
;    61          0061  1 ! 1-010 - Put L_LINE and L_CHANGE back TMV 7-Dec-81
;    62          0062  1 ! 1-011 - Add control C handling for line mode.  JBS 21-Dec-1981
;    63          0063  1 ! 1-012 - Debug control C handling.  JBS 24-Dec-1981
;    64          0064  1 ! 1-013 - Before returning load MACCAL into memory, in case we are returning
;    65          0065  1 !           to it on the PDP-11.  JBS 10-Mar-1982
;    66          0066  1 ! 1-014 - Revise the nooverlay logic to avoid undefined symbols at build time.
;    67          0067  1 !           JBS 15-Mar-1982
;    68          0068  1 ! 1-015 - Remove the reference to EDT$$LOAD_MACCAL.  JBS 18-Mar-1982
;    69          0069  1 ! 1-016 - Add alternative control C message.  JBS 24-May-1982
;    70          0070  1 ! 1-017 - Remove L_LINE and L_CHANGE.  JBS 03-Jun-1982
;    71          0071  1 ! 1-018 - Reset command buffer if ^c seen from terminal. STS 15-Jul-1982
;    72          0072  1 ! 1-019 - Don't clear the control C counters unless we are reading from
;    73          0073  1 !           a terminal.  Also, treat the startup file like a macro for
;    74          0074  1 !           control C processing.  JBS 28-Jul-1982
;    75          0075  1 ! 1-020 - Improve the control Z ignoring logic: don't journal ignored
;    76          0076  1 !           control Z's, since three in a row will be mistaken for the
;    77          0077  1 !           end of the journal file on a /RECOVER.  JBS 29-Jul-1982
;    78          0078  1 ! 1-021 - Simplify the call to initialize the keypad.  JBS 13-Jul-1982
;    79          0079  1 ! 1-022 - Clear control C if we don't have a macro. STS 06-Oct-1982
;    80          0080  1 ! 1-023 - Improve the appearance of the listing.  JBS 14-Jun-1983
;    81          0081  1 ! 1-024 - Exit on control Z from terminal.  JBS 20-Jun-1983
;    82          0082  1 !--
;    83          0083  1
```

```
;    85          0084  1 %SBTTL 'Declarations'
;    86          0085  1 !
;    87          0086  1 ! TABLE OF CONTENTS:
;    88          0087  1 !
;    89          0088  1
;    90          0089  1 REQUIRE 'EDTSRC:TRAROUNAM';
;    91          0528  1
;    92          0529  1 FORWARD ROUTINE
;    93          0530  1     EDT$$EXE_CMD;
;    94          0531  1
;    95          0532  1 !
;    96          0533  1 ! INCLUDE FILES:
;    97          0534  1 !
;    98          0535  1
;    99          0536  1 REQUIRE 'EDTSRC:EDTREQ';
;   100          0671  1
;   101          0672  1 !
;   102          0673  1 ! MACROS:
;   103          0674  1 !
;   104          0675  1 !       NONE
;   105          0676  1 !
;   106          0677  1 ! EQUATED SYMBOLS:
;   107          0678  1 !
;   108          0679  1 !+
;   109          0680  1 ! EDT$$MACCAL uses this symbol indirectly to refer to this routine, so that it
;   110          0681  1 ! can be excluded from the overlay analysis.
;   111          0682  1 !-
;   112          0683  1
;   113          0684  1 GLOBAL BIND
;   114          0685  1     ROUTINE
;   115          0686  1     EDT$$EXE_CMD_NOOVERLAY_REF = EDT$$EXE_CMD;
;   116          0687  1
;   117          0688  1 !
;   118          0689  1 ! OWN STORAGE:
;   119          0690  1 !
;   120          0691  1 !       NONE
;   121          0692  1 !
;   122          0693  1 ! EXTERNAL REFERENCES:
;   123          0694  1 !
;   124          0695  1 !       In the routine
```

```
;   126          0696  1 %SBTTL 'EDT$$EXE_CMD  - execute commands'
;   127          0697  1
;   128          0698  1 GLOBAL ROUTINE EDT$$EXE_CMD (                    ! Execute commands
;   129          0699  1     SOURCE                                       ! New input source
;   130          0700  1     ) =
;   131          0701  1
;   132          0702  1 !++
;   133          0703  1 ! FUNCTIONAL DESCRIPTION:
;   134          0704  1 !
;   135          0705  1 !       Enter and exit line and change mode.
;   136          0706  1 !
;   137          0707  1 ! FORMAL PARAMETERS:
;   138          0708  1 !
;   139          0709  1 !  SOURCE                new input source
;   140          0710  1 !
;   141          0711  1 ! IMPLICIT INPUTS:
;   142          0712  1 !
;   143          0713  1 !       EDT$$G_TRN_TBLINIT
;   144          0714  1 !       EDT$$A_CMD_END
;   145          0715  1 !       EDT$$T_CMD_BUF
;   146          0716  1 !       EDT$$G_CMD_LEN
;   147          0717  1 !       EDT$$G_EXITD
;   148          0718  1 !       EDT$$G_EDIT_MOD
;   149          0719  1 !       EDT$$G_TI_TYP
;   150          0720  1 !       EDT$$G_TXT_ONSCR
;   151          0721  1 !       EDT$$Z_PA_STK
;   152          0722  1 !       EDT$$T_PMT_LINE
;   153          0723  1 !       EDT$$G_CC_DONE
;   154          0724  1 !
;   155          0725  1 ! IMPLICIT OUTPUTS:
;   156          0726  1 !
;   157          0727  1 !       EDT$$G_INP_SRC
;   158          0728  1 !       EDT$$A_CMD_BUF
;   159          0729  1 !       EDT$$G_TXT_ONSCR
;   160          0730  1 !       EDT$$A_CMD_END
;   161          0731  1 !       EDT$$G_CC_DONE
;   162          0732  1 !
;   163          0733  1 ! ROUTINE VALUE:
;   164          0734  1 !
;   165          0735  1 !       0 - We have seen an EXIT command; EDT$$G_EXITD will be set
;   166          0736  1 !       1 - We have reached end of file.
;   167          0737  1 !       2 - We saw a control C
;   168          0738  1 !
;   169          0739  1 ! SIDE EFFECTS:
;   170          0740  1 !
;   171          0741  1 !       NONE
;   172          0742  1 !
;   173          0743  1 !--
;   174          0744  1
;   175          0745  2     BEGIN
;   176          0746  2
;   177          0747  2     EXTERNAL ROUTINE
;   178          0748  2         EDT$$DEF_DFLTK,                          ! Define the default keys
;   179          0749  2         EDT$$CHM_EXE,                            ! Execute change mode commands
;   180          0750  2         EDT$$GET_LN,                             ! Get a line from the input device
;   181          0751  2         EDT$$PA_CMD,                             ! Parse a line-mode command
;   182          0752  2         EDT$$LNM_CMD,                            ! Execute a line mode command
```

```
;  183          0753  2          EDT$$CHK_CC,                        ! Check for control C
;  184          0754  2          EDT$$MSG_TOSTR : NOVALUE,           ! Convert a message code to a string
;  185          0755  2          EDT$$OUT_FMTBUF : NOVALUE,          ! Print the format buffer
;  186          0756  2          EDT$$CLR_CC : NOVALUE,              ! Clear control C flag
;  187          0757  2          EDT$$FMT_CRLF : NOVALUE;            ! End of output line
;  188          0758  2
;  189          0759  2      EXTERNAL
;  190          0760  2          EDT$$G_TRN_TBLINIT,                 ! Translate table init flag
;  191          0761  2          EDT$$A_CMD_END,                     ! Pointer to end of command line
;  192          0762  2          EDT$$T_CMD_BUF,                     ! Command line buffer
;  193          0763  2          EDT$$A_CMD_BUF,                     ! Pointer into command line buffer
;  194          0764  2          EDT$$G_CMD_LEN,                     ! Length of current command
;  195          0765  2          EDT$$G_EXITD,                       ! Did we exit with the last command?
;  196          0766  2          EDT$$G_INP_SRC,                     ! Source of input commands
;  197          0767  2          EDT$$G_EDIT_MOD,                    ! Current editing mode
;  198          0768  2          EDT$$G_TI_TYP,                      ! Type of terminal
;  199          0769  2          EDT$$G_TXT_ONSCR,                   ! Text was written to screen
;  200          0770  2          EDT$$Z_PA_STK,                      ! Parser semantic stack
;  201          0771  2          EDT$$T_PMT_LINE : VECTOR [, BYTE],  ! Counted ASCII string of line-mode prompt
;  202          0772  2          EDT$$G_CC_DONE,                     ! Set if control C actually aborted anything
;  203          0773  2          EDT$$G_TIN_OBUFPOS,                 ! Current length of journal record
;  204          0774  2          EDT$$G_JOU_VALID;                   ! Set to 1 if there is a record in the journal buffer
;  205          0775  2
;  206          0776  2  !+
;  207          0777  2  ! Specify the messages used in this routine.
;  208          0778  2  !-
;  209          0779  2      MESSAGES ((ABOBYCC, CTRC__IGN));
;  210          0780  2
;  211          0781  2      LOCAL
;  212          0782  2          SAVE_SOURCE,
;  213          0783  2          EOF_FLAG;                                   ! 1 = reached end of file, 2 = control C
;  214          0784  2
;  215          0785  2  !+
;  216          0786  2  ! If we are not coming from the terminal but a control C has been typed, get out now.
;  217          0787  2  !-
;  218          0788  2
;  219          0789  3      IF ((.EDT$$G_INP_SRC NEQ INP_TERM) AND (.EDT$$G_INP_SRC NEQ INP_JOURNAL))
;  220          0790  2      THEN
;  221          0791  2
;  222          0792  2          IF EDT$$CHK_CC ()
;  223          0793  2          THEN
;  224          0794  3              BEGIN
;  225          0795  3              EDT$$G_CC_DONE = 1;
;  226          0796  3              RETURN (2);
;  227          0797  2              END;
;  228          0798  2
;  229          0799  2      SAVE_SOURCE = .EDT$$G_INP_SRC;
;  230          0800  2      EDT$$G_INP_SRC = .SOURCE;
;  231          0801  2      EDT$$A_CMD_BUF = CH$PTR (EDT$$T_CMD_BUF);
;  232          0802  2      CH$WCHAR ('!', EDT$$T_CMD_BUF);
;  233          0803  2      EOF_FLAG = 0;
;  234          0804  2
;  235          0805  2      WHILE ((.EOF_FLAG EQL 0) AND ( NOT .EDT$$G_EXITD)) DO
;  236          0806  2
;  237          0807  3          IF (.EDT$$G_EDIT_MOD EQL CHANGE_MODE)
;  238          0808  2          THEN
;  239          0809  2  !+
```

```
;  240          0810  2 ! We are in change mode, go into the change mode processor.
;  241          0811  2 !-
;  242          0812  3             BEGIN
;  243          0813  3
;  244          0814  3             LOCAL
;  245          0815  3                 STATUS;
;  246          0816  3
;  247          0817  3 !+
;  248          0818  3 ! Init the keypad translation table if necessary.
;  249          0819  3 ! If the initialization fails, drop into line mode.
;  250          0820  3 !-
;  251          0821  3             STATUS = 1;
;  252          0822  3
;  253          0823  3             IF ( NOT .EDT$$G_TRN_TBLINIT) THEN STATUS = EDT$$DEF_DFLTK ();
;  254          0824  3
;  255          0825  3             IF .STATUS
;  256          0826  3             THEN
;  257          0827  3                 EOF_FLAG = EDT$$CHM_EXE ()
;  258          0828  3             ELSE
;  259          0829  4                 BEGIN
;  260          0830  4                 EDT$$MSG_TOSTR (.STATUS);
;  261          0831  4                 EDT$$FMT_CRLF ();
;  262          0832  4                 EDT$$G_EDIT_MOD = LINE_MODE;
;  263          0833  3                 END;
;  264          0834  3
;  265          0835  3 !+
;  266          0836  3 ! EDT$$CHM_EXE returns after a series of commands.  It returns on the EXIT
;  267          0837  3 ! command (which sets EDT$$G_EDIT_MOD to LINE_MODE) and at the end of the
;  268          0838  3 ! journal file.  In the latter case it returns a 1; if it returns because
;  269          0839  3 ! of exiting it returns a 0.
;  270          0840  3 !-
;  271          0841  3             END
;  272          0842  2         ELSE
;  273          0843  3             BEGIN
;  274          0844  3
;  275          0845  3             IF (.EDT$$G_INP_SRC NEQ INP_MACRO) THEN EDT$$CLR_CC ();
;  276          0846  3
;  277          0847  3             ASSERT (.EDT$$G_EDIT_MOD EQL LINE_MODE);    ! We are in line mode
;  278          0848  3 !+
;  279          0849  3 ! See if we must read a line.
;  280          0850  3 !-
;  281          0851  3
;  282          0852  4             IF (CH$RCHAR (.EDT$$A_CMD_BUF) EQL %C'!')
;  283          0853  3             THEN
;  284          0854  4                 BEGIN
;  285          0855  4
;  286          0856  4                 LOCAL
;  287          0857  4                     LOOP_DONE,
;  288          0858  4                     EOF_COUNTER;
;  289          0859  4
;  290          0860  4                 EOF_COUNTER = 0;
;  291          0861  4                 LOOP_DONE = 0;
;  292          0862  4 !+
;  293          0863  4 ! We stay in the following loop until we either have a command to process or we have
;  294          0864  4 ! end of file from somewhere other than the terminal.  In the later case EOF_FLAG is set.
;  295          0865  4 !-
;  296          0866  4
```

```
;  297          0867  4                   WHILE ( NOT .LOOP_DONE) DO
;  298          0868  5                       BEGIN
;  299          0869  5
;  300          0870  6                       IF (EDT$$GET_LN (EDT$$T_PMT_LINE [1], .EDT$$T_PMT_LINE [0]) EQL 0)
;  301          0871  5                       THEN
;  302          0872  5                           LOOP_DONE = 1
;  303          0873  5                       ELSE
;  304          0874  6                           BEGIN
;  305          0875  6
;  306          0876  7                           IF (.EDT$$G_INP_SRC NEQ INP_TERM)
;  307          0877  6                           THEN
;  308          0878  7                               BEGIN
;  309          0879  7                               EOF_FLAG = 1;
;  310          0880  7                               LOOP_DONE = 1;
;  311          0881  7                               END
;  312          0882  6                           ELSE
;  313          0883  7                               BEGIN
;  314          0884  7 !+
;  315          0885  7 ! This is an EOF from the terminal, invalidate the journal buffer.
;  316          0886  7 ! If we journal this it may be confused with the end of the journal file.
;  317          0887  7 !-
;  318          0888  7                               EDT$$G_TIN_OBUFPOS = 0;
;  319          0889  7                               EDT$$G_JOU_VALID = 0;
;  320          0890  7 !+
;  321          0891  7 ! Convert three EOFs in a row into a QUIT/SAVE command.   The purpose
;  322          0892  7 ! of this feature is to prevent a batch job from getting into an endless
;  323          0893  7 ! loop if EDT runs into the end of the command file.
;  324          0894  7 ! Note that the command is not placed in the journal file.
;  325          0895  7 !-
;  326          0896  7                               EOF_COUNTER = .EOF_COUNTER + 1;
;  327          0897  7
;  328          0898  8                               IF (.EOF_COUNTER GEQ 3)
;  329          0899  7                               THEN
;  330          0900  8                                   BEGIN
;  331          0901  8                                   CH$MOVE (10, UPLIT (BYTE ('QUIT/SAVE!')), .EDT$$A_CMD_END);
;  332          0902  8                                   EDT$$A_CMD_END = CH$PLUS (.EDT$$A_CMD_END, 9);
;  333          0903  8                                   EDT$$G_CMD_LEN = .EDT$$G_CMD_LEN + 9;
;  334          0904  8                                   LOOP_DONE = 1;
;  335          0905  7                                   END;
;  336          0906  7
;  337          0907  6                               END;
;  338          0908  6
;  339          0909  5                           END;
;  340          0910  5
;  341          0911  4                       END;
;  342          0912  4
;  343          0913  3                   END;
;  344          0914  3
;  345          0915  3 !+
;  346          0916  3 ! Parse and execute the command. (Unless it is a comment or we have reached EOF.)
;  347          0917  3 !-
;  348          0918  3
;  349          0919  4               IF ( NOT .EOF_FLAG)
;  350          0920  3               THEN
;  351          0921  4                   BEGIN
;  352          0922  4
;  353          0923  5                   IF ((.EDT$$G_CMD_LEN EQL 0) OR (CH$RCHAR (EDT$$T_CMD_BUF) NEQ %C'!'))
```

```
;   354          0924  4                    THEN
;   355          0925  4
;   356          0926  4                        IF EDT$$PA_CMD ()
;   357          0927  4                        THEN
;   358          0928  4                            EDT$$LNM_CMD (EDT$$Z_PA_STK)
;   359          0929  4                        ELSE
;   360          0930  4                            EDT$$G_TXT_ONSCR = .EDT$$G_TXT_ONSCR + 1;
;   361          0931  4
;   362          0932  4 !+
;   363          0933  4 ! If the control C flag is set, it is likely that the command was aborted
;   364          0934  4 ! by a control C.
;   365          0935  4 !-
;   366          0936  4
;   367          0937  4                    IF EDT$$CHK_CC ()
;   368          0938  4                    THEN
;   369          0939  5                        BEGIN
;   370          0940  5
;   371          0941  6                        IF ((.EDT$$G_INP_SRC EQL INP_TERM) OR (.EDT$$G_INP_SRC EQL INP_JOURNAL))
;   372          0942  5                        THEN
;   373          0943  6                            BEGIN
;   374          0944  6
;   375          0945  7                            IF (.EDT$$G_CC_DONE)
;   376          0946  6                            THEN
;   377          0947  6                                EDT$$MSG_TOSTR (EDT$_ABOBYCC)
;   378          0948  6                            ELSE
;   379          0949  6                                EDT$$MSG_TOSTR (EDT$_CTRC__IGN);
;   380          0950  6
;   381          0951  6                            EDT$$OUT_FMTBUF ();
;   382          0952  6 !+
;   383          0953  6 ! We've seen a control C from the terminal, so clear out the command buffer.
;   384          0954  6 !-
;   385          0955  6                            EDT$$A_CMD_BUF = EDT$$T_CMD_BUF;
;   386          0956  6                            EDT$$A_CMD_END = .EDT$$A_CMD_BUF;
;   387          0957  6                            CH$WCHAR ('!', .EDT$$A_CMD_END);
;   388          0958  6                            END
;   389          0959  5                    ELSE
;   390          0960  5 !+
;   391          0961  5 ! We have seen a control C but we are not reading from the terminal.  Return to our caller, who
;   392          0962  5 ! will arrange to print an appropriate message.
;   393          0963  5 !-
;   394          0964  6                        BEGIN
;   395          0965  6                        EDT$$G_CC_DONE = 1;
;   396          0966  6                        EOF_FLAG = 2;
;   397          0967  5                        END;
;   398          0968  5
;   399          0969  4                    END;
;   400          0970  4
;   401          0971  3                END;
;   402          0972  3
;   403          0973  2            END;
;   404          0974  2
;   405          0975  2 !+
;   406          0976  2 ! When the loop falls through we have either reached an end of file or
;   407          0977  2 ! seen an EXIT command.  Tell the caller which.
;   408          0978  2 !-
;   409          0979  2     EDT$$G_INP_SRC = .SAVE_SOURCE;
;   410          0980  2     RETURN (.EOF_FLAG);
```

```
;  411          0981  1     END;                                        ' of routine EDT$$EXE_CMD


                                                                       .TITLE  EDT$EXEC EDT$EXEC - enter and exit line and cha
;                                                                                      nge mode
                                                                       .IDENT  \V04-000\
                                                                       .PSECT  _EDT$CODE,NOWRT,  SHR,  PIC,2
                    21  45  56  41  53  2F  54  49  55  51  00000 P.AAA:  .ASCII  \QUIT/SAVE!\                                ;
                                                                       .EXTRN  EDT$$DEF_DFLTK, EDT$$CHM_EXE
                                                                       .EXTRN  EDT$$GET_LN, EDT$$PA_CMD
                                                                       .EXTRN  EDT$$LNM_CMD, EDT$$CHK_CC
                                                                       .EXTRN  EDT$$MSG_TOSTR, EDT$$OUT_FMTBUF
                                                                       .EXTRN  EDT$$CLR_CC, EDT$$FMT_CRLF
                                                                       .EXTRN  EDT$$G_TRN_TBLINIT
                                                                       .EXTRN  EDT$$A_CMD_END, EDT$$T_CMD_BUF
                                                                       .EXTRN  EDT$$A_CMD_BUF, EDT$$G_CMD_LEN
                                                                       .EXTRN  EDT$$G_EXITD, EDT$$G_INP_SRC
                                                                       .EXTRN  EDT$$G_EDIT_MOD
                                                                       .EXTRN  FDT$$G_TI_TYP, EDT$$G_TXT_ONSCR
                                                                       .EXTRN  EDT$$Z_PA_STK, EDT$$T_PMT_LINE
                                                                       .EXTRN  EDT$$G_CC_DONE, EDT$$G_TIR_OBUFPOS
                                                                       .EXTRN  EDT$$G_JOU_VALID
                                                                       .EXTRN  EDT$_ABOBYCC, EDT$_CTRC__IGN
                                                                       .EXTRN  EDT$$INTER_ERR

                                                OFFC 00000             .ENTRY  EDT$$EXE_CMD, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 0698
                                                                               R10,R11
                    5B 00000000G   00   9E 00002                       MOVAB   EDT$$T_CMD_BUF, R11
                    5A 00000000G   00   9E 00009                       MOVAB   EDT$$G_INP_SRC, R10
                    50             6A   D0 00010                       MOVL    EDT$$G_INP_SRC, R0                          ; 0789
                                   1A   13 00013                       BEQL    1$
                    03             50   D1 00015                       CMPL    R0, #3
                                   15   13 00018                       BEQL    1$
00000000G           00             00   FB 0001A                       CALLS   #0, EDT$$CHK_CC                             ; 0792
                    08             50   E9 00021                       BLBC    R0, 1$
00000000G           00             01   D0 00024                       MOVL    #1, EDT$$G_CC_DONE                          ; 0795
                    50             02   D0 0002B                       MOVL    #2, R0                                      ; 0796
                                   04   0002E                           RET
                    59             6A   D0 0002F 1$:                    MOVL    EDT$$G_INP_SRC, SAVE_SOURCE                 ; 0799
                    6A       04    AC   D0 00032                       MOVL    SOURCE, EDT$$G_INP_SRC                      ; 0800
00000000G           00             6B   9E 00036                       MOVAB   EDT$$T_CMD_BUF, EDT$$A_CMD_BUF              ; 0801
                    6B             21   90 0003D                       MOVB    #33, EDT$$T_CMD_BUF                         ; 0802
                                   58   D4 00040                       CLRL    EOF_FLAG                                    ; 0803
                                   58   D5 00042 2$:                    TSTL    EOF_FLAG                                    ; 0805
                                   03   13 00044                       BEQL    4$
                                 015A   31 00046 3$:                    BRW     23$
                    F6 00000000G   00   E8 00049 4$:                    BLBS    EDT$$G_EXITD, 3$
                       00000000G   00   D5 00050                       TSTL    EDT$$G_EDIT_MOD                             ; 0807
                                   39   12 00056                       BNEQ    7$
                    50             01   D0 00058                       MOVL    #1, STATUS                                  ; 0821
                    07 00000000G   00   E8 0005B                       BLBS    EDT$$G_TRN_TBLINIT, 5$                      ; 0823
00000000G           00             00   FB 00062                       CALLS   #0, EDT$$DEF_DFLTK
                    0C             50   E9 00069 5$:                    BLBC    STATUS, 6$                                  ; 0825
00000000G           00             00   FB 0006C                       CALLS   #0, EDT$$CHM_EXE                            ; 0827
```

```
                  58               50  D0 00073         MOVL    R0, EOF_FLAG                    ;
                                   CA  11 00076         BRB     2$                              ;
                                   50  DD 00078 6$:     PUSHL   STATUS                          ; 0830
         00000000G  00             01  FB 0007A         CALLS   #1, EDT$$MSG_TOSTR              ;
         00000000G  00             00  FB 00081         CALLS   #0, EDT$$FMT_CRLF               ; 0831
         00000000G  00             01  D0 00088         MOVL    #1, EDT$$G_EDIT_MOD             ; 0832
                                   B1  11 0008F         BRB     2$                              ; 0807
                  01               6A  D1 00091 7$:     CMPL    EDT$$G_INP_SRC, #1              ; 0845
                                   07  13 00094         BEQL    8$                              ;
         00000000G  00             00  FB 00096         CALLS   #0, EDT$$CLR_CC                 ;
                  01  00000000G    00  D1 0009D 8$:     CMPL    EDT$$G_EDIT_MOD, #1             ; 0847
                                   07  13 000A4         BEQL    9$                              ;
         00000000G  00             00  FB 000A6         CALLS   #0, EDT$$INTER_ERR              ;
                  50  00000000G    00  D0 000AD 9$:     MOVL    EDT$$A_CMD_BUF, R0              ; 0852
                  21               60  91 000B4         CMPB    (R0), #33                       ;
                                   59  12 000B7         BNEQ    13$                             ;
                                   56  7C 000B9         CLRQ    LOOP_DONE                       ; 0861
                  54               56  E8 000BB 10$:    BLBS    LOOP_DONE, 13$                  ; 0867
                  7E  00000000G    00  9A 000BE         MOVZBL  EDT$$T_PMT_LINE, -(SP)          ; 0870
                      00000000G    00  9F 000C5         PUSHAB  EDT$$T_PMT_LINE+1               ;
         00000000G  00             02  FB 000CB         CALLS   #2, EDT$$GET_LN                 ;
                                   50  D5 000D2         TSTL    R0                              ;
                                   37  13 000D4         BEQL    12$                             ;
                                   6A  D5 000D6         TSTL    EDT$$G_INP_SRC                  ; 0876
                                   05  13 000D8         BEQL    11$                             ;
                  58               01  D0 000DA         MOVL    #1, EOF_FLAG                    ; 0879
                                   2E  11 000DD         BRB     12$                             ; 0880
                      00000000G    00  D4 000DF 11$:    CLRL    EDT$$G_TIN_OBUFPOS              ; 0888
                      00000000G    00  D4 000E5         CLRL    EDT$$G_JOU_VALID                ; 0889
                                   57  D6 000EB         INCL    EOF_COUNTER                     ; 0896
                  03               57  D1 000ED         CMPL    EOF_COUNTER, #3                 ; 0898
                                   C9  19 000F0         BLSS    10$                             ;
                  50  00000000G    00  D0 000F2         MOVL    EDT$$A_CMD_END, R0              ; 0901
     60      FEF8   CF             0A  28 000F9         MOVC3   #10, P.AAA, (R0)                ;
         00000000G  00             09  C0 000FF         ADDL2   #9, EDT$$A_CMD_END              ; 0902
         00000000G  00             09  C0 00106         ADDL2   #9, EDT$$G_CMD_LEN              ; 0903
                  56               01  D0 0010D 12$:    MOVL    #1, LOOP_DONE                   ; 0904
                                   A9  11 00110         BRB     10$                             ; 0867
                  7F               58  E8 00112 13$:    BLBS    EOF_FLAG, 20$                   ; 0919
                      00000000G    00  D5 00115         TSTL    EDT$$G_CMD_LEN                  ; 0923
                                   05  13 0011B         BEQL    14$                             ;
                  21               6B  91 0011D         CMPB    EDT$$T_CMD_BUF, #33             ;
                                   1F  13 00120         BEQL    16$                             ;
         00000000G  00             00  FB 00122 14$:    CALLS   #0, EDT$$PA_CMD                 ; 0926
                  0F               50  E9 00129         BLBC    R0, 15$                         ;
                      00000000G    00  9F 0012C         PUSHAB  EDT$$Z_PA_STK                   ; 0928
         00000000G  00             01  FB 00132         CALLS   #1, EDT$$CNM_CMD                ;
                                   06  11 00139         BRB     16$                             ;
                      00000000G    00  D6 0013B 15$:    INCL    EDT$$G_TXT_ONSCR                ; 0930
         00000000G  00             00  FB 00141 16$:    CALLS   #0, EDT$$CHK_CC                 ; 0937
                  55               50  E9 00148         BLBC    R0, 22$                         ;
                  50               6A  D0 0014B         MOVL    EDT$$G_INP_SRC, R0              ; 0941
                                   05  13 0014E         BEQL    17$                             ;
                  03               50  D1 00150         CMPL    R0, #3                          ;
                                   41  12 00153         BNEQ    21$                             ;
                  08  00000000G    00  E9 00155 17$:    BLBC    EDT$$G_CC_DONE, 18$             ; 0945
                      00000000G    8F  DD 0015C         PUSHL   #EDT$_ABOBYCC                   ; 0947
```

```
                                   06  11 00162           BRB     19$                                        ;
                        00000000G  8F  DD 00164 18$:      PUSHL   #EDT$_CTRC__IGN                            ; 0949
        00000000G  00              01  FB 0016A 19$:      CALLS   #1, EDT$$MSG_TOSTR                         ;
        00000000G  00              00  FB 00171           CALLS   #0, EDT$$OUT_FMTBUF                        ; 0951
        00000000G  00              6B  9E 00178           MOVAB   EDT$$T_CMD_BUF, EDT$$A_CMD_BUF             ; 0955
        00000000G  00 00000000G    00  D0 0017F           MOVL    EDT$$A_CMD_BUF, EDT$$A_CMD_END             ; 0956
                   50 00000000G    00  D0 0018A           MOVL    EDT$$A_CMD_END, R0                         ; 0957
                   60              21  90 00191           MOVB    #33, (R0)                                  ;
                                   0A  11 00194 20$:      BRB     22$                                        ; 0941
        00000000G  00              01  D0 00196 21$:      MOVL    #1, EDT$$G_CC_DONE                         ; 0965
                   58              02  D0 0019D           MOVL    #2, EOF_FLAG                               ; 0966
                                 FE9F  31 001A0 22$:      BRW     2$                                         ; 0937
                   6A              59  D0 001A3 23$:      MOVL    SAVE_SOURCE, EDT$$G_INP_SRC                ; 0979
                   50              58  D0 001A6           MOVL    EOF_FLAG, R0                               ; 0980
                                       04 001A9           RET                                                ; 0981

; Routine Size:  426 bytes,    Routine Base:  _EDT$CODE + 000A


;   412          0982  1
;   413          0983  1 !<BLF/PAGE>
```

```
;  415          0984  1 END                                          . of module EDT$EXEC
;  416          0985  1
;  417          0986  0 ELUDOM


                                                    EDT$$EXE_CMD_NOOVERLAY_REF==
                                                                  EDT$$EXE_CMD

;                                  PSECT SUMMARY

;       Name                    Bytes                           Attributes

;   _EDT$CODE                       436  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         7         1          40       00:00.2
;   _$255$DUA28:[EDT.SRC]PSECTS.L32;1          2         1        50           7       00:00.1


;                                     COMMAND QUALIFIERS

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

; Size:          426 code + 10 data bytes
; Run Time:          00:21.3
; Elapsed Time:      00:27.9
; Lines/CPU Min:      2778
; Lexemes/CPU-Min:    8023
; Memory Used:  137 pages
; Compilation Complete
```

EXTEND
LIS

FDEC
LIS

FILL
LIS

FINDPARA
LIS

FORLF
LIS

EDT
LIS

EXEC
LIS

EXECNOO
LIS

FILEIO
LIS

EDTVECTOR
LIS

FINDKEY
LIS

FCOLINC
LIS

FINAL
LIS

FIOHNDLR
LIS

DEFKEY
LIS

ERRMSG
LIS

FCHAR
LIS