


```
1 0001 0 %TITLE 'EDT$LXPRINT - type the current line on the terminal'  
2 0002 0 MODULE EDT$LXPRINT ( ! Type the current line on the terminal  
3 0003 0 IDENT = 'V04-000' ! File: LXPRINT.BLI Edit: JBS1010  
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 displays the current line on the terminal.  
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: February 3, 1978  
41 0041 1  
42 0042 1 MODIFIED BY:  
43 0043 1  
44 0044 1 1-001 - Original. DJS 30-JAN-81. This module was created by  
45 0045 1 extracting the routine EDT$$TY CURLN from the module EXEC.BLI.  
46 0046 1 1-002 - Regularize headers. JBS 23-Mar-1981  
47 0047 1 1-003 - Change [EOB] to user defined string. STS 06-Oct-1981  
48 0048 1 1-004 - Use EDT$$FMT_CRLF instead of EDT$$OUT_FMTBUF. JBS 05-Jul-1982  
49 0049 1 1-005 - Turn off the working message. JBS 07-Jul-1982  
50 0050 1 1-006 - Modify the alphabetic processing. JBS 19-Jul-1982  
51 0051 1 1-007 - Put edt$$st_eob in line. STS 22-Sep-1982  
52 0052 1 1-008 - Don't print beyond the terminal's width. JBS 22-Oct-1982  
53 0053 1 1-009 - Use EDT$$G_FMT_LNPOS instead of EDT$$G PRV_COL to measure against terminal's width. JBS 01-Nov-19  
54 0054 1 1-010 - Add VT220 support conditional. JBS 11-Feb-1983  
55 0055 1 --  
56 0056 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 'EDT SRC:TRAROUNAM';  
.. 64 0501 1  
.. 65 0502 1 FORWARD ROUTINE  
.. 66 0503 1 EDT$STY_CURLN;  
.. 67 0504 1  
.. 68 0505 1 :  
.. 69 0506 1 : INCLUDE FILES:  
.. 70 0507 1 :  
.. 71 0508 1  
.. 72 0509 1 REQUIRE 'EDT SRC:EDTREQ';  
.. 73 0644 1  
.. 74 0645 1 LIBRARY 'EDT SRC:SUPPORTS';  
.. 75 0646 1  
.. 76 0647 1 :  
.. 77 0648 1 : MACROS:  
.. 78 0649 1 :  
.. 79 0650 1 : NONE  
.. 80 0651 1 :  
.. 81 0652 1 : EQUATED SYMBOLS:  
.. 82 0653 1 :  
.. 83 0654 1 : NONE  
.. 84 0655 1 :  
.. 85 0656 1 : OWN STORAGE:  
.. 86 0657 1 :  
.. 87 0658 1 : NONE  
.. 88 0659 1 :  
.. 89 0660 1 : EXTERNAL REFERENCES:  
.. 90 0661 1 :  
.. 91 0662 1 : In the routine
```

! Display the current line

```

: 93 0663 1 %SBTTL 'EDT$$TY_CURLN - type the current line on the terminal'
: 94 0664 1
: 95 0665 1 GLOBAL ROUTINE EDT$$TY_CURLN ! Type the current line on the terminal
: 96 0666 1 =
: 97 0667 1
: 98 0668 1
: 99 0669 1 ++
100 0670 1 FUNCTIONAL DESCRIPTION:
101 0671 1 This routine displays the current line on the terminal. If the
102 0672 1 EDT$$G_NOS flag is on the line number is displayed before the line.
103 0673 1 If case flagging is enabled it is handled by this routine.
104 0674 1
105 0675 1 FORMAL PARAMETERS:
106 0676 1
107 0677 1 NONE
108 0678 1
109 0679 1 IMPLICIT INPUTS:
110 0680 1
111 0681 1 EDT$$G_CAS_FLG
112 0682 1 EDT$$G_NOS
113 0683 1 EDT$$A_WK_LN
114 0684 1 EDT$$G_EXE_SBITS
115 0685 1 EDT$$Z_EXE_SBLK
116 0686 1 EDT$$G_TI_WID
117 0687 1 EDT$$G_FMT_LNPOS
118 0688 1
119 0689 1 IMPLICIT OUTPUTS:
120 0690 1
121 0691 1 NONE
122 0692 1
123 0693 1 ROUTINE VALUE:
124 0694 1
125 0695 1 Same as EDT$$FMT_CRLF
126 0696 1
127 0697 1 SIDE EFFECTS:
128 0698 1
129 0699 1 NONE
130 0700 1
131 0701 1 --
132 0702 1
133 0703 2 BEGIN
134 0704 2
135 0705 2 EXTERNAL ROUTINE
136 0706 2 EDT$$FMT_CH : NOVALUE, ! Output a character
137 0707 2 EDT$$FMT_CHWID, ! Compute a character's width
138 0708 2 EDT$$FILC_LNO : NOVALUE, ! Output a line number
139 0709 2 EDT$$FMT_TEXT : NOVALUE, ! Output an [EOB] or other text selected by the user
140 0710 2 EDT$$FMT_CRLF, ! End of the output line
141 0711 2 EDT$$STOP_WKINGMSG : NOVALUE; ! Turn off the "working" message
142 0712 2
143 0713 2 EXTERNAL
144 0714 2 EDT$$G_CAS_FLG, ! Upper and lower-case flagging
145 0715 2 EDT$$G_NOS, ! Print line numbers
146 0716 2 EDT$$A_WK_LN : REF LIN_BLOCK, ! Current line of text in the work file
147 0717 2 EDT$$Z_EOB_LN, ! [EOB] line
148 0718 2 EDT$$G_EXE_SBITS, ! The options switches.
149 0719 2 EDT$$Z_EXE_SBLK : REF NODE_BLOCK, ! The option switch value block.

```

```
150 0720 2
151 L 0721 2 %IF SUPPORT_VT220
152 0722 2 %THEN
153 0723 2 EDT$$B_CHAR_INFO : BLOCKVECTOR [256, 1, BYTE], ! Information about characters
154 0724 2 %FI
155 0725 2
156 0726 2 EDT$$G_TI_WID, ! Width of the terminal
157 0727 2 EDT$$G_FMT_LNPOS; ! Horizontal position for formatting purposes
158 0728 2
159 0729 2 LOCAL
160 0730 2 POINTER, ! Pointer into the line
161 0731 2 LEN; ! Number of characters to print
162 0732 2
163 0733 2 !+
164 0734 2 Make sure the "working" message is not still printing.
165 0735 2 !-
166 0736 2 EDT$$STOP_WKINGMSG ();
167 0737 2 !+
168 0738 2 Check for the special case of end of buffer and display [EOB] or user's string
169 0739 2 !-
170 0740 2
171 0741 2 IF (.EDT$$A_WK_LN EQLA EDT$$Z_EOB_LN)
172 0742 2 THEN
173 0743 2 EDT$$FMT_TEXT (0)
174 0744 2 ELSE
175 0745 2 BEGIN
176 0746 2 !+
177 0747 2 Print the line number if numbers are enabled.
178 0748 2 !-
179 0749 2
180 0750 2 IF .EDT$$G_NOS THEN EDT$$FILL_LNO ();
181 0751 2
182 0752 2 !+
183 0753 2 Get the length of the line to display. Normally this is just the
184 0754 2 length of the line, but if the /BRIEF option is on, we need to
185 0755 2 minimize it with the value of the qualifier.
186 0756 2 !-
187 0757 2 LEN = .EDT$$A_WK_LN [LIN_LENGTH];
188 0758 2
189 0759 2 IF .EDT$$G_EXE_SBITS<OPB_BRIEF>
190 0760 2 THEN
191 0761 2 BEGIN
192 0762 2 !+
193 0763 2 If no value was specified, use 10 for the length.
194 0764 2 !-
195 0765 2
196 0766 2 IF (.EDT$$Z_EXE_SBLK [SW_VAL1] EQL 0) THEN EDT$$Z_EXE_SBLK [SW_VAL1] = 10;
197 0767 2
198 0768 2 IF (.EDT$$Z_EXE_SBLK [SW_VAL1] LSSU .LEN) THEN LEN = .EDT$$Z_EXE_SBLK [SW_VAL1];
199 0769 2
200 0770 2 END;
201 0771 2
202 0772 2 !+
203 0773 2 Get a pointer to the line.
204 0774 2 !-
205 0775 2 POINTER = EDT$$A_WK_LN [LIN_TEXT];
206 0776 2 !+
```

```
207 0777 3 | Increment our way through the line.
208 0778 3 | -
209 0779 3 |
210 0780 3 |     INCR I FROM 1 TO .LEN DO
211 0781 4 |     BEGIN
212 0782 4 |
213 0783 4 |     LOCAL
214 0784 4 |         FLAG_CHAR,           ! 1 = we must flag the character
215 0785 4 |         CHAR,               ! The character to print
216 0786 4 |         NEW_POS;
217 0787 4 |
218 0788 4 |     CHAR = CH$RCHAR_A (POINTER);
219 0789 4 |
220 0790 4 |     Determine if the character needs to be flagged with
221 0791 4 |     a leading apostrophe.
222 0792 4 |     -
223 0793 4 |
224 0794 4 |     CASE .EDT$$G_CAS_FLG FROM 0 TO 2 OF
225 0795 4 |     SET
226 0796 4 |
227 0797 4 |     [0] :                               ! No case flagging
228 0798 5 |         BEGIN
229 0799 5 |         FLAG_CHAR = 0;
230 0800 4 |         END;
231 0801 4 |
232 0802 4 |     [1] :                               ! Flag upper case letters
233 0803 5 |         BEGIN
234 0804 5 |
235 L 0805 5 | %IF SUPPORT_VT220
236 0806 5 | %THEN
237 0807 5 |
238 0808 5 |         IF .EDT$$B_CHAR_INFO [.CHAR, 0, 1, 1, 0] ! Upper case letter
239 0809 5 |         THEN
240 U 0810 5 | %ELSE
241 U 0811 5 |
242 U 0812 5 |         IF ((.CHAR GEQ %C'A') AND (.CHAR LEQ %C'Z'))
243 U 0813 5 |         THEN
244 0814 5 |         %FI
245 0815 5 |
246 0816 5 |         FLAG_CHAR = 1
247 0817 5 |         ELSE
248 0818 5 |         FLAG_CHAR = 0;
249 0819 5 |
250 0820 4 |         END;
251 0821 4 |
252 0822 4 |     [2] :                               ! Flag lower case letters
253 0823 5 |         BEGIN
254 0824 5 |
255 L 0825 5 | %IF SUPPORT_VT220
256 0826 5 | %THEN
257 0827 5 |
258 0828 5 |         IF .EDT$$B_CHAR_INFO [.CHAR, 0, 0, 1, 0] ! Lower case letter
259 0829 5 |         THEN
260 U 0830 5 | %ELSE
261 U 0831 5 |
262 U 0832 5 |         IF ((.CHAR GEQ %C'a') AND (.CHAR LEQ %C'z'))
263 U 0833 5 |         THEN
```

```

: 264 0834 5 %FI
: 265 0835 5
: 266 0836 5
: 267 0837 5
: 268 0838 5
: 269 0839 5
: 270 0840 4
: 271 0841 4
: 272 0842 4
: 273 0843 4
: 274 0844 4
: 275 0845 4
: 276 0846 4
: 277 0847 4
: 278 0848 4
: 279 0849 4
: 280 0850 4
: 281 0851 4
: 282 0852 4
: 283 0853 4
: 284 0854 4
: 285 0855 4
: 286 0856 4
: 287 0857 4
: 288 0858 4
: 289 0859 4
: 290 0860 4
: 291 0861 4
: 292 0862 4
: 293 0863 3
: 294 0864 3
: 295 0865 2
: 296 0866 2
: 297 0867 2
: 298 0868 2
: 299 0869 2
: 300 0870 2
: 301 0871 1

        FLAG_CHAR = 1
    ELSE
        FLAG_CHAR = 0;

    END;

[OUTRANGE] :
    ASSERT (0);
TES;

+ See if there is room for this character on the line.  If not, break the line here.
-
    NEW_POS = .EDT$$G_FMT_LNPOS + .FLAG_CHAR + EDT$$FMT_CHWID (.CHAR, .EDT$$G_FMT_LNPOS + .FLAG_CHAR
    IF (.NEW_POS GEQ .EDT$$G_TI_WID) THEN EDT$$FMT_CRLF ();

+ Output the flag if needed.
-
    IF .FLAG_CHAR THEN EDT$$FMT_CH (%C'');

+ And, finally, output the character.
-
    EDT$$FMT_CH (.CHAR);
    END;

    END;

+ Terminate the line.
-
    RETURN (EDT$$FMT_CRLF ());

    END;

```

! of routine EDT\$TY_CURLN

```

.TITLE EDT$LXPRINT EDT$LXPRINT - type the current line
        on the term
.IDENT  \V04-000\
.EXTRN EDT$$FMT_CH, EDT$$FMT_CHWID
.EXTRN EDT$$FILC_LNO, EDT$$FMT_TEXT
.EXTRN EDT$$FMT_CRLF, EDT$$STOP_WKINGMSG
.EXTRN EDT$$G_CAS_FLG, EDT$$G_NOS
.EXTRN EDT$$A_WK_CN, EDT$$Z_EOB_LN
.EXTRN EDT$$G_EXE_SBITS
.EXTRN EDT$$Z_EXE_SBLK
.EXTRN EDT$$B_CHAR_INFO
.EXTRN EDT$$G_TI_WID, EDT$$G_FMT_LNPOS
.EXTRN EDT$$INTER_ERR

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

```


	OFFC	00000	.ENTRY	EDT\$STY_CURLN, Save R2,R3,R4,R5,R6,R7,R8,-				
				R9,R10,R11	0665			
	5B	00000000G	00	9E 00002	MOVAB	EDT\$FMT_CH, R11		
	5A	00000000G	00	9E 00009	MOVAB	EDT\$FMT_CRLF, R10		
	59	00000000G	00	9E 00010	MOVAB	EDT\$B_CHAR_INFO, R9		
	58	00000000G	00	9E 00017	MOVAB	EDT\$A_WK_LN, R8		
00000000G	00		00	FB 0001E	CALLS	#0, EDT\$STOP_WKINGMSG	0736	
	50	00000000G	00	9E 00025	MOVAB	EDT\$Z_EOB_LN, R0	0741	
	50		68	D1 0002C	CPL	EDT\$A_WK_LN, R0		
			0C	12 0002F	BNEQ	1\$		
			7E	D4 00031	CLRL	-(SP)	0743	
00000000G	00		01	FB 00033	CALLS	#1, EDT\$FMT_TEXT		
			009D	31 0003A	BRW	15\$		
	07	00000000G	00	E9 0003D	1\$:	BLBC	EDT\$G_NOS, 2\$	0750
00000000G	00		00	FB 00044	CALLS	#0, EDT\$FILL_LNO		
	52		68	D0 0004B	2\$:	MOVL	EDT\$A_WK_LN, R2	0757
	56		62	9A 0004E	MOVZBL	(R2), [EN		
19 00000000G	00		02	E1 00051	BBC	#2, EDT\$G_EXE_SBITS, 4\$	0759	
	50	00000000G	00	D0 00059	MOVL	EDT\$Z_EXE_SBLK, R0	0766	
	50		08	C0 00060	ADDL2	#8, R0		
			60	D5 00063	TSTL	(R0)		
			03	12 00065	BNEQ	3\$		
	60		0A	D0 00067	MOVL	#10, (R0)		
	56		60	D1 0006A	3\$:	CPL	(R0), LEN	0768
			03	1E 0006D	BGEQU	4\$		
	56		60	D0 0006F	MOVL	(R0), LEN		
	52		07	C0 00072	4\$:	ADDL2	#7, POINTER	0775
			57	D4 00075	CLRL	I	0780	
			5D	11 00077	BRB	14\$		
	53		82	9A 00079	5\$:	MOVZBL	(POINTER)+, CHAR	0788
02	00	00000000G	00	CF 0007C	CASEL	EDT\$G_CAS_FLG, #0, #2	0794	
0016	000F		0020	0C084	6\$:	.WORD	10\$-6\$, -	
							7\$-6\$, -	
							8\$-6\$	
00000000G	00		00	FB 0008A	CALLS	#0, EDT\$INTER_ERR	0843	
			13	11 00091	BRB	11\$	0794	
07	6943		01	E0 00093	7\$:	BBS	#1, EDT\$B_CHAR_INFO[CHAR], 9\$	0808
			0A	11 00098	BRB	10\$	0818	
05	6943		00	E1 0009A	8\$:	BBC	#0, EDT\$B_CHAR_INFO[CHAR], 10\$	0828
	55		01	D0 0009F	9\$:	MOVL	#1, FLAG_CHAR	0836
			02	11 000A2	BRB	11\$		
			55	D4 000A4	10\$:	CLRL	FLAG_CHAR	0838
	50	00000000G	00	D0 000A6	11\$:	MOVL	EDT\$G_FMT_LNPOS, R0	0849
54	50		55	C1 000AD	ADDL3	FLAG_CHAR, R0, R4		
			18	BB 000B1	PUSHR	#^M<R3,R4>		
00000000G	00		02	FB 000B3	CALLS	#2, EDT\$FMT_CHWID		
	50		54	C0 000BA	ADDL2	R4, NEW_POS		
00000000G	00		50	D1 000BD	CPL	NEW_POS, EDT\$G_TI_WID	0851	
			03	19 000C4	BLSS	12\$		
	6A		00	FB 000C6	CALLS	#0, EDT\$FMT_CRLF		
	05		55	E9 000C9	12\$:	BLBC	FLAG_CHAR, 13\$	0857
			27	DD 000CC	PUSHL	#39		
	6B		01	FB 000CE	CALLS	#1, EDT\$FMT_CH		
			53	DD 000D1	13\$:	PUSHL	CHAR	0862
	6B		01	FB 000D3	CALLS	#1, EDT\$FMT_CH		
9F	57		56	F3 000D6	14\$:	AOBLEQ	LEN, I, 5\$	0780
	6A		00	FB 000DA	15\$:	CALLS	#0, EDT\$FMT_CRLF	0870

EDT\$LXPRINT
V04-000

EDT\$LXPRINT - type the current line on the term ^{C 5} 16-Sep-1984 01:06:29
EDT\$\$TY_CURLN - type the current line on the t 14-Sep-1984 12:23:49

VAX-11 Bliss-32 V4.0-742 Page 8
DISK\$VMSMASTER:[EDT.SRC]LXPRINT.BLI;1 (3)

04 000DD RET

; 0871

; Routine Size: 222 bytes, Routine Base: _EDT\$CODE + 0000

: 302 0872 1
: 303 0873 1 !<BLF/PAGE>

: 305 0874 1 END
: 306 0875 1
: 307 0876 0 ELUDOM

' of module EDTSLXPRINT

PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	222	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	59	15	40	00:00.2
-\$255\$DUA28:[EDT.SRC]PSECTS.L32;1	2	1	50	7	00:00.1
-\$255\$DUA28:[EDT.SRC]SUPPORTS.L32;1	2	1	50	5	00:00.1

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LISS:LXPRINT/OBJ=OBJ\$:LXPRINT MSRCS:LXPRINT.BLI/UPDATE=(ENHS:LXPRINT)

: Size: 222 code + 0 data bytes
: Run Time: 00:16.0
: Elapsed Time: 00:19.9
: Lines/CPU Min: 3285
: Lexemes/CPU-Min: 9821
: Memory Used: 109 pages
: Compilation Complete

MCGETLIN LIS	MCRIGHT LIS	MCCHANGE LIS	MACCAL LIS	NOOPEN LIS	PRAPPNUM LIS	PRGETTOK LIS	PRISTOK LIS
MCDOWN LIS	MCLEFT LIS	MCBOTTOM LIS	LXPRINT LIS	MCTOP LIS	MSGTXT LIS	PAUDIT LIS	PRGETCHR LIS
LXCOM LIS	MAIN LIS	MCUP LIS					