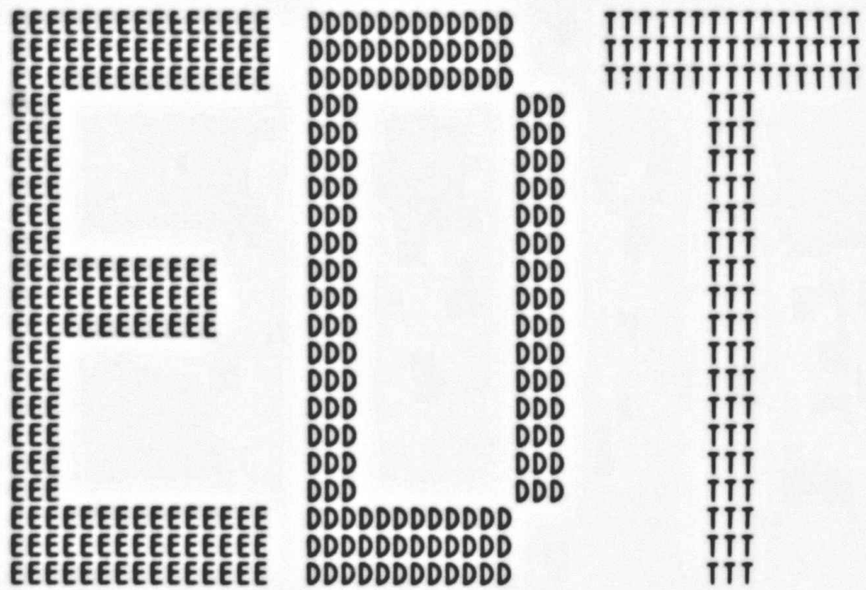
EEEEEEEEEEEEEEE DDDDDDDDDDD TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEE DDDDDDDDDDDD TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEE DDDDDDDDDDDD TTTTTTTTTTTTTTTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEEEEEEEEEEE DDD DDD TTT
EEEEEEEEEEEE DDD DDD TTT
EEEEEEEEEEEE DDD DDD 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

FILEID**LMOVE

```
LL          MM      MM    000000    VV      VV  EEEEEEEEEE
LL          MM      MM    000000    VV      VV  EEEEEEEEEE
LL          MMMM  MMMM  OO      OO  VV      VV  EE
LL          MMMM  MMMM  OO      OO  VV      VV  EE
LL          MM  MM  MM  OO      OO  VV      VV  EE
LL          MM  MM  MM  OO      OO  VV      VV  EE
LL          MM      MM  OO      OO  VV      VV  EEEEEEEE
LL          MM      MM  OO      OO  VV      VV  EEEEEEEE
LL          MM      MM  OO      OO  VV      VV  EE
LL          MM      MM  OO      OO  VV      VV  EE
LL          MM      MM  OO      OO    VV  VV    EE                ....
LL          MM      MM  OO      OO    VV  VV    EE                ....
LLLLLLLLLL  MM      MM    000000        VV      EEEEEEEEEE        ....
LLLLLLLLLL  MM      MM    000000        VV      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$LMOVE - MOVE and COPY line-mode commands'
;     2        0002  0 MODULE EDT$LMOVE (                      ! MOVE and COPY line-mode commands
;     3        0003  0                 IDENT = 'V04-000'                ! File: LMOVE.BLI Edit: STS1021
;     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 executes the line mode MOVE and COPY commands.
;    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: February 3, 1978
;    41        0041  1 !
;    42        0042  1 ! MODIFIED BY:
;    43        0043  1 !
;    44        0044  1 ! 1-001 - Original.  DJS 30-JAN-81.  This module was created by
;    45        0045  1 !       extracting the routine from the module EXEC.BLI.
;    46        0046  1 ! 1-002 - Regularize headers.  JBS 20-Mar-1981
;    47        0047  1 ! 1-003 - Use new message codes.  JBS 04-Aug-1981
;    48        0048  1 ! 1-004 - Abort on control C.  JBS 04-Jan-1982
;    49        0049  1 ! 1-005 - Change size of move/copy counters.  SMB 05-Feb-1982
;    50        0050  1 ! 1-006 - Change DUP count to 48-bits.  SMB 07-Feb-1982
;    51        0051  1 ! 1-007 - Limit DUP count to 65535 or less.  SMB 08-Feb-1982
;    52        0052  1 ! 1-008 - Fix bug in return address for DUP range check.  SMB 09-Feb-1982
;    53        0053  1 ! 1-009 - Set a flag if control C actually aborts something.  JBS 25-May-1982
;    54        0054  1 ! 1-010 - Correct a message error introduced by edit 1-009.  JBS 26-May-1982
;    55        0055  1 ! 1-011 - Check the status returned by EDT$$CPY_LN.  JBS 09-Jun-1982
;    56        0056  1 ! 1-012 - Stop working message before messages.  SMB 30-Jun-1982
;    57        0057  1 ! 1-013 - Stop processing on bad select range.  SMB 01-Jul-1982
```

```
;    58          0058  1 ! 1-014 - Use EDT$$FMT_CRLF instead of EDT$$OUT_FMTBUF.  JBS 05-Jul-1982
;    59          0059  1 ! 1-015 - Make sure the screen is repainted if any lines are changed.  For
;    60          0060  1 !           compatibility with EDT version 2, don't go through "Press return
;    61          0061  1 !           to continue".  JBS 06-Jul-1982
;    62          0062  1 ! 1-016 - Move edt$$tst_eob in line. STS 22-Sep-1982
;    63          0063  1 ! 1-017 - Modify to use new 48 bit macro. STS 01-Oct-1982
;    64          0064  1 ! 1-018 - Remove EDT$$G_SCR_CHGD, new screen update logic doesn't need it.  JBS 09-Oct-1982
;    65          0065  1 ! 1-019 - Put code for edt$$rng_posfrst in line. STS 11-Oct-1982
;    66          0066  1 ! 1-020 - Modify to use new compare macro. STS 20-Oct-1982
;    67          0067  1 ! 1-021 - Modify control c checking. STS 01-Dec-1982
;    68          0068  1 !--
;    69          0069  1
```

```
;    71          0070  1 %SBTTL 'Declarations'
;    72          0071  1 !
;    73          0072  1 ! TABLE OF CONTENTS:
;    74          0073  1 !
;    75          0074  1
;    76          0075  1 REQUIRE 'EDTSRC:TRAROUNAM';
;    77          0514  1
;    78          0515  1 FORWARD ROUTINE
;    79          0516  1     EDT$$MOVCPY_CMD : NOVALUE;                  ! Process the MOVE and COPY commands
;    80          0517  1
;    81          0518  1 !
;    82          0519  1 ! INCLUDE FILES:
;    83          0520  1 !
;    84          0521  1
;    85          0522  1 REQUIRE 'EDTSRC:EDTREQ';
;    86          0657  1
;    87          0658  1 !
;    88          0659  1 ! MACROS:
;    89          0660  1 !
;    90          0661  1 !       NONE
;    91          0662  1 !
;    92          0663  1 ! EQUATED SYMBOLS:
;    93          0664  1 !
;    94          0665  1 !       NONE
;    95          0666  1 !
;    96          0667  1 ! OWN STORAGE:
;    97          0668  1 !
;    98          0669  1 !       NONE
;    99          0670  1 !
;   100          0671  1 ! EXTERNAL REFERENCES:
;   101          0672  1 !
;   102          0673  1 !       In the routine
```

```
;  104          0674  1 %SBTTL 'EDT$$MOVCPY_CMD  - MOVE and COPY line-mode commands'
;  105          0675  1
;  106          0676  1 GLOBAL ROUTINE EDT$$MOVCPY_CMD (                ! MOVE and COPY line-mode commands
;  107          0677  1     DELETE                                      ! 1 = delete lines as moved (0 = copy)
;  108          0678  1     ) : NOVALUE =
;  109          0679  1
;  110          0680  1 !++
;  111          0681  1 ! FUNCTIONAL DESCRIPTION:
;  112          0682  1 !
;  113          0683  1 !       Command processing for MOVE and COPY.   First, find the destination
;  114          0684  1 !       range to make sure we can position there.  Next process the source
;  115          0685  1 !       range, copying all the lines to an intermediate buffer.  Then position
;  116          0686  1 !       to the destination range again, and copy the temporary buffer the
;  117          0687  1 !       indicated number of times.  If the command is MOVE then the  DELETE parameter
;  118          0688  1 !       will be true, causing us to delete the source lines as they are processed.
;  119          0689  1 !
;  120          0690  1 ! FORMAL PARAMETERS:
;  121          0691  1 !
;  122          0692  1 !  DELETE                   1 = delete lines (MOVE command) 0 = don't (COPY command)
;  123          0693  1 !
;  124          0694  1 ! IMPLICIT INPUTS:
;  125          0695  1 !
;  126          0696  1 !       EDT$$A_CUR_BUF
;  127          0697  1 !       EDT$$A_WK_LN
;  128          0698  1 !       EDT$$A_EXE_CURCMD
;  129          0699  1 !       EDT$$G_EXE_QRYQUIT
;  130          0700  1 !       EDT$$G_EXE_SBITS
;  131          0701  1 !       EDT$$Z_EXE_SBLK
;  132          0702  1 !
;  133          0703  1 ! IMPLICIT OUTPUTS:
;  134          0704  1 !
;  135          0705  1 !       EDT$$G_CC_DONE
;  136          0706  1 !
;  137          0707  1 ! ROUTINE VALUE:
;  138          0708  1 !
;  139          0709  1 !       NONE
;  140          0710  1 !
;  141          0711  1 ! SIDE EFFECTS:
;  142          0712  1 !
;  143          0713  1 !       NONE
;  144          0714  1 !
;  145          0715  1 !--
;  146          0716  1
;  147          0717  2     BEGIN
;  148          0718  2
;  149          0719  2     EXTERNAL ROUTINE
;  150          0720  2         EDT$$STOP_WKINGMSG,
;  151          0721  2         EDT$$FMT_STRCNT : NOVALUE,
;  152          0722  2         EDT$$FMT_CRLF,                          ! Terminate an output line
;  153          0723  2         EDT$$FMT_MSG,
;  154          0724  2         EDT$$FMT_STR,
;  155          0725  2         EDT$$PROC_QRYQAL,
;  156          0726  2         EDT$$NXT_LNRNG,
;  157          0727  2         EDT$$RNG_REPOS,
;  158          0728  2         EDT$$CPY_LN,
;  159          0729  2         EDT$$END_CPY,
;  160          0730  2         EDT$$LOC_LN,
```

```
;  161     0731  2          EDT$$NEW_BUF,
;  162     0732  2          EDT$$RD_CURLN,
;  163     0733  2          EDT$$RD_NXTLN,
;  164     0734  2          EDT$$START_CPY,
;  165     0735  2          EDT$$TOP_BUF,
;  166     0736  2          EDT$$CHK_CC;                                ! Check for a control C having been typed
;  167     0737  2
;  168     0738  2      EXTERNAL
;  169     0739  2          EDT$$G_EXT_MOD,                             ! Are we in EXT mode
;  170     0740  2          EDT$$L_LNOO : LN_BLOCK,
;  171     0741  2          EDT$$L_LNO_ZERO : LN_BLOCK,
;  172     0742  2          EDT$$A_CUR_BUF : REF POS_BLOCK,
;  173     0743  2          EDT$$A_WK_LN : REF LIN_BLOCK,
;  174     0744  2          EDT$$A_EXE_CURCMD : REF NODE_BLOCK,         ! Pointer to the current command.
;  175     0745  2          EDT$$G_EXE_QRYQUIT,                         ! Quit flag for /QUERY operations.
;  176     0746  2          EDT$$G_RNG_FRSTLN,
;  177     0747  2          EDT$$Z_RNG_ORIGPOS: POS_BLOCK,
;  178     0748  2          EDT$$G_EXE_SBITS,                           ! The options switches.
;  179     0749  2          EDT$$Z_EXE_SBLK : REF NODE_BLOCK,           ! The option switch value block.
;  180     0750  2          EDT$$G_CC_DONE,                             ! Set to 1 if control C actually aborts something
;  181     0751  2          EDT$$Z_EOB_LN;
;  182     0752  2
;  183     0753  2      MESSAGES ((DSTMOVCOP, BADRANGE, INSMEM));
;  184     0754  2
;  185     0755  2      LOCAL
;  186     0756  2          NLINES : LN_BLOCK,                          ! Number of lines copied.
;  187     0757  2          DEST_LINE : LN_BLOCK,                       ! Line number of the destination.
;  188     0758  2          S_TBCB,                                     ! Address of TBCB for the source.
;  189     0759  2          D_TBCB,                                     ! Address of TBCB for the destination.
;  190     0760  2          NUM_COPIES : LN_BLOCK,                      ! Number of times to copy it.
;  191     0761  2          START_POS : POS_BLOCK,                      ! The start position.
;  192     0762  2          TEMP : TBCB_BLOCK,                          ! Temporary buffer TBCB.
;  193     0763  2          COPY_COUNTER : LN_BLOCK,                    ! Number of copies made
;  194     0764  2          CONTROL_C;                                  ! 1 = a control C was typed.
;  195     0765  2
;  196     0766  2      !+
;  197     0767  2      ! Save the current position for future restoration.
;  198     0768  2      !-
;  199     0769  2          EDT$$CPY_MEM (POS_SIZE, .EDT$$A_CUR_BUF, START_POS);
;  200     0770  2      !+
;  201     0771  2      ! Save the current TBCB address so we can detect a buffer change
;  202     0772  2      !-
;  203     0773  2          S_TBCB = .EDT$$A_CUR_BUF;
;  204     0774  2      !+
;  205     0775  2      ! First check to make sure we can locate the destination line.
;  206     0776  2      !-
;  207     0777  2
;  208     0778  2          EDT$$G_RNG_FRSTLN = 1;
;  209     0779  2          EDT$$CPY_MEM(POS_SIZE, .EDT$$A_CUR_BUF, EDT$$Z_RNG_ORIGPOS);
;  210     0780  3          IF ( NOT EDT$$RNG_REPOS (.EDT$$A_EXE_CURCMD [RANGE1]))
;  211     0781  2          THEN
;  212     0782  3              BEGIN
;  213     0783  3              EDT$$FMT_MSG (EDT$_DSTMOVCOP);
;  214     0784  3              RETURN;
;  215     0785  2              END;
;  216     0786  2
;  217     0787  2          CONTROL_C = EDT$$CHK_CC();
```

```
;  218          0788  2      IF .CONTROL_C THEN RETURN;
;  219          0789  2  !+
;  220          0790  2  ! Save the line number of the destination line so we are able to reposition
;  221          0791  2  ! there when we are ready to move it.
;  222          0792  2  !-
;  223          0793  2      MOVELINE (EDT$$A_WK_LN [LIN_NUM], DEST_LINE);
;  224          0794  2      D_TBCB = .EDT$$A_CUR_BUF;
;  225          0795  2  !+
;  226          0796  2  ! Create a temporary buffer to copy the lines.
;  227          0797  2  !-
;  228          0798  2      EDT$$A_CUR_BUF = TEMP;
;  229          0799  2      CH$FILL (0, TBCB_SIZE, TEMP);
;  230          0800  2      EDT$$NEW_BUF ();
;  231          0801  2      EDT$$A_CUR_BUF = .S_TBCB;
;  232          0802  2      EDT$$CPY_MEM (POS_SIZE, START_POS, .EDT$$A_CUR_BUF);
;  233          0803  2      EDT$$RD_CURLN ();
;  234          0804  2  !+
;  235          0805  2  ! Now position to the front of the source range.
;  236          0806  2  !-
;  237          0807  2
;  238          0808  2      EDT$$G_RNG_FRSTLN = 1;
;  239          0809  2      EDT$$CPY_MEM(POS_SIZE,.EDT$$A_CUR_BUF, EDT$$Z_RNG_ORIGPOS);
;  240          0810  2      IF ( NOT EDT$$RNG_REPOS (.EDT$$A_EXE_CURCMD [RANGE2])) THEN RETURN;
;  241          0811  2
;  242          0812  2  !+
;  243          0813  2  ! Copy the source lines to a temporary buffer.
;  244          0814  2  !-
;  245          0815  2      EDT$$START_CPY (TEMP);
;  246          0816  2      MOVELINE (EDT$$L_LNO_ZERO, NLINES);
;  247          0817  2
;  248          0818  2      IF (.DELETE EQL 0) THEN CONTROL_C = EDT$$CHK_CC();
;  249          0819  2
;  250          0820  2      WHILE (EDT$$NXT_LNRNG (.DELETE) AND (.EDT$$A_WK_LN NEQA EDT$$Z_EOB_LN) AND (.EDT$$G_EXE_QRYQUIT EQL 0))
;  251          0821  3          BEGIN
;  252          0822  3          IF EDT$$PROC_QRYQAL (0, 0)
;  253          0823  3          THEN
;  254          0824  4              BEGIN
;  255          0825  4              ADDLINE (NUMBER_ONE, NLINES);
;  256          0826  4              IF .DELETE THEN EDT$$A_CUR_BUF [POS_CHAR_POS] = 0;
;  257          0827  5              IF ( NOT EDT$$CPY_LN (.DELETE))
;  258          0828  4              THEN
;  259          0829  5                  BEGIN
;  260          0830  5                  EDT$$END_CPY (0);
;  261          0831  5                  EDT$$FMT_MSG (EDT$_INSMEM);
;  262          0832  5                  RETURN;
;  263          0833  4                  END;
;  264          0834  4
;  265          0835  4              END
;  266          0836  3          ELSE
;  267          0837  3
;  268          0838  3              IF .DELETE THEN EDT$$RD_NXTLN ();
;  269          0839  3
;  270          0840  3      IF (.DELETE EQL 0) THEN CONTROL_C = EDT$$CHK_CC();
;  271          0841  2      END;
;  272          0842  2
;  273          0843  2      EDT$$END_CPY (0);
;  274          0844  2  !+
```

```
;   275         0845  2 ! Now, re-position to the destination range.
;   276         0846  2 !-
;   277         0847  2     EDT$$A_CUR_BUF = .D_TBCB;
;   278         0848  2     EDT$$RD_CURLN ();
;   279         0849  2     EDT$$LOC_LN (DEST_LINE);
;   280         0850  2 !+
;   281         0851  2 ! Switch to the temp buffer we just created.
;   282         0852  2 !-
;   283         0853  2     EDT$$A_CUR_BUF = TEMP;
;   284         0854  2     EDT$$RD_CURLN ();
;   285         0855  2 !+
;   286         0856  2 ! Get the number of times to copy.
;   287         0857  2 !-
;   288         0858  2
;   289         0859  2     IF .EDT$$G_EXE_SBITS<OPB_DUPL>
;   290         0860  2     THEN
;   291         0861  3         MOVELINE (EDT$$Z_EXE_SBLK [SW_VAL1], NUM_COPIES)
;   292         0862  2     ELSE
;   293         0863  2         MOVELINE (EDT$$L_LNO0, NUM_COPIES);
;   294         0864  2
;   295         0865  2 !+
;   296         0866  2 ! Check that DUP is less than 65536
;   297         0867  2 !-
;   298         0868  2
;   299         0869  3     IF ((.NUM_COPIES [LN_HI] NEQ 0) OR (.NUM_COPIES [LN_MD] NEQ 0))
;   300         0870  2     THEN
;   301         0871  3         BEGIN
;   302         0872  3         EDT$$FMT_MSG (EDT$_BADRANGE);
;   303         0873  3         MOVELINE (EDT$$L_LNO_ZERO, NUM_COPIES);
;   304         0874  4         MOVELINE (EDT$$L_LNO_ZERO, NLINES)
;   305         0875  2         END;
;   306         0876  2
;   307         0877  2 !+
;   308         0878  2 ! And finally, do the copy.
;   309         0879  2 !-
;   310         0880  2     EDT$$START_CPY (.D_TBCB);
;   311         0881  2     MOVELINE (EDT$$L_LNO_ZERO, COPY_COUNTER);
;   312         0882  2     IF (.DELETE EQL 0) THEN CONTROL_C = EDT$$CHK_CC ();
;   313         0883  2
;   314         0884  2     WHILE ((CMPLNO (COPY_COUNTER, NUM_COPIES) LSS 0) AND ( NOT .CONTROL_C)) DO
;   315         0885  3         BEGIN
;   316         0886  3         EDT$$TOP_BUF ();
;   317         0887  3
;   318         0888  3         WHILE (.EDT$$A_WK_LN NEQA EDT$$Z_EOB_LN) DO
;   319         0889  4             BEGIN
;   320         0890  4
;   321         0891  5             IF ( NOT EDT$$CPY_LN (0))
;   322         0892  4             THEN
;   323         0893  5                 BEGIN
;   324         0894  5                 EDT$$END_CPY (1);
;   325         0895  5                 EDT$$FMT_MSG (EDT$_INSMEM);
;   326         0896  5                 RETURN;
;   327         0897  4                 END;
;   328         0898  4
;   329         0899  4             EDT$$RD_NXTLN ();
;   330         0900  3             END;
;   331         0901  3
```

```
;   332     0902  3          ADDLINE (NUMBER_ONE, COPY_COUNTER);
;   333     0903  3          IF (.DELETE EQL 0) THEN CONTROL_C = EDT$$CHK_CC ();
;   334     0904  2          END;
;   335     0905  2
;   336     0906  2      EDT$$END_CPY (1);
;   337     0907  2
;   338     0908  2      IF (.EDT$$G_EXT_MOD) THEN EDT$$STOP_WKINGMSG ();
;   339     0909  2
;   340     0910  2  !+
;   341     0911  2  ! Report the number of lines moved or copied.
;   342     0912  2  !-
;   343     0913  2      EDT$$FMT_STRCNT (NLINES, UPLIT (%STRING (' line')), 5);
;   344     0914  2
;   345     0915  3      IF (.DELETE NEQ 0)
;   346     0916  2      THEN
;   347     0917  2          EDT$$FMT_STR (UPLIT (%STRING (' moved')), 6)
;   348     0918  2      ELSE
;   349     0919  2          EDT$$FMT_STR (UPLIT (%STRING (' copied')), 7);
;   350     0920  2
;   351     0921  3      IF NOT (LINNOEQL (COPY_COUNTER, EDT$$L_LNOO))       !
;   352     0922  3      THEN
;   353     0923  3          BEGIN
;   354     0924  3          EDT$$FMT_STR (UPLIT (%STRING (' ')), 1);
;   355     0925  3          EDT$$FMT_STRCNT (COPY_COUNTER, UPLIT (%STRING (' time')), 5);
;   356     0926  2          END;
;   357     0927  2
;   358     0928  2      EDT$$FMT_CRLF ();
;   359     0929  2
;   360     0930  2      IF ( NOT (LINNOEQL (COPY_COUNTER, NUM_COPIES)) AND (.CONTROL_C)) THEN EDT$$G_CC_DONE = 1;
;   361     0931  2
;   362     0932  1      END;                                            ! of routine EDT$$MOVCPY_CMD

                                                                .TITLE  EDT$LMOVE EDT$LMOVE - MOVE and COPY line-mode c
;                                                                       ommands
                                                                .IDENT  \V04-000\

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

                        00  00  00  65  6E  69  6C  20  00000 P.AAA:   .ASCII  \ line\<0><0><0>                                ;
                        00  00  64  65  76  6F  6D  20  00008 P.AAB:   .ASCII  \ moved\<0><0>                                 ;
                        00  64  65  69  70  6F  63  20  00010 P.AAC:   .ASCII  \ copied\<0>                                   ;
                                        00  00  00  20  00018 P.AAD:   .ASCII  \ \<0><0><0>                                  ;
                        00  00  00  65  6D  69  74  20  0001C P.AAE:   .ASCII  \ time\<0><0><0>                               ;

                                                                .EXTRN  EDT$$STOP_WKINGMSG
                                                                .EXTRN  EDT$$FMT_STRCNT
                                                                .EXTRN  EDT$$FMT_CRLF, EDT$$FMT_MSG
                                                                .EXTRN  EDT$$FMT_STR, EDT$$PROC_QRYQAL
                                                                .EXTRN  EDT$$NXT_LNRNG, EDT$$RNG_REPOS
                                                                .EXTRN  EDT$$CPY_LN, EDT$$END_CPY
                                                                .EXTRN  EDT$$LOC_LN, EDT$$NEW_BUF
                                                                .EXTRN  EDT$$RD_CURLN, EDT$$RD_NXTLN
                                                                .EXTRN  EDT$$START_CPY, EDT$$TOP_BUF
                                                                .EXTRN  EDT$$CHK_CC, EDT$$G_EXT_MOD
                                                                .EXTRN  EDT$$L_LNOO, EDT$$L_LNO_ZERO
                                                                .EXTRN  EDT$$A_CUR_BUF, EDT$$A_QK_LN
```

```
                                                                .EXTRN  EDT$$A_EXE_CURCMD
                                                                .EXTRN  EDT$$G_EXE_QRYQUIT
                                                                .EXTRN  EDT$$G_RNG_FRSTLN
                                                                .EXTRN  EDT$$Z_RNG_ORIGPOS
                                                                .EXTRN  EDT$$G_EXE_SBITS
                                                                .EXTRN  EDT$$Z_EXE_SBLK
                                                                .EXTRN  EDT$$G_CC_DONE, EDT$$Z_EOB_LN
                                                                .EXTRN  EDT$_DSTMOVCOP, EDT$_BADRANGE
                                                                .EXTRN  EDT$_INSMEM

                                                0FFC 00000      .ENTRY  EDT$$MOVCPY_CMD, Save R2,R3,R4,R5,R6,R7,R8,-; 0676
                                                                        R9,R10,R11
                                    5B 00000000G   00   9E 00002        MOVAB   EDT$$L_LNO_ZERO, R11
                                    5A 00000000G   00   9E 00009        MOVAB   EDT$$A_CUR_BUF, R10
                                    5E       A0    AE   9E 00010        MOVAB   -96(SP), SP
                                    56             6A   D0 00014        MOVL    EDT$$A_CUR_BUF, R6                      ; 0769
                38  AE                         66       0E   28 00017   MOVC3   #14, (R6), START_POS
                                    57             56   D0 0001C        MOVL    R6, S_TBCB                              ; 0773
                    00000000G   00             01   D0 0001F        MOVL    #1, EDT$$G_RNG_FRSTLN                   ; 0778
00000000G   00                      66         0E   28 00026        MOVC3   #14, (R6), EDT$$Z_RNG_ORIGPOS           ; 0779
                                    50 00000000G   00   D0 0002E        MOVL    EDT$$A_EXE_CURCMD, R0                   ; 0780
                                          04       A0   DD 00035        PUSHL   4(R0)
                    00000000G   00             01   FB 00038        CALLS   #1, EDT$$RNG_REPOS
                                09             50   E8 0003F        BLBS    R0, 1$
                                       00000000G   8F   DD 00042        PUSHL   #EDT$_DSTMOVCOP                         ; 0783
                                             01C6  31   00048        BRW     23$
                    00000000G   00             00   FB 0004B 1$:    CALLS   #0, EDT$$CHK_CC                         ; 0787
                                59             50   D0 00052        MOVL    R0, CONTROL_C
                                01             59   E9 00055        BLBC    CONTROL_C, 2$                           ; 0788
                                               04   00058        RET
                                50 00000000G   00   D0 00059 2$:    MOVL    EDT$$A_WK_LN, R0                        ; 0793
          50  AE        01       A0             06   28 00060        MOVC3   #6, 1(R0), DEST_LINE
                                56             6A   D0 00066        MOVL    EDT$$A_CUR_BUF, D_TBCB                  ; 0794
                                6A       08    AE   9E 00069        MOVAB   TEMP, EDT$$A_CUR_BUF                    ; 0798
2D              00              6E             00   2C 0006D        MOVC5   #0, (SP), #0, #45, TEMP                 ; 0799
                                          08   AE       00072
                    00000000G   00             00   FB 00074        CALLS   #0, EDT$$NEW_BUF                        ; 0800
                                6A             57   D0 0007B        MOVL    S_TBCB, EDT$$A_CUR_BUF                  ; 0801
                                50             6A   D0 0007E        MOVL    EDT$$A_CUR_BUF, R0                      ; 0802
                60      38      AE             0E   28 00081        MOVC3   #14, START_POS, (R0)
                    00000000G   00             00   FB 00086        CALLS   #0, EDT$$RD_CURLN                       ; 0803
                    00000000G   00             01   D0 0008D        MOVL    #1, EDT$$G_RNG_FRSTLN                   ; 0808
                                50             6A   D0 00094        MOVL    EDT$$A_CUR_BUF, R0                      ; 0809
    00000000G   00              60             0E   28 00097        MOVC3   #14, (R0), EDT$$Z_RNG_ORIGPOS
                                50 00000000G   00   D0 0009F        MOVL    EDT$$A_EXE_CURCMD, R0                   ; 0810
                                          08   A0   DD 000A6        PUSHL   8(R0)
                    00000000G   00             01   FB 000A9        CALLS   #1, EDT$$RNG_REPOS
                                01             50   E8 000B0        BLBS    R0, 3$
                                               04   000B3        RET
                                          08   AE   9F 000B4 3$:    PUSHAB  TEMP                                    ; 0815
                    00000000G   00             01   FB 000B7        CALLS   #1, EDT$$START_CPY
          58  AE                6B             06   28 000BE        MOVC3   #6, EDT$$L_LNO_ZERO, NLINES             ; 0816
                                57       04    AC   D0 000C3        MOVL    DELETE, R7                              ; 0818
                                               58   D4 000C7        CLRL    R8
                                               57   D5 000C9        TSTL    R7
                                               0C   12 000CB        BNEQ    5$
                                               58   D6 000CD        INCL    R8
```

```
                 00000000G  00         00  FB 000CF 4$:   CALLS   #0, EDT$$CHK_CC                    ;
                            59         50  D0 000D6       MOVL    R0, CONTROL_C                      ;
                                       57  DD 000D9 5$:   PUSHL   R7                                 ; 0820
                 00000000G  00         01  FB 000DB       CALLS   #1, EDT$$NXT_LNRNG                 ;
                            55         50  E9 000E2       BLBC    R0, 10$                            ;
                            50 00000000G  00  9E 000E5    MOVAB   EDT$$Z_EOB_LN, R0                  ;
                            50 00000000G  00  D1 000EC    CMPL    EDT$$A_WK_LN, R0                   ;
                                       45  13 000F3       BEQL    10$                                ;
                               00000000G  00  D5 000F5    TSTL    EDT$$G_EXE_QRYQUIT                 ;
                                       3D  12 000FB       BNEQ    10$                                ;
                                       7E  7C 000FD       CLRQ    -(SP)                              ; 0822
                 00000000G  00         02  FB 000FF       CALLS   #2, EDT$$PROC_QRYQAL               ;
                            22         50  E9 00106       BLBC    R0, 8$                             ;
                                  58   AE  D6 00109       INCL    FIRST_LWORD                        ; 0825
                                       03  12 0010C       BNEQ    6$                                 ;
                                  5C   AE  B6 0010E       INCW    NEXT_WORD                          ;
                            06         57  E9 00111 6$:   BLBC    R7, 7$                             ; 0826
                            50         6A  D0 00114       MOVL    EDT$$A_CUR_BUF, R0                 ;
                                  0C   A0  B4 00117       CLRW    12(R0)                             ;
                                       57  DD 0011A 7$:   PUSHL   R7                                 ; 0827
                 00000000G  00         01  FB 0011C       CALLS   #1, EDT$$CPY_LN                    ;
                            0F         50  E8 00123       BLBS    R0, 9$                             ;
                                       7E  D4 00126       CLRL    -(SP)                              ; 0830
                                     00D9  31 00128       BRW     22$                                ;
                            07         57  E9 0012B 8$:   BLBC    R7, 9$                             ; 0838
                 00000000G  00         00  FB 0012E       CALLS   #0, EDT$$RD_NXTLN                  ;
                            A1         58  E9 00135 9$:   BLBC    R8, 5$                             ; 0840
                                       95  11 00138       BRB     4$                                 ;
                                       7E  D4 0013A 10$:  CLRL    -(SP)                              ; 0843
                 00000000G  00         01  FB 0013C       CALLS   #1, EDT$$END_CPY                   ;
                            6A         56  D0 00143       MOVL    D_TBCB, EDT$$A_CUR_BUF             ; 0847
                 00000000G  00         00  FB 00146       CALLS   #0, EDT$$RD_CURLN                  ; 0848
                                  50   AE  9F 0014D       PUSHAB  DEST_LINE                          ; 0849
                 00000000G  00         01  FB 00150       CALLS   #1, EDT$$LOC_LN                    ;
                            6A    08   AE  9E 00157       MOVAB   TEMP, EDT$$A_CUR_BUF               ; 0853
                 00000000G  00         00  FB 0015B       CALLS   #0, EDT$$RD_CURLN                  ; 0854
             0F  00000000G  00         05  E1 00162       BBC     #5, EDT$$G_EXE_SBITS, 11$          ; 0859
                            50 00000000G  00  D0 0016A    MOVL    EDT$$Z_EXE_SBLR, R0                ; 0861
      48    AE         08   A0         06  28 00171       MOVC3   #6, 8(R0), NUM_COPIES              ;
                                       09  11 00177       BRB     12$                                ;
      48    AE   00000000G  00         06  28 00179 11$:  MOVC3   #6, EDT$$L_LNO0, NUM_COPIES        ; 0863
                                  4C   AE  B5 00182 12$:  TSTW    NUM_COPIES+4                       ; 0869
                                       05  12 00185       BNEQ    13$                                ;
                                  4A   AE  B5 00187       TSTW    NUM_COPIES+2                       ;
                                       17  13 0018A       BEQL    14$                                ;
                               00000000G  8F  DD 0018C 13$: PUSHL #EDT$_BADRANGE                     ; 0872
                 00000000G  00         01  FB 00192       CALLS   #1, EDT$$FMT_MSG                   ;
      48    AE              6B         06  28 00199       MOVC3   #6, EDT$$L_LNO_ZERO, NUM_COPIES    ; 0873
      58    AE              6B         06  28 0019E       MOVC3   #6, EDT$$L_LNO_ZERO, NLINES        ; 0874
                                       56  DD 001A3 14$:  PUSHL   D_TBCB                             ; 0880
                 00000000G  00         01  FB 001A5       CALLS   #1, EDT$$START_CPY                 ;
            6E              6B         06  28 001AC       MOVC3   #6, EDT$$L_LNO_ZERO, COPY_COUNTER  ; 0881
                            0A         58  E9 001B0 15$:  BLBC    R8, 16$                            ; 0882
                 00000000G  00         00  FB 001B3       CALLS   #0, EDT$$CHK_CC                    ;
                            59         50  D0 001BA       MOVL    R0, CONTROL_C                      ;
                 4C         AE    04   AE  B1 001BD 16$:  CMPW    HIGH_1, HIGH_2                     ; 0884
                                       08  1F 001C2       BLSSU   17$                                ;
```

```
                                                 11  12 001C4          BNEQ    19$                                                                 ;
                                        48   AE      6E  D1 001C6          CMPL    LOW_1, LOW_2                                                        ;
                                                     05  1E 001CA          BGEQU   18$                                                                 ;
                                             50      01  CE 001CC 17$:      MNEGL   #1, R0                                                              ;
                                                     09  11 001CF          BRB     20$                                                                 ;
                                                     04  12 001D1 18$:      BNEQ    19$                                                                 ;
                                                     50  D4 001D3          CLRL    R0                                                                  ;
                                                     03  11 001D5          BRB     20$                                                                 ;
                                             50      01  D0 001D7 19$:      MOVL    #1, R0                                                              ;
                                                     4F  18 001DA 20$:      BGEQ    26$                                                                 ;
                                             4C      59  E8 001DC          BLBS    CONTROL_C, 26$                                                      ;
                                00000000G    00      00  FB 001DF          CALLS   #0, EDT$$TOP_BUF                                                    ; 0886
                                             50 00000000G 00  9E 001E6 21$:      MOVAB   EDT$$Z_EOB_LN, R0                                           ; 0888
                                             50 00000000G 00  D1 001ED          CMPL    EDT$$A_WK_LN, R0                                                    ;
                                                     2C  13 001F4          BEQL    25$                                                                 ;
                                                     7E  D4 001F6          CLRL    -(SP)                                                               ; 0891
                                00000000G    00          01  FB 001F8          CALLS   #1, EDT$$CPY_LN                                                     ;
                                             17          50  E8 001FF          BLBS    R0, 24$                                                             ;
                                                         01  DD 00202          PUSHL   #1                                                                  ; 0894
                                00000000G    00          01  FB 00204 22$:      CALLS   #1, EDT$$END_CPY                                                    ;
                                                 00000000G  8F  DD 0020B          PUSHL   #EDT$_INSMEM                                                        ; 0895
                                00000000G    00          01  FB 00211 23$:      CALLS   #1, EDT$$FMT_MSG                                                    ;
                                                         04  00218          RET                                                                         ; 0893
                                00000000G    00          00  FB 00219 24$:      CALLS   #0, EDT$$RD_NXTLN                                                   ; 0899
                                                         C4  11 00220          BRB     21$                                                                 ; 0888
                                                         6E  D6 00222 25$:      INCL    FIRST_LWORD                                                         ; 0902
                                                         8A  12 00224          BNEQ    15$                                                                 ;
                                                     04  AE  B6 00226          INCW    NEXT_WORD                                                           ;
                                                         85  11 00229          BRB     15$                                                                 ; 0903
                                                         01  DD 0022B 26$:      PUSHL   #1                                                                  ; 0906
                                00000000G    00          01  FB 0022D          CALLS   #1, EDT$$END_CPY                                                    ;
                                             07 00000000G 00  E9 00234          BLBC    EDT$$G_EXT_MOD, 27$                                                 ; 0908
                                00000000G    00          00  FB 0023B          CALLS   #0, EDT$$STOP_WKINGMSG                                              ;
                                                         05  DD 00242 27$:      PUSHL   #5                                                                  ; 0913
                                                 FD94    CF  9F 00244          PUSHAB  P.AAA                                                               ;
                                                   60    AE  9F 00248          PUSHAB  NLINES                                                              ;
                                00000000G    00          03  FB 0024B          CALLS   #3, EDT$$FMT_STRCNT                                                 ;
                                                         57  D5 00252          TSTL    R7                                                                  ; 0915
                                                         08  13 00254          BEQL    28$                                                                 ;
                                                         06  DD 00256          PUSHL   #6                                                                  ; 0917
                                                 FD88    CF  9F 00258          PUSHAB  P.AAB                                                               ;
                                                         06  11 0025C          BRB     29$                                                                 ;
                                                         07  DD 0025E 28$:      PUSHL   #7                                                                  ; 0919
                                                 FD88    CF  9F 00260          PUSHAB  P.AAC                                                               ;
                                00000000G    00          02  FB 00264 29$:      CALLS   #2, EDT$$FMT_STR                                                    ;
                                00000000G    00          6E  D1 0026B          CMPL    LOW_1, LOW_2                                                        ; 0921
                                                         0A  12 00272          BNEQ    30$                                                                 ;
                                00000000G    00      04  AE  B1 00274          CMPW    HIGH_1, HIGH_2                                                      ;
                                                         1D  13 0027C          BEQL    31$                                                                 ;
                                                         01  DD 0027E 30$:      PUSHL   #1                                                                  ; 0924
                                                 FD70    CF  9F 00280          PUSHAB  P.AAD                                                               ;
                                00000000G    00          02  FB 00284          CALLS   #2, EDT$$FMT_STR                                                    ;
                                                         05  DD 0028B          PUSHL   #5                                                                  ; 0925
                                                 FD67    CF  9F 0028D          PUSHAB  P.AAE                                                               ;
                                                   08    AE  9F 00291          PUSHAB  COPY_COUNTER                                                        ;
                                00000000G    00          03  FB 00294          CALLS   #3, EDT$$FMT_STRCNT                                                 ;
                                00000000G    00          00  FB 0029B 31$:      CALLS   #0, EDT$$FMT_CRLF                                                   ; 0928
```

```
                          48   AE             6E  D1 002A2           CMPL    LOW_1, LOW_2                                    ; 0930
                                              07  12 002A6           BNEQ    32$                                             ;
                          4C   AE        04   AE  B1 002A8           CMPW    HIGH_1, HIGH_2                                  ;
                                              0A  13 002AD           BEQL    33$                                             ;
                               07             59  E9 002AF 32$:      BLBC    CONTROL_C, 33$                                  ;
                    00000000G  00             01  D0 002B2           MOVL    #1, EDT$$G_CC_DONE                              ;
                                                  04 002B9 33$:      RET                                                     ; 0932

; Routine Size:  698 bytes,    Routine Base:  _EDT$CODE + 0024

;  363          0933  1
;  364          0934  1 !<BLF/PAGE>
```

```
;  366          0935  1 END                                        ! of module EDT$LMOVE
;  367          0936  1
;  368          0937  0 ELUDOM
```

```
;                                  PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  _EDT$CODE                        734  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       102        27           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$:LMOVE/OBJ=OBJ$:LMOVE MSRC$:LMOVE.BLI/UPDATE=(ENH$:LMOVE)

; Size:         698 code + 36 data bytes
; Run Time:         00:28.4
; Elapsed Time:     00:34.5
; Lines/CPU Min:     1977
; Lexemes/CPU-Min: 10666
; Memory Used:  239 pages
; Compilation Complete
```