C 10

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

```
EXE

Mod
---
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
SYS
LBI
LII
```

```
**FILE**ID**WFINSLIN
```

```
WW      WW  FFFFFFFFFF    IIIIII    NN      NN    SSSSSSSS  LL            IIIIII    NN      NN
WW      WW  FFFFFFFFFF    IIIIII    NN      NN    SSSSSSSS  LL            IIIIII    NN      NN
WW      WW  FF              II      NN      NN  SS          LL              II      NN      NN
WW      WW  FF              II      NN      NN  SS          LL              II      NN      NN
WW      WW  FF              II      NNNN    NN  SS          LL              II      NNNN    NN
WW      WW  FF              II      NNNN    NN  SS          LL              II      NNNN    NN
WW      WW  FFFFFFFF        II      NN  NN  NN    SSSSSS    LL              II      NN  NN  NN
WW      WW  FFFFFFFF        II      NN  NN  NN    SSSSSS    LL              II      NN  NN  NN
WW  WW  WW  FF              II      NN    NNNN          SS  LL              II      NN    NNNN
WW  WW  WW  FF              II      NN    NNNN          SS  LL              II      NN    NNNN
WWWW  WWWW  FF              II      NN      NN          SS  LL              II      NN      NN      ....
WWWW  WWWW  FF              II      NN      NN          SS  LL              II      NN      NN      ....
WW      WW  FF            IIIIII    NN      NN  SSSSSSSS    LLLLLLLLLL    IIIIII    NN      NN      ....
WW      WW  FF            IIIIII    NN      NN  SSSSSSSS    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
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 %TITLE 'EDT$WFINSLIN - insert a new line'
0 MODULE EDT$WFINSLIN (                              ! ınsert a new line
0               IDENT = 'V04-000'                                ! File: WFINSLIN.BLI Edit: JBS1017
0               ) =
1 BEGIN
1 !
1 !****************************************************************' '*******
1 !*                                                                         *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                *
1 !*  DIuITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
1 !*  ALL RIGHTS RESERVED.                                                   *
1 !*                                                                         *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
1 !*  TRANSFERRED.                                                           *
1 !*                                                                         *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGF WITHOUT NOTICE  *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT P' u.GITAL EQUIPMENT  *
1 !*  CORPORATION.                                                           *
1 !*                                                                         *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
1 !*                                                                         *
1 !*                                                                         *
1 !***************************************************************************
1 !
1
1 !++
1 ! FACILITY:     EDT -- The DEC Standard Editor
1 !
1 ! ABSTRACT:
1 !
1 !       Insert a new line into the current text buffer.
1 !
1 ! ENVIRONMENT:  Runs at any access mode - AST reentrant
1 !
1 ! AUTHOR: Bob Kushlis, CREATION DATE: October 16, 1978
1 !
1 ! MODIFIED BY:
1 !
1 ! 1-001 - Original.  DJS 23-Feb-1981.  This module was created by
1 !       extracting routine EDT$$INS_LN  from module EDTWF.
1 ! 1-002 - Regularize headers.  JBS 17-Mar-1981
1 ! 1-003 - Change index for line numbers from 10 to 15.  SMB 18-Jan-1982
1 ! 1-004 - Add line number error checks and return value.  SMB 3-Feb-1982
1 ! 1-005 - Handle error return from splitting a bucket.  JBS 09-Jun-1982
1 ! 1-006 - New screen update logic.  JBS 13-Sep-1982
1 ! 1-007 - Remove EDT$$SET_WKLN.  JBS 14-Sep-1982
1 ! 1-008 - Change the call to insert lines.  SMB 21-Sep-1982
1 ! 1-009 - Modify to use new 48 bit macros. STS 01-Oct-1982
1 ! 1-010 - Modify to use new compare macro. 20-Oct-1982
1 ! 1-011 - Count inserted lines, and stop maintaining the screen data
1 !          base if we have too many.  JBS 21-Oct-1982
1 ! 1-012 - The counter also counts deleted lines.  JBS 27-Oct-1982
```

```
;    58          0058  1 ! 1-013 - Add parameters to SC_INSLN.  SMB 02-Dec-1982
;    59          0059  1 ! 1-014 - Improve code, add consistency checking.  JBS 28-Dec-1982
;    60          0060  1 ! 1-015 - Up the number of recs that can be inserted before SCR_REBUILD is set.  SMB 16-Feb-1983
;    61          0061  1 ! 1-016 - Be more defensive with select position updating.  JBS 28-Feb-1983
;    62          0062  1 ! 1-017 - Watch out for negative record lengths.  JBS 11-Jul-1983
;    63          0063  1 !--
;    64          0064  1
;     66          0065  1 %SBTTL 'Declarations'
;     67          0066  1 !
;     68          0067  1 ! TABLE OF CONTENTS:
;     69          0068  1 !
;     70          0069  1
;     71          0070  1 REQUIRE 'EDTSRC:TRAROUNAM';
;     72          0509  1
;     73          0510  1 FORWARD ROUTINE
;     74          0511  1     EDT$$INS_LN;
;     75          0512  1
;     76          0513  1 !
;     77          0514  1 ! INCLUDE FILES:
;     78          0515  1 !
;     79          0516  1
;     80          0517  1 REQUIRE 'EDTSRC:EDTREQ';
;     81          0652  1
;     82          0653  1 !
;     83          0654  1 ! MACROS:
;     84          0655  1 !
;     85          0656  1 !       NONE
;     86          0657  1 !
;     87          0658  1 ! EQUATED SYMBOLS:
;     88          0659  1 !
;     89          0660  1 !       NONE
;     90          0661  1 !
;     91          0662  1 ! OWN STORAGE:
;     92          0663  1 !
;     93          0664  1 !       NONE
;     94          0665  1 !
;     95          0666  1 ! EXTERNAL REFERENCES:
;     96          0667  1 !
;     97          0668  1 !       In the routine
;   99          0669  1 %SBTTL 'EDT$$INS_LN  - insert a new line'
;  100          0670  1
;  101          0671  1 GLOBAL ROUTINE EDT$$INS_LN (                        ! Insert a new line
;  102          0672  1     REC_ADDR,                                       ! Address of new line
;  103          0673  1     REC_LEN                                         ! Length of new line
;  104          0674  1     ) =
;  105          0675  1
;  106          0676  1 !++
;  107          0677  1 ! FUNCTIONAL DESCRIPTION:
;  108          0678  1 !
;  109          0679  1 !       Insert a new line in the text buffer.  If there is enough space in the
;  110          0680  1 !       current bucket, text is moved around to make a hole and the new line is
;  111          0681  1 !       inserted, otherwise, the bucket is split into two parts.
;  112          0682  1 !
;  113          0683  1 ! FORMAL PARAMETERS:
;  114          0684  1 !
;  115          0685  1 !  REC_ADDR             the address of the new line
;  116          0686  1 !
;  117          0687  1 !  REC_LEN              the length of the new line
;  118          0688  1 !
;  119          0689  1 ! IMPLICIT INPUTS:
;  120          0690  1 !
;  121          0691  1 !       EDT$$A_CUR_BUF
;  122          0692  1 !       EDT$$A_WK_BUK
;  123          0693  1 !       EDT$$G_WK_CURBUK
;  124          0694  1 !       EDT$$L_WK_INSCNT
;  125          0695  1 !       EDT$$A_WK_LN
;  126          0696  1 !       EDT$$L_LNO0
;  127          0697  1 !       EDT$$L_SEL_LN
;  128          0698  1 !       EDT$$A_LN_PTR
;  129          0699  1 !       EDT$$G_RECS_INSERTED
;  130          0700  1 !       EDT$$G_SCR_LNS
;  131          0701  1 !
;  132          0702  1 ! IMPLICIT OUTPUTS:
;  133          0703  1 !
;  134          0704  1 !       EDT$$A_CUR_BUF
;  135          0705  1 !       EDT$$A_WK_LN
;  136          0706  1 !       EDT$$A_WK_BUK
;  137          0707  1 !       EDT$$G_WK_MODFD
;  138          0708  1 !       EDT$$L_WK_INSCNT
;  139          0709  1 !       EDT$$A_SEL_POS
;  140          0710  1 !       EDT$$G_RECS_INSERTED
;  141          0711  1 !       EDT$$G_SCR_REBUILD
;  142          0712  1 !
;  143          0713  1 ! ROUTINE VALUE:
;  144          0714  1 !
;  145          0715  1 !       0 = insertion did not occur , 1 = successful insertion
;  146          0716  1 !
;  147          0717  1 ! SIDE EFFECTS:
;  148          0718  1 !
;  149          0719  1 !       NONE
;  150          0720  1 !
;  151          0721  1 !--
;  152          0722  1
;  153          0723  2     BEGIN
;  154          0724  2
;  155          0725  2     EXTERNAL ROUTINE
```

```
2           EDT$$FMT MSG,
2           EDT$$WF_MAKECUR : NOVALUE,
2           EDT$$WF_SPLTBUK,
2           EDT$$SEL_RNGPOS,
2           EDT$$SC_INSLN : NOVALUE;
2
2       EXTERNAL
2           EDT$$L_MAX_LINES : LN_BLOCK,          ! Maximum lines EDT can handle
2           EDT$$A_CUR_BUF : REF TBCB_BLOCK,      ! Current text buffer control block
2           EDT$$A_WK_BUK :                       ! Pointer to current bucket
2               REF BLOCK [WF_BUKT_SIZE, BYTE] FIELD (WFB_FIELDS),
2           EDT$$G_WK_CURBUK,                     ! Number of the current bucket
2           EDT$$L_WK_INSCNT : LN_BLOCK,          ! The count of inserted lines
2           EDT$$A_WK_LN : REF LIN_BLOCK,         ! Pointer to current line
2           EDT$$G_WK_MODFD,                      ! Flag indicating bucket was modified
2           EDT$$L_LNO0 : LNOVECTOR [14],
2           EDT$$A_SEL_POS,                       ! Select position
2           EDT$$A_LN_PTR,                        ! Pointer into line buffer
2           EDT$$T_LN_BUF,                        ! The line buffer
2           EDT$$L_SEL_LN,                        ! Select line
2           EDT$$G_SCR_LNS,                       ! The number of text lines on the screen
2           EDT$$G_RECS_INSERTED,   ! Number of records inserted and deleted since the last screen update
2           EDT$$G_SCR_REBUILD;             ! 1 = the screen data structures must be rebuilt from the work file
2
2       MESSAGES ((MAXLINVAL, INSMEM));
2
2       LOCAL
2           OLD_NEXT,
2           INS_LEN;
2
2       ASSERT (.REC_LEN GEQ 0);
2 !+
2 ! Make sure we aren't exceeding the maximum number of lines allowed.
2 !-
2
3       IF (CMPLNO (EDT$$A_CUR_BUF [TBCB_LINE_COUNT], EDT$$L_MAX_LINES) GEQ 0)
2       THEN
3           BEGIN
3           EDT$$FMT_MSG (EDT$_MAXLINVAL);
3           RETURN (0);
3           END;
2
2 !+
2 ! Fix up select range if necessary.
2 !-
2
2       CASE EDT$$SEL_RNGPOS () FROM -1 TO 1 OF
2           SET
2
2           [0] :                                ! Current line is select line
2
3               IF CH$PTR_GTR (.EDT$$A_SEL_POS, .EDT$$A_LN_PTR)
2               THEN
3                   BEGIN
3                   EDT$$A_SEL_POS = MINA (EDT$$T_LN_BUF, CH$PLUS (.EDT$$A_SEL_POS, -.REC_LEN));
3                   ADDLINE (NUMBER_ONE, EDT$$L_SEL_LN);
2                   END;
```

```
2
2           [1] :                                   ! Current line is before select line
2               ADDLINE (NUMBER_ONE, EDT$$L_SEL_LN);
2
2           [-1] :                                  ! Current line is after select line, or no select
3               BEGIN
3               0
2               END;
2
2           [OUTRANGE] :
2               ASSERT (0);
2           TES;
2
2   !+
2   ! If we are not going to rebuild the screen data base from the work file,
2   ! tell the screen updater that we have inserted a line.
2   !-
2
3       IF ( NOT .EDT$$G_SCR_REBUILD)
2       THEN
3           BEGIN
3           EDT$$G_RECS_INSERTED = .EDT$$G_RECS_INSERTED + 1;
3
4           IF (.EDT$$G_RECS_INSERTED GTR (.EDT$$G_SCR_LNS*2))
3           THEN
3               EDT$$G_SCR_REBUILD = 1
3           ELSE
3               EDT$$SC_INSLN (.REC_ADDR, .REC_LEN);
3
2           END;
2
2   !+
2   ! Update the various counters for this text buffer.
2   !-
2       ADDLINE (NUMBER_ONE, EDT$$A_CUR_BUF [TBCB_LINE_COUNT]);
2       EDT$$A_CUR_BUF [TBCB_CHAR_COUNT] = .EDT$$A_CUR_BUF [TBCB_CHAR_COUNT] + .REC_LEN;
2       ADDLINE (NUMBER_ONE, EDT$$A_CUR_BUF [TBCB_CUR_LIN]);
2       EDT$$A_WK_LN = CH$PTR (.EDT$$A_WK_BUK, .EDT$$A_CUR_BUF [TBCB_LINE_ADDR]);
2   !+
2   ! Compute the length of the line to be inserted, including the line number information.
2   !-
2       INS_LEN = .REC_LEN + LIN_FIXED_SIZE + 1;
2   !+
2   ! Will it fit in this bucket?
2   !-
2
3       IF ((.EDT$$A_WK_BUK [WFB_END] + .INS_LEN) GTRU WF_BUKT_SIZE)
2       THEN
2   !+
2   ! If we are at the beginning of a bucket and it is not the
2   ! first bucket, then check out the previous bucket.
2   !-
2
3           IF ((.EDT$$A_CUR_BUF [TBCB_LINE_ADDR] EQL WFB_FIXED_SIZE) AND   !
3               (.EDT$$A_WK_BUK [WFB_PREV_BUKT] NEQ 0))
2           THEN
3               BEGIN
```

```
   3 !+
   3 ! Read the previous bucket and position to it's end.
   2 !-
   3             EDT$$WF_MAKECUR (.EDT$$A_WK_BUK [WFB_PREV_BUKT]);
   3             EDT$$A_CUR_BUF [TBCB_LINE_ADDR] = .EDT$$A_WK_BUK [WFB_END];
   3             EDT$$A_CUR_BUF [TBCB_CUR_BUKT] = .EDT$$G_WK_CURBUK;
   2             END;
   2
   2 !+
   2 ! If it still doesn't fit, then split the bucket.  Note that
   2 ! the while loop is here because it may not fit after the first
   2 ! split.  In this case the second split is guaranteed to create
   2 ! a new bucket and the line must fit.
   2 !-
   2
   2     WHILE ((.EDT$$A_WK_BUK [WFB_END] + .INS_LEN) GTRU WF_BUKT_SIZE) DO
   3         BEGIN
   3
   4         IF ( NOT EDT$$WF_SPLTBUK ())
   3         THEN
   4             BEGIN
   4             EDT$$FMT_MSG (EDT$_INSMEM);
   4             RETURN (0);
   3             END;
   3
   2         END;
   2
   2 !+
   2 ! Make a hole for the line to be inserted.
   2 !-
   2     EDT$$A_WK_LN = CH$PTR (.EDT$$A_WK_BUK, .EDT$$A_CUR_BUF [TBCB_LINE_ADDR]);
P  2     EDT$$CPY_MEM (.EDT$$A_WK_BUK [WFB_END] - .EDT$$A_CUR_BUF [TBCB_LINE_ADDR],  !
P  2         .EDT$$A_WK_LN,                                                          !
   2         CH$PLUS (.EDT$$A_WK_LN, .INS_LEN));
   2 !+
   2 ! Update the end of bucket field to reflect new size.
   2 !-
   2     EDT$$A_WK_BUK [WFB_END] = .EDT$$A_WK_BUK [WFB_END] + .INS_LEN;
   2 !+
   2 ! And move the line into the bucket buffer.
   2 !-
   2     EDT$$A_WK_LN [LIN_LENGTH] = .REC_LEN;
   2     CH$WCHAR (.REC_LEN, EDT$$CPY_MEM (.REC_LEN, .REC_ADDR, EDT$$A_WK_LN [LIN_TEXT]));
   2 !+
   2 ! Update the record pointer
   2 !-
   2     EDT$$A_CUR_BUF [TBCB_LINE_ADDR] = .EDT$$A_CUR_BUF [TBCB_LINE_ADDR] + .INS_LEN;
   2 !+
   2 ! Set the "modified" flag, and bump the count of inserted lines.
   2 !-
   2     EDT$$G_WK_MODFD = 1;
   2     ADDLINE (NUMBER_ONE, EDT$$L_WK_INSCNT);
   2 !+
   2 ! Now make sure we are positioned correctly.
   2 !-
   2
   3     IF (.EDT$$A_CUR_BUF [TBCB_LINE_ADDR] GEQ .EDT$$A_WK_BUK [WFB_END])
```

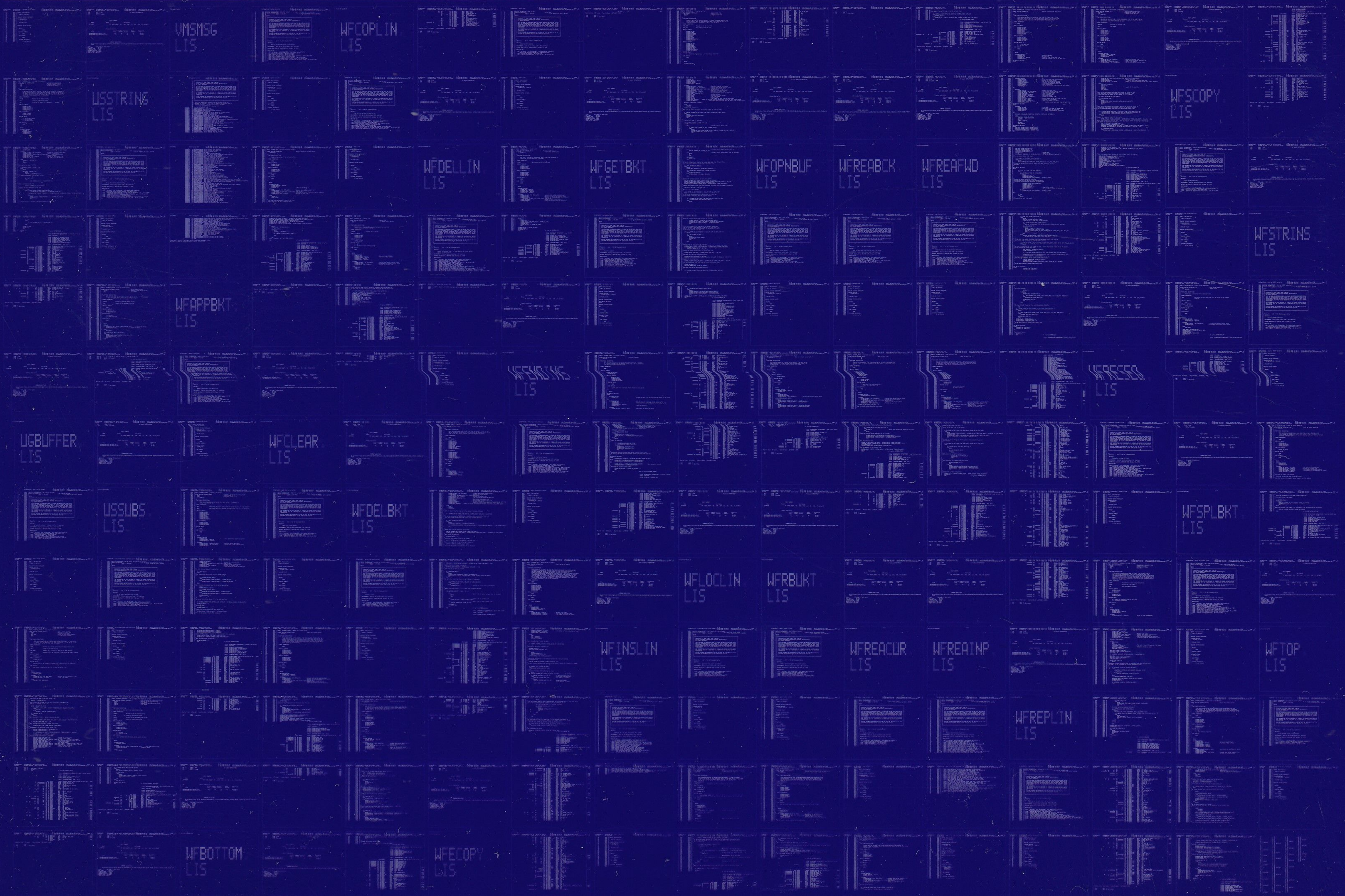
```
;  327          0897  2      THEN
;  328          0898  2
;  329          0899  3          IF (.EDT$$A_WK_BUK [WFB_NEXT_BUKT] NEQ 0)
;  330          0900  2          THEN
;  331          0901  3              BEGIN
;  332          0902  3              EDT$$WF_MAKECUR (.EDT$$A_WK_BUK [WFB_NEXT_BUKT]);
;  333          0903  3              EDT$$A_CUR_BUF [TBCB_LINE_ADDR] = WFB_FIXED_SIZE;
;  334          0904  3              EDT$$A_CUR_BUF [TBCB_CUR_BUKT] = .EDT$$G_WK_CURBUK;
;  335          0905  2              END;
;  336          0906  2
;  337          0907  2      EDT$$A_WK_LN = CH$PTR (.EDT$$A_WK_BUK, .EDT$$A_CUR_BUF [TBCB_LINE_ADDR]);
;  338          0908  3      RETURN (1)
,  339          0909  1      END;                                          ! of routine EDT$$INS_LN


                                                            .TITLE  EDT$WFINSLIN EDT$WFINSLIN - insert a new line
                                                            .IDENT  \V04-000\

                                                            .EXTRN  EDT$$FMT_MSG, EDT$$WF_MAKECUR
                                                            .EXTRN  EDT$$WF_SPLTBUK
                                                            .EXTRN  EDT$$SEL_RNGPOS
                                                            .EXTRN  EDT$$SC_INSLN, EDT$$L_MAX_LINES
                                                            .EXTRN  EDT$$A_CUR_BUF, EDT$$A_WK_BUK
                                                            .EXTRN  EDT$$G_WK_CURBUK
                                                            .EXTRN  EDT$$L_WK_INSCNT
                                                            .EXTRN  EDT$$A_WK_LN, EDT$$G_WK_MODFD
                                                            .EXTRN  EDT$$L_LNO0, EDT$$A_SEL_POS
                                                            .EXTRN  EDT$$A_LN_PTR, EDT$$T_LN_BUF
                                                            .EXTRN  EDT$$L_SEL_LN, EDT$$G_SCR_LNS
                                                            .EXTRN  EDT$$G_RECS_INSERTED
                                                            .EXTRN  EDT$$G_SCR_REBUILD
                                                            .EXTRN  EDT$_MAXLINVAL, EDT$_INSMEM
                                                            .EXTRN  EDT$$INTER_ERR

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

                                   0FFC 00000               .ENTRY  EDT$$INS_LN, Save R2,R3,R4,R5,R6,R7,R8,R9,- ; 0671
                                                                    R10,R11                                      ;
                     5B 00000000G   00   9E 00002           MOVAB   EDT$$A_CUR_BUF, R11                          ;
                     5A        08   AC   D0 00009           MOVL    REC_LEN, R10                                 ; 0756
                                    07   18 0000D           BGEQ    1$                                           ;
          00000000G  00             00   FB 0000F           CALLS   #0, EDT$$INTER_ERR                           ;
   50                6B             18   C1 00016 1$:       ADDL3   #24, EDT$$A_CUR_BUF, R0                      ; 0761
                     51 00000000G   00   3C 0001A           MOVZWL  HIGH_2, R1                                   ;
                     51        04   A0   B1 00021           CMPW    4(R0), R1                                    ;
                                    0E   1F 00025           BLSSU   2$                                           ;
                                    17   12 00027           BNEQ    4$                                           ;
                     51 00000000G   00   D0 00029           MOVL    LOW_2, R1                                    ;
                     51             60   D1 00030           CMPL    (R0), R1                                     ;
                                    05   1E 00033           BGEQU   3$                                           ;
                     50             01   CE 00035 2$:       MNEGL   #1, R0                                       ;
                                    09   11 00038           BRB     5$                                           ;
                                    04   12 0003A 3$:       BNEQ    4$                                           ;
                                    50   D4 0003C           CLRL    R0                                           ;
                                    03   11 0003E           BRB     5$                                           ;
                     50             01   D0 00040 4$:       MOVL    #1, R0                                       ;
                                    09   19 00043 5$:       BLSS    6$                                           ;
```

```
                                          00000000G   8F  DD 00045         PUSHL   #EDT$_MAXLINVAL                              ; 0764
                                                    010E  31 0004B         BRW     17$                                          ;
                            00000000G  00             00  FB 0004E 6$:     CALLS   #0, EDT$$SEL_RNGPOS                          ; 0772
                         02 FFFFFFFF   8F             50  CF 00055         CASEL   R0, #-1, #2                                  ;
                       0038          000F           0046     0005D 7$:     .WORD   11$-7$,-                                     ;
                                                                                   8$-7$,-                                      ;
                                                                                   10$-7$                                       ;
                            00000000G  00             00  FB 00063         CALLS   #0, EDT$$INTER_ERR                           ; 0793
                                                      37  11 0006A         BRB     11$                                          ; 0772
                                       50 00000000G   00  D0 0006C 8$:     MOVL    EDT$$A_SEL_POS, R0                           ; 0777
                            00000000G  00             50  D1 00073         CMPL    R0, EDT$$A_LN_PTR                            ;
                                                      27  1B 0007A         BLEQU   11$                                          ;
                                       50             5A  C2 0007C         SUBL2   R10, R0                                      ; 0780
                                       51 00000000G   00  9E 0007F         MOVAB   EDT$$T_LN_B'F, R1                            ;
                                       51             50  D1 00086         CMPL    R0, R1                                       ;
                                                      03  1B 00089         BLEQU   9$                                           ;
                                       50             51  D0 0008B         MOVL    R1, R0                                       ;
                            00000000G  00             50  D0 0008E 9$:     MOVL    R0, EDT$$A_SEL_POS                           ;
                                          00000000G   00  D6 00095 10$:    INCL    FIRST_LWORD                                  ; 0785
                                                      06  12 0009B         BNEQ    11$                                          ;
                                          00000000G   00  B6 0009D         INCW    NEXT_WORD                                    ;
                                       2C 00000000G   00  E8 000A3 11$:    BLBS    EDT$$G_SCR_REBUILD, 13$                      ; 0801
                                          00000000G   00  D6 000AA         INCL    EDT$$G_RECS_INSERTED                         ; 0804
                         50 00000000G  00             01  78 000B0         ASHL    #1, EDT$$G_SCR_LNS, R0                       ; 0806
                                       50 00000000G   00  D1 000B8         CMPL    EDT$$G_RECS_INSERTED, R0                     ;
                                                      09  15 000BF         BLEQ    12$                                          ;
                            00000000G  00             01  D0 000C1         MOVL    #1, EDT$$G_SCR_REBUILD                       ; 0808
                                                      0C  11 000C8         BRB     13$                                          ;
                                                      5A  DD 000CA 12$:    PUSHL   R10                                          ; 0810
                                                 04   AC  DD 000CC         PUSHL   REC_ADDR                                     ;
                            00000000G  00             02  FB 000CF         CALLS   #2, EDT$$SC_INSLN                            ;
                                       50             6B  D0 000D6 13$:    MOVL    EDT$$A_CUR_BUF, R0                           ; 0817
                                                 18   A0  D6 000D9         INCL    24(R0)                                       ;
                                                      03  12 000DC         BNEQ    14$                                          ;
                                                 1C   A0  B6 000DE         INCW    28(R0)                                       ;
                                  1E   A0             5A  C0 000E1 14$:    ADDL2   R10, 30(R0)                                  ; 0818
                                                 06   A0  D6 000E5         INCL    6(R0)                                        ; 0819
                                                      03  12 000E8         BNEQ    15$                                          ;
                                                 0A   A0  B6 000EA         INCW    10(R0)                                       ;
                                       51 00000000G   00  D0 000ED 15$:    MOVL    EDT$$A_WK_BUK, R1                            ; 0820
              00000000G  00            51             60  C1 000F4         ADDL3   (R0), R1, EDT$$A_WK_LN                       ;
                                       56        08   AA  9E 000FC         MOVAB   8(R10), INS_LEN                              ; 0824
                         52            56        04   A1  C1 00100         ADDL3   4(R1), INS_LEN, R2                           ; 0829
                            00000200   8F             52  D1 00105         CMPL    R2, #512                                     ;
                                                      29  1B 0010C         BLEQU   16$                                          ;
                                       08             60  D1 0010E         CMPL    (R0), #8                                     ; 0836
                                                      24  12 00111         BNEQ    16$                                          ;
                                                      61  B5 00113         TSTW    (R1)                                         ; 0837
                                                      20  13 00115         BEQL    16$                                          ;
                                       7E             61  3C 00117         MOVZWL  (R1), -(SP)                                  ; 0843
                            00000000G  00             01  FB 0011A         CALLS   #1, EDT$$WF_MAKECUR                          ;
                                       51             6B  D0 00121         MOVL    EDT$$A_CUR_BUF, R1                           ; 0844
                                       50 00000000G   00  D0 00124         MOVL    EDT$$A_WK_BUK, R0                            ;
                                       61        04   A0  D0 0012B         MOVL    4(R0), (R1)                                  ;
                                  04   A1 00000000G   00  B0 0012F         MOVW    EDT$$G_WK_CURBUK, 4(R1)                      ; 0845
                                       50 00000000G   00  D0 00137 16$:    MOVL    EDT$$A_WK_BUK, R0                            ; 0855
                         50            56        04   A0  C1 0013E         ADDL3   4(R0), INS_LEN, R0                           ;
```

```
                     00000200  8F             50 D1 00143        CMPL    R0, #512                            ;
                                              1; 1B 0014A        BLEQU   18$                                 ;
                     00000000G 00             00 FB 0014C        CALLS   #0, EDT$$WF_SPLTBUK                 ; 0858
                               E1             50 E8 00153        BLBS    R0, 16$                             ;
                                    00000000G 8F DD 00156        PUSHL   #EDT$_INSMEM                        ; 0861
                     00000000G 00             01 FB 0015C 17$:   CALLS   #1, EDT$$FMT_MSG                    ;
                                            0082 31 00163        BRW     21$                                 ; 0862
                               57 00000000G   00 D0 00166 18$:   MOVL    EDT$$A_WK_BUK, R7                   ; 0870
                               59             6B D0 0016D        MOVL    EDT$$A_CUR_BUF, R9                  ;
            000000C0G 00       57             69 C1 00170        ADDL3   (R9), R7, EDT$$A_WK_LN              ;
                   50      04  A7             69 C3 00178        SUBL3   (R9), 4(R7), R0                     ; 0873
                               58 00000000G   00 D0 0017D        MOVL    EDT$$A_WK_LN, R8                    ;
                 6648          68             50 28 00184        MOVC3   R0, (R8), (INS_LEN)[R8]             ;
                           04  A7             56 C0 00189        ADDL2   INS_LEN, 4(R7)                      ; 0877
                               68             5A 90 0018D        MOVB    R10, (R8)                           ; 0881
               07   A8     04  BC             5A 28 00190        MOVC3   R10, @REL_ADDR, 7(R8)               ; 0882
                               63             5A 90 00196        MOVB    R10, (R3)                           ;
                               69             56 C0 00199        ADDL2   INS_LEN, (R9)                       ; 0886
                     00000000G 00             01 D0 0019C        MOVL    #1, EDT$$G_WK_MODFD                 ; 0890
                                    00000000G 00 D6 001A3        INCL    FIRST_LWORD                         ; 0891
                                              06 12 001A9        BNEQ    19$                                 ;
                                    00000000G 00 B6 001AB        INCW    NEXT_WORD                           ;
                           04  A7             69 D1 001B1 19$:   CMPL    (R9), 4(R7)                         ; 0896
                                              1E 19 001B5        BLSS    20$                                 ;
                                         02   A7 B5 001B7        TSTW    2(R7)                               ; 0899
                                              19 13 001BA        BEQL    20$                                 ;
                               7E        02   A7 3C 001BC        MOVZWL  2(R7), -(SP)                        ; 0902
                     00000000G 00             01 FB 001C0        CALLS   #1, EDT$$WF_MAKECUR                 ;
                               50             6B D0 001C7        MOVL    EDT$$A_CUR_BUF, R0                  ; 0903
                               60             08 D0 001CA        MOVL    #8, (R0)                            ;
                           04  A0 00000000G   00 B0 001CD        MOVW    EDT$$G_WK_CURBUK, 4(R0)             ; 0904
                               50             6B D0 001D5 20$:   MOVL    EDT$$A_CUR_BUF, R0                  ; 0907
            00000000G 00 00000000G 00         60 C1 001D8        ADDL3   (R0), EDT$$A_WK_BUK, EDT$$A_WK_LN   ;
                               50             01 D0 001E4        MOVL    #1, R0                              ; 0908
                                                 04 001E7        RET                                         ;
                                              50 D4 001E8 21$:   CLRL    R0                                  ; 0909
                                                 04 001EA        RET                                         ;

; Routine Size:  491 bytes,    Routine Base:  _EDT$CODE + 0000


;  340          0910  1
;  341          0911  1 !<BLF/PAGE>
;  343           0912  1 END                                              ! of module EDT$WFINSLIN
;  344           0913  1
;  345           0914  0 ELUDOM



;                                         PSECT SUMMARY
;
;       Name                         Bytes                          Attributes
;
;   _EDT$CODE                             491  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        51        13           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$:WFINSLIN/OBJ=OBJ$:WFINSLIN MSRC$:WFINSLIN.BLI/UPDATE=(ENH$:WFIN
;       SLIN)

; Size:          491 code + 0 data bytes
; Run Time:         00:25.2
; Elapsed Time:     00:29.5
; Lines/CPU Min:     2178
; Lexemes/CPU-Min: 13866
; Memory Used:  189 pages
; Compilation Complete
```

0141 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



VMSMSG
LIS

WFCOPLIN
LIS

USSTRING
LIS

WFSCOPY
LIS

WFDELLIN
LIS

WFGETBKT
LIS

WFOPNBUF
LIS

WFREABCK
LIS

WFREAFWD
LIS

WFSTRINS
LIS

WFAPPBKT
LIS

WFENDINS
LIS

WFRESEQ
LIS

UGBUFFER
LIS

WFCLEAR
LIS

USSUBS
LIS

WFDELBKT
LIS

WFSPLBKT
LIS

WFLOCLIN
LIS

WFRBUKT
LIS

WFINSLIN
LIS

WFREACUR
LIS

WFREAINP
LIS

WFTOP
LIS

WFREPLIN
LIS

WFBOTTOM
LIS

WFECOPY
LIS