


```

WW      WW      FFFFFFFFFF  RRRRRRRR  EEEEEEEEEEE  AAAAAA  BBBB88888  CCCCCCCC  KK      KK
WW      WW      FFFFFFFFFF  P' ,RRRRR  EEEEEEEEEEE  AAAAAA  BBBB88888  CCCCCCCC  KK      KK
WW      WW      FF          RK          RR  EE          AA      AA  BB      BB  CC          KK      KK
WW      WW      FF          RR          RR  EE          AA      AA  BB      BB  CC          KK      KK
WW      WW      FF          RR          RR  EE          AA      AA  BB      BB  CC          KK      KK
WW      WW      FF          RR          RR  EE          AA      AA  BB      BB  CC          KK      KK
WW      WW      FFFFFFFF    RRRRRRRR  EEEEEEEEEEE  AA      AA  BBBB88888  CC          KKKKKK
WW      WW      FFFFFFFF    RRRRRRRR  EEEEEEEEEEE  AA      AA  BBBB88888  CC          KKKKKK
WW      WW      FF          RR  RR      EE          ^AAAAAAAAA  BB      BB  CC          KK      KK
WW      WW      FF          RR  RR      EE          AAAAAAAAAA  BB      BB  CC          KK      KK
WWW     WWW     FF          RR      RR  EE          AA      AA  BB      BB  CC          KK      KK
WWW     WWW     FF          RR      RR  EE          AA      AA  BB      BB  CC          KK      KK
WW      WW      FF          RR          RR  EEEEEEEEEEE  AA      AA  BBBB88888  CCCCCCCC  KK      KK
WW      WW      FF          RR          RR  EEEEEEEEEEE  AA      AA  BBBB88888  CCCCCCCC  KK      KK

```

```

LL      111111  SSSSSSSS
LL      111111  SSSSSSSS
LL      11      SS
LL      11      SS
LL      11      SS
LL      11      SS
LL      11      SSSSSS
LL      11      SSSSSS
LL      11      SS
LL      11      SS
LL      11      SS
LL      11      SS
LLLLLLLLLLLL 111111  SSSSSSSS
LLLLLLLLLLLL 111111  SSSSSSSS

```

```

1 0001 0 %TITLE 'EDT$WFREABCK - read previous line'
2 0002 0 MODULE EDT$WFREABCK ( ! Read previous line
3 0003 0 IDENT = 'V04-000' ! File: WFREABCK.BLI Edit: TSS1005
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 Read the previous line from the work file system.
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: October 16, 1978
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original. DJS 23-Feb-1981. This module was created by
45 0045 1 extracting routine EDT$SRD_PRVLN from module EDTWF.
46 0046 1 1-002 - regularize headers. JBS 19-Mar-1981
47 0047 1 1-003 - Change index for line numbers from 10 to 15. SMB 18-Jan-1982
48 0048 1 1-004 - Remove EDT$SET_WKLN. JBS 14-Sep-1982
49 0049 1 1-005 - Modify to use new 48 bit macro. STS 01-Oct-1982
50 0050 1 --
51 0051 1

```

```
53 0052 1 %SBTTL 'Declarations'  
54 0053 1  
55 0054 1 TABLE OF CONTENTS:  
56 0055 1  
57 0056 1  
58 0057 1 REQUIRE 'EDT$SRC:TRAROUNAM';  
59 0496 1  
60 0497 1 FORWARD ROUTINE  
61 0498 1 EDT$SRD_PRVLN;  
62 0499 1  
63 0500 1  
64 0501 1 INCLUDE FILES:  
65 0502 1  
66 0503 1  
67 0504 1 REQUIRE 'EDT$SRC:EDTREQ';  
68 0639 1  
69 0640 1  
70 0641 1 MACROS:  
71 0642 1  
72 0643 1 NONE  
73 0644 1  
74 0645 1 EQUATED SYMBOLS:  
75 0646 1  
76 0647 1 NONE  
77 0648 1  
78 0649 1 OWN STORAGE:  
79 0650 1  
80 0651 1 NONE  
81 0652 1  
82 0653 1 EXTERNAL REFERENCES:  
83 0654 1  
84 0655 1 In the routine
```

```
86 0656 1 %SBTTL 'EDT$$RD_PRVLN - read previous line'
87 0657 1
88 0658 1 GLOBAL ROUTINE EDT$$RD_PRVLN          ! Read previous line
89 0659 1 =
90 0660 1
91 0661 1 !++
92 0662 1 ! FUNCTIONAL DESCRIPTION:
93 0663 1
94 0664 1     Read a line backwards.  The line preceding the current line becomes the
95 0665 1     current line.  If we are already at the top, return a 0 otherwise return
96 0666 1     a 1.
97 0667 1
98 0668 1 ! FORMAL PARAMETERS:
99 0669 1
100 0670 1     NONE
101 0671 1
102 0672 1 ! IMPLICIT INPUTS:
103 0673 1
104 0674 1     EDT$$A_CUR_BUF
105 0675 1     EDT$$A_WK_BUK
106 0676 1     EDT$$G_WK_CURBUK
107 0677 1     EDT$$L_LN00
108 0678 1
109 0679 1 ! IMPLICIT OUTPUTS:
110 0680 1
111 0681 1     EDT$$A_CUR_BUF
112 0682 1     EDT$$A_WK_LN
113 0683 1
114 0684 1 ! ROUTINE VALUE:
115 0685 1
116 0686 1     1 = previous line read successfully
117 0687 1     0 = there is no previous line
118 0688 1
119 0689 1 ! SIDE EFFECTS:
120 0690 1
121 0691 1     NONE
122 0692 1
123 0693 1 --
124 0694 1
125 0695 2     BEGIN
126 0696 2
127 0697 2     EXTERNAL ROUTINE
128 0698 2     EDT$$WF_MAKECUR : NOVALUE;
129 0699 2
130 0700 2     EXTERNAL
131 0701 2     EDT$$A_CUR_BUF : REF TBCB_BLOCK,          ! Current text buffer control block
132 0702 2     EDT$$A_WK_BUK :                          ! Pointer to current bucket
133 0703 2     REF BLOCK [WF_BUKT_SIZE, BYTE] FIELD (WFB_FIELDS),
134 0704 2     EDT$$G_WK_CURBUK,                          ! Number of the current bucket
135 0705 2     EDT$$L_LN00 : LNOVECTOR [14],
136 0706 2     EDT$$A_WK_LN : REF LIN_BLOCK;             ! Pointer to work line
137 0707 2
138 0708 2     EDT$$A_CUR_BUF [TBCB_CHAR_POS] = 0;
139 0709 2 !+
140 0710 2 ! See if we are at the beginning of a bucket.
141 0711 2 !-
142 0712 2
```

```

143 0713 IF (.EDT$$A_CUR_BUF [TBCB_LINE_ADDR] EQL WFB_FIXED_SIZE)
144 0714 THEN
145 0715
146 0716 IF (.EDT$$A_WK_BUK [WFB_PREV_BUKT] EQL 0)
147 0717 THEN
148 0718 RETURN (0) ! cannot read backward at beginning of buffer.
149 0719 ELSE
150 0720 BEGIN
151 0721
152 0722 + Read the previous bucket and position to it's end.
153 0723
154 0724 EDT$$WF_MAKECUR (.EDT$$A_WK_BUK [WFB_PREV_BUKT]);
155 0725 EDT$$A_CUR_BUF [TBCB_LINE_ADDR] = .EDT$$A_WK_BUK [WFB_END];
156 0726 EDT$$A_CUR_BUF [TBCB_CUR_BUKT] = .EDT$$G_WK_CURBUK;
157 0727 END;
158 0728
159 0729 + Now, move back a line.
160 0730
161 0731
162 0732 EDT$$A_CUR_BUF [TBCB_LINE_ADDR] = .EDT$$A_CUR_BUF [TBCB_LINE_ADDR] -
163 0733 CHSRCHR (CHSPTR (.EDT$$A_WK_BUK, .EDT$$A_CUR_BUF [TBCB_LINE_ADDR] - 1)) -
164 0734 LIN_FIXED_SIZE - 1;
165 0735 SUBLINE (NUMBER_ONE, EDT$$A_CUR_BUF [TBCB_CUR_LIN]);
166 0736
167 0737 + Get the address of the current line.
168 0738
169 0739 EDT$$A_WK_LN = CHSPTR (.EDT$$A_WK_BUK, .EDT$$A_CUR_BUF [TBCB_LINE_ADDR]);
170 0740 RETURN (1);
171 0741 END; ! of routine EDT$$RD_PRVLN

```

.TITLE EDT\$WFREABCK EDT\$WFREABCK - read previous line
.IDENT \V04-000\

.EXTRN EDT\$\$WF_MAKECUR
.EXTRN EDT\$\$A_CUR_BUF, EDT\$\$A_WK_BUK
.EXTRN EDT\$\$G_WK_CURBUK
.EXTRN EDT\$\$L_LN00, EDT\$\$A_WK_LN

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

			001C 00000	.ENTRY EDT\$\$RD_PRVLN, Save R2,R3,R4	: 0658
54	00000000G	00	9E 00002	MOVAB EDT\$\$A_CUR_BUF, R4	
53	00000000G	00	9E 00009	MOVAB EDT\$\$A_WK_BUK, R3	
50		64	D0 00010	MOVL EDT\$\$A_CUR_BUF, R0	: 0708
		0C	A0 B4 00013	CLRW 12(R0)	
08		60	D1 00016	CPL (R0), #8	: 0713
		23	12 00019	BNEQ 1\$	
50		63	D0 0001B	MOVL EDT\$\$A_WK_BUK, R0	: 0716
		60	B5 0001E	TSTW (R0)	
		4E	13 00020	BEQL 3\$	
7E		60	3C 00022	MOVZWL (R0), -(SP)	: 0724
00000000G	00	01	FB 00025	CALLS #1, EDT\$\$WF_MAKECUR	
	51	64	D0 0002C	MOVL EDT\$\$A_CUR_BUF, R1	: 0725
	50	63	D0 0002F	MOVL EDT\$\$A_WK_BUK, R0	
	61	04	A0 D0 00032	MOVL 4(R0), -(R1)	
04	A1 00000000G	00	B0 00036	MOVW EDT\$\$G_WK_CURBUK, 4(R1)	: 0726

	50		64	D0	0003E	1\$:	MOVL	EDT\$\$A_CUR_BUF, R0	:	0732
	52		63	D0	00041		MOVL	EDT\$\$A_WK_BUK, R2	:	0733
51	52		60	C1	00044		ADDL3	(R0), R2, -R1	:	
	51	FF	A1	9A	00048		MOVZBL	-1(R1), R1	:	
51	60		51	C3	0004C		SUBL3	R1, (R0), R1	:	
	60	F8	A1	9E	00050		MOVAB	-8(R1), (R0)	:	0734
	51	06	A0	D0	00054		MOVL	6(R0), SAVE	:	0735
		06	AC	D7	00058		DECL	6(R0)	:	
	51	06	A0	D1	0005B		CMPL	6(R0), SAVE	:	
			03	1B	0005F		BLEQU	2\$:	
		0A	A0	B7	00061		DECW	10(R0)	:	
00000000G 00	52		60	C1	00064	2\$:	ADDL3	(R0), R2, EDT\$\$A_WK_LN	:	0739
	50		01	D0	0006C		MOVL	#1, R0	:	0740
				04	0006F		RET		:	
			50	D4	00070	3\$:	CLRL	R0	:	0741
			04	00072			RET		:	

: Routine Size: 115 bytes, Routine Base: _EDT\$CODE + 0000

: 172 0742 1
: 173 0743 1 !<BLF/PAGE>

: R
:
:

EDT\$WFREABCK
V04-000

EDT\$WFREABCK - read previous line
EDT\$\$RD_PRVLN - read previous line

J 11
16-Sep-1984 02:10:18
14-Sep-1984 12:25:39

VAX-11 Bliss-32 V4.0-742
DISK\$VMMASTER:[EDT.SRC]WFREABCK.BLI;1 (4)

EDT
V04

: 175 0744 1 END
: 176 0745 1
: 177 0746 0 ELUDOM

! of module EDT\$WFREABCK

PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	115	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[EDT.SRC]EDT.L32;1	377	42	11	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\$:WFREABCK/OBJ=OBJ\$:WFREABCK MSRC\$:WFREABCK.BLI/UPDATE=(ENH\$:WFREABCK)

: Size: 115 code + 0 data bytes
: Run Time: 00:12.5
: Elapsed Time: 00:16.9
: Lines/CPU Min: 3589
: Lexemes/CPU-Min: 12880
: Memory Used: 85 pages
: Compilation Complete

S
R
E
L
M
C

