```
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**CHMCHANGE

L 10

```
  CCCCCCCC  HH      HH  MM      MM    CCCCCCCC  HH      HH    AAAAAA    NN      NN    GGGGGGGG  EEEEEEEEEE
  CCCCCCCC  HH      HH  MM      MM    CCCCCCCC  HH      HH    AAAAAA    NN      NN    GGGGGGGG  EEEEEEEEEE
CC          HH      HH  MMMM  MMMM  CC          HH      HH  AA      AA  NN      NN  GG          EE
CC          HH      HH  MMMM  MMMM  CC          HH      HH  AA      AA  NN      NN  GG          EE
CC          HH      HH  MM  MM  MM  CC          HH      HH  AA      AA  NNNN    NN  GG          EE
CC          HH      HH  MM  MM  MM  CC          HH      HH  AA      AA  NNNN    NN  GG          EE
CC          HHHHHHHHHH  MM      MM  CC          HHHHHHHHHH  AA      AA  NN  NN  NN  GG          EEEEEEEE
CC          HHHHHHHHHH  MM      MM  CC          HHHHHHHHHH  AA      AA  NN  NN  NN  GG          EEEEEEEE
CC          HH      HH  MM      MM  CC          HH      HH  AAAAAAAAAA  NN    NNNN  GG  GGGGGG  EE
CC          HH      HH  MM      MM  CC          HH      HH  AAAAAAAAAA  NN    NNNN  GG  GGGGGG  EE
CC          HH      HH  MM      MM  CC          HH      HH  AA      AA  NN      NN  GG      GG  EE              ....
CC          HH      HH  MM      MM  CC          HH      HH  AA      AA  NN      NN  GG      GG  EE              ....
  CCCCCCCC  HH      HH  MM      MM    CCCCCCCC  HH      HH  AA      AA  NN      NN    GGGGGG    EEEEEEEEEE      ....
  CCCCCCCC  HH      HH  MM      MM    CCCCCCCC  HH      HH  AA      AA  NN      NN    GGGGGG    EEEEEEEEEE      ....

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$CHMCHANGE - change mode execution'
;    2          0002  0 MODULE EDT$CHMCHANGE (                          ! Change mode execution
;    3          0003  0                 IDENT = 'V04-000'                       ! File: CHMCHANGE.BLI Edit: REM1040
;    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 contains the main routine for change mode execution.
;   37          0037  1 !
;   38          0038  1 ! ENVIRONMENT:  Runs at any access mode - AST reentrant
;   39          0039  1 !
;   40          0040  1 ! AUTHOR: Bob Kushlis, CREATION DATE: Unknown
;   41          0041  1 !
;   42          0042  1 ! MODIFIED BY:
;   43          0043  1 !
;   44          0044  1 ! 1-001 - Original.  DJS 04-Feb-1981.  This module was created by
;   45          0045  1 !       extracting the routine EDT$$CHM_EXE  from module CHANGE.BLI.
;   46          0046  1 ! 1-002 - Regularize headers and remove control C.  JBS 27-Feb-1981
;   47          0047  1 ! 1-003 - Fix module name.  JBS 02-Mar-1981
;   48          0048  1 ! 1-004 - Revise journaling.  JBS 22-Jun-1981
;   49          0049  1 ! 1-005 - Fix a journaling bug.  JBS 08-Jul-1981
;   50          0050  1 ! 1-006 - Use EDT$_ for message codes.  JBS 04-Aug-1981
;   51          0051  1 ! 1-007 - Fix a comment which stated that EDT$$GZ_COMMAND_RAB is an implicit input;
;   52          0052  1 !         actually, it is not used.  JBS 16-Aug-1981
;   53          0053  1 ! 1-008 - Return if the journal file terminates, and tell the caller.  JBS 01-Oct-1981
;   54          0054  1 ! 1-009 - Change EOB to user defined string. STS 06-Oct-1981
;   55          0055  1 ! 1-010 - Change literal prompt to use global string.  JBS 20-Oct-1981
;   56          0056  1 ! 1-011 - Remove length of prompt string.  JBS 23-Oct-1981
;   57          0057  1 ! 1-012 - Add control C handling.  JBS 21-Dec-1981
```

```
;    58          0058  1 ! 1-013 - Debug control C handling.  JBS 24-Dec-1981
;    59          0059  1 ! 1-014 - Add EDT$$G_JOU_VALID.  JBS 09-Apr-1982
;    60          0060  1 ! 1-015 - Simplify the call to EDT$$SC_RESET.  JBS 22-Apr-1982
;    61          0061  1 ! 1-016 - Add alternative control C message.  JBS 25-May-1982
;    62          0062  1 ! 1-017 - Clear EDT$$G_EXI before exiting.  JBS 02-Jun-1982
;    63          0063  1 ! 1-018 - Clear error indicator after an error in hardcopy
;    64          0064  1 !         change mode.  JBS 02-Jun-1982
;    65          0065  1 ! 1-019 - Remove reference to SET_FMTWRRUT and other message changes.
;    66          0066  1 !         SMB 29-Jun-1982
;    67          0067  1 ! 1-020 - Make FIRST_TIME_ENTERED a global.  SMB 1-Jul-1982
;    68          0068  1 ! 1-021 - Remove EDT$$G_CHM_FRST_ENTRY, use instead EDT$$G_LASTMSG.  JBS 05-Jul-1982
;    69          0069  1 ! 1-022 - Remove EDT$$G_LN_NO.  SMB 24-Sep-1982
;    70          0070  1 ! 1-023 - Change the call to screen update and add a LOAD entry point, so this module
;    71          0071  1 !         can be displaced by the screen update modules on the PDP-11.  JBS 25-Sep-1982
;    72          0072  1 ! 1-024 - Change the screen update call again, to improve overlay size.  JBS 27-Sep-1982
;    73          0073  1 ! 1-025 - Remove the call to SC_INIT, set a flag instead.  SMB 06-Oct-1982
;    74          0074  1 ! 1-026 - Convert to new journalling scheme. STS 06-Oct-1982
;    75          0075  1 ! 1-027 - Change the name of the cursor positioning routine, to obsolete a
;    76          0076  1 !         redundent module.  JBS 07-Oct-1982
;    77          0077  1 ! 1-028 - Correct a comment.  JBS 09-Oct-1982
;    78          0078  1 ! 1-029 - Set EDT$$G_SCR_REBUILD on exit, so line mode will be more efficient.  JBS 21-Oct-1982
;    79          0079  1 ! 1-030 - First time through - ignore typeahead. STS 10-Nov-1982
;    80          0080  1 ! 1-031 - First time through - don't update screen if typeahead. STS 01-Dec-1982
;    81          0081  1 ! 1-032 - Clear screen first time through. STS 02-Dec-1982
;    82          0082  1 ! 1-033 - Fix problem with hardcopy recoveries. STS 13-Dec-1982
;    83          0083  1 ! 1-034 - Only initialize the screen once.  JBS 20-Dec-1982
;    84          0084  1 ! 1-035 - Be more careful about printing the owed message.  JBS 18-Jan-1983
;    85          0085  1 ! 1-036 - Don't call EDT$$ERA_MSGLN unnecessarily.  JBS 20-Jan-1983
;    86          0086  1 ! 1-037 - Don't call EDT$$RD_ECHO unnecessarily.  JBS 21-Jan-1983
;    87          0087  1 ! 1-038 - Only update the screen for every 10 characters when doing
;    88          0088  1 !         a recovery.  JBS 21-Jan-1983
;    89          0089  1 ! 1-039 - Add new value for EDT$$G_SCR_CHGD.  JBS 02-Mar-1983
;    90          0090  1 ! 1-040 - Added logic to maintain EDT$$G_TIN_OBUFPOS durring /RECOVERY mode.
;    91          0091  1 !         REM 10-Oct-1983
;    92          0092  1 !--
;    93          0093  1
```

```
;    95          0094  1 %SBTTL 'Declarations'
;    96          0095  1 !
;    97          0096  1 ! TABLE OF CONTENTS:
;    98          0097  1 !
;    99          0098  1
;   100          0099  1 REQUIRE 'EDTSRC:TRAROUNAM';
;   101          0538  1
;   102          0539  1 FORWARD ROUTINE
;   103          0540  1     EDT$$CHM_EXE,                       ! Driver for change mode processing
;   104          0541  1     EDT$$LOAD_CHMCHANGE : NOVALUE;      ! Load this module into memory
;   105          0542  1
;   106          0543  1 !
;   107          0544  1 ! INCLUDE FILES:
;   108          0545  1 !
;   109          0546  1
;   110          0547  1 REQUIRE 'EDTSRC:EDTREQ';
;   111          0682  1
;   112          0683  1 !
;   113          0684  1 ! MACROS:
;   114          0685  1 !
;   115          0686  1 !       NONE
;   116          0687  1 !
;   117          0688  1 ! EQUATED SYMBOLS:
;   118          0689  1 !
;   119          0690  1 !       NONE
;   120          0691  1 !
;   121          0692  1 ! OWN STORAGE:
;   122          0693  1 !
;   123          0694  1 !       NONE
;   124          0695  1 !
;   125          0696  1 ! EXTERNAL REFERENCES:
;   126          0697  1 !
;   127          0698  1 !       In the routine
```

```
;  129          0699  1 %SBTTL 'EDT$$CHM_EXE  - change mode execution'
;  130          0700  1
;  131          0701  1 GLOBAL ROUTINE EDT$$CHM_EXE                     ! Change mode execution
;  132          0702  1     =
;  133          0703  1
;  134          0704  1 !++
;  135          0705  1 ! FUNCTIONAL DESCRIPTION:
;  136          0706  1 !
;  137          0707  1 !       This is the main routine for change mode execution.  First we initialize
;  138          0708  1 !       for change mode, then execute any change mode commands on the current
;  139          0709  1 !       command line.  After we have finished with them, we check to see if an
;  140          0710  1 !       exit was done and if so get out.  Otherwise we enter into the change mode
;  141          0711  1 !       comma d loop, getting commands and executing them until an exit is done,
;  142          0712  1 !       or until the journal file ends.
;  143          0713  1 !
;  144          0714  1 ! FORMAL PARAMETERS:
;  145          0715  1 !
;  146          0716  1 !       NONE
;  147          0717  1 !
;  148          0718  1 ! IMPLICIT INPUTS:
;  149          0719  1 !
;  150          0720  1 !       EDT$$T_CMD_BUF
;  151          0721  1 !       EDT$$G_CUR_COL
;  152          0722  1 !       EDT$$G_CS_CNO
;  153          0723  1 !       EDT$$G_EXI
;  154          0724  1 !       EDT$$G_CMD_LEN
;  155          0725  1 !       EDT$$A_CMD_BUF
;  156          0726  1 !       EDT$$G_INP_SRC
;  157          0727  1 !       EDT$$G_RCOV_MOD
;  158          0728  1 !       EDT$$G_TI_TYP
;  159          0729  1 !       EDT$$T_LN_BUF
;  160          0730  1 !       EDT$$A_LN_PTR
;  161          0731  1 !       EDT$$A_LN_END
;  162          0732  1 !       EDT$$T_PMT_HCCHG
;  163          0733  1 !       EDT$$G_CC_DONE
;  164          0734  1 !       EDT$$G_LASTMSG
;  165          0735  1 !       EDT$$G_TIN_ECHOFLG
;  166          0736  1 !       EDT$$G_RECSCRUPD
;  167          0737  1 !
;  168          0738  1 ! IMPLICIT OUTPUTS:
;  169          0739  1 !
;  170          0740  1 !       EDT$$A_CMD_BUF
;  171          0741  1 !       EDT$$A_CMD_END
;  172          0742  1 !       EDT$$A_CUR_BUF
;  173          0743  1 !       EDT$$G_EDIT_MOD
;  174          0744  1 !       EDT$$G_EXI
;  175          0745  1 !       EDT$$G_JOU_VALID
;  176          0746  1 !       EDT$$G_RECSCRUPD
;  177          0747  1 !       EDT$$G_SCR_CHGD
;  178          0748  1 !       EDT$$G_SCR_REBUILD
;  179          0749  1 !       EDT$$G_TIN_OBUFPOS
;  180          0750  1 !
;  181          0751  1 ! ROUTINE VALUE:
;  182          0752  1 !
;  183          0753  1 !       1 = reached the end of the journal file
;  184          0754  1 !       0 = executed an exit command
;  185          0755  1 !
```

```
;  186          0756  1 ! SIDE EFFECTS:
;  187          0757  1 !
;  188          0758  1 !       MANY
;  189          0759  1 !
;  190          0760  1 !--
;  191          0761  1
;  192          0762  2     BEGIN
;  193          0763  2
;  194          0764  2     EXTERNAL ROUTINE
;  195          0765  2         EDT$$MSG_BELL : NOVALUE,            ! Output a message to the terminal with a warning bell
;  196          0766  2         EDT$$SC_POSCSIF : NOVALUE,          ! Position the cursor
;  197          0767  2         EDT$$RD_ECHO,                       ! Try to optimize terminal input
;  198          0768  2         EDT$$INIT_CHM : NOVALUE,            ! Initialization on entering change mode
;  199          0769  2         EDT$$FMT_CH,                        ! Format a charcter
;  200          0770  2         EDT$$OUT_FMTBUF,                    ! Dump the format buffer
;  201          0771  2         EDT$$FMT_STR,                       ! Format a string
;  202          0772  2         EDT$$RD_CMDLN,                      ! Get a command line
;  203          0773  2         EDT$$TI_BUFCH : NOVALUE,            ! Put a character in the journal file buffer
;  204          0774  2         EDT$$TI_FLUSHJOUFI : NOVALUE,       ! Empty the journal file's buffer
;  205          0775  2         EDT$$RD_JOUTXT,                     ! Read a text record from the journal file
;  206          0776  2         EDT$$TI_WRLN,                       ! Write to terminal
;  207          0777  2         EDT$$TI_WRSTR,                      ! Write to terminal unformatted
;  208          0778  2         EDT$$GET_KPADCMD,                   ! Get a keypad command
;  209          0779  2         EDT$$FMT_TEXT : NOVALUE,            ! output eob string
;  210          0780  2         EDT$$SC_INIT,                       ! initialize terminal for change mode                    ;
;  211          0781  2         EDT$$SC_ERAALL,                     ! erase the screen
;  212          0782  2         EDT$$RPL_CHGDLN,                    ! Declare current line as changed
;  213          0783  2         EDT$$CHM_PAREXE,                    ! Parse and execute a change mode command string         ;
;  214          0784  2         EDT$$SC_RESET,                      ! Reset screen parameters
;  215          0785  2         EDT$$SC_UPD_NOOVERLAY1,             ! Update the screen, no overlay checking on PDP-11
;  216          0786  2         EDT$$START_WKINGMSG,                ! Set up working AST
;  217          0787  2         EDT$$STOP_WKINGMSG,                 ! Terminate working AST
;  218          0788  2         EDT$$FMT_MSG,                       ! Print message text
;  219          0789  2         EDT$$ERA_MSGLN,                     ! Erase the message line
;  220          0790  2         EDT$$TI_TSTTYAHED,                  ! Check for type ahead
;  221          0791  2         EDT$$CHR_CC,                        ! Check for control C
;  222          0792  2         EDT$$CLR_CC : NOVALUE;              ! Clear control C flag
;  223          0793  2
;  224          0794  2     EXTERNAL
;  225          0795  2         EDT$$G_EDIT_DFLTMOD,                ! Editing default mode
;  226          0796  2         EDT$$G_PUT_JOU,
;  227          0797  2         EDT$$G_MESSAGE_LINE,                ! Error message line is 1 more
;  228          0798  2         EDT$$G_MSGFLG,                      ! 1 = there is a message on the last line
;  229          0799  2         EDT$$G_LASTMSG,                     ! The last message printed
;  230          0800  2         EDT$$A_FMT_WRRUT,                   ! Holds address of write routine
;  231          0801  2         EDT$$T_CMD_BUF,                     ! Command string buffer
;  232          0802  2         EDT$$G_CUR_COL,                     ! current column
;  233          0803  2         EDT$$G_CS_LNO,                      ! cursor line.
;  234          0804  2         EDT$$A_CMD_END,                     ! End of command pointer
;  235          0805  2         EDT$$G_EXI,                         ! Change mode has been exited.
;  236          0806  2         EDT$$G_CMD_LEN,                     ! Length of command buffer
;  237          0807  2         EDT$$A_CMD_BUF,                     ! Command string pointer
;  238          0808  2         EDT$$G_INP_SRC,                     ! Source of command input.
;  239          0809  2         EDT$$G_RCOV_MOD,                    ! In recovery mode?
;  240          0810  2         EDT$$A_CUR_BUF : REF TBCB_BLOCK,    ! The current buffer tbcb
;  241          0811  2         EDT$$G_TI_TYP,                      ! Terminal type.
;  242          0812  2         EDT$$T_LN_BUF,                      ! Current line buffer
```

```
;  243          0813  2          EDT$$A_LN_PTR,                          ! Current character pointer
;  244          0814  2          EDT$$A_LN_END,                          ! End of current line pointer
;  245          0815  2          EDT$$G_EDIT_MOD,                        ! Editing mode: line or change
;  246          0816  2          EDT$$T_PMT_ACCHG : VECTOR [, BYTE],     ! Counted ASCII prompt string for hardcopy change mode
;  247          0817  2          EDT$$G_JOU_VALID,                       ! 1 = journal record is valid
;  248          0818  2          EDT$$A_WK_LN : REF LIN_BLOCK,
;  249          0819  2          EDT$$Z_EOB_LN,
;  250          0820  2          EDT$$G_CC_DONE,                         ! 1 = Control C actually aborted something
;  251          0821  2          EDT$$G_SCR_CHGD,                        ! The screen must be repainted
;  252          0822  2          EDT$$G_SCR_REBUILD,                     ! The screen must be rebuilt from the work file
;  253          0823  2          EDT$$G_TIN_ECHOFLG,
;  254          0824  2          EDT$$G_RECSCRUPD,                       ! Counter for screen updates in recovery mode
;  255          0825  2          EDT$$G_TIN_OBUFPOS;                     ! Position in journal output buffer
;  256          0826  2
;  257          0827  2      LOCAL
;  258          0828  2          FIRST_TIME,                             ! 1 = first time, at least initialize the screen
;  259          0829  2          STATUS,                                 ! 0 = error, 1 = ok, 2 = journal file ended
;  260          0830  2          CC_MSG,                                 ! 0 = no message, 1 = aborted msg, 2 = ignored msg
;  261          0831  2          OWED_MESSAGE;                           ! 1 = no message, else the code for the message to print
;  262          0832  2
;  263          0833  2  !+
;  264          0834  2  ! Specify messages used in this routine.
;  265          0835  2  !-
;  266          0836  2      MESSAGES ((CHGMODTER, ABOBYCC, CTRC__IGN));
;  267          0837  2  !+
;  268          0838  2  ! Since we entered this routine, we must be in change mode.
;  269          0839  2  !-
;  270          0840  2      ASSERT (.EDT$$G_EDIT_MOD EQL CHANGE_MODE);
;  271          0841  2  !+
;  272          0842  2  ! Perform initialization.
;  273          0843  2  !-
;  274          0844  2      EDT$$INIT_CHM ();
;  275          0845  2      FIRST_TIME = 1;
;  276          0846  2      CC_MSG = 0;
;  277          0847  2  !+
;  278          0848  2  ! make sure any previous commands have been written out to the journal file
;  279          0849  2  !-
;  280          0850  2
;  281          0851  2      IF .edt$$g_rcov_mod THEN
;  282          0852  2          edt$$g_tin_obufpos = 0
;  283          0853  2      ELSE
;  284          0854  2          edt$$ti_flushjoufi (%C'T');
;  285          0855  2
;  286          0856  2  !+
;  287          0857  2  ! Execute remainder of command line if there is a semicolon.
;  288          0858  2  !-
;  289          0859  2
;  290          0860  2      IF (CH$RCHAR (.EDT$$A_CMD_BUF) NEQ %C'!') THEN STATUS = EDT$$CHM_PAREXE (1) ELSE STATUS = 1;
;  291          0861  2
;  292          0862  2  !+
;  293          0863  2  ! If an exit command was seen, get out now.
;  294          0864  2  !-
;  295          0865  2
;  296          0866  2      IF .EDT$$G_EXI
;  297          0867  2      THEN
;  298          0868  3          BEGIN
;  299          0869  3          EDT$$G_EDIT_MOD = LINE_MODE;
```

```
;  300     0870  3          EDT$$RPL_CHGDLN ();
;  301     0871  3          EDT$$G_EXI = 0;
;  302     0872  3          RETURN (0);
;  303     0873  2          END;
;  304     0874  2
;  305     0875  2      !+
;  306     0876  2      ! Before entering the command loop, make sure the input is either coming
;  307     0877  2      ! from the terminal or from a recovery file.  We do not allow the startup
;  308     0878  2      ! file or macros to enter change mode.
;  309     0879  2      !-
;  310     0880  2
;  311     0881  3          IF (((.EDT$$G_INP_SRC NEQ INP_TERM) AND ( NOT .EDT$$G_RCOV_MOD)) OR          !
;  312     0882  3              (.EDT$$G_TI_TYP EQL TERM_UNKNOWN))
;  313     0883  2          THEN
;  314     0884  3              BEGIN
;  315     0885  3              EDT$$G_EDIT_MOD = LINE_MODE;
;  316     0886  3              EDT$$FMT_MSG (EDT$_CHGMODTER);
;  317     0887  3              RETURN (0);
;  318     0888  2              END;
;  319     0889  2
;  320     0890  2      !+
;  321     0891  2      ! Set up for change mode editing.
;  322     0892  2      !-
;  323     0893  2          OWED_MESSAGE = 1;
;  324     0894  2
;  325     0895  3          IF ((.EDT$$G_TI_TYP EQL TERM_VT52) OR (.EDT$$G_TI_TYP EQL TERM_VT100))
;  326     0896  2          THEN
;  327     0897  3              BEGIN
;  328     0898  3              EDT$$A_FMT_WRRUT = EDT$$TI_WRSTR;
;  329     0899  3      !+
;  330     0900  3      ! Since we are about to refresh the screen, remember if there is a message,
;  331     0901  3      ! so we can display it.  This lets us display, for example, "Input file does not
;  332     0902  3      ! have standard text format" even if the first thing we do is enter change mode.
;  333     0903  3      !-
;  334     0904  3              OWED_MESSAGE = .EDT$$G_LASTMSG;
;  335     0905  3              EDT$$G_SCR_CHGD = 2;                    ! Initialize terminal and repaint screen
;  336     0906  2              END;
;  337     0907  2
;  338     0908  2      !+
;  339     0909  2      ! Now loop through, getting commands until an exit is seen or the journal file ends.
;  340     0910  2      !-
;  341     0911  2
;  342     0912  2          WHILE ((.EDT$$G_EXI EQL 0) AND (.STATUS NEQ 2)) DO
;  343     0913  3              BEGIN
;  344     0914  3
;  345     0915  3              IF .edt$$g_put_jou
;  346     0916  3              THEN
;  347     0917  4                  BEGIN
;  348     0918  4
;  349     0919  4                  IF .edt$$g_rcov_mod THEN
;  350     0920  4                      edt$$g_tin_obufpos = 0
;  351     0921  4                  ELSE
;  352     0922  4                      edt$$ti_flushjoufi (%C'T');
;  353     0923  4
;  354     0924  4                  edt$$clr_cc ();
;  355     0925  4                  edt$$g_put_jou = 0
;  356     0926  3                  END;
```

```
;  357          0927  3
;  358          0928  3 !+
;  359          0929  3 ! Check for a hardcopy terminal.
;  360          0930  3 !-
;  361          0931  3
;  362          0932  4         IF (.EDT$$G_TI_TYP EQL TERM_HCPY)
;  363          0933  3         THEN
;  364          0934  4             BEGIN
;  365          0935  4 !+
;  366          0936  4 ! Hard copy change mode.  Output a control C message if we owe one.
;  367          0937  4 !-
;  368          0938  4
;  369          0939  4             CASE .CC_MSG FROM 0 TO 2 OF
;  370          0940  4                 SET
;  371          0941  4
;  372          0942  4                 [0] :                           ! We don't owe a message
;  373          0943  5                     BEGIN
;  374          0944  5                     0
;  375          0945  4                     END;
;  376          0946  4
;  377          0947  4                 [1] :                           ! The previous operation was aborted by a control C
;  378          0948  5                     BEGIN
;  379          0949  5                     EDT$$MSG_BELL (EDT$_ABOBYCC);
;  380          0950  5                     CC_MSG = 0;
;  381          0951  5                     END;
;  382          0952  4
;  383          0953  4                 [2] :                           ! The previous control C was ignored
;  384          0954  5                     BEGIN
;  385          0955  5                     EDT$$MSG_BELL (EDT$_CTRC__IGN);
;  386          0956  5                     CC_MSG = 0;
;  387          0957  4                     END;
;  388          0958  4                 TES;
;  389          0959  4
;  390          0960  4 !+
;  391          0961  4 ! Type the current line with the cursor bracketed.
;  392          0962  4 !-
;  393          0963  4
;  394          0964  5             IF (.EDT$$A_WK_LN EQLA EDT$$Z_EOB_LN)
;  395          0965  4             THEN
;  396          0966  4                 EDT$$FMT_TEXT (0)
;  397          0967  4             ELSE
;  398          0968  5                 BEGIN
;  399          0969  5                 EDT$$FMT_STR (EDT$$T_LN_BUF, CH$DIFF (.EDT$$A_LN_PTR, CH$PTR (EDT$$T_LN_BUF)));
;  400          0970  5                 EDT$$FMT_CH ('[');
;  401          0971  5                 EDT$$FMT_CH (CH$RCHAR (.EDT$$A_LN_PTR));
;  402          0972  5                 EDT$$FMT_CH (']');
;  403          0973  5
;  404          0974  6                 IF CH$PTR_NEQ (.EDT$$A_LN_PTR, .EDT$$A_LN_END)
;  405          0975  5                 THEN
;  406          0976  5                     EDT$$FMT_STR (CH$PLUS (.EDT$$A_LN_PTR, 1), CH$DIFF (.EDT$$A_LN_END, .EDT$$A_LN_PTR) - 1)
;  407          0977  5
;  408          0978  4                 END;
;  409          0979  4
;  410          0980  4             EDT$$OUT_FMTBUF ();
;  411          0981  4 !+
;  412          0982  4 ! Now get the next command string.
;  413          0983  4 !-
```

```
;  414          0984  4              STATUS = 1;
;  415          0985  4
;  416          0986  4              IF .EDT$$G_RCOV_MOD
;  417          0987  4              THEN
;  418          0988  5                  BEGIN
;  419          0989  5
;  420          0990  6                  IF ( NOT EDT$$RD_JOUTXT (EDT$$T_CMD_BUF, EDT$$G_CMD_LEN))
;  421          0991  5                  THEN
;  422          0992  5                      STATUS = 2                  ! Journal file ended
;  423          0993  5                  ELSE
;  424          0994  6                      BEGIN
;  425          0995  6
;  426          0996  7                      IF (.EDT$$G_CMD_LEN EQL 2)
;  427          0997  6                      THEN
;  428          0998  6
;  429          0999  7                          IF (CH$RCHAR (CH$PTR (EDT$$T_CMD_BUF)) EQL %C'^')
;  430          1000  6                          THEN
;  431          1001  6
;  432          1002  7                              IF ((CH$RCHAR (CH$PTR (EDT$$T_CMD_BUF, 1)) EQL %C'Z') OR    !
;  433          1003  7                                  (CH$RCHAR (CH$PTR (EDT$$T_CMD_BUF, 1)) EQL %C'z'))
;  434          1004  6                              THEN
;  435          1005  6                                  EDT$$G_EXI = 1;
;  436          1006  6
;  437          1007  6                      EDT$$A_CMD_END = CH$PTR (EDT$$T_CMD_BUF, .EDT$$G_CMD_LEN);
;  438          1008  6                      EDT$$G_TIN_OBUFPOS = .EDT$$G_TIN_OBUFPOS + .EDT$$G_CMD_LEN
;  439          1009  6                      END
;  440          1010  6
;  441          1011  5                  END
;  442          1012  4              ELSE
;  443          1013  5                  BEGIN
;  444          1014  5          !+
;  445          1015  5          ! We are not recovering.
;  446          1016  5          !-
;  447          1017  5          !+
;  448          1018  5          ! Since we are about to read from the terminal, make sure the last
;  449          1019  5          ! line has been written to the journal file.
;  450          1020  5          !-
;  451          1021  5                  EDT$$G_EXI = EDT$$RD_CMDLN (EDT$$T_PMT_HCCHG [1], .EDT$$T_PMT_HCCHG [0], EDT$$T_CMD_BUF,
;  452          1022  5                      EDT$$G_CMD_LEN, 255);
;  453          1023  5                  EDT$$A_CMD_END = CH$PTR (EDT$$T_CMD_BUF, .EDT$$G_CMD_LEN);
;  454          1024  5          !+
;  455          1025  5          ! Put the new line in the journal file buffer.
;  456          1026  5          !-
;  457          1027  5
;  458          1028  5                  IF .EDT$$G_EXI
;  459          1029  5                  THEN
;  460          1030  6                      BEGIN
;  461          1031  6                      EDT$$TI_BUFCH ('^');
;  462          1032  6                      EDT$$TI_BUFCH ('Z');
;  463          1033  6                      END
;  464          1034  5                  ELSE
;  465          1035  6                      BEGIN
;  466          1036  6
;  467          1037  6                      INCR COUNTER FROM 0 TO .EDT$$G_CMD_LEN - 1 DO
;  468          1038  6                          EDT$$TI_BUFCH (CH$RCHAR (CH$PTR (EDT$$T_CMD_BUF, .COUNTER)));
;  469          1039  6
;  470          1040  5                      END;
```

```
;   471          1041  5
;   472          1042  5                    EDT$$G_JOU_VALID = 1;
;   473          1043  4                    END;
;   474          1044  4
;   475          1045  4                END
;   476          1046  3            ELSE
;   477          1047  4                BEGIN
;   478          1048  4 !+
;   479          1049  4 ! This is not a hard copy terminal.
;   480          1050  4 ! Erase the message line, unless it is scheduled to be erased by the next keystroke.
;   481          1051  4 !-
;   482          1052  4
;   483          1053  4                IF (( NOT .EDT$$G_MSGFLG) AND .EDT$$G_TIN_ECHOFLG) THEN EDT$$ERA_MSGLN ();
;   484          1054  4
;   485          1055  4 !+
;   486          1056  4 ! Check for characters in type ahead.  Do not update if there
;   487          1057  4 ! are more characters to handle.
;   488          1058  4 !-
;   489          1059  4
;   490          1060  5                IF ( NOT EDT$$TI_TSTTYAHED ())
;   491          1061  4                THEN
;   492          1062  5                    BEGIN
;   493          1063  5                    FIRST_TIME = 0;
;   494          1064  5 !+
;   495          1065  5 ! Update the screen.  This call may bring in the screen update overlay;
;   496          1066  5 ! this module will be loaded back into memory before the return.
;   497          1067  5 ! In recovery mode we update the screen only 1/10 as often, to make
;   498          1068  5 ! recovery go faster, particularly when an overlay is needed.
;   499          1069  5 !-
;   500          1070  5
;   501          1071  5                    IF .EDT$$G_RCOV_MOD
;   502          1072  5                    THEN
;   503          1073  6                        BEGIN
;   504          1074  6                        EDT$$G_RECSCRUPD = .EDT$$G_RECSCRUPD - 1;
;   505          1075  6
;   506          1076  7                        IF (.EDT$$G_RECSCRUPD LEQ 0)
;   507          1077  6                        THEN
;   508          1078  7                            BEGIN
;   509          1079  7                            EDT$$G_RECSCRUPD = 10;
;   510          1080  7                            EDT$$SC_UPD_NOOVERLAY1 ();
;   511          1081  7                            END
;   512          1082  7
;   513          1083  6                        END
;   514          1084  5                    ELSE
;   515          1085  5                        EDT$$SC_UPD_NOOVERLAY1 ();
;   516          1086  5
;   517          1087  5 !+
;   518          1088  5 ! Re-display the last message we saw before entering change mode.
;   519          1089  5 !-
;   520          1090  5
;   521          1091  6                    IF (.OWED_MESSAGE NEQ 1)        ! If there is a message
;   522          1092  5                    THEN
;   523          1093  6                        BEGIN
;   524          1094  6                        EDT$$SC_POSCSIF (.EDT$$G_MESSAGE_LINE + 1, 0);
;   525          1095  6                        EDT$$MSG_BELL (.OWED_MESSAGE);
;   526          1096  6                        EDT$$G_MSGFLG = 1;          ! Erase it on next keystroke
;   527          1097  6                        OWED_MESSAGE = 1;           ! We no longer owe the message
```

```
;   528          1098  6                    END
;   529          1099  5                ELSE
;   530          1100  6                    BEGIN
;   531          1101  6 !+
;   532          1102  6 ! Output a control C message if we owe one.
;   533          1103  6 !-
;   534          1104  6
;   535          1105  6                    CASE .CC_MSG FROM 0 TO 2 OF
;   536          1106  6                        SET
;   537          1107  6
;   538          1108  6                        [0] :                   ! We don't owe a message
;   539          1109  7                            BEGIN
;   540          1110  7                            0
;   541          1111  6                            END;
;   542          1112  6
;   543          1113  6                        [1] :                   ! The previous operation was aborted by a control C
;   544          1114  7                            BEGIN
;   545          1115  7                            EDT$$MSG_BELL (EDT$_ABOBYCC);
;   546          1116  7                            CC_MSG = 0;
;   547          1117  6                            END;
;   548          1118  6
;   549          1119  6                        [2] :                   ! The previous control C was ignored
;   550          1120  7                            BEGIN
;   551          1121  7                            EDT$$MSG_BELL (EDT$_CTRC__IGN);
;   552          1122  7                            CC_MSG = 0;
;   553          1123  6                            END;
;   554          1124  6                        TES;
;   555          1125  5
;   556          1126  5                    END;
;   557          1127  5
;   558          1128  5 !+
;   559          1129  5 ! Check for the optimized input applying.
;   560          1130  5 !-
;   561          1131  5
;   562          1132  6                    IF ( NOT .EDT$$G_RCOV_MOD)
;   563          1133  5                    THEN
;   564          1134  5
;   565          1135  5                        IF ( NOT EDT$$RD_ECHO ()) THEN EDT$$SC_POSCSIF (.EDT$$G_CS_LNO, .EDT$$G_CUR_COL);
;   566          1136  5
;   567          1137  5                    END
;   568          1138  4                ELSE
;   569          1139  4
;   570          1140  4                    IF .FIRST_TIME
;   571          1141  4                    THEN
;   572          1142  5                        BEGIN
;   573          1143  5                        FIRST_TIME = 0;
;   574          1144  5                        EDT$$SC_INIT ();
;   575          1145  5                        EDT$$SC_ERAALL ();
;   576          1146  4                        END;
;   577          1147  4
;   578          1148  4 !+
;   579          1149  4 ! Get the next command string.
;   580          1150  4 !-
;   581          1151  4                EDT$$A_CMD_BUF = CH$PTR (EDT$$T_CMD_BUF);
;   582          1152  4                EDT$$A_CMD_END = CH$PTR (EDT$$T_CMD_BUF, 256);
;   583          1153  4
;   584          1154  4                IF EDT$$CHK_CC () THEN STATUS = 1 ELSE STATUS = EDT$$GET_KPADCMD ();
```

```
;   585          1155  4
;   586          1156  3              END;
;   587          1157  3
;   588          1158  4          IF ((.STATUS EQL 1) AND ( NOT EDT$$CHK_CC ()))
;   589          1159  3          THEN
;   590          1160  4              BEGIN
;   591          1161  4      !+
;   592          1162  4      ! Start up the 'working' AST.
;   593          1163  4      !-
;   594          1164  4
;   595          1165  4              IF (.EDT$$G_TI_TYP NEQ TERM_HCPY) THEN EDT$$START_WKINGMSG ();
;   596          1166  4
;   597          1167  4      !+
;   598          1168  4      ! Execute the command string in EDT$$T_CMD_BUF .
;   599          1169  4      !-
;   600          1170  4              EDT$$A_CMD_BUF = CH$PTR (EDT$$T_CMD_BUF);
;   601          1171  4              STATUS = EDT$$CHM_PAREXE (1);
;   602          1172  4      !+
;   603          1173  4      ! Turn off the 'working' AST.
;   604          1174  4      !-
;   605          1175  4
;   606          1176  4              IF (.EDT$$G_TI_TYP NEQ TERM_HCPY) THEN EDT$$STOP_WKINGMSG ();
;   607          1177  4
;   608          1178  3              END;
;   609          1179  3
;   610          1180  3      !+
;   611          1181  3      ! If the control C flag is set, the command was probably aborted by a control C.
;   612          1182  3      ! Invalidate the screen (since control C can sometimes cause a scroll) and arrange
;   613          1183  3      ! to show an appropriate message.
;   614          1184  3      !-
;   615          1185  3
;   616          1186  3          IF EDT$$CHK_CC ()
;   617          1187  3          THEN
;   618          1188  4              BEGIN
;   619          1189  4
;   620          1190  4              IF (EDT$$G_TI_TYP NEQ TERM_HCPY) THEN EDT$$G_SCR_CHGD = 1;
;   621          1191  4
;   622          1192  4              IF .EDT$$G_CC_DONE THEN CC_MSG = 1 ELSE CC_MSG = 2;
;   623          1193  4
;   624          1194  4              END
;   625          1195  3          ELSE
;   626          1196  3              CC_MSG = 0;
;   627          1197  3
;   628          1198  2          END;
;   629          1199  2
;   630          1200  2      EDT$$RPL_CHGDLN ();
;   631          1201  2      !+
;   632          1202  2      ! Clean up after ourselves.  If we will be back here because this is
;   633          1203  2      ! just the journal file ending we will put everything back for change
;   634          1204  2      ! mode again.
;   635          1205  2      !
;   636          1206  2      ! Reset the terminal as required.
;   637          1207  2      !-
;   638          1208  2      EDT$$SC_RESET ();
;   639          1209  2      !+
;   640          1210  2      ! Flag that the screen must be rebuilt from the work file.  This makes line
;   641          1211  2      ! mode more efficient, since it does not need to maintain the screen data base.
```

```
;  642          1212  2 !-
;  643          1213  2     EDT$$G_SCR_REBUILD = 1;
;  644          1214  2 !+
;  645          1215  2 ! Reset the formatted write routine for line mode.
;  646          1216  2 !-
;  647          1217  2     EDT$$A_FMT_WRRUT = EDT$$TI_WRLN;
;  648          1218  2     EDT$$A_CUR_BUF [TBCB_CHAR_POS] = CH$DIFF (.EDT$$A_LN_PTR, CH$PTR (EDT$$T_LN_BUF));
;  649          1219  2     EDT$$A_CMD_BUF = CH$PTR (EDT$$T_CMD_BUF);
;  650          1220  2     CH$WCHAR (%C'!', .EDT$$A_CMD_BUF);
;  651          1221  2
;  652          1222  3     IF (.EDT$$G_EXI NEQ 0)
;  653          1223  2     THEN
;  654          1224  3         BEGIN
;  655          1225  3         EDT$$G_EDIT_MOD = LINE_MODE;
;  656          1226  3         EDT$$G_EXI = 0;
;  657          1227  3         END;
;  658          1228  2
;  659          1229  2     IF (.STATUS EQL 2) THEN RETURN (1) ELSE RETURN (0);
;  660          1230  2
;  661          1231  1     END;                                    ! of routine EDT$$CHM_EXE


;                                                                   .TITLE  EDT$CHMCHANGE EDT$CHMCHANGE - change mode execu
                                                                                        tion
                                                                    .IDENT  \V04-000\

                                                                    .EXTRN  EDT$$MSG_BELL, EDT$$SC_POSCSIF
                                                                    .EXTRN  EDT$$RD_ECHO, EDT$$INIT_CHM
                                                                    .EXTRN  EDT$$FMT_CH, EDT$$OUT_FMTBUF
                                                                    .EXTRN  EDT$$FMT_STR, EDT$$RD_CMDLN
                                                                    .EXTRN  EDT$$TI_BUFCH, EDT$$TI_FLUSHJOUFI
                                                                    .EXTRN  EDT$$RD_JOUTXT, EDT$$TI_WRLN
                                                                    .EXTRN  EDT$$TI_WRSTR, EDT$$GET_KPADCMD
                                                                    .EXTRN  EDT$$FMT_TEXT, EDT$$SC_INIT
                                                                    .EXTRN  EDT$$SC_ERAALL, EDT$$RPL_CHGDLN
                                                                    .EXTRN  EDT$$CHM_PAREXE
                                                                    .EXTRN  EDT$$SC_RESET, EDT$$SC_UPD_NOOVERLAY1
                                                                    .EXTRN  EDT$$START_WKINGMSG
                                                                    .EXTRN  EDT$$STOP_WKINGMSG
                                                                    .EXTRN  EDT$$FMT_MSG, EDT$$ERA_MSGLN
                                                                    .EXTRN  EDT$$TI_TSTTYAHED
                                                                    .EXTRN  EDT$$CHR_CC, EDT$$CLR_CC
                                                                    .EXTRN  EDT$$G_EDIT_DFLTMOD
                                                                    .EXTRN  EDT$$G_PUT_JOU, EDT$$G_MESSAGE_LINE
                                                                    .EXTRN  EDT$$G_MSGFLG, EDT$$G_LASTMSG
                                                                    .EXTRN  EDT$$A_FMT_WRRUT
                                                                    .EXTRN  EDT$$T_CMD_BUF, EDT$$G_CUR_COL
                                                                    .EXTRN  EDT$$G_CS_LNO, EDT$$A_CMD_END
                                                                    .EXTRN  EDT$$G_EXI, EDT$$G_CMD_LEN
                                                                    .EXTRN  EDT$$A_CMD_BUF, EDT$$G_INP_SRC
                                                                    .EXTRN  EDT$$G_RCOV_MOD
                                                                    .EXTRN  EDT$$A_CUR_BUF, EDT$$G_TI_TYP
                                                                    .EXTRN  EDT$$T_LN_BUF, EDT$$A_LN_PTR
                                                                    .EXTRN  EDT$$A_LN_END, EDT$$G_EDIT_MOD
                                                                    .EXTRN  EDT$$T_PMT_HCCHG
                                                                    .EXTRN  EDT$$G_JOU_VALID
                                                                    .EXTRN  EDT$$A_WK_LN, EDT$$Z_EOB_LN
```

```
                                                        .EXTRN  EDT$$G_CC_DONE, EDT$$G_SCR_CHGD
                                                        .EXTRN  EDT$$G_SCR_REBUILD
                                                        .EXTRN  EDT$$G_TIN_ECHOFLG
                                                        .EXTRN  EDT$$G_RECSCRUPD
                                                        .EXTRN  EDT$$G_TIN_OBUFPOS
                                                        .EXTRN  EDT$_CHGMODTER, EDT$_ABOBYCC
                                                        .EXTRN  EDT$_CTRC__IGN, EDT$$INTER_ERR

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

                                        0FFC 00000      .ENTRY  EDT$$CHM_EXE, Save R2,R3,R4,R5,R6,R7,R8,R9,-  ; 0701
                                                                R10,R11                                       :
                    5B 00000000G   00    9E 00002      MOVAB   EDT$$G_TI_TYP, R11                            :
                    5A 00000000G   00    9E 00009      MOVAB   EDT$$G_RCOV_MOD, R10                          :
                    59 00000000G   00    9E 00010      MOVAB   EDT$$G_EXI, R9                                :
                    58 00000000G   00    9E 00017      MOVAB   EDT$$T_CMD_BUF, R8                            :
                       00000000G   00    D5 0001E      TSTL    EDT$$G_EDIT_MOD                               ; 0840
                                   07    13 00024      BEQL    1$                                            :
00000000G   00                     00    FB 00026      CALLS   #0, EDT$$INTER_ERR                            :
00000000G   00                     00    FB 0002D 1$:  CALLS   #0, EDT$$INIT_CHM                             ; 0844
                    57             01    D0 00034      MOVL    #1, FIRST_TIME                                ; 0845
                                   54    D4 00037      CLRL    CC_MSG                                        ; 0846
                    08             6A    E9 00039      BLBC    EDT$$G_RCOV_MOD, 2$                           ; 0851
                       00000000G   00    D4 0003C      CLRL    EDT$$G_TIN_OBUFPOS                            ; 0852
                                   0B    11 00042      BRB     3$                                            :
                    7E         54  8F    9A 00044 2$:  MOVZBL  #84, -(SP)                                    ; 0854
00000000G   00                     01    FB 00048      CALLS   #1, EDT$$TI_FLUSHJOUFI                        :
                    50 00000000G   00    D0 0004F 3$:  MOVL    EDT$$A_CMD_BUF, R0                            ; 0860
                    21             60    91 00056      CMPB    (R0), #33                                     :
                                   0E    13 00059      BEQL    4$                                            :
                                   01    DD 0005B      PUSHL   #1                                            :
00000000G   00                     01    FB 0005D      CALLS   #1, EDT$$CHM_PAREXE                           :
                    55             50    D0 00064      MOVL    R0, STATUS                                    :
                                   03    11 00067      BRB     5$                                            :
                    55             01    D0 00069 4$:  MOVL    #1, STATUS                                    :
                    12             69    E9 0006C 5$:  BLBC    EDT$$G_EXI, 6$                                ; 0866
00000000G   00                     01    D0 0006F      MOVL    #1, EDT$$G_EDIT_MOD                           ; 0869
00000000G   00                     00    FB 00076      CALLS   #0, EDT$$RPL_CHGDLN                           ; 0870
                                   69    D4 0007D      CLRL    EDT$$G_EXI                                    ; 0871
                                   23    11 0007F      BRB     9$                                            ; 0872
                       00000000G   00    D5 00081 6$:  TSTL    EDT$$G_INP_SRC                                ; 0881
                                   03    13 00087      BEQL    7$                                            :
                    04             6A    E9 00089      BLBC    EDT$$G_RCOV_MOD, 8$                           :
                                   6B    D5 0008C 7$:  TSTL    EDT$$G_TI_TYP                                 ; 0882
                                   17    12 0008E      BNEQ    10$                                           :
00000000G   00                     01    D0 00090 8$:  MOVL    #1, EDT$$G_EDIT_MOD                           ; 0885
                       00000000G   8F    DD 00097      PUSHL   #EDT$_CHGMODTER                               ; 0886
00000000G   00                     01    FB 0009D      CALLS   #1, EDT$$FMT_MSG                              :
                              037F 31    000A4 9$:     BRW     59$                                           ; 0887
                    56             01    D0 000A7 10$: MOVL    #1, OWED_MESSAGE                              ; 0893
                    50             6B    D0 000AA      MOVL    EDT$$G_TI_TYP, R0                             ; 0895
                    01             50    D1 000AD      CMPL    R0, #1                                        :
                                   05    13 000B0      BEQL    11$                                           :
                    02             50    D1 000B2      CMPL    R0, #2                                        :
                                   19    12 000B5      BNEQ    12$                                           :
00000000G   00 00000000G   00    9E 000B7 11$:  MOVAB   EDT$$TI_WRSTR, EDT$$A_FMT_WRRUT                      ; 0898
                    56 00000000G   00    D0 000C2      MOVL    EDT$$G_LASTMSG, OWED_MESSAGE                  ; 0904
```

```
                    00000000G  00            02  D0 000C9          MOVL    #2, EDT$$G_SCR_CHGD                     ; 0905
                                              69  D5 000D0 12$:     TSTL    EDT$$G_EXI                              ; 0912
                                              03  13 000D2          BEQL    14$                                     ;
                                           02F1   31 000D4 13$:     BRW     57$                                     ;
                                      02        55  D1 000D7 14$:     CMPL    STATUS, #2                              ;
                                                F8  13 000DA          BEQL    13$                                     ;
                                      23 00000000G  00  E9 000DC          BLBC    EDT$$G_PUT_JOU, 17$                     ; 0915
                                      08            6A  E9 000E3          BLBC    EDT$$G_RCOV_MOD, 15$                    ; 0919
                                         00000000G  00  D4 000E6          CLRL    EDT$$G_TIN_OBUFPOS                      ; 0920
                                                   0B  11 000EC          BRB     16$                                     ;
                                   7E        54     8F  9A 000EE 15$:     MOVZBL  #84, -(SP)                              ; 0922
                    00000000G  00                  01  FB 000F2          CALLS   #1, EDT$$TI_FLUSHJOUFI                  ;
                    00000000G  00                  00  FB 000F9 16$:     CALLS   #0, EDT$$CLR_CC                         ; 0924
                                         00000000G  00  D4 00100          CLRL    EDT$$G_PUT_JOU                          ; 0925
                                      03            6B  D1 00106 17$:     CMPL    EDT$$G_TI_TYP, #3                       ; 0932
                                                   03  13 00109          BEQL    18$                                     ;
                                                 0169  31 0010B          BRW     36$                                     ;
                          02            00         54  CF 0010E 18$:     CASEL   CC_MSG, #0, #2                          ; 0939
                        0010          0008       001F      00112 19$:     .WORD   23$-19$,-                               ;
                                                                                 20$-19$,-                               ;
                                                                                 21$-19$                                 ;
                                                   17  11 00118          BRB     23$                                     ; 0943
                                         00000000G  8F  DD 0011A 20$:     PUSHL   #EDT$_ABOBYCC                           ; 0949
                                                   06  11 00120          BRB     22$                                     ;
                                         00000000G  8F  DD 00122 21$:     PUSHL   #EDT$_CTRC_IGN                          ; 0955
                    00000000G  00                  01  FB 00128 22$:     CALLS   #1, EDT$$MSG_BELL                       ;
                                                   54  D4 0012F          CLRL    CC_MSG                                  ; 0956
                                      50 00000000G  00  9E 00131 23$:     MOVAB   EDT$$Z_EOB_LN, R0                       ; 0964
                                      50 00000000G  00  D1 00138          CMPL    EDT$$A_WK_LN, R0                        ;
                                                   0B  12 0013F          BNEQ    24$                                     ;
                                                   7E  D4 00141          CLRL    -(SP)                                   ; 0966
                    00000000G  00                  01  FB 00143          CALLS   #1, EDT$$FMT_TEXT                       ;
                                                   66  11 0014A          BRB     25$                                     ;
                                      50 00000000G  00  9E 0014C 24$:     MOVAB   EDT$$T_LN_BUF, R0                       ; 0969
                 7E 00000000G  00                  50  C3 00153          SUBL3   R0, EDT$$A_LN_PTR, -(SP)                ;
                                         00000000G  00  9F 0015B          PUSHAB  EDT$$T_LN_BUF                           ;
                    00000000G  00                  02  FB 00161          CALLS   #2, EDT$$FMT_STR                        ;
                                   7E        5B     8F  9A 00168          MOVZBL  #91, -(SP)                              ; 0970
                    00000000G  00                  01  FB 0016C          CALLS   #1, EDT$$FMT_CH                         ;
                                      50 00000000G  00  D0 00173          MOVL    EDT$$A_LN_PTR, R0                       ; 0971
                                                   7E  60 9A 0017A          MOVZBL  (R0), -(SP)                             ;
                    00000000G  00                  01  FB 0017D          CALLS   #1, EDT$$FMT_CH                         ;
                                   7E        5D     8F  9A 00184          MOVZBL  #93, -(SP)                              ; 0972
                    00000000G  00                  01  FB 00188          CALLS   #1, EDT$$FMT_CH                         ;
                                      50 00000000G  00  D0 0018F          MOVL    EDT$$A_LN_PTR, R0                       ; 0974
                                      51 00000000G  00  D0 00196          MOVL    EDT$$A_LN_END, R1                       ;
                                      51            50  D1 0019D          CMPL    R0, R1                                  ;
                                                   10  13 001A0          BEQL    25$                                     ;
                                      51           50  C2 001A2          SUBL2   R0, R1                                  ; 0976
                                               FF  A1  9F 001A5          PUSHAB  -1(R1)                                  ;
                                               01  A0  9F 001A8          PUSHAB  1(R0)                                   ;
                    00000000G  00                  02  FB 001AB          CALLS   #2, EDT$$FMT_STR                        ;
                    00000000G  00                  00  FB 001B2 25$:     CALLS   #0, EDT$$OUT_FMTBUF                     ; 0980
                                                   55  01  D0 001B9          MOVL    #1, STATUS                          ; 0984
                                                   4D  6A  E9 001BC          BLBC    EDT$$G_RCOV_MOD, 30$                ; 0986
                                         00000000G  00  9F 001BF          PUSHAB  EDT$$G_CMD_CEN                          ; 0990
                                                   58  DD 001C5          PUSHL   R8
```

```
                    00000000G  00              02  FB 001C7          CALLS   #2, EDT$$RD_JOUTXT                        ;
                               05              50  E8 001CE          BLBS    R0, 26$                                  ;
                               55              02  D0 001D1          MOVL    #2, STATUS                               ; 0992
                                               34  11 001D4          BRB     29$                                      ;
                               50 00000000G    00  D0 001D6 26$:     MOVL    EDT$$G_CMD_LEN, R0                       ; 0996
                               02              50  D1 001DD          CMPL    R0, #2                                   ;
                                               19  12 001E0          BNEQ    28$                                      ;
                          5E   8F              68  91 001E2          CMPB    EDT$$T_CMD_BUF, #94                      ; 0999
                                               13  12 001E6          BNEQ    28$                                      ;
                               51      01      AB  9A 001E8          MOVZBL  EDT$$T_CMD_BUF+1, R1                     ; 1002
                          5A   8F              51  91 001EC          CMPB    R1, #90                                  ;
                                               06  13 001F0          BEQL    27$                                      ;
                          7A   8F              51  91 001F2          CMPB    R1, #122                                 ; 1003
                                               03  12 001F6          BNEQ    28$                                      ;
                               69              01  D0 001F8 27$:     MOVL    #1, EDT$$G_EXI                           ; 1005
          00000000G  00        50              58  C1 001FB 28$:     ADDL3   R8, R0, EDT$$A_CMD_END                   ; 1007
                     00000000G 00              50  C0 00203          ADDL2   R0, EDT$$G_TIN_OBUFPOS                   ; 1008
                                               68  11 0020A 29$:     BRB     35$                                      ; 0988
                               7E      FF      8F  9A 0020C 30$:     MOVZBL  #255, -(SP)                              ; 1021
                                  00000000G    00  9F 00210          PUSHAB  EDT$$G_CMD_LEN                           ;
                                               58  DD 00216          PUSHL   R8                                       ;
                               7E 00000000G    00  9A 00218          MOVZBL  EDT$$T_PMT_HCCHG, -(SP)                  ;
                                  00000000G    00  9F 0021F          PUSHAB  EDT$$T_PMT_HCCHG+1                       ;
                    00000000G  00              05  FB 00225          CALLS   #5, EDT$$RD_CMDLN                        ;
                               69              50  D0 0022C          MOVL    R0, EDT$$G_EXI                           ;
                               53 00000000G    00  D0 0022F          MOVL    EDT$$G_CMD_LEN, R3                       ; 1023
          00000000G  00        53              58  C1 00236          ADDL3   R8, R3, EDT$$A_CMD_END                   ;
                               18              69  E9 0023E          BLBC    EDT$$G_EXI, 31$                          ; 1028
                               7E      5E      8F  9A 00241          MOVZBL  #94, -(SP)                               ; 1031
                    00000000G  00              01  FB 00245          CALLS   #1, EDT$$TI_BUFCH                        ;
                               7E      5A      8F  9A 0024C          MOVZBL  #90, -(SP)                               ; 1032
                    00000000G  00              01  FB 00250          CALLS   #1, EDT$$TI_BUFCH                        ;
                                               14  11 00257          BRB     34$                                      ; 1028
                               52              01  CE 00259 31$:     MNEGL   #1, COUNTER                              ; 1037
                                               0B  11 0025C          BRB     33$                                      ;
                               7E            6842  9A 0025E 32$:     MOVZBL  EDT$$T_CMD_BUF[COUNTER], -(SP)           ; 1038
                    00000000G  00              01  FB 00262          CALLS   #1, EDT$$TI_BUFCH                        ;
               F1   52                         53  F2 00269 33$:     AOBLSS  R3, COUNTER, 32$                         ;
                    00000000G  00              01  D0 0026D 34$:     MOVL    #1, EDT$$G_JOU_VALID                     ; 1042
                                            00E8  31 00274 35$:     BRW     50$                                      ; 0932
                               0E 00000000G 00    E8 00277 36$:     BLBS    EDT$$G_MSGFLG, 37$                       ; 1053
                               07 00000000G 00    E9 0027E          BLBC    EDT$$G_TIN_ECHOFLG, 37$                  ;
                    00000000G  00              00  FB 00285          CALLS   #0, EDT$$ERA_MSGLN                       ;
                    00000000G  00              00  FB 0028C 37$:     CALLS   #0, EDT$$TI_TSTTYAHED                    ; 1060
                               03              50  E9 00293          BLBC    R0, 38$                                  ;
                                            008A  31 00296          BRW     47$                                      ;
                                               57  D4 00299 38$:     CLRL    FIRST_TIME                               ; 1063
                               0E              6A  E9 0029B          BLBC    EDT$$G_RCOV_MOD, 39$                     ; 1071
                               0E 00000000G    00  F5 0029E          SOBGTR  EDT$$G_RECSCRUPD, 40$                    ; 1074
                    00000000G  00              0A  D0 002A5          MOVL    #10, EDT$$G_RECSCRUPD                    ; 1079
                    00000000G  00              00  FB 002AC 39$:     CALLS   #0, EDT$$SC_UPD_NOOVERLAY1               ; 1085
                               01              56  D1 002B3 40$:     CMPL    OWED_MESSAGE, #1                         ; 1091
                                               26  13 002B6          BEQL    41$                                      ;
                                               7E  D4 002B8          CLRL    -(SP)                                    ; 1094
               7E 00000000G    00              01  C1 002BA          ADDL3   #1, EDT$$G_MESSAGE_LINE, -(SP)           ;
                  00000000G    00              02  FB 002C2          CALLS   #2, EDT$$SC_POSCSIF                      ;
                                               56  DD 002C9          PUSHL   OWED_MESSAGE                             ; 1095
```

```
                    00000000G  00            01  FB 002CB          CALLS   #1, EDT$$MSG_BELL                                ;
                    00000000G  00            01  D0 002D2          MOVL    #1, EDT$$G_MSGFLG                               ; 1096
                               56            01  D0 002D9          MOVL    #1, OWED_MESSAGE                                ; 1097
                                             23  11 002DC          BRB     46$                                             ; 1091
                         02                  54  CF 002DE 41$:     CASEL   CC_MSG, #0, #2                                  ; 1105
                       0010          0008          001F  002E2 42$:     .WORD   46$-42$,-                                       ;
                                                                            43$-42$,-                                       ;
                                                                            44$-42$                                         ;
                                             17  11 002E8          BRB     46$                                             ; 1109
                                  00000000G  8F  DD 002EA 43$:     PUSHL   #EDT$_ABOBYCC                                   ; 1115
                                             06  11 002F0          BRB     45$
                                  00000000G  8F  DD 002F2 44$:     PUSHL   #EDT$_CTRC_IGN                                  ; 1121
                    00000000G  00            01  FB 002F8 45$:     CALLS   #1, EDT$$MSG_BELL
                                             54  D4 002FF          CLRL    CC_MSG                                          ; 1122
                               32            6A  E8 00301 46$:     BLBS    EDT$$G_RCOV_MOD, 48$                            ; 1132
                    00000000G  00            00  FB 00304          CALLS   #0, EDT$$RD_ECHO                                ; 1135
                               28            50  E8 0030B          BLBS    R0, 48$
                                  00000000G  00  DD 0030E          PUSHL   EDT$$G_CUR_COL
                                  00000000G  00  DD 00314          PUSHL   EDT$$G_CS_CNO
                    00000000G  00            02  FB 0031A          CALLS   #2, EDT$$SC_POSCSIF
                                             13  11 00321          BRB     48$                                             ; 1060
                               10            57  E9 00323 47$:     BLBC    FIRST_TIME, 48$                                 ; 1140
                                             57  D4 00326          CLRL    FIRST_TIME                                      ; 1143
                    00000000G  00            00  FB 00328          CALLS   #0, EDT$$SC_INIT                                ; 1144
                    00000000G  00            00  FB 0032F          CALLS   #0, EDT$$SC_ERAALL                              ; 1145
                    00000000G  00            68  9E 00336 48$:     MOVAB   EDT$$T_CMD_BUF, EDT$$A_CMD_BUF                  ; 1151
                    00000000G  00      0100  C8  9E 0033D          MOVAB   EDT$$T_CMD_BUF+256, EDT$$A_CMD_END              ; 1152
                    00000000G  00            00  FB 00346          CALLS   #0, EDT$$CHK_CC                                 ; 1154
                               05            50  E9 0034D          BLBC    R0, 49$
                               55            01  D0 00350          MOVL    #1, STATUS
                                             0A  11 00353          BRB     50$
                    00000000G  00            00  FB 00355 49$:     CALLS   #0, EDT$$GET_KPADCMD
                               55            50  D0 0035C          MOVL    R0, STATUS
                               01            55  D1 0035F 50$:     CMPL    STATUS, #1                                      ; 1158
                                             35  12 00362          BNEQ    52$
                    00000000G  00            00  FB 00364          CALLS   #0, EDT$$CHK_CC
                               2B            50  E8 0036B          BLBS    R0, 52$
                               03            6B  D1 0036E          CMPL    EDT$$G_TI_TYP, #3                               ; 1165
                                             07  13 00371          BEQL    51$
                    00000000G  00            00  FB 00373          CALLS   #0, EDT$$START_WKINGMSG
                    00000000G  00            68  9E 0037A 51$:     MOVAB   EDT$$T_CMD_BUF, EDT$$A_CMD_BUF                  ; 1170
                                             01  DD 00381          PUSHL   #1                                              ; 1171
                    00000000G  00            01  FB 00383          CALLS   #1, EDT$$CHM_PAREXE
                               55            50  D0 0038A          MOVL    R0, STATUS
                               03            6B  D1 0038D          CMPL    EDT$$G_TI_TYP, #3                               ; 1176
                                             07  13 00390          BEQL    52$
                    00000000G  00            00  FB 00392          CALLS   #0, EDT$$STOP_WKINGMSG
                    00000000G  00            00  FB 00399 52$:     CALLS   #0, EDT$$CHK_CC                                 ; 1186
                               20            50  E9 003A0          BLBC    R0, 55$
                               50            6B  9E 003A3          MOVAB   EDT$$G_TI_TYP, R0                               ; 1190
                               03            50  D1 003A6          CMPL    R0, #3
                                             07  13 003A9          BEQL    53$
                    00000000G  00            01  D0 003AB          MOVL    #1, EDT$$G_SCR_CHGD
                               05  00000000G  00  E9 003B2 53$:     BLBC    EDT$$G_CC_DONE, 54$                             ; 1192
                               54            01  D0 003B9          MOVL    #1, CC_MSG
                                             07  11 003BC          BRB     56$
                               54            02  D0 003BE 54$:     MOVL    #2, CC_MSG                                      ;
```

```
                                                  02  11 003C1          BRB     56$                                     ; 1186
                                                  54  D4 003C3 55$:     CLRL    CC_MSG                                  ; 1196
                                                FD08  31 003C5 56$:     BRW     12$                                     ; 0912
                          00000000G   00            00  FB 003C8 57$:     CALLS   #0, EDT$$RPL_CHGDLN                     ; 1200
                          00000000G   00            00  FB 003CF          CALLS   #0, EDT$$SC_RESET                       ; 1208
                          00000000G   00            01  D0 003D6          MOVL    #1, EDT$$G_SCR_REBUILD                  ; 1213
                          00000000G   00 00000000G  00  9E 003DD          MOVAB   EDT$$TI_WRLN, EDT$$A_FMT_WRRUT          ; 1217
                                      50 00000000G  00  D0 003E8          MOVL    EDT$$A_CUR_BUF, R0                      ; 1218
                                      51 00000000G  00  9E 003EF          MOVAB   EDT$$T_LN_BUF, R1                       ;
                     0C     A0 00000000G   00            51  A3 003F6          SUBW3   R1, EDT$$A_LN_PTR, 12(R0)       ;
                          00000000G   00            68  9E 003FF          MOVAB   EDT$$T_CMD_BUF, EDT$$A_CMD_BUF          ; 1219
                                      50 00000000G  00  D0 00406          MOVL    EDT$$A_CMD_BUF, R0                      ; 1220
                                      60            21  90 0040D          MOVB    #33, (R0)                               ;
                                                    69  D5 00410          TSTL    EDT$$G_EXI                              ; 1222
                                                    09  13 00412          BEQL    58$                                     ;
                          00000000G   00            01  D0 00414          MOVL    #1, EDT$$G_EDIT_MOD                     ; 1225
                                                    69  D4 0041B          CLRL    EDT$$G_EXI                              ; 1226
                                      02            55  D1 0041D 58$:     CMPL    STATUS, #2                              ; 1229
                                                    04  12 00420          BNEQ    59$                                     ;
                                      50            01  D0 00422          MOVL    #1, R0                                  ;
                                                        04 00425          RET                                             ;
                                                    50  D4 00426 59$:     CLRL    R0                                      ; 1231
                                                        04 00428          RET                                             ;

; Routine Size:  1065 bytes,    Routine Base:  _EDT$CODE + 0000

;  662          1232  1
```

```
;  664          1233  1 %SBTTL 'EDT$$LOAD_CHMCHANGE - load this module into memory'
;  665          1234  1
;  666          1235  1 GLOBAL ROUTINE EDT$$LOAD_CHMCHANGE                ! Load this module into memory
;  667          1236  1     : NOVALUE =
;  668          1237  1
;  669          1238  1 !++
;  670          1239  1 ! FUNCTIONAL DESCRIPTION:
;  671          1240  1 !
;  672          1241  1 !       This is a do-nothing entry point, which serves to get this module
;  673          1242  1 !       loaded back into memory in case it was displaced by the screen update modules.
;  674          1243  1 !
;  675          1244  1 ! FORMAL PARAMETERS:
;  676          1245  1 !
;  677          1246  1 !       NONE
;  678          1247  1 !
;  679          1248  1 ! IMPLICIT INPUTS:
;  680          1249  1 !
;  681          1250  1 !       NONE
;  682          1251  1 !
;  683          1252  1 ! IMPLICIT OUTPUTS:
;  684          1253  1 !
;  685          1254  1 !       NONE
;  686          1255  1 !
;  687          1256  1 ! ROUTINE VALUE:
;  688          1257  1 !
;  689          1258  1 !       NONE
;  690          1259  1 !
;  691          1260  1 ! SIDE EFFECTS:
;  692          1261  1 !
;  693          1262  1 !       NONE
;  694          1263  1 !
;  695          1264  1 !--
;  696          1265  1
;  697          1266  2     BEGIN
;  698          1267  2     0
;  699          1268  1     END;                                    ! of routine EDT$$LOAD_CHMCHANGE


                                        0000 00000          .ENTRY  EDT$$LOAD_CHMCHANGE, Save nothing       ; 1235
                                          04 00002          RET                                             ; 1268

; Routine Size:  3 bytes,    Routine Base:  _EDT$CODE + 0429

;  700          1269  1
;  701          1270  1 !<BLF/PAGE
```

```
;  703          1271  1 END                                          ! of module EDT$CHMCHANGE
;  704          1272  1
;  705          1273  0 ELUDOM
```

PSECT SUMMARY

Name	Bytes	Attributes
_EDT$CODE	1068	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	Symbols Total	Symbols Loaded	Symbols Percent	Pages Mapped	Processing Time
_255DUA28:[EDT.SRC]EDT.L32;1	377	42	11	40	00:00.2
_255DUA28:[EDT.SRC]PSECTS.L32;1	2	1	50	7	00:00.1

COMMAND QUALIFIERS

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

; Size:          1068 code + 0 data bytes
; Run Time:         00:37.0
; Elapsed Time:     00:42.6
; Lines/CPU Min:     2065
; Lexemes/CPU-Min:  6240
; Memory Used:  231 pages
; Compilation Complete
```

ERRMSG REQ

TRACEOFF REQ

SUPPORTS REQ

TRAROUNAM REQ

BADKEY LIS

CHMCMBLIN LIS

TRACEON REQ

TRANSLATE REQ

CALLWIQ LIS

CHMBEGSEN LIS

CHMBELL LIS

SYSSYM REQ

TRANNAMES REQ

CHMCHGC LIS

CHMDELLIN LIS

TRACELIT REQ

EDTREQ REQ

KEYPADDEF REQ

CALLFIO LIS

CHMBEGWRD LIS

CHMCLRCC LIS

CHMBEEP LIS

CHMCHANGE LIS

TRACEMAC REQ

CHMDELCHR LIS

PSECTS REQ

VERSION REQ

CHMCHKCC LIS

CHMEINPUT LIS