


```

WW      WW      FFFFFFFFFF      RRRRRRRR      EEEEEEEEEEE      AAAAAA      FFFFFFFFFF      WW      WW      DDDDDDDD
WW      WW      FFFFFFFFFF      RRRRRRRR      EEEEEEEEEEE      AAAAAA      FFFFFFFFFF      WW      WW      DDDDDDDD
WW      WW      FF              RR          RR      EE              AA          AA      FF              WW      WW      DD          DD
WW      WW      FF              RR          RR      EE              AA          AA      FF              WW      WW      DD          DD
WW      WW      FF              RR          RR      EE              AA          AA      FF              WW      WW      DD          DD
WW      WW      FF              RR          RR      EE              AA          AA      FF              WW      WW      DD          DD
WW      WW      FF              RR          RR      EE              AA          AA      FF              WW      WW      DD          DD
WW      WW      FFFFFFFFFF      RRRRRRRR      EEEEEEEEEEE      AAAAAA      FFFFFFFFFF      WW      WW      DD          DD
WW      WW      FFFFFFFFFF      RRRRRRRR      EEEEEEEEEEE      AAAAAA      FFFFFFFFFF      WW      WW      DD          DD
WW      WW      FF              RR          RR      EE              AA          AA      FF              WW      WW      DD          DD
WW      WW      FF              RR          RR      EE              AA          AA      FF              WW      WW      DD          DD
WWW     WWW     FF              RR          RR      EE              AA          AA      FF              WWW     WWW     DD          DD
WWW     WWW     FF              RR          RR      EE              AA          AA      FF              WWW     WWW     DD          DD
WW      WW      FF              RR          RR      EEEEEEEEEEE      AA          AA      FF              WW      WW      DDDDDDDD
WW      WW      FF              RR          RR      EEEEEEEEEEE      AA          AA      FF              WW      WW      DDDDDDDD

```

```

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
LLLLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLLLL      IIIIII      SSSSSSSS

```



```

1 0001 0 %TITLE 'EDT$WFREAFWD - read the next line'
2 0002 0 MODULE EDT$WFREAFWD ( ! Read the next line
3 0003 0 IDENT = 'V04-000' ! File: WFREAFWD.BLI Edit: STS1008
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 next line in the forward direction.
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$$RD_NXTLN from module EDTWF.
46 0046 1 1-002 - Regularized the headers. JBS 25-Feb-1981
47 0047 1 1-003 - Fix module name. JBS 19-Mar-1981
48 0048 1 1-004 - Change EOB_LINE to EDT$$Z_EOB_LN. JBS 31-Mar-1981
49 0049 1 1-005 - Correct a typo in a subtitle. JBS 02-Jun-1981
50 0050 1 1-006 - Change index for line numbers from 10 to 15. SMB 18-Jan-1982
51 0051 1 1-007 - Remove EDT$$SET_WKLN. JBS 14-Sep-1982
52 0052 1 1-008 - Modify to use new 48 bit macro. STS 01-Oct-1982
53 0053 1 --
54 0054 1

```

```
56 0055 1 %SBTTL 'Declarations'  
57 0056 1  
58 0057 1 : TABLE OF CONTENTS:  
59 0058 1 :  
60 0059 1  
61 0060 1 REQUIRE 'EDTSRC:TRAROUNAM';  
62 0499 1  
63 0500 1 FORWARD ROUTINE  
64 0501 1 EDT$SRD_NXTLN;  
65 0502 1  
66 0503 1 :  
67 0504 1 : INCLUDE FILES:  
68 0505 1 :  
69 0506 1  
70 0507 1 REQUIRE 'EDTSRC:EDTREQ';  
71 0642 1  
72 0643 1 :  
73 0644 1 : MACROS:  
74 0645 1  
75 0646 1 : NONE  
76 0647 1  
77 0648 1 : EQUATED SYMBOLS:  
78 0649 1  
79 0650 1 : NONE  
80 0651 1  
81 0652 1 : OWN STORAGE:  
82 0653 1  
83 0654 1 : NONE  
84 0655 1  
85 0656 1 : EXTERNAL REFERENCES:  
86 0657 1  
87 0658 1 : In the routine
```

```

89 0659 1 %SBTTL 'EDTSSRD_NXTLN - read the next line'
90 0660 1
91 0661 1 GLOBAL ROUTINE EDTSSRD_NXTLN          ! Read the next line
92 0662 1 =
93 0663 1
94 0664 1 !++
95 0665 1 FUNCTIONAL DESCRIPTION:
96 0666 1
97 0667 1     Read the next line in the forward direction. The line following the
98 0668 1     current line becomes the new line. This routine may have the effect
99 0669 1     of reading a line from the input file. If we are already at the end
100 0670 1     of the buffer, then return a 0 otherwise return a 1.
101 0671 1
102 0672 1 FORMAL PARAMETERS:
103 0673 1
104 0674 1     NONE
105 0675 1
106 0676 1 IMPLICIT INPUTS:
107 0677 1
108 0678 1     EDTSSG_EXITD
109 0679 1     EDTSSA_WK_BUK
110 0680 1     EDTSSG_WK_CURBUK
111 0681 1     EDTSSA_WK_LN
112 0682 1     EDTSSZ_EOB_LN
113 0683 1     EDTSSL_LNOO
114 0684 1
115 0685 1 IMPLICIT OUTPUTS:
116 0686 1
117 0687 1     EDTSSA_CUR_BUF
118 0688 1     EDTSSA_WK_LN
119 0689 1
120 0690 1 ROUTINE VALUE:
121 0691 1
122 0692 1     1           Not at end of buffer
123 0693 1     0           At end of buffer
124 0694 1
125 0695 1 SIDE EFFECTS:
126 0696 1
127 0697 1     NONE
128 0698 1
129 0699 1 --
130 0700 1
131 0701 2 BEGIN
132 0702 2
133 0703 2 EXTERNAL ROUTINE
134 0704 2     EDTSSF_MAKECUR : NOVALUE,
135 0705 2     EDTSSRD_ILN;
136 0706 2
137 0707 2 EXTERNAL
138 0708 2     EDTSSA_CUR_BUF : REF TCB_BLOCK,          ! Current text buffer control block
139 0709 2     EDTSSG_EXITD,                          ! Exit flag (on if we are exiting)
140 0710 2     EDTSSA_WK_BUK :                       ! Pointer to current bucket
141 0711 2     REF BLOCK [WF_BUKT_SIZE, BYTE] FIELD (WFB_FIELDS),
142 0712 2     EDTSSG_WK_CURBUK,                       ! Number of the current bucket
143 0713 2     EDTSSA_WK_LN : REF LIN_BLOCK,          ! Pointer to current line
144 0714 2     EDTSSZ_EOB_LN,
145 0715 2     EDTSSL_LNOO : LNOVECTOR [14];

```

```
146 0716 2
147 0717 2
148 0718 2
149 0719 2
150 0720 2
151 0721 2
152 0722 2
153 0723 2
154 0724 2
155 0725 2
156 0726 2
157 0727 2
158 0728 2
159 0729 2
160 0730 2
161 0731 2
162 0732 2
163 0733 2
164 0734 2
165 0735 2
166 0736 2
167 0737 2
168 0738 2
169 0739 2
170 0740 2
171 0741 2
172 0742 3
173 0743 4
174 0744 3
175 0745 4
176 0746 4
177 0747 4
178 0748 4
179 0749 4
180 0750 4
181 0751 4
182 0752 3
183 0753 2
184 0754 2
185 0755 2
186 0756 2
187 0757 2
188 0758 2
189 0759 2
190 0760 2
191 0761 2
192 0762 2
193 0763 2
194 0764 2
195 0765 2
196 0766 2
197 0767 2
198 0768 2
199 0769 2
200 0770 1

EDTSSA_CUR_BUF [TBCB_CHAR_POS] = 0;
+
Point to the next line in the bucket.
-
IF (.EDTSSA_CUR_BUF [TBCB_LINE_ADDR] NEQA .EDTSSA_WK_BUK [WFB_END])
THEN
BEGIN
EDTSSA_CUR_BUF [TBCB_LINE_ADDR] = .EDTSSA_CUR_BUF [TBCB_LINE_ADDR] + .EDTSSA_WK_LN [LIN_LENGTH] +
LIN_FIXED_SIZE + 1;
EDTSSA_WK_LN = CHSPTR (.EDTSSA_WK_BUK, .EDTSSA_CUR_BUF [TBCB_LINE_ADDR]);
ADDLINE (NUMBER_ONE, EDTSSA_CUR_BUF [TBCB_CUR_IN]);
END;
+
If this is out of the scope of the bucket, then we better read the
next bucket.
-
IF (.EDTSSA_CUR_BUF [TBCB_LINE_ADDR] GEQA .EDTSSA_WK_BUK [WFB_END])
THEN
IF (.EDTSSA_WK_BUK [WFB_NEXT_BUKT] EQL 0)
THEN
BEGIN
IF (.EDTSSG_EXITD NEQ 0)
THEN
BEGIN
+
Return zero so we don't read any more.
-
EDTSSA_WK_LN = EDTSSZ_EOB_LN;
RETURN (0);
END;
+
End of buffer, try reading the next record.
-
RETURN (EDTSSRD_ILN ());
END
ELSE
BEGIN
EDTSSWF_MAKECUR (.EDTSSA_WK_BUK [WFB_NEXT_BUKT]);
EDTSSA_CUR_BUF [TBCB_LINE_ADDR] = WFB_FIXED_SIZE;
EDTSSA_CUR_BUF [TBCB_CUR_BUKT] = .EDTSSG_WK_CURBUK;
END;
+
Update the current line pointer.
-
EDTSSA_WK_LN = CHSPTR (.EDTSSA_WK_BUK, .EDTSSA_CUR_BUF [TBCB_LINE_ADDR]);
RETURN (1)
END;
! of routine EDTSSRD_NXTLN
```

				.TITLE	EDTSWFREAFWD EDTSWFREAFWD - read the next line			
				.IDENT	\V04-000\			
				.EXTRN	EDTSSWF_MAKECUR			
				.EXTRN	EDTSSRD_ILN, EDTSSA_CUR_BUF			
				.EXTRN	EDTSSG_EXITD, EDTSSA_WK_BUK			
				.EXTRN	EDTSSG_WK_CURBUK			
				.EXTRN	EDTSSA_WK_LN, EDTSSZ_EOB_LN			
				.EXTRN	EDTSSL_LN00			
				.PSECT	_EDT\$CODE, NOWRT, SHR, PIC, 2			
				.ENTRY	EDTSSRD_NXTLN, Save R2,R3,R4,R5	0661		
		55	00000000G	00	9E 00002	MOVAB	EDTSSA_WK_BUK, R5	
		54	00000000G	00	9E 00009	MOVAB	EDTSSA_CUR_BUF, R4	
		53	00000000G	00	9E 00010	MOVAB	EDTSSA_WK_LN, R3	
		50		64	D0 00017	MOVL	EDTSSA_CUR_BUF, R0	0717
						CLR	12(R0)	
		51		65	D0 0001D	MOVL	EDTSSA_WK_BUK, R1	0722
	04	A1		60	D1 00020	CMPL	(R0), 4(RT)	
				19	13 00024	BEQL	1\$	
		52		63	D0 00026	MOVL	EDTSSA_WK_LN, R2	0725
		52		62	9A 00029	MOVZBL	(R2), R2	
		52		60	C0 0002C	ADDL2	(R0), R2	
		60		A2	9E 0002F	MOVAB	8(R2), (R0)	0726
63		51		60	C1 00033	ADDL3	(R0), R1, EDTSSA_WK_LN	0727
						INCL	6(R0)	0728
				03	12 0003A	BNEQ	1\$	
				0A	A0 B6 0003C	INCL	10(R0)	
	04	A1		60	D1 0003F 1\$:	CMPL	(R0), 4(R1)	0736
				37	1F 00043	BLSSU	4\$	
				02	A1 B5 00045	TSTW	2(R1)	0739
				19	12 00048	BNEQ	3\$	
			00000000G	00	D5 0004A	TSTL	EDTSSG_EXITD	0743
				09	13 00050	BEQL	2\$	
		63	00000000G	00	9E 00052	MOVAB	EDTSSZ_EOB_LN, EDTSSA_WK_LN	0749
				2C	11 00059	BRB	5\$	0750
		00000000G	00	00	FB 0005B 2\$:	CALLS	#0, EDTSSRD_ILN	0756
						RET		
		7E		02	A1 3C 00063 3\$:	MOVZWL	2(R1), -(SP)	0760
		00000000G	00	01	FB 00067	CALLS	#1, EDTSSWF_MAKECUR	
		50		64	D0 0006E	MOVL	EDTSSA_CUR_BUF, R0	0761
		60		08	D0 00071	MOVL	#8, (R0)	
		04	00000000G	00	B0 00074	MOVW	EDTSSG_WK_CURBUK, 4(R0)	0762
		50		64	D0 0007C 4\$:	MOVL	EDTSSA_CUR_BUF, R0	0768
63		65		60	C1 0007F	ADDL3	(R0), EDTSSA_WK_BUK, EDTSSA_WK_LN	
		50		01	D0 00083	MOVL	#1, R0	0769
						RET		
				50	D4 00087 5\$:	CLRL	R0	0770
				04	00089	RET		

: Routine Size: 138 bytes, Routine Base: _EDT\$CODE + 0000

: 201 0771 1
: 202 0772 1 !<BLF/PAGE>

EDT\$WFREAFWD
V04-000

EDT\$WFREAFWD - read the next line
EDT\$\$RD_NXTLN - read the next line

J 12
16-Sep-1984 02:11:20
14-Sep-1984 12:25:40

VAX-11 Bliss-32 V4.0-742 Page 6
DISK\$VMMASTER:[EDT.SRC]WFREAFWD.BLI;1 (4)

: 204 0773 1 END
: 205 0774 1
: 206 0775 0 ELUDOM

! of module EDT\$WFREAFWD

PSECT SUMMARY

: Name Bytes Attributes
: _EDT\$CODE 138 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\$:WFREAFWD/OBJ=OBJ\$:WFREAFWD MSRCS:WFREAFWD.BLI/UPDATE=(ENHS:WFREAFWD)

: Size: 138 code + 0 data bytes
: Run Time: 00:13.3
: Elapsed Time: 00:15.8
: Lines/CPU Min: 3496
: Lexemes/CPU-Min: 13069
: Memory Used: 95 pages
: Compilation Complete

