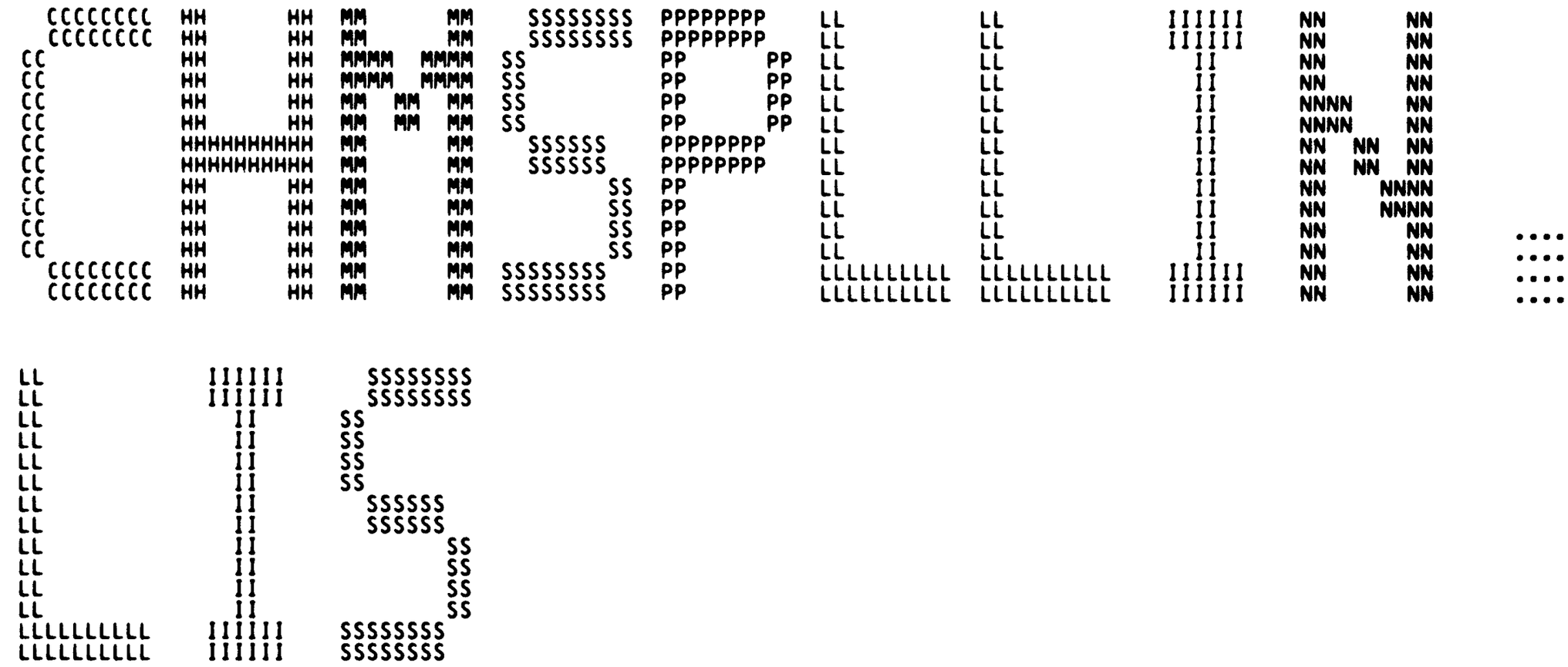
```
EEEEEEEEEEEEEEE  DDDDDDDDDDD     TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD    TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD    TTTTTTTTTTTTTTT
EEE              DDD         DDD       TTT
EEE              DDD          DDD      TTT
EEE              DDD          DDD      TTT
EEE              DDD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


CCCCCCCC HH HH MM MM SSSSSSSS PPPPPPPP LL LL IIIIII NN NN
CCCCCCCC HH HH MM MM SSSSSSSS PPPPPPPP LL LL IIIIII NN NN
CC HH HH MMMM MMMM SS PP PP LL LL II NN NN
CC HH HH MMMM MMMM SS PP PP LL LL II NN NN
CC HH HH MM MM MM SS PP PP LL LL II NNNN NN
CC HH HH MM MM MM SS PP PP LL LL II NNNN NN
CC HHHHHHHHHH MM MM SSSSSS PPPPPPPP LL LL II NN NN NN
CC HHHHHHHHHH MM MM SSSSSS PPPPPPPP LL LL II NN NN NN
CC HH HH MM MM SS PP LL LL II NN NNNN
CC HH HH MM MM SS PP LL LL II NN NNNN
CC HH HH MM MM SS PP LL LL II NN NN
CC HH HH MM MM SS PP LL LL II NN NN
CCCCCCCC HH HH MM MM SSSSSSSS PP LLLLLLLLLL LLLLLLLLLL IIIIII NN NN
CCCCCCCC HH HH MM MM SSSSSSSS PP LLLLLLLLLL LLLLLLLLLL IIIIII NN NN
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
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

```
;  1          0001  0 %TITLE 'EDT$CHMSPLLIN - split line'
;  2          0002  0 MODULE EDT$CHMSPLLIN (                          ! Split line
;  3          0003  0                 IDENT = 'V04-000'                       . File: CHMSPLLIN.BLI Edit: JBS1013
;  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 spiits a line of text at the current cursor position.
; 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 routines EDT$$SPLT_LN  and SPLIT_LINE from
; 46          0046  1 !       module CHANGE.BLI.
; 47          0047  1 ! 1-002 - Regularize headers.  JBS 03-Mar-1981
; 48          0048  1 ! 1-003 - Change SPLIT_LINE to EDT$$SPLT_LNINS .  JBS 27-Mar-1981
; 49          0049  1 ! 1-004 - Use the ASSERT macro.  JBS 01-Jun-1981
; 50          0050  1 ! 1-005 - EDT$$SEL_RNGPOS  has 3 values.  JBS 02-Jun-1981
; 51          0051  1 ! 1-006 - New screen update logic.  JBS 13-Sep-1982
; 52          0052  1 ! 1-007 - Remove EDT$$G_LN_NO for new screen update logic.  JBS 29-Sep-1982
; 53          0053  1 ! 1-008 - Make code dependent on length of inserted line.  SMB 16-Nov-1982
; 54          0054  1 ! 1-009 - Reorganize to fix bugs in line numbering.  SMB 17-Nov-1982
; 55          0055  1 ! 1-010 - Save the end of the old line in heap storage rather than
; 56          0056  1 !         on the stack.  Saving it on the stack uses too much
; 57          0057  1 !         stack on the PDP-11.  JBS 18-Nov-1982
```

```
;    58          0058  1 ! 1-011 - Change the call to EDT$$MRK_LNCHG.  JBS 27-Dec-1982
;    59          0059  1 ! 1-012 - Do things the stupid way if there is a select active, so that
;    60          0060  1 !           the select range will be updated properly.  JBS 22-Feb-1983
;    61          0061  1 ! 1-013 - Set EDT$$A_LN_PTR before inserting the new line, so that the
;    62          0062  1 !           screen data base will get the line inserted in the right place
;    63          0063  1 !           even if the current record is being displayed as two lines.
;    64          0064  1 !           JBS 26-Apr-1983
;    65          0065  1 !--
;    66          0066  1
```

```
;    68          0067  1 %SBTTL 'Declarations'
;    69          0068  1 !
;    70          0069  1 ! TABLE OF CONTENTS:
;    71          0070  1 !
;    72          0071  1
;    73          0072  1 REQUIRE 'EDTSRC:TRAROUN4M';
;    74          0511  1
;    75          0512  1 FORWARD ROUTINE
;    76          0513  1     EDT$$SPLT_LN : NOVALUE,                 ! Split a line of text at the current cursor position
;    77          0514  1     EDT$$SPLT_LNINS : NOVALUE;              ! Split a line of text at the current cursor position
;    78          0515  1
;    79          0516  1 !
;    80          0517  1 ! INCLUDE FILES:
;    81          0518  1 !
;    82          0519  1
;    83          0520  1 REQUIRE 'EDTSRC:EDTREQ';
;    84          0655  1
;    85          0656  1 !
;    86          0657  1 ! MACROS:
;    87          0658  1 !
;    88          0659  1 !       NONE
;    89          0660  1 !
;    90          0661  1 ! EQUATED SYMBOLS:
;    91          0662  1 !
;    92          0663  1 !       NONE
;    93          0664  1 !
;    94          0665  1 ! OWN STORAGE:
;    95          0666  1 !
;    96          0667  1 !       NONE
;    97          0668  1 !
;    98          0669  1 ! EXTERNAL REFERENCES:
;    99          0670  1 !
;   100          0671  1 !       In the routines
```

```
;  102          0672  1 %SBTTL 'EDT$$SPLT_LN  - split line'
;  103          0673  1
;  104          0674  1 GLOBAL ROUTINE EDT$$SPLT_LN                    ! Split line
;  105          0675  1     : NOVALUE =
;  106          0676  1
;  107          0677  1 !++
;  108          0678  1 ! FUNCTIONAL DESCRIPTION:
;  109          0679  1 !
;  110          0680  1 !       Split a line of text at the current cursor position.
;  111          0681  1 !       The characters in front of the cursor become a new line, inserted
;  112          0682  1 !       in front of the current one.  The cursor is left on the original
;  113          0683  1 !       line.
;  114          0684  1 !
;  115          0685  1 ! FORMAL PARAMETERS:
;  116          0686  1 !
;  117          0687  1 !       NONE
;  118          0688  1 !
;  119          0689  1 ! IMPLICIT INPUTS:
;  120          0690  1 !
;  121          0691  1 !       EDT$$T_LN_BUF
;  122          0692  1 !       EDT$$A_LN_END
;  123          0693  1 !       EDT$$A_LN_PTR
;  124          0694  1 !
;  125          0695  1 ! IMPLICIT OUTPUTS:
;  126          0696  1 !
;  127          0697  1 !       EDT$$A_LN_PTR
;  128          0698  1 !
;  129          0699  1 ! ROUTINE VALUE:
;  130          0700  1 !
;  131          0701  1 !       NONE
;  132          0702  1 !
;  133          0703  1 ! SIDE EFFECTS:
;  134          0704  1 !
;  135          0705  1 !       NONE
;  136          0706  1 !
;  137          0707  1 !--
;  138          0708  1
;  139          0709  2     BEGIN
;  140          0710  2
;  141          0711  2     EXTERNAL ROUTINE
;  142          0712  2         EDT$$MRK_LNCHG : NOVALUE,          ! Track line updates
;  143          0713  2         EDT$$UPD_LNLEN : NOVALUE,          ! Update the length of the current line
;  144          0714  2         EDT$$INS_LN : NOVALUE;             ! Insert a line into buffer
;  145          0715  2
;  146          0716  2     EXTERNAL
;  147          0717  2         EDT$$T_LN_BUF,                     ! Current line buffer
;  148          0718  2         EDT$$A_LN_END,                     ! End of current line pointer
;  149          0719  2         EDT$$A_LN_PTR;                     ! Current character pointer
;  150          0720  2
;  151          0721  2     LOCAL
;  152          0722  2         LEN;
;  153          0723  2
;  154          0724  2 !+
;  155          0725  2 ! Get the length of the new line.
;  156          0726  2 !-
;  157          0727  2     LEN = CH$DIFF (.EDT$$A_LN_PTR, CH$PTR (EDT$$T_LN_BUF));
;  158          0728  2 !+
```

```
;  159          0729  2 ! Insert the new line.
;  160          0730  2 !-
;  161          0731  2     EDT$$INS_LN (CH$PTR (EDT$$T_LN_BUF), .LEN);
;  162          0732  2
;  163          0733  2     IF (.LEN NEQ 0)
;  164          0734  2     THEN
;  165          0735  3         BEGIN
;  166          0736  3         EDT$$CPY_MEM (CH$DIFF (.EDT$$A_LN_END, .EDT$$A_LN_PTR), .EDT$$A_LN_PTR, CH$PTR (EDT$$T_LN_BUF));
;  167          0737  3         EDT$$MRK_LNCHG (SCR_EDIT_MODIFY, 0);
;  168          0738  3         EDT$$UPD_LNLEN (-.LEN);
;  169          0739  2         END;
;  170          0740  2
;  171          0741  2     EDT$$A_LN_PTR = CH$PTR (EDT$$T_LN_BUF);
;  172          0742  1     END;                                          ! of routine EDT$$SPLT_LN


                                                                        .TITLE  EDT$CHMSPLLIN EDT$CHMSPLLIN - split line
                                                                        .IDENT  \V04-000\

                                                                        .EXTRN  EDT$$MRK_LNCHG, EDT$$UPD_LNLEN
                                                                        .EXTRN  EDT$$INS_LN, EDT$$T_LN_BUF
                                                                        .EXTRN  EDT$$A_LN_END, EDT$$A_LN_PTR

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

                                                  01FC 00000            .ENTRY  EDT$$SPLT_LN, Save R2,R3,R4,R5,R6,R7,R8       ; 0674
                                58 00000000G   00   9E 00002            MOVAB   EDT$$A_LN_PTR, R8                             ;
                                57 00000000G   00   9E 00009            MOVAB   EDT$$T_LN_BUF, R7                             ;
                                50             67   9E 00010            MOVAB   EDT$$T_LN_BUF, R0                             ;
                             56                68   50 C3 00013            SUBL3   R0, EDT$$A_LN_PTR, LEN                        ; 0727
                                                   56 DD 00017            PUSHL   LEN                                           ; 0731
                                                   57 DD 00019            PUSHL   R7                                            ;
                                00000000G   00     02 FB 0001B            CALLS   #2, EDT$$INS_LN                               ;
                                                   56 D5 00022            TSTL    LEN                                           ; 0733
                                                   23 13 00024            BEQL    1$                                            ;
                                            50     68 D0 00026            MOVL    EDT$$A_LN_PTR, R0                             ; 0736
                             51 00000000G   00     50 C3 00029            SUBL3   R0, EDT$$A_LN_END, R1                         ;
                             67             60     51 28 00031            MOVC3   R1, (R0), EDT$$T_LN_BUF                       ;
                                            7E     01 7D 00035            MOVQ    #1, -(SP)                                     ; 0737
                                00000000G   00     02 FB 00038            CALLS   #2, EDT$$MRK_LNCHG                            ;
                                            7E     56 CE 0003F            MNEGL   LEN, -(SP)                                    ; 0738
                                00000000G   00     01 FB 00042            CALLS   #1, EDT$$UPD_LNLEN                            ;
                                            68     67 9E 00049 1$:        MOVAB   EDT$$T_LN_BUF, EDT$$A_LN_PTR                  ; 0741
                                                   04 0004C               RET                                                   ; 0742

; Routine Size:  77 bytes,    Routine Base:  _EDT$CODE + 0000

;  173          0743  1
```

```
;  175          0744  1 %SBTTL 'EDT$$SPLT_LNINS  - split a line'
;  176          0745  1
;  177          0746  1 GLOBAL ROUTINE EDT$$SPLT_LNINS (                ! Split a line
;  178          0747  1     LENGTH_OPT                                   ! 1 = use length optimizer
;  179          0748  1     ) : NOVALUE =
;  180          0749  1
;  181          0750  1 !++
;  182          0751  1 ! FUNCTIONAL DESCRIPTION:
;  183          0752  1 !
;  184          0753  1 !       Split a line of text at the current cursor position.
;  185          0754  1 !
;  186          0755  1 ! FORMAL PARAMETERS:
;  187          0756  1 !
;  188          0757  1 !       LENGTH_OPT              ! 1 = length is greater than zero so insert
;  189          0758  1 !                               ! after current line
;  190          0759  1 !
;  191          0760  1 ! IMPLICIT INPUTS:
;  192          0761  1 !
;  193          0762  1 !       EDT$$G_LN_LEN
;  194          0763  1 !       EDT$$T_LN_BUF
;  195          0764  1 !       EDT$$A_LN_END
;  196          0765  1 !       EDT$$A_LN_PTR
;  197          0766  1 !       EDT$$A_WK_LN
;  198          0767  1 !       EDT$$Z_EOB_LN
;  199          0768  1 !
;  200          0769  1 ! IMPLICIT OUTPUTS:
;  201          0770  1 !
;  202          0771  1 !       EDT$$G_LN_LEN
;  203          0772  1 !       EDT$$T_LN_BUF
;  204          0773  1 !       EDT$$A_LN_END
;  205          0774  1 !       EDT$$A_LN_PTR
;  206          0775  1 !
;  207          0776  1 ! ROUTINE VALUE:
;  208          0777  1 !
;  209          0778  1 !       NONE
;  210          0779  1 !
;  211          0780  1 ! SIDE EFFECTS:
;  212          0781  1 !
;  213          0782  1 !       Calls EDT$$SPLT_LN
;  214          0783  1 !
;  215          0784  1 !--
;  216          0785  1
;  217          0786  2     BEGIN
;  218          0787  2
;  219          0788  2     EXTERNAL ROUTINE
;  220          0789  2         EDT$$MRK_LNCHG : NOVALUE,           ! Track line updates
;  221          0790  2         EDT$$UPD_LNLEN : NOVALUE,           ! Update the length of the current line
;  222          0791  2         EDT$$INS_LN : NOVALUE,              ! Insert a line into buffer
;  223          0792  2         EDT$$RPL_CHGDLN : NOVALUE,          ! Replace a changed workfile line
;  224          0793  2         EDT$$RD_NXTLN,                      ! Read next work file record
;  225          0794  2         EDT$$RD_PRVLN,                      ! Read previous record
;  226          0795  2         EDT$$START_INS : NOVALUE,           ! Start an insert sequence
;  227          0796  2         EDT$$END_INS : NOVALUE,             ! End an insert sequence
;  228          0797  2         EDT$$ALO_HEAP,                      ! Allocate heap storage
;  229          0798  2         EDT$$DEA_HEAP : NOVALUE;            ! Deallocate heap storage
;  230          0799  2
;  231          0800  2     EXTERNAL
```

```
;  232     0801  2          EDT$$G_LN_LEN,                      ! Length of the current line
;  233     0802  2          EDT$$T_LN_BUF,                      ! Current line buffer
;  234     0803  2          EDT$$A_LN_END,                      ! End of current line pointer
;  235     0804  2          EDT$$A_LN_PTR,                      ! Current character pointer
;  236     0805  2          EDT$$A_WK_LN,                       ! Current work line
;  237     0806  2          EDT$$Z_EOB_LN,                      ! EOB line
;  238     0807  2          EDT$$A_SEL_BUF : REF TBCB_BLOCK;    ! Select buffer, 0 = none
;  239     0808  2
;  240     0809  2      LOCAL
;  241     0810  2          LEN,                                ! New length of old line
;  242     0811  2          ENDLEN,                             ! Ending line length
;  243     0812  2          INSERT_AFTER,                       ! Insert after current record
;  244     0813  2          INSERT_DONE;                        ! 1 = the insert has been done
;  245     0814  2
;  246     0815  2      INSERT_DONE = 0;
;  247     0816  2  !+
;  248     0817  2  ! Don't do the length optinization if we are at [EOB] or if there is an active select.
;  249     0818  2  !-
;  250     0819  2
;  251     0820  3      IF ((.EDT$$A_WK_LN EQLA EDT$$Z_EOB_LN) OR (.EDT$$A_SEL_BUF NEQA 0))
;  252     0821  2      THEN
;  253     0822  2          INSERT_AFTER = 0
;  254     0823  2      ELSE
;  255     0824  2          INSERT_AFTER = .LENGTH_OPT;
;  256     0825  2
;  257     0826  3      IF (.INSERT_AFTER NEQ 0)
;  258     0827  2      THEN
;  259     0828  3          BEGIN
;  260     0829  3          LEN = CH$DIFF (.EDT$$A_LN_PTR, CH$PTR (EDT$$T_LN_BUF));
;  261     0830  3          ENDLEN = CH$DIFF (.EDT$$A_LN_END, .EDT$$A_LN_PTR);
;  262     0831  3  !+
;  263     0832  3  ! Use the length optimizer, which says insert below the current line
;  264     0833  3  ! if we're positioned anywhere except at the beginning of a line.
;  265     0834  3  !-
;  266     0835  3
;  267     0836  4          IF (.LEN EQL 0)
;  268     0837  3          THEN
;  269     0838  4              BEGIN
;  270     0839  4  !+
;  271     0840  4  ! We are at the beginning of the line.  Just insert a 0-length line before this one.
;  272     0841  4  !-
;  273     0842  4              EDT$$START_INS ();                  ! Perform initialization for text insertion
;  274     0843  4              EDT$$INS_LN (CH$PTR (EDT$$T_LN_BUF), 0);
;  275     0844  4              EDT$$END_INS ();                    ! End the insertion sequence
;  276     0845  4              INSERT_DONE = 1;
;  277     0846  4              END
;  278     0847  3          ELSE
;  279     0848  4              BEGIN
;  280     0849  4  !+
;  281     0850  4  ! Adjust the old line and mark for repaint.
;  282     0851  4  !-
;  283     0852  4
;  284     0853  4              LOCAL
;  285     0854  4                  SAV_LN_BUF,                     ! Place to save the end of the old line
;  286     0855  4                  GOT_HEAP;                       ! 1 = we got (or don't need) heap storage
;  287     0856  4
;  288     0857  4  !+
```

```
;  289          0858  4 ! Copy the end of the old line into heap storage, so we can insert it later.
;  290          0859  4 !-
;  291          0860  4
;  292          0861  4             IF (.ENDLEN NEQ 0) THEN GOT_HEAP = EDT$$ALO_HEAP (ENDLEN, SAV_LN_BUF) ELSE GOT_HEAP = 1;
;  293          0862  4
;  294          0863  4             IF .GOT_HEAP
;  295          0864  4             THEN
;  296          0865  5                 BEGIN
;  297          0866  5
;  298          0867  6                 IF (.ENDLEN NEQ 0)
;  299          0868  5                 THEN
;  300          0869  6                     BEGIN
;  301          0870  6                     EDT$$CPY_MEM (.ENDLEN, .EDT$$A_LN_PTR, CH$PTR (.SAV_LN_BUF));
;  302          0871  6 !+
;  303          0872  6 ! Shorten the current line.
;  304          0873  6 !-
;  305          0874  6                     EDT$$UPD_LNLEN (-.ENDLEN);
;  306          0875  6                     EDT$$MRK_LNCHG (SCR_EDIT_MODIFY, .LEN);
;  307          0876  5                     END;
;  308          0877  5
;  309          0878  5                 EDT$$RPL_CHGDLN ();
;  310          0879  5 !+
;  311          0880  5 ! Insert the tail as the new line.
;  312          0881  5 !-
;  313          0882  5
;  314          0883  6                 IF (.ENDLEN NEQ 0)
;  315          0884  5                 THEN
;  316          0885  6                     BEGIN
;  317          0886  6                     EDT$$CPY_MEM (.ENDLEN, CH$PTR (.SAV_LN_BUF), CH$PTR (EDT$$T_LN_BUF));
;  318          0887  6                     EDT$$DEA_HEAP (ENDLEN, SAV_LN_BUF);
;  319          0888  5                     END;
;  320          0889  5
;  321          0890  5                 EDT$$RD_NXTLN ();
;  322          0891  5                 EDT$$A_LN_PTR = CH$PTR (EDT$$T_LN_BUF);
;  323          0892  5                 EDT$$START_INS ();              ! Perform initialization for text insertion
;  324          0893  5 !+
;  325          0894  5 ! Adjust the line pointer to the new line.
;  326          0895  5 !-
;  327          0896  5                 EDT$$INS_LN (CH$PTR (EDT$$T_LN_BUF), .ENDLEN);
;  328          0897  5                 EDT$$G_LN_LEN = .ENDLEN + .LEN;
;  329          0898  5                 EDT$$UPD_LNLEN (-.LEN);
;  330          0899  5                 EDT$$END_INS ();                ! End the insertion sequence
;  331          0900  5                 EDT$$RD_PRVLN ();
;  332          0901  5                 INSERT_DONE = 1;
;  333          0902  4                 END;
;  334          0903  4
;  335          0904  3             END;
;  336          0905  3
;  337          0906  2         END;
;  338          0907  2
;  339          0908  2 !+
;  340          0909  2 ! If we haven't done the insert, perhaps because we have run out of heap storage,
;  341          0910  2 ! do it the stupid way.
;  342          0911  2 !-
;  343          0912  2
;  344          0913  3     IF ( NOT .INSERT_DONE)
;  345          0914  2     THEN
```

```
;  346         0915  3         BEGIN
;  347         0916  3         EDT$$START_INS ();                         ! Perform initialization for text insertion
;  348         0917  3         EDT$$SPLT_LN ();                           ! Split the line
;  349         0918  3         EDT$$END_INS ();                           ! End the insertion sequence
;  350         0919  2         END;
;  351         0920  2
;  352         0921  1     END;                                           ! of routine EDT$$SPLT_LNINS


                                                   .EXTRN  EDT$$RPL_CHGDLN
                                                   .EXTRN  EDT$$RD_NXTLN, EDT$$RD_PRVLN
                                                   .EXTRN  EDT$$START_INS, EDT$$END_INS
                                                   .EXTRN  EDT$$ALO_HEAP, EDT$$DEA_HEAP
                                                   .EXTRN  EDT$$G_LN_LEN, EDT$$A_WK_LN
                                                   .EXTRN  EDT$$Z_EOB_LN, EDT$$A_SEL_BUF

                                 0FFC 00000        .ENTRY  EDT$$SPLT_LNINS, Save R2,R3,R4,R5,R6,R7,R8,-;  0746
                                                           R9,R10,R11                             ;
           5B 00000000G  00   9E 00002             MOVAB   EDT$$A_LN_PTR, R11                     ;
           5A 00000000G  00   9E 00009             MOVAB   EDT$$T_LN_BUF, R10                     ;
           5E            08   C2 00010             SUBL2   #8, SP                                 ;
                         59   D4 00013             CLRL    INSERT_DONE                            ;  0815
           50 00000000G  00   9E 00015             MOVAB   EDT$$Z_EOB_LN, R0                      ;  0820
           50 00000000G  00   D1 0001C             CMPL    EDT$$A_WK_LN, R0                       ;
                         08   13 00023             BEQL    1$                                     ;
              00000000G  00   D5 00025             TSTL    EDT$$A_SEL_BUF                         ;
                         04   13 0002B             BEQL    2$                                     ;
                         50   D4 0002D 1$:         CLRL    INSERT_AFTER                           ;  0822
                         04   11 0002F             BRB     3$                                     ;
           50         04 AC   D0 00031 2$:         MOVL    LENGTH_OPT, INSERT_AFTER               ;  0824
                         03   12 00035 3$:         BNEQ    5$                                     ;  0826
                       00D1   31 00037 4$:         BRW     12$                                    ;
           51            6B   D0 0003A 5$:         MOVL    EDT$$A_LN_PTR, R1                      ;  0829
           50            6A   9E 0003D             MOVAB   EDT$$T_LN_BUF, R0                      ;
     56    51            50   C3 00040             SUBL3   R0, R1, LEN                            ;
04   AE 00000000G  00    51   C3 00044             SUBL3   R1, EDT$$A_LN_END, ENDLEN              ;  0830
                         56   D5 0004D             TSTL    LEN                                    ;  0836
                         1C   12 0004F             BNEQ    6$                                     ;
        00000000G  00    00   FB 00051             CALLS   #0, EDT$$START_INS                     ;  0842
                         7E   D4 00058             CLRL    -(SP)                                  ;  0843
                         5A   DD 0005A             PUSHL   R10                                    ;
        00000000G  00    02   FB 0005C             CALLS   #2, EDT$$INS_LN                        ;
        00000000G  00    00   FB 00063             CALLS   #0, EDT$$END_INS                       ;  0844
                       009B   31 0006A             BRW     11$                                    ;  0845
                      04 AE   D5 0006D 6$:         TSTL    ENDLEN                                 ;  0861
                         0E   13 00070             BEQL    7$                                     ;
                         5E   DD 00072             PUSHL   SP                                     ;
                      08 AE   9F 00074             PUSHAB  ENDLEN                                 ;
        00000000G  00    02   FB 00077             CALLS   #2, EDT$$ALO_HEAP                      ;
                         03   11 0007E             BRB     8$                                     ;
           50            01   D0 00080 7$:         MOVL    #1, GOT_HEAP                           ;
           B1            50   E9 00083 8$:         BLBC    GOT_HEAP, 4$                           ;  0863
           57         04 AE   D0 00086             MOVL    ENDLEN, R7                             ;  0867
                         58   D4 0008A             CLRL    R8                                     ;
                         57   D5 0008C             TSTL    R7                                     ;
                         1F   13 0008E             BEQL    9$                                     ;
```

```
                                   58  D6 00090          INCL    R8                                   ;
                         50        6B  D0 00092          MOVL    EDT$$A_LN_PTR, R0                    ; 0870
         00    BE        60        57  28 00095          MOVC3   R7, (R0), @SAV_LN_BUF                ;
                         7E        57  CE 0009A          MNEGL   R7, -(SP)                            ; 0874
               00000000G 00        01  FB 0009D          CALLS   #1, EDT$$UPD_LNLEN                   ;
                                   56  DD 000A4          PUSHL   LEN                                  ; 0875
                                   01  DD 000A6          PUSHL   #1                                   ;
               00000000G 00        02  FB 000A8          CALLS   #2, EDT$$MRK_LNCHG                   ;
               000  ^00G 00        00  FB 000AF 9$:      CALLS   #0, EDT$$RPL_CHGDLN                  ; 0878
                         11        58  E9 000B6          BLBC    R8, 10$                              ; 0883
           6A       00   BE        57  28 000B9          MOVC3   R7, @SAV_LN_BUF, EDT$$T_LN_BUF       ; 0886
                                   5E  DD 000BE          PUSHL   SP                                   ; 0887
                              08   AE  9F 000C0          PUSHAB  ENDLEN                               ;
               00000000G 00        02  FB 000C3          CALLS   #2, EDT$$DEA_HEAP                    ;
               00000000G 00        00  FB 000CA 10$:     CALLS   #0, EDT$$RD_NXTLN                    ; 0890
                         6B        6A  9E 000D1          MOVAB   EDT$$T_LN_BUF, EDT$$A_LN_PTR         ; 0891
               00000000G 00        00  FB 000D4          CALLS   #0, EDT$$START_INS                   ; 0892
                              04   AE  DD 000DB          PUSHL   ENDLEN                               ; 0896
                                   5A  DD 000DE          PUSHL   R10                                  ;
               00000000G 00        02  FB 000E0          CALLS   #2, EDT$$INS_LN                      ;
               00000000G 00   04 BE46  9E 000E7          MOVAB   @ENDLEN[LEN], EDT$$G_LN_LEN          ; 0897
                         7E        56  CE 000F0          MNEGL   LEN, -(SP)                           ; 0898
               00000000G 00        01  FB 000F3          CALLS   #1, EDT$$UPD_LNLEN                   ;
               00000000G 00        00  FB 000FA          CALLS   #0, EDT$$END_INS                     ; 0899
               00000000G 00        00  FB 00101          CALLS   #0, EDT$$RD_PRVLN                    ; 0900
                         59        01  D0 00108 11$:     MOVL    #1, INSERT_DONE                      ; 0901
                         13        59  E8 0010B 12$:     BLBS    INSERT_DONE, 13$                     ; 0913
               00000000G 00        00  FB 0010E          CALLS   #0, EDT$$START_INS                   ; 0916
                    FE99 CF        00  FB 00115          CALLS   #0, EDT$$SPLT_LN                     ; 0917
               00000000G 00        00  FB 0011A          CALLS   #0, EDT$$END_INS                     ; 0918
                                       04 00121 13$:     RET                                          ; 0921

; Routine Size:  290 bytes,    Routine Base:  _EDT$CODE + 004D


;  353          0922  1
;  354          0923  1 !<BLF/PAGE>
```

```
;  356        0924  1 END                                   ! of module EDT$CHMSPLLIN
;  357        0925  1
;  358        0926  0 ELUDOM
```

```
;                              PSECT SUMMARY
;
;       Name                         Bytes                        Attributes
;
;    _EDT$CODE                          367  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        25         6         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$:CHMSPLLIN/OBJ=OBJ$:CHMSPLLIN MSRC$:CHMSPLLIN.BLI/UPDATE=(ENH$:C
;       HMSPLLIN)

; Size:          367 code + 0 data bytes
; Run Time:         00:18.5
; Elapsed Time:     00:21.6
; Lines/CPU Min:     3008
; Lexemes/CPU-Min:  8881
; Memory Used:  110 pages
; Compilation Complete
```

CHMMRKCHG LIS

CHMPARSEN LIS

CHMSELPOS LIS

CHMSPLLIN LIS

DATA LIS

CHMPAREN LIS

CHMUNDEL LIS

COMMAND LIS

CHMONSTR LIS

CHMSAVPOS LIS

CHMSCHSTR LIS

CHMREPOS LIS

CHMMESS LIS

CHMPASTE LIS

CHMSENDEL LIS

CHMTADJ LIS

CHMSAVTXT LIS

CLRKEY LIS

CHMNEWLEN LIS

CHMPARSE LIS

CHMSAVLIN LIS

CHMSUBS LIS