```
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
```

```
EX

Mo
--
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
ED
SY
LB
LI
```

```
**FILE**ID**WFSPLBKT                                   I 15
```

```
WW      WW  FFFFFFFFFF    SSSSSSSS  PPPPPPPP    LL          BBBBBBBB    KK      KK  TTTTTTTTTT
WW      WW  FFFFFFFFFF    SSSSSSSS  PPPPPPPP    LL          BBBBBBBB    KK      KK  TTTTTTTTTT
WW      WW  FF        SS          PP      PP  LL          BB      BB  KK      KK      TT
WW      WW  FF        SS          PP      PP  LL          BB      BB  KK      KK      TT
WW      WW  FF        SS          PP      PP  LL          BB      BB  KK    KK        TT
WW      WW  FF        SS          PP      PP  LL          BB      BB  KK    KK        TT
WW      WW  FFFFFFFF      SSSSSS    PPPPPPPP    LL          BBBBBBBB    KKKKKK          TT
WW      WW  FFFFFFFF      SSSSSS    PPPPPPPP    LL          BBBBBBBB    KKKKKK          TT
WW  WW  WW  FF                SS  PP          LL          BB      BB  KK    KK        TT
WW  WW  WW  FF                SS  PP          LL          BB      BB  KK    KK        TT
WWWW  WWWW  FF                SS  PP          LL          BB      BB  KK      KK      TT          ....
WWWW  WWWW  FF                SS  PP          LL          BB      BB  KK      KK      TT          ....
WW      WW  FF        SSSSSSSS    PP          LLLLLLLLLL  BBBBBBBB    KK      KK      TT          ....
WW      WW  FF        SSSSSSSS    PP          LLLLLLLLLL  BBBBBBBB    KK      KK      TT          ....

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
```

ED
VO

```
0 %TITLE 'EDT$WFSPLBKT - split the current bucket'
0 MODULE EDT$WFSPLBKT (                                ! Split the current bucket
0               IDENT = 'V04-000'                              ! File: WFSPLBKT.BLI Edit: JBS1010
0               ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL 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 CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL 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 !       Split the current bucket.
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 SPLIT_BUKT from module EDTWF.
1 ! 1-002 - Regularize headers.  JBS 19-Mar-1981
1 ! 1-003 - Modify to use EDT$WORKIO. STS 15-Feb-1982
1 ! 1-004 - Copy entire old bucket before split. STS 18-Feb-1982
1 ! 1-005 - Don't copy on 11's since it uses too much stack. STS 01-Mar-1982
1 ! 1-006 - Change stack storage to heap storage. STS 05-Mar-1982
1 ! 1-007 - Add literals for callable EDT. STS 10-Mar-1982
1 ! 1-008 - Give an error return if heap storage is exhausted.  JBS 09-Jun-1982
1 ! 1-009 - Put code for edt$$wf_nxt_buf in line. STS 11-oct-1982
1 ! 1-010 - Improve listing appearance.  JBS 20-Jun-1983
1 !--
1
```

```
;    58          0057  1 %SBTTL 'Declarations'
;    59          0058  1 !
;    60          0059  1 ! TABLE OF CONTENTS:
;    61          0060  1 !
;    62          0061  1
;    63          0062  1 REQUIRE 'EDTSRC:TRAROUNAM';
;    64          0501  1
;    65          0502  1 FORWARD ROUTINE
;    66          0503  1     EDT$$WF_SPLTBUK;
;    67          0504  1
;    68          0505  1 !
;    69          0506  1 ! INCLUDE FILES:
;    70          0507  1 !
;    71          0508  1
;    72          0509  1 REQUIRE 'EDTSRC:EDTREQ';
;    73          0644  1
;    74          0645  1 !
;    75          0646  1 ! MACROS:
;    76          0647  1 !
;    77          0648  1 !       NONE
;    78          0649  1 !
;    79          0650  1 ! EQUATED SYMBOLS:
;    80          0651  1 !
;    81          0652  1
;    82          0653  1 EXTERNAL LITERAL
;    83          0654  1     EDT$K_PUT;
;    84          0655  1
;    85          0656  1 !
;    86          0657  1 ! OWN STORAGE:
;    87          0658  1 !
;    88          0659  1 !       NONE
;    89          0660  1 !
;    90          0661  1 ! EXTERNAL REFERENCES:
;    91          0662  1 !
;    92          0663  1 !       In the routine
```

```
1 %SBTTL 'EDT$$WF_SPLTBUK  - split the current bucket'
1
1 GLOBAL ROUTINE EDT$$WF_SPLTBUK                        ! Split the current bucket
1     =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine splits the current bucket at the current position into
1 !       two buckets.  In the special case that we are at the end of the bucket
1 !       this is done simply by appending a new bucket, otherwise, we must allocate
1 !       a new bucket and copy all the text from the current line to the end of the
1 !       bucket into the new bucket.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       EDT$$A_CUR_BUF
1 !       EDT$$Z_WF_DESC
1 !       EDT$$A_WK_BUK
1 !       EDT$$G_WK_CURBUK
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       EDT$$G_WK_MODFD
1 !       EDT$$A_WK_BUK
1 !
1 ! ROUTINE VALUE:
1 !
1 !       1 = OK
1 !       0 = heap storage exhausted
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2    BEGIN
2
2    EXTERNAL ROUTINE
2        EDT$$ALO_HEAP,                          ! allocate heap storage
2        EDT$$DEA_HEAP : NOVALUE,                ! deallocate heap storage
2        EDT$$CAL[WIO,
2        EDT$$WF_NEWBUK : NOVALUE,
2        EDT$$WF_ALOBUF : NOVALUE,
2        EDT$$WF_MAKECUR : NOVALUE;
2
2    EXTERNAL
2        EDT$$Z_WF_DESC : BLOCK [, BYTE],        ! descriptor for the workfile record
2        EDT$$G_WK_AVAIL,
2        EDT$$G_WK_GRTSTBUK,
2        EDT$$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$$G_WK_CURBUK,                       ! Number of the current bucket
2             EDT$$G_WK_MODFD;                        ! Flag indicating bucket was modified
2
2         LOCAL
2             OLD_NEXT,
2             LEN,
2             ORIG_BUKT;
2
2 !+
2 ! Remember the next bucket number, and the original one.
2 !-
2         OLD_NEXT = .EDT$$A_WK_BUK [WFB_NEXT_BUKT];
2         ORIG_BUKT = .EDT$$G_WK_CURBUK;
2 !+
2 ! First check to see if we are at the end of a bucket.
2 !-
2
3         IF (.EDT$$A_CUR_BUF [TBCB_LINE_ADDR] EQL .EDT$$A_WK_BUK [WFB_END])
2         THEN
2 !+
2 ! We are at the end, just link a new bucket to this one
2 !-
3             BEGIN
3
4             IF (.EDT$$G_WK_AVAIL NEQ 0)
3             THEN
3                 EDT$$A_WK_BUK [WFB_NEXT_BUKT] = .EDT$$G_WK_AVAIL
3             ELSE
3                 EDT$$A_WK_BUK [WFB_NEXT_BUKT] = .EDT$$G_WK_GRTSTBUK;
3
3             EDT$$G_WK_MODFD = 1;
3             EDT$$WF_NEWBUK (.OLD_NEXT, .ORIG_BUKT)
3             END
2         ELSE
3             BEGIN
3 !+
3 ! We are not at the end of a bucket.
3 ! Split the bucket into two, at the beginning of the current line.
3 !-
3
3             LOCAL
3                 OLD_BUKT,
3                 NEW_BUKT;
3
3             LEN = .EDT$$A_WK_BUK [WFB_END] - .EDT$$A_CUR_BUF [TBCB_LINE_ADDR];
3             EDT$$A_WK_BUK [WFB_END] = .EDT$$A_CUR_BUF [TBCB_LINE_ADDR];
3
4             IF (.EDT$$G_WK_AVAIL NEQ 0)
3             THEN
3                 EDT$$A_WK_BUK [WFB_NEXT_BUKT] = NEW_BUKT = .EDT$$G_WK_AVAIL
3             ELSE
3                 EDT$$A_WK_BUK [WFB_NEXT_BUKT] = NEW_BUKT = .EDT$$G_WK_GRTSTBUK;
3
3             EDT$$CALLWIO (EDT$K_PUT, .EDT$$G_WK_CURBUK, EDT$$Z_WF_DESC);
3 !+
3 ! Save the bucket contents so that later we can extract a portion.
```

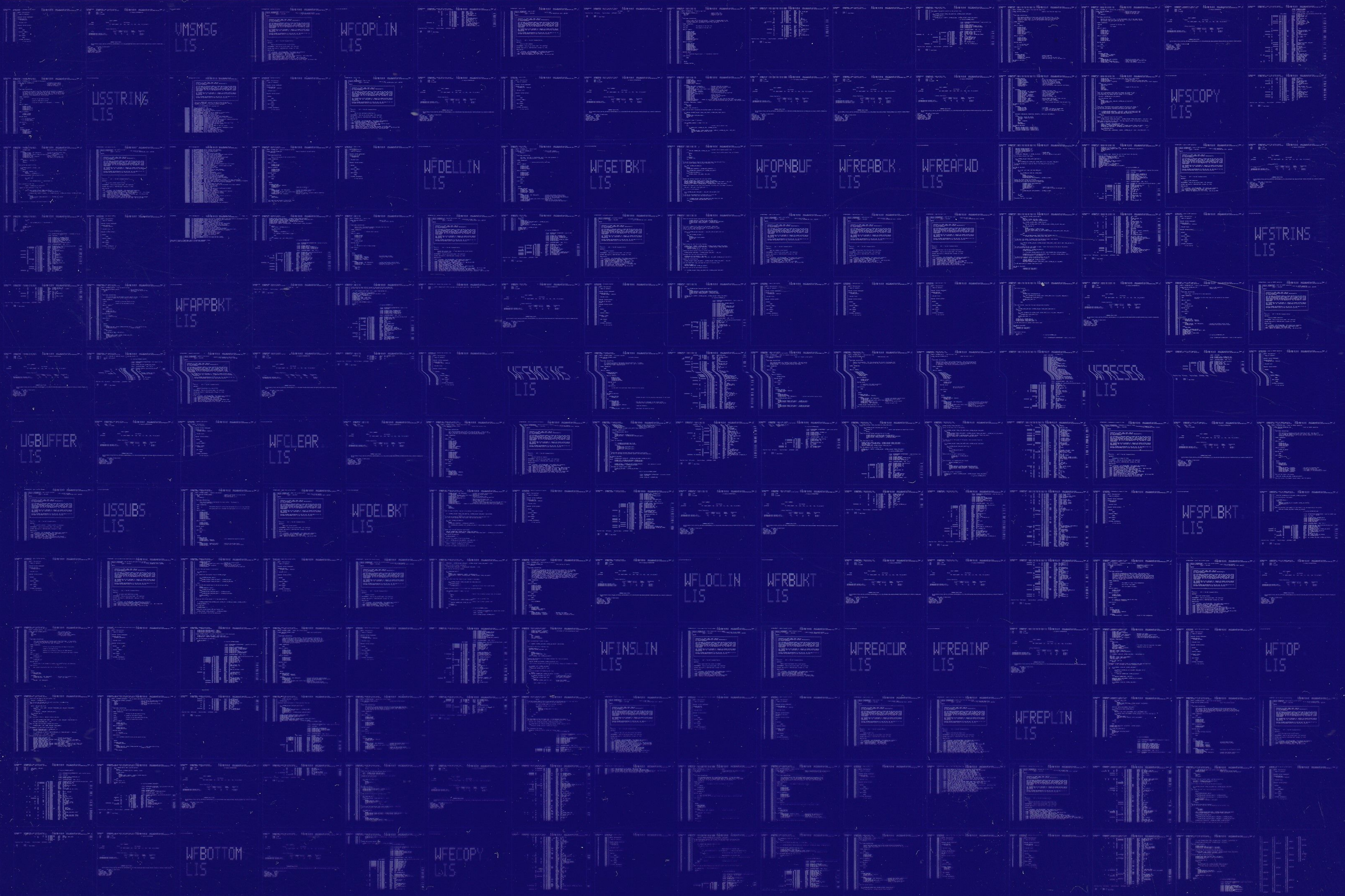
```
    3 !-
    3
    3         IF ( NOT EDT$$ALO_HEAP (%REF (WF_BUKT_SIZE), OLD_BUKT)) THEN RETURN (0);
    3
    3         EDT$$CPY_MEM (WF_BUKT_SIZE, .EDT$$A_WK_BUK, .OLD_BUKT);
    3 !+
    3 ! Now get a fresh buffer.
    3 !-
    3         EDT$$WF_ALOBUF ();
    3         EDT$$A_WK_BUK [WFB_NEXT_BUKT] = .OLD_NEXT;
    3         EDT$$A_WK_BUK [WFB_PREV_BUKT] = .ORIG_BUKT;
    3 !+
    3 ! Copy a portion of the old buffer into the new buffer.
    3 !-
P   3         EDT$$CPY_MEM (.LEN, CH$PTR (.OLD_BUKT, .EDT$$A_CUR_BUF [TBCB_LINE_ADDR]),        !
    3             CH$PTR (.EDT$$A_WK_BUK, WFB_FIXED_SIZE));
    3 !+
    3 ! Discard the copy of the old bucket.
    3 !-
    3         EDT$$DEA_HEAP (%REF (WF_BUKT_SIZE), OLD_BUKT);
    3         EDT$$A_WK_BUK [WFB_END] = .LEN + WFB_FIXED_SIZE;
    3         EDT$$G_WK_MODFD = 1;
    3
    4         IF (.OLD_NEXT EQL 0)
    3         THEN
    4             BEGIN
    4
    5             IF (.ORIG_BUKT EQL .EDT$$A_CUR_BUF [TBCB_LAST_BUKT])
    4             THEN
    4                 EDT$$A_CUR_BUF [TBCB_LAST_BUKT] = .NEW_BUKT;
    4
    4             END
    3         ELSE
    4             BEGIN
    4             EDT$$WF_MAKECUR (.OLD_NEXT);
    4             EDT$$A_WK_BUK [WFB_PREV_BUKT] = .NEW_BUKT;
    4             EDT$$G_WK_MODFD = 1;
    3             END;
    3
    3         EDT$$WF_MAKECUR (.ORIG_BUKT)
    2         END;
    2
    2     RETURN (1);
    1     END;                                          ! of routine EDT$$WF_SPLTBUK


                                                           .TITLE  EDT$WFSPLBKT EDT$WFSPLBKT - split the current b
;                                                                              ucket
                                                           .IDENT  \V04-000\

                                                           .EXTRN  EDT$K_PUT, EDT$$ALO_HEAP
                                                           .EXTRN  EDT$$DEA_HEAP, EDT$$CALLWIO
                                                           .EXTRN  EDT$$WF_NEWBUK, EDT$$WF_ALOBUF
                                                           .EXTRN  EDT$$WF_MAKECUR
                                                           .EXTRN  EDT$$Z_WF_DESC, EDT$$G_WK_AVAIL
                                                           .EXTRN  EDT$$G_WK_GRTSTBUK
                                                           .EXTRN  EDT$$A_CUR_BUF, EDT$$A_WK_BUK
```

```
                                                               .EXTRN  EDT$$G_WK_CURBUK
                                                               .EXTRN  EDT$$G_WK_MODFD

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

                                            0FFC 00000         .ENTRY  EDT$$WF_SPLTBUK, Save R2,R3,R4,R5,R6,R7,R8,-; 0666
                                                                       R9,R10,R11                                    ;
                         5B 00000000G  00   9E 00002           MOVAB   EDT$$A_CUR_BUF, R11                           ;
                         5A 00000000G  00   9E 00009           MOVAB   EDT$$A_WK_BUK, R10                            ;
                         5E            08   C2 00010           SUBL2   #8, SP                                        ;
                         50            6A   D0 00013           MOVL    EDT$$A_WK_BUK, R0                             ; 0733
                         53      02    A0   9E 00016           MOVAB   2(R0), R3                                     ;
                         59            63   3C 0001A           MOVZWL  (R3), OLD_NEXT                                ;
                         54 00000000G  00   D0 0001D           MOVL    EDT$$G_WK_CURBUK, R4                          ; 0734
                         58            54   D0 00024           MOVL    R4, ORIG_BUKT                                 ;
                         52 00000000G  00   D0 00027           MOVL    EDT$$G_WK_AVAIL, R2                           ; 0746
                         51            6B   D0 0002E           MOVL    EDT$$A_CUR_BUF, R1                            ; 0739
               04        A0            61   D1 00031           CMPL    (R1), 4(R0)                                   ;
                                       25   12 00035           BNEQ    3$                                            ;
                                       52   D5 00037           TSTL    R2                                            ; 0746
                                       05   13 00039           BEQL    1$                                            ;
                         63            52   B0 0003B           MOVW    R2, (R3)                                      ; 0748
                                       07   11 0003E           BRB     2$                                            ;
                         63 00000000G  00   B0 00040 1$:       MOVW    EDT$$G_WK_GRTSTBUK, (R3)                      ; 0750
          00000000G      00            01   D0 00047 2$:       MOVL    #1, EDT$$G_WK_MODFD                           ; 0752
                                       58   DD 0004E           PUSHL   ORIG_BUKT                                     ; 0753
                                       59   DD 00050           PUSHL   OLD_NEXT                                      ;
          00000000G      00            02   FB 00052           CALLS   #2, EDT$$WF_NEWBUK                            ;
                                     00CB   31 00059           BRW     9$                                            ;
      57       04        A0            61   C3 0005C 3$:       SUBL3   (R1), 4(R0), LEN                              ; 0766
               04        A0            61   D0 00061           MOVL    (R1), 4(R0)                                   ; 0767
                                       52   D5 00065           TSTL    R2                                            ; 0769
                                       08   13 00067           BEQL    4$                                            ;
                         56            52   D0 00069           MOVL    R2, NEW_BUKT                                  ; 0771
                         63            52   B0 0006C           MOVW    R2, (R3)                                      ;
                                       0A   11 0006F           BRB     5$                                            ;
                         56 00000000G  00   D0 00071 4$:       MOVL    EDT$$G_WK_GRTSTBUK, NEW_BUKT                  ; 0773
                         63            56   B0 00078           MOVW    NEW_BUKT, (R3)                                ;
                            00000000G  00   9F 0007B 5$:       PUSHAB  EDT$$Z_WF_DESC                                ; 0775
                                       54   DD 00081           PUSHL   R4                                            ;
                            00000000G  8F   DD 00083           PUSHL   #EDT$K_PUT                                    ;
          00000000G      00            03   FB 00089           CALLS   #3, EDT$$CALLWIO                              ;
                                 04    AE   9F 00090           PUSHAB  OLD_BUKT                                      ; 0780
               04        AE    0200    8F   3C 00093           MOVZWL  #512, 4(SP)                                   ;
                                 04    AE   9F 00099           PUSHAB  4(SP)                                         ;
          00000000G      00            02   FB 0009C           CALLS   #2, EDT$$ALO_HEAP                             ;
                         03            50   E8 000A3           BLBS    R0, 6$                                        ;
                                     0082   31 000A6           BRW     10$                                           ;
                         50            6A   D0 000A9 6$:       MOVL    EDT$$A_WK_BUK, R0                             ; 0782
      04        BE       60    0200    8F   28 000AC           MOVC3   #512, (R0), @OLD_BUKT                         ;
          00000000G      00            00   FB 000B3           CALLS   #0, EDT$$WF_ALOBUF                            ; 0786
                         50            6A   D0 000BA           MOVL    EDT$$A_WK_BUK, R0                             ; 0787
               02        A0            59   B0 000BD           MOVW    OLD_NEXT, 2(R0)                               ;
                         60            58   B0 000C1           MOVW    ORIG_BUKT, (R0)                               ; 0788
                         51            6B   D0 000C4           MOVL    EDT$$A_CUR_BUF, R1                            ; 0793
      51       04        AE            61   C1 000C7           ADDL3   (R1), OLD_BUKT, R1                            ;
      08        A0       61            57   28 000CC           MOVC3   LEN, (R1), 8(R0)                              ;
```

```
                                               04   AE   9F 000D1          PUSHAB   OLD_BUKT                                      ; 0797
                            04   AE          0200   8F   3C 000D4          MOVZWL   #512, 4(SP)                                   ;
                                               04   AE   9F 000DA          PUSHAB   4(SP)                                         ;
                     00000000G   00                 02   FB 000DD          CALLS    #2, EDT$$DEA_HEAP                             ;
                                 50                 6A   D0 000E4          MOVL     EDT$$A_WK_BUK, R0                             ; 0798
                            04   A0            08   A7   9E 000E7          MOVAB    8(R7), 4(R0)                                  ;
                     00000000G   00                 01   D0 000EC          MOVL     #1, EDT$$G_WK_MODFD                           ; 0799
                                                    59   D5 000F3          TSTL     OLD_NEXT                                      ; 0801
                                                    11   12 000F5          BNEQ     7$                                            ;
                                 50                 6B   D0 000F7          MOVL     EDT$$A_CUR_BUF, R0                            ; 0805
        58        10   A0        10                 00   ED 000FA          CMPZV    #0, #8, 16(R0), ORIG_BUKT                     ;
                                                    1C   12 00100          BNEQ     8$                                            ;
                            10   A0                 56   B0 00102          MOVW     NEW_BUKT, 16(R0)                              ; 0807
                                                    16   11 00106          BRB      8$                                            ; 0801
                                                    59   DD 00108 7$:      PUSHL    OLD_NEXT                                      ; 0812
                     00000000G   00                 01   FB 0010A          CALLS    #1, EDT$$WF_MAKECUR                           ;
                                 50                 6A   D0 00111          MOVL     EDT$$A_WK_BUK, R0                             ; 0813
                                 60                 56   B0 00114          MOVW     NEW_BUKT, (R0)                                ;
                     00000000G   00                 01   D0 00117          MOVL     #1, EDT$$G_WK_MODFD                           ; 0814
                                                    58   DD 0011E 8$:      PUSHL    ORIG_BUKT                                     ; 0817
                     00000000G   00                 01   FB 00120          CALLS    #1, EDT$$WF_MAKECUR                           ;
                                 50                 01   D0 00127 9$:      MOVL     #1, R0                                        ; 0820
                                                         04 0012A          RET                                                    ;
                                                    50   D4 0012B 10$:     CLRL     R0                                            ; 0821
                                                         04 0012D          RET                                                    ;

; Routine Size:  302 bytes,    Routine Base:  _EDT$CODE + 0000


;  252          0822  1
;  253          0823  1 !<BLF/PAGE>
```

```
:  255          0824  1 END                                              ! of module EDT$WFSPLBKT
:  256          0825  1
:  257          0826  0 ELUDOM



;                                    PSECT SUMMARY
;
;       Name                         Bytes                        Attributes
;
;  _EDT$CODE                            302  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        31         8         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$:WFSPLBKT/OBJ=OBJ$:WFSPLBKT MSRC$:WFSPLBKT.BLI/UPDATE=(ENH$:WFSP
;       LBKT)

; Size:          302 code + 0 data bytes
; Run Time:         00:16.9
; Elapsed Time:     00:20.8
; Lines/CPU Min:     2939
; Lexemes/CPU-Min: 10256
; Memory Used:  120 pages
; Compilation Complete
```

0141 AH-BT13A-SE DIGITAL EQUIPMENT CORPORATION
VAX/VMS V4.0 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