

.....

```

LL      CCCCCCCC LL      EEEEEEEEEE  AAAAAA  RRRRRRRR
LL      CCCCCCCC LL      EEEEEEEEEE  AAAAAA  RRRRRRRR
LL      CC          LL      EE          AA      AA  RR      RR
LL      CC          LL      EE          AA      AA  RR      RR
LL      CC          LL      EE          AA      AA  RR      RR
LL      CC          LL      EE          AA      AA  RR      RR
LL      CC          LL      EEEEEEEEEE  AA      AA  RRRRRRRR
LL      CC          LL      EEEEEEEEEE  AA      AA  RRRRRRRR
LL      CC          LL      EE          AAAAAAAAAA RR      RR
LL      CC          LL      EE          AAAAAAAAAA RR      RR
LL      CC          LL      EE          AA      AA  RR      RR
LL      CC          LL      EE          AA      AA  RR      RR
LLLLLLLLLLLL  CCCCCCCC LLLLLLLLLL  EEEEEEEEEE  AA      AA  RR      RR
LLLLLLLLLLLL  CCCCCCCC LLLLLLLLLL  EEEEEEEEEE  AA      AA  RR      RR

```

```

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$LCLEAR - CLEAR command'
2 0002 0 MODULE EDT$LCLEAR ( ! CLEAR command
3 0003 0 IDENT = 'V04-000' ! File: LCLEAR.BLI Edit: JBS1009
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 executes the line mode CLEAR command.
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 02-FEB-1981. This module was created by
45 0045 1 extracting the routine EDT$CLR_CMD from the routine EXEC.BLI.
46 0046 1 1-002 - Regularize headers. JBS 19-Mar-1981
47 0047 1 1-003 - Implement virtual memory deallocation TMV 5-Aug-81
48 0048 1 1-004 - Close the input buffer on a CLEAR MAIN. SMB 3-May-1982
49 0049 1 1-005 - Regularize format. JBS 04-May-1982
50 0050 1 1-006 - Flag the screen as changed if we delete the current buffer. JBS 07-Jul-1982
51 0051 1 1-007 - Change EDT$SG_SCR_CHGD to EDT$SG_SCR_REBUILD. JBS 09-Oct-1982
52 0052 1 1-008 - Don't let EDT$SA_PRV_BUF point to limbo. JBS 14-Apr-1983
53 0053 1 1-009 - When deleting a buffer, put all its blocks on the available list. JBS 20-Jun-1983
54 0054 1 --
55 0055 1

```

```
57 0056 1 %SBTTL 'Declarations'  
58 0057 1  
59 0058 1 : TABLE OF CONTENTS:  
60 0059 1 :  
61 0060 1  
62 0061 1 REQUIRE 'EDT$SRC:TRAROUNAM';  
63 0500 1  
64 0501 1 FORWARD ROUTINE  
65 0502 1 EDT$CLR_CMD : NOVALUE;  
66 0503 1  
67 0504 1 :  
68 0505 1 : INCLUDE FILES:  
69 0506 1 :  
70 0507 1  
71 0508 1 REQUIRE 'EDT$SRC:EDTREQ';  
72 0643 1  
73 0644 1 :  
74 0645 1 : MACROS:  
75 0646 1  
76 0647 1 : NONE  
77 0648 1 :  
78 0649 1 : EQUATED SYMBOLS:  
79 0650 1 :  
80 0651 1  
81 0652 1 EXTERNAL LITERAL  
82 0653 1 EDT$K_INPUT_FILE,  
83 0654 1 EDT$K_CLOSE;  
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
```

! Process the CLEAR command

```

: 93 0663 1 %SBTTL 'EDT$$CLR_CMD - CLEAR command'
: 94 0664 1
: 95 0665 1 GLOBAL ROUTINE EDT$$CLR_CMD          ! CLEAR line-mode command
: 96 0666 1   : NOVALUE =
: 97 0667 1
: 98 0668 1 !++
: 99 0669 1 ! FUNCTIONAL DESCRIPTION:
100 0670 1
101 0671 1     Command processing for CLEAR.  Position to the buffer to
102 0672 1     be cleared, then call the work file system to clear it.
103 0673 1     If we are deleting the buffer header, delete the first block
104 0674 1     of the buffer also.
105 0675 1
106 0676 1 ! FORMAL PARAMETERS:
107 0677 1
108 0678 1     NONE
109 0679 1
110 0680 1 ! IMPLICIT INPUTS:
111 0681 1
112 0682 1     EDT$$A_CUR_BUF
113 0683 1     EDT$$A_EXE_CURCMD
114 0684 1     EDT$$A_WK_BUK
115 0685 1     EDT$$G_WK_AVAIL
116 0686 1     EDT$$G_WK_CURBUK
117 0687 1
118 0688 1 ! IMPLICIT OUTPUTS:
119 0689 1
120 0690 1     EDT$$G_SCR_REBUILD
121 0691 1     EDT$$A_WK_BUK
122 0692 1     EDT$$G_WK_AVAIL
123 0693 1     EDT$$G_WK_CURBUK
124 0694 1     EDT$$A_PRV_BUF
125 0695 1
126 0696 1 ! ROUTINE VALUE:
127 0697 1
128 0698 1     NONE
129 0699 1
130 0700 1 ! SIDE EFFECTS:
131 0701 1
132 0702 1     The input file is closed (if not already) when a CLEAR MAIN occurs.
133 0703 1
134 0704 1 !--
135 0705 1
136 0706 2 BEGIN
137 0707 2
138 0708 2 EXTERNAL ROUTINE
139 0709 2     EDT$$FMT_MSG,          ! Print a message
140 0710 2     EDT$$CALCFIO,        ! Do file I/O
141 0711 2     EDT$$CEA_HEAP,      ! Deallocate heap storage
142 0712 2     EDT$$FND_BUF,      ! Find a buffer
143 0713 2     EDT$$WF_CLRBUF,     ! Clear a buffer
144 0714 2     EDT$$RD_CURLN;     ! Read the current line
145 0715 2
146 0716 2 EXTERNAL
147 0717 2     EDT$$G_INPUT_OPN,      ! input file is still open
148 0718 2     EDT$$G_SCR_REBUILD,   ! Set if the screen must be rebuilt
149 0719 2     EDT$$A_BUF_LST,       ! The list of buffers

```

```
150 0720 2      EDT$$A_CUR_BUF : REF TBCB_BLOCK,      ! Pointer to the current buffer
151 0721 2      EDT$$A_EXE_CURCMD : REF NODE_BLOCK, ! Pointer to the current command.
152 0722 2      EDT$$A_PRV_BUF : REF TBCB_BLOCK, ! Pointer to the previous buffer
153 0723 2      EDT$$A_WK_BUK : REF BLOCK-[WF_BUKT_SIZE, BYTE] FIELD (WFB_FIELDS), ! Current bucket
154 0724 2      EDT$$G_WK_AVAIL, ! Next available bucket
155 0725 2      EDT$$G_WK_CURBUK; ! Current bucket
156 0726 2
157 0727 2      MESSAGES ((INPFILCLO));
158 0728 2
159 0729 2      LOCAL
160 0730 2      LEN : BYTE,
161 0731 2      SAVE_TBCB,
162 0732 2      PREV : REF TBCB_BLOCK,
163 0733 2      NEXT : REF TBCB_BLOCK,
164 0734 2      RANGE : REF NODE_BLOCK;
165 0735 2
166 0736 2      BIND
167 0737 2      MAIN_TXT = UPLIT (BYTE ('MAIN')),
168 0738 2      PASTE_TXT = UPLIT (BYTE ('PASTE'));
169 0739 2
170 0740 2      !+
171 0741 2      ! Save the current TBCB so we can get back.
172 0742 2      !-
173 0743 2      SAVE_TBCB = .EDT$$A_CUR_BUF;
174 0744 2      RANGE = .EDT$$A_EXE_CURCMD [RANGE1];
175 0745 2      !+
176 0746 2      ! Did we find or allocate a buffer successfully?
177 0747 2      !-
178 0748 2
179 0749 2      IF EDT$$FND_BUF (.RANGE [BUF_NAME], .RANGE [BUF_LEN])
180 0750 2      THEN
181 0751 2      BEGIN
182 0752 2      !+
183 0753 2      ! The MAIN and the PASTE buffers cannot be deleted, they can be cleared only.
184 0754 2      !-
185 0755 2
186 0756 3      IF (CH$EQL (.RANGE [BUF_LEN], .RANGE [BUF_NAME], 4, MAIN_TXT)) OR !
187 0757 4      (CH$EQL (.RANGE [BUF_LEN], .RANGE [BUF_NAME], 5, PASTE_TXT))
188 0758 3      THEN
189 0759 4      BEGIN
190 0760 4      !+
191 0761 4      ! If MAIN is being CLEARed and the input file is still open, then close it.
192 0762 4      !-
193 0763 4
194 0764 5      IF ((CH$EQL (.RANGE [BUF_LEN], .RANGE [BUF_NAME], 4, MAIN_TXT)) AND .EDT$$G_INPUT_OPN)
195 0765 4      THEN
196 0766 4
197 0767 4      IF EDT$$CALLFIO (EDT$K_CLOSE, EDT$K_INPUT_FILE, 0, 0)
198 0768 4      THEN
199 0769 5      BEGIN
200 0770 5      EDT$$A_CUR_BUF [TBCB_INPUT_RAB] = 0;
201 0771 5      EDT$$G_INPUT_OPN = 0;
202 0772 5      END
203 0773 4      ELSE
204 0774 4      EDT$$FMT_MSG (EDT$ _INPFILCLO);
205 0775 4
206 0776 4      EDT$$WF_CLRBUF (); ! Delete all but first block
```

```
207 0777 4      END
208 0778 3      ELSE
209 0779 4      BEGIN
210 0780 4      EDT$$WF_CLRBUF ();          ! Delete all but first block
211 0781 4      +
212 0782 4      | Put the first block of the buffer on the available list.
213 0783 4      |
214 0784 4      | EDT$$A_WK_BUK [WFB_NEXT_BUKT] = .EDT$$G_WK_AVAIL;
215 0785 4      | EDT$$G_WK_AVAIL = .EDT$$G_WK_CURBUK;
216 0786 4      +
217 0787 4      | Delete the buffer by disconnecting the TBCB header from the list
218 0788 4      |
219 0789 4      | PREV = .EDT$$A_CUR_BUF [TBCB_PREV_BUF];
220 0790 4      | NEXT = .EDT$$A_CUR_BUF [TBCB_NEXT_BUF];
221 0791 4      +
222 0792 4      | If we want to clear the first buffer which is not MAIN we have to
223 0793 4      | readjust the buffer list header pointer.
224 0794 4      |
225 0795 4      |
226 0796 4      | IF (.PREV EQLA 0) THEN EDT$$A_BUF_LST = .NEXT;
227 0797 4      |
228 0798 4      +
229 0799 4      | Unlink the deleted buffer header
230 0800 4      |
231 0801 4      |
232 0802 4      | IF (.NEXT NEQA 0) THEN NEXT [TBCB_PREV_BUF] = .PREV;
233 0803 4      |
234 0804 4      | IF (.PREV NEQA 0) THEN PREV [TBCB_NEXT_BUF] = .NEXT;
235 0805 4      |
236 0806 4      +
237 0807 4      | Get the name length, and the beginning address, then deallocate header.
238 0808 4      |
239 0809 4      | LEN = .EDT$$A_CUR_BUF [TBCB_NAME_LEN];
240 0810 4      | EDT$$DEA_HEAP (%REF (TBCB_SIZE + .LEN), EDT$$A_CUR_BUF);
241 0811 3      | END;
242 0812 3      |
243 0813 3      +
244 0814 3      | If the current buffer other than MAIN was deleted, go to the MAIN buffer and don't let
245 0815 3      | LAST point to a deleted buffer. Also, if the current buffer was deleted then the
246 0816 3      | text part of the screen must be rebuilt.
247 0817 3      |
248 0818 3      |
249 0819 4      | IF (.EDT$$A_CUR_BUF EQLA .SAVE_TBCB)
250 0820 3      | THEN
251 0821 4      | BEGIN
252 0822 4      | EDT$$FND_BUF (MAIN_TXT, 4);
253 0823 4      | EDT$$A_PRV_BUF = .EDT$$A_CUR_BUF;
254 0824 4      | EDT$$G_SCR_REBUILD = 1;
255 0825 4      | END
256 0826 4      +
257 0827 4      | Otherwise go back to the original buffer.
258 0828 4      |
259 0829 3      | ELSE
260 0830 3      | EDT$$A_CUR_BUF = .SAVE_TBCB;
261 0831 3      |
262 0832 2      | END;
263 0833 2
```

: 264
: 265

0834 2 EDT\$\$RD_CURLN ();
0835 1 END;

! of routine EDT\$\$CLR_CMD

.TITLE EDT\$LCLEAR EDT\$LCLEAR - CLEAR command
.IDENT \V04-000\

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

45 4E 49 41 4D 00000 P.AAA: .ASCII \MAIN\
54 53 41 50 00004 P.AAB: ASCII \PASTE\
:

MAIN_TXT= P.AAA
PASTE_TXT= P.AAB

.EXTRN EDT\$K_INPUT_FILE
.EXTRN EDT\$K_CLOSE, EDT\$\$FMT MSG
.EXTRN EDT\$\$CALLFIO, EDT\$\$DEA HEAP
.EXTRN EDT\$\$FND_BUF, EDT\$\$WF_CLRBUF
.EXTRN EDT\$\$RD_CURLN, EDT\$\$G_INPUT_OPN
.EXTRN EDT\$\$G_SCR_REBUILD
.EXTRN EDT\$\$A_BUF_LST, EDT\$\$A_CUR_BUF
.EXTRN EDT\$\$A_EXE_CURCMD
.EXTRN EDT\$\$A_PRV_BUF, EDT\$\$A_WK_BUK
.EXTRN EDT\$\$G_WK_AVAIL
.EXTRN EDT\$\$G_WK_CURBUK
.EXTRN EDT\$_INPFILCLO

OFFC 00000

.ENTRY EDT\$\$CLR_CMD, Save R2,R3,R4,R5,R6,R7,R8,R9,-; R10,R11 0665

5B 00000000G 00 9E 00002
5A 00000000G 00 9E 00009
59 00000000G 00 9E 00010
58 DD AF 9E 00017
57 00000000G 00 9E 0001B
56 00000000G 00 9E 00022
5E 04 C2 00029
55 66 D0 0002C
50 00000000G 00 D0 0002F
54 04 A0 D0 00036
7E 08 A4 7D 0003A
67 02 FB 0003E
03 50 E8 00041
00C8 31 00044
04 00 08 B4 0C A4 2D 00047 1\$:
68 0004E
0B 13 0004F
05 00 08 B4 0C A4 2D 00051
04 A8 00058
41 12 0005A
04 00 08 B4 0C A4 2D 0005C 2\$:
68 00063
32 12 00064
2F 69 E9 00066
7E 7C 00069
00000000G 8F DD 0006B
00000000G 8F DD 00071
00000000G 00 04 FB 00077

MOVAB EDT\$\$G_WK_AVAIL, R11
MOVAB EDT\$\$WF_CLRBUF, R10
MOVAB EDT\$\$G_INPUT_OPN, R9
MOVAB MAIN_TXT, R8
MOVAB EDT\$\$FND_BUF, R7
MOVAB EDT\$\$A_CUR_BUF, R6
SUBL2 #4, SP
MOVL EDT\$\$A_CUR_BUF, SAVE TBCB 0743
MOVL EDT\$\$A_EXE_CURCMD, R0 0744
MOVL 4(R0), RANGE
MOVQ 8(RANGE), -(SP) 0749
CALLS #2, EDT\$\$FND_BUF
BLBS R0, 1\$
BRW 11\$
CMPCS 12(RANGE), @8(RANGE), #0, #4, MAIN_TXT 0756
BEQL 2\$
CMPCS 12(RANGE), @8(RANGE), #0, #5, PASTE_TXT 0757
BNEQ 5\$
CMPCS 12(RANGE), @8(RANGE), #0, #4, MAIN_TXT 0764
BNEQ 4\$
BLBC EDT\$\$G_INPUT_OPN, 4\$
CLRQ -(SP) 0767
PUSHL #EDT\$K_INPUT_FILE
PUSHL #EDT\$K_CLOSE
CALLS #4, EDT\$\$CALLFIO

0A	50	E9	0007E	BLBC	R0, 3\$	
50		66	D0 00081	MOVL	EDT\$\$A_CUR_BUF, R0	0770
	2A	A0	94 00084	CLRB	42(R0)	
		69	D4 00087	CLRL	EDT\$\$G_INPUT_OPN	0771
		0D	11 00089	BRB	4\$	0767
00000000G	00	8F	DD 0008B	3\$: PUSHL	#EDT\$ INPFILCLO	0774
		01	FB 00091	CALLS	#1, EDT\$\$FMT_MSG	
	6A	00	FB 00098	4\$: CALLS	#0, EDT\$\$WF_CLRBUF	0776
		53	11 0009B	BRB	9\$	0756
	6A	00	FB 0009D	5\$: CALLS	#0, EDT\$\$WF_CLRBUF	0780
	50	00	D0 000A0	MOVL	EDT\$\$A_WK_BUF, R0	0784
02	A0	6B	B0 000A7	MOVW	EDT\$\$G_WK_AVAIL, 2(R0)	
	6B	00	D0 000AB	MOVL	EDT\$\$G_WK_CURBUF, EDT\$\$G_WK_AVAIL	0785
	50	66	D0 000B2	MOVL	EDT\$\$A_CUR_BUF, R0	0789
	52	22	A0 D0 000B5	MOVL	34(R0), PREV	
	51	26	A0 D0 000B9	MOVL	38(R0), NEXT	0790
		52	D5 000BD	TSTL	PREV	0796
		07	12 000BF	BNEQ	6\$	
00000000G	00	51	D0 000C1	MOVL	NEXT, EDT\$\$A_BUF_LST	
		51	D5 000C8	6\$: TSTL	NEXT	0802
		04	13 000CA	BEQL	7\$	
	22	A1	52 D0 000CC	MOVL	PREV, 34(NEXT)	
		52	D5 000D0	7\$: TSTL	PREV	0804
		04	13 000D2	BEQL	8\$	
	26	A2	51 D0 000D4	MOVL	NEXT, 38(PREV)	
	51	2C	A0 90 000D8	8\$: MOVB	44(R0), LEN	0809
		56	DD 000DC	PUSHL	R6	0810
	04	AE	51 9A 000DE	MOVZBL	LEN, 4(SP)	
	04	AE	2D C0 000E2	ADDL2	#45, 4(SP)	
		04	AE 9F 000E6	PUSHAB	4(SP)	
00000000G	00	02	FB 000E9	CALLS	#2, EDT\$\$DEA_HEAP	
	55	66	D1 000F0	9\$: CMPL	EDT\$\$A_CUR_BUF, SAVE_TBCB	0819
		17	12 000F3	BNEQ	10\$	
		04	DD 000F5	PUSHL	#4	0822
		58	DD 000F7	PUSHL	R8	
	67	02	FB 000F9	CALLS	#2, EDT\$\$FND_BUF	
00000000G	00	66	D0 000FC	MOVL	EDT\$\$A_CUR_BUF, EDT\$\$A_PRV_BUF	0823
00000000G	00	01	D0 00103	MOVL	#1, EDT\$\$G_SCR_REBUILD	0824
		03	11 0010A	BRB	11\$	0819
	66	55	D0 0010C	10\$: MOVL	SAVE_TBCB, EDT\$\$A_CUR_BUF	0830
00000000G	00	00	FB 0010F	11\$: CALLS	#0, EDT\$\$RD_CURLN	0834
		04	00116	RET		0835

; Routine Size: 279 bytes, Routine Base: _EDT\$CODE + 0009

; 266 0836 1
; 267 0837 1 !<BLF/PAGE

EDT\$LCLEAR
V04-000

EDT\$LCLEAR - CLEAR command
EDT\$\$CLR_CMD - CLEAR command

C 13
16-Sep-1984 00:46:39 VAX-11 Bliss-32 V4.0-742 Page 8
14-Sep-1984 12:23:30 DISK\$VMSMASTER:[EDT.SRC]LCLEAR.BLI;1 (4)

: 269 0838 1 END
: 270 0839 1
: 271 0840 0 ELUDOM

! of module EDT\$LCLEAR

EDT
V04

PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	288	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	80	21	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=LISS:LCLEAR/OBJ=OBJ\$:LCLEAR MSRCS:LCLEAR.BLI/UPDATE=(ENHS:LCLEAR)

: Size: 279 code + 9 data bytes
: Run Time: 00:16.9
: Elapsed Time: 00:19.6
: Lines/CPU Min: 2991
: Lexemes/CPU-Min: 10433
: Memory Used: 123 pages
: Compilation Complete

