


```

LL      IIIIII  NN      NN      SSSSSSSS  EEEEEEEEEEE  RRRRRRRR  TTTTTTTTTT
LL      IIIIII  NN      NN      SSSSSSSS  EEEEEEEEEEE  RRRRRRRR  TTTTTTTTTT
LL      II      NN      NN      SS          EE          RR          TT
LL      II      NN      NN      SS          EE          RR          TT
LL      II      NNNN     NN      SS          EE          RR          TT
LL      II      NNNN     NN      SS          EE          RR          TT
LL      II      NN      NN      SSSSSS      EEEEEEEEE  RRRRRRRR  TT
LL      II      NN      NN      SSSSSS      EEEEEEEEE  RRRRRRRR  TT
LL      II      NN      NN      SS          EE          RR          TT
LL      II      NN      NN      SS          EE          RR          TT
LL      II      NN      NN      SS          EE          RR          TT
LL      II      NN      NN      SS          EE          RR          TT
LL      II      NN      NN      SS          EE          RR          TT
LLLLLLLLLLLL IIIIII  NN      NN      SSSSSSSS  EEEEEEEEEEE  RRRRRRRR  TT
LLLLLLLLLLLL IIIIII  NN      NN      SSSSSSSS  EEEEEEEEEEE  RRRRRRRR  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
LLLLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLLLL IIIIII  SSSSSSSS

```



```
0001 0 %TITLE 'EDT$INSERT - insert text'
0002 0 MODULE EDT$INSERT (
0003 0 IDENT = 'V04-000'
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1
0031 1 ++
0032 1 FACILITY: EDT -- The DEC Standard Editor
0033 1
0034 1 ABSTRACT:
0035 1
0036 1 This module executes the line mode INSERT command.
0037 1 It also provides support for the insert part of the
0038 1 REPLACE command.
0039 1
0040 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0041 1
0042 1 AUTHOR: Bob Kushlis, CREATION DATE: February 3, 1978
0043 1
0044 1 MODIFIED BY:
0045 1
0046 1 1-001 - Original. DJS 30-JAN-81. This module was created by
0047 1 extracting routines INSERT and INS_COM from EXEC.BLI.
0048 1 1-002 - Regularize headers. JBS 20-Mar-1981
0049 1 1-003 - Change INS_COM to EDT$$INS_CMD. JBS 30-Mar-1981
0050 1 1-004 - Do prompting from a global. JBS 23-Oct-1981
0051 1 1-005 - Add error check for too many lines. SMB 5-Feb-1982
0052 1 1-006 - Reset command buffer after EOF. STS 23-Jun-1982
0053 1 1-007 - Add CRLF for EXT insert and replace commands. SMB 07-Jul-1982
0054 1 1-008 - Change indentation for large line numbers. SMB 14-Jul-1982
0055 1 1-009 - Use previous line size instead of current. SMB 15-Jul-1982
0056 1 1-010 - Put code for edt$$rng_posfrst in line. STS 11-Oct-1982
0057 1 1-011 - Use new macro for comparing line numbers. STS 20-Oct-1982
```


EDT\$INSERT
V04-000

EDT\$INSERT - insert text

C 3
16-Sep-1984 00:52:19
14-Sep-1984 12:23:36

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]INSERT.BLI;1 Page 2
(1)

: 58
: 59
: 60
: 61
0058 1 : 1-012 - Don't print CRLF unless we are coming from the terminal. JBS 19-Jan-1983
0059 1 : 1-013 - Make sure select is at the beginning of the line before inserting. JBS 12-Jul-1983
0060 1 :--
0061 1

EDT\$INSERT
V04-000

EDT\$INSERT - insert text
Declarations

D 3
16-Sep-1984 00:52:19
14-Sep-1984 12:23:36

VAX-11 Bliss-32 V4.0-742
DISK\$VMMASTER:[EDT.SRC]LINSERT.BLI;1
Page 3
(2)

```
: 63      0062 1 %SBTTL 'Declarations'
: 64      0063 1
: 65      0064 1  TABLE OF CONTENTS:
: 66      0065 1
: 67      0066 1
: 68      0067 1  REQUIRE 'EDTSRC:TRAROUNAM';
: 69      0506 1
: 70      0507 1  FORWARD ROUTINE
: 71      0508 1      EDT$$INS_TXT : NOVALUE,
: 72      0509 1      EDT$$INS_CMD : NOVALUE;
: 73      0510 1
: 74      0511 1
: 75      0512 1  INCLUDE FILES:
: 76      0513 1
: 77      0514 1
: 78      0515 1  REQUIRE 'EDTSRC:EDTREQ';
: 79      0650 1
: 80      0651 1
: 81      0652 1  MACROS:
: 82      0653 1
: 83      0654 1      NONE
: 84      0655 1
: 85      0656 1  EQUATED SYMBOLS:
: 86      0657 1
: 87      0658 1      NONE
: 88      0659 1
: 89      0660 1  OWN STORAGE:
: 90      0661 1
: 91      0662 1      NONE
: 92      0663 1
: 93      0664 1  EXTERNAL REFERENCES:
: 94      0665 1
: 95      0666 1      In the routines
```

```
: Insert text
: Process the INSERT command
```


EDT\$LININSERT
V04-000

EDT\$LININSERT - insert text
EDT\$\$INS_TXT - insert text

E 3
16-Sep-1984 00:52:19
14-Sep-1984 12:23:36

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]LININSERT.BLI;1
Page 4
(3)

```

: 97 0667 1 %SBTTL 'EDT$$INS_TXT - insert text'
: 98 0668 1
: 99 0669 1 GLOBAL ROUTINE EDT$$INS_TXT          ! Insert text for INSERT and REPLACE commands
:100 0670 1 : NOVALUE =
:101 0671 1
:102 0672 1 ++
:103 0673 1 FUNCTIONAL DESCRIPTION:
:104 0674 1
:105 0675 1 This routine handles the insertion of text from an INSERT or REPLACE
:106 0676 1 command. If the current token from the parser is a ';', then the text
:107 0677 1 to be inserted is the remainder of the command line. Otherwise, loop
:108 0678 1 reading a line and inserting it until an end of file is detected.
:109 0679 1
:110 0680 1 FORMAL PARAMETERS:
:111 0681 1
:112 0682 1 NONE
:113 0683 1
:114 0684 1 IMPLICIT INPUTS:
:115 0685 1
:116 0686 1 EDT$$A_CMD_END
:117 0687 1 EDT$$G_CMD_LEN
:118 0688 1 EDT$$A_CMD_BUF
:119 0689 1 EDT$$G_INP_SRC
:120 0690 1 EDT$$G_NOS
:121 0691 1 EDT$$A_PA_CURTOK
:122 0692 1 EDT$$T_PMT_INS
:123 0693 1 EDT$$T_PMT_INSN
:124 0694 1 EDT$$T_LN_BUF
:125 0695 1
:126 0696 1 IMPLICIT OUTPUTS:
:127 0697 1
:128 0698 1 EDT$$T_CMD_BUF
:129 0699 1 EDT$$A_SEL_POS
:130 0700 1
:131 0701 1 ROUTINE VALUE:
:132 0702 1
:133 0703 1 NONE
:134 0704 1
:135 0705 1 SIDE EFFECTS:
:136 0706 1
:137 0707 1 NONE
:138 0708 1
:139 0709 1 --
:140 0710 1
:141 0711 2 BEGIN
:142 0712 2
:143 0713 2 EXTERNAL ROUTINE
:144 0714 2 EDT$$FMT_CRLF,          ! Start a new line
:145 0715 2 EDT$$TI_ORSTR,
:146 0716 2 EDT$$GET_LN,          ! Get the next input line
:147 0717 2 EDT$$END_INS,
:148 0718 2 EDT$$INS_LN,
:149 0719 2 EDT$$START_INS,
:150 0720 2 EDT$$TY_CURLN;        ! Type out the current line
:151 0721 2
:152 0722 2 EXTERNAL
:153 0723 2 EDT$$L_WK_STARTNO : LN_BLOCK,
```



```
154 0724 2 EDT$LN00 : LNOVECTOR [14],
155 0725 2 EDT$A_FMT_WRRUT,
156 0726 2 EDT$G_EXT_MOD,
157 0727 2 EDT$T_CMD_BUF,
158 0728 2 EDT$A_CMD_END,
159 0729 2 EDT$G_CMD_LEN,
160 0730 2 EDT$A_CMD_BUF,
161 0731 2 EDT$G_INP_SRC,
162 0732 2 EDT$G_NOS, ! 1 = numbers, 0 = nonnumbers
163 0733 2 EDT$A_PA_CURTOK,
164 0734 2 EDT$T_PMT_INS : VECTOR [, BYTE], ! Counted ASCII string for line-mode insert prompt
165 0735 2 EDT$T_PMT_INSN : VECTOR [, BYTE], ! Counted ASCII string for line-mode insert nonnumbers prompt
166 0736 2 EDT$T_LN_BUF, ! Line buffer
167 0737 2 EDT$A_SEC_POS, ! Position of select point
168 0738 2
169 0739 2 LOCAL
170 0740 2 BIG_LINE_PMT : VECTOR [32, BYTE], ! Prompt for lines larger than 10**5
171 0741 2 STATUS,
172 0742 2 EOF;
173 0743 2
174 0744 2 !+
175 0745 2 !- Set up the work-file system for insertion
176 0746 2
177 0747 2 EDT$START_INS ();
178 0748 2 !+
179 0749 2 !- Look for the immediate insertion signaled by a ';' following the command.
180 0750 2
181 0751 2
182 0752 2 IF (CH$RCHAR (.EDT$A_PA_CURTOK) EQL %C';')
183 0753 2 THEN
184 0754 2 BEGIN
185 0755 2 EDT$INS_LN (.EDT$A_PA_CURTOK + 1, .EDT$A_CMD_END - .EDT$A_PA_CURTOK - 1);
186 0756 2 EDT$A_CMD_BUF = .EDT$A_CMD_END;
187 0757 2 END
188 0758 2 ELSE
189 0759 2 !+
190 0760 2 !- Insert mode loop. Keep reading and inserting lines until EOF or until
191 0761 2 the maximum number of lines has been reached for this buffer
192 0762 2
193 0763 2 BEGIN
194 0764 2
195 0765 2 IF (.EDT$G_EXT_MOD AND (.EDT$A_FMT_WRRUT EQL EDT$TI_WRSTR) AND (.EDT$G_INP_SRC EQL INP_TERM))
196 0766 2 THEN
197 0767 2 EDT$FMT_CRLF ();
198 0768 2
199 0769 2 !+
200 0770 2 !- Set up the prompt for large lines just in case we need it later
201 0771 2 (only want to do this once. To do this we add 5 spaces after the
202 0772 2 <CR><LF> in the prompt.
203 0773 2
204 0774 2
205 0775 2 IF (.EDT$G_NOS EQL 1)
206 0776 2 THEN
207 0777 2 BEGIN
208 0778 2 BIG_LINE_PMT [0] = .EDT$T_PMT_INS [0] + 5;
209 0779 2 CH$MOVE 72, EDT$T_PMT_INS [1], BIG_LINE_PMT [1];
210 0780 2 CH$MOVE (.EDT$T_PMT_INS [0] - 2, EDT$T_PMT_INS [3], BIG_LINE_PMT [8]);
```



```
211 0781 4 CH$FILL (%C' ', 5, BIG_LINE_PMT [3]);
212 0782 4 END;
213 0783 3
214 0784 3 DO
215 0785 4 BEGIN
216 0786 4
217 0787 4 LOCAL
218 0788 4 PROMPT_ADDR : REF VECTOR [, BYTE];
219 0789 4
220 0790 5 IF ((CMPLNO (EDT$$L_WK_STARTNO, EDT$$L_LN00 [10]) GEQ 0) AND (.EDT$$G_NOS EQL 1))
221 0791 4 THEN
222 0792 4 PROMPT_ADDR = BIG_LINE_PMT
223 0793 4 ELSE
224 0794 5 PROMPT_ADDR = (CASE .EDT$$G_NOS FROM 0 TO 1 OF
225 0795 5 SET
226 0796 5 [0] : EDT$$T_PMT_INSN;
227 0797 5 [1] : EDT$$T_PMT_INS;
228 0798 5 [OUTRANGE] :
229 0799 6 BEGIN
230 0800 6 ASSERT (0);
231 0801 6 0
232 0802 5 END;
233 0803 4 TES);
234 0804 4
235 0805 4 EOF = EDT$$GET_LN (PROMPT_ADDR [1], .PROMPT_ADDR [0]);
236 0806 4
237 0807 5 IF (( NOT .EOF) OR (.EDT$$G_CMD_LEN NEQ 0)) !
238 0808 4 THEN
239 0809 5 BEGIN
240 0810 5 EDT$$A_SEL_POS = EDT$$T_LN_BUF;
241 0811 5 STATUS = EDT$$INS_LN (EDT$$T_CMD_BUF, .EDT$$G_CMD_LEN);
242 0812 4 END;
243 0813 4
244 0814 4 END
245 0815 3 UNTIL ((.EOF) OR ( NOT .STATUS));
246 0816 3
247 0817 2 END;
248 0818 2
249 0819 2 IF .EOF
250 0820 2 THEN
251 0821 3 BEGIN
252 0822 3 EDT$$A_CMD_END = EDT$$T_CMD_BUF;
253 0823 3 EDT$$A_CMD_BUF = EDT$$T_CMD_BUF;
254 0824 3 CH$WCHAR ('!', .EDT$$A_CMD_END);
255 0825 2 END;
256 0826 2
257 0827 2 !+
258 0828 2 Inform the work file system that the insert is complete, and print
259 0829 2 the next line.
260 0830 2 !-
261 0831 2 EDT$$END_INS ();
262 0832 2
263 0833 2 IF (.EDT$$G_INP_SRC EQL INP_TERM) THEN EDT$$TY_CURLN ();
264 0834 2
265 0835 1 END;
! of routine EDT$$INS_TXT
```


EDT\$INSERT
V04-000

EDT\$INSERT - insert text
EDT\$\$INS_TXT - insert text

H 3
16-Sep-1984 00:52:19
14-Sep-1984 12:23:36

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]INSERT.BLI;1
Page 7
(3)

.TITLE EDT\$INSERT EDT\$INSERT - insert text
.IDENT \V04-000\

.EXTRN EDT\$\$FMT_CRLF, EDT\$\$TI_WSTR
.EXTRN EDT\$\$GET_LN, EDT\$\$END_INS
.EXTRN EDT\$\$INS_LN, EDT\$\$START_INS
.EXTRN EDT\$\$TY_CURLN, EDT\$\$L_WR_STARTNO
.EXTRN EDT\$\$L_CNOO, EDT\$\$A_FMT_WRRUT
.EXTRN EDT\$\$G_EXT_MOD, EDT\$\$T_CMD_BUF
.EXTRN EDT\$\$A_CMD_END, EDT\$\$G_CMD_LEN
.EXTRN EDT\$\$A_CMD_BUF, EDT\$\$G_INP_SRC
.EXTRN EDT\$\$G_NOS, EDT\$\$A_PA_CURTOK
.EXTRN EDT\$\$T_PMT_INS, EDT\$\$T_PMT_INSN
.EXTRN EDT\$\$T_LN_BUF, EDT\$\$A_SEL_POS
.EXTRN EDT\$\$INTER_ERR

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

.ENTRY EDT\$\$INS_TXT, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 0669
R10,R11
MOVAB EDT\$\$A_CMD_BUF, R11
MOVAB EDT\$\$INS_LN, R10
MOVAB EDT\$\$T_CMD_BUF, R9
MOVAB EDT\$\$G_NOS, R8
MOVAB EDT\$\$T_PMT_INS, R7
MOVAB EDT\$\$A_CMD_END, R6
SUBL2 #32, SP
CALLS #0, EDT\$\$START_INS 0747
MOVL EDT\$\$A_PA_CURTOK, R1 0752
CMPB (R1), #59
BNEQ 1\$
SUBL3 R1, EDT\$\$A_CMD_END, R0 0755
PUSHAB -1(R0)
PUSHAB 1(R1)
CALLS #2, EDT\$\$INS_LN
MOVL EDT\$\$A_CMD_END, EDT\$\$A_CMD_BUF 0756
BRW 15\$ 0752
BLBC EDT\$\$G_EXT_MOD, 2\$ 0765
MOVAB EDT\$\$TI_WSTR, R0
CMPL EDT\$\$A_FMT_WRRUT, R0
BNEQ 2\$
TSTL EDT\$\$G_INP_SRC
BNEQ 2\$
CALLS #0, EDT\$\$FMT_CRLF 0767
CMPL EDT\$\$G_NOS, #1 0775
BNEQ 3\$
MOVZBL EDT\$\$T_PMT_INS, R0 0778
ADDB3 #5, R0, BIG_LINE_PMT
MOVW EDT\$\$T_PMT_INS+1, BIG_LINE_PMT+1 0779
SUBL2 #2, R0 0780
MOVC3 R0, EDT\$\$T_PMT_INS+3, BIG_LINE_PMT+8
MOVC5 #0, (SP), #32, #5, BIG_LINE_PMT+3 0781
MOVZWL HIGH_1, R1 0790
MOVZWL HIGH_2, R0
CMPL R1, R0
BLSSU 4\$

OFFC 00000
5B 00000000G 00 9E 00002
5A 00000000G 00 9E 00009
59 00000000G 00 9E 00010
58 00000000G 00 9E 00017
57 00000000G 00 9E 0001E
56 00000000G 00 9E 00025
5E 20 C2 0002C
00 00 00 0002F
51 00000000G 00 D0 00036
3B 61 91 0003D
13 12 00040
50 66 51 C3 00042
FF A0 9F 00046
01 A1 9F 00049
6A 02 FB 0004C
6B 66 D0 0004F
00E5 31 00052
1F 00000000G 00 E9 00055 1\$:
50 00000000G 00 9E 0005C
50 00000000G 00 D1 00063
0F 12 0006A
00000000G 00 D5 0006C
07 12 00072
50 00 00 FB 00074
01 68 D1 0007B 2\$:
1C 12 0007E
50 67 9A 00080
6E 50 05 81 00083
01 AE 01 A7 B0 00087
50 02 C2 0008C
08 AE 03 A7 50 28 0008F
20 6E 00 2C 00095
03 AE 0009A
51 00000000G 00 3C 0009C 3\$:
50 00000000G 00 3C 000A3
50 51 D1 000AA
15 1F 000AD

51	00000000G	00	1E 12 000AF	BNEQ	6\$		
50	00000000G	00	D0 000B1	MOVL	LOW_1, R1		
50		00	D0 000B8	MOVL	LOW_2, R0		
		51	D1 000BF	CMPL	R1, R0		
		05	1E 000C2	BGEQU	5\$		
50		01	CE 000C4 4\$:	MNEGL	#1, R0		
		09	11 000C7	BRB	7\$		
		04	12 000C9 5\$:	BNEQ	6\$		
		50	D4 000CB	CLRL	R0		
		03	11 000CD	BRB	7\$		
50		01	D0 000CF 6\$:	MOVL	#1, R0		
		0A	19 000D2 7\$:	BLSS	8\$		
01		68	D1 000D4	CMPL	EDT\$\$G_NOS, #1		
		05	12 000D7	BNEQ	8\$		
52		6E	9E 000D9	MOVAB	BIG_LINE_PMT, PROMPT_ADDR		0792
		1F	11 000DC	BRB	12\$		
01	00	68	CF 000DE 8\$:	CASEL	EDT\$\$G_NOS, #0, #1		0794
	0018	000F	000E2 9\$:	.WORD	10\$-9\$,-		
					11\$-9\$		
00000000G	00	00	FB 000E6	CALLS	#0, EDT\$\$INTER_ERR		0800
		52	D4 000ED	CLRL	PROMPT_ADDR		0799
		0C	11 000EF	BRB	12\$		
		52	00 9E 000F1 10\$:	MOVAB	EDT\$\$T_PMT_INSN, PROMPT_ADDR		0794
		03	11 000F8	BRB	12\$		
		52	67 9E 000FA 11\$:	MOVAB	EDT\$\$T_PMT_INSN, PROMPT_ADDR		
		7E	62 9A 000FD 12\$:	MOVZBL	(PROMPT_ADDR), -(SP)		0805
			A2 9F 00100	PUSHAB	1(PROMPT_ADDR)		
00000000G	00	02	FB 00103	CALLS	#2, EDT\$\$GET_LN		
		50	D0 0010A	MOVL	R0, EOF		
		08	53 E9 0010D	BLBC	EOF, 13\$		0807
			00 D5 00110	TSTL	EDT\$\$G_CMD_LEN		
		19	13 00116	BEQL	14\$		
00000000G	00	00	9E 00118 13\$:	MOVAB	EDT\$\$T_LN_BUF, EDT\$\$A_SEL_POS		0810
		00	DD 00123	PUSHL	EDT\$\$G_CMD_LEN		0811
		59	DD 00129	PUSHL	R9		
		02	FB 0012B	CALLS	#2, EDT\$\$INS_LN		
		50	D0 0012E	MOVL	R0, STATUS		
		09	53 E8 00131 14\$:	BLBS	EOF, 16\$		0815
		03	54 E9 00134	BLBC	STATUS, 15\$		
		FF62	31 00137	BRW	3\$		
		0C	53 E9 0013A 15\$:	BLBC	EOF, 17\$		0819
		66	69 9E 0013D 16\$:	MOVAB	EDT\$\$T_CMD_BUF, EDT\$\$A_CMD_END		0822
		68	69 9E 00140	MOVAB	EDT\$\$T_CMD_BUF, EDT\$\$A_CMD_BUF		0823
		50	66 D0 00143	MOVL	EDT\$\$A_CMD_END, R0		0824
		60	21 90 00146	MOVB	#33, (R0)		
00000000G	00	00	FB 00149 17\$:	CALLS	#0, EDT\$\$END_INS		0831
		00	D5 00150	TSTL	EDT\$\$G_INP_SRC		0833
		07	12 00156	BNEQ	18\$		
00000000G	00	00	FB 00158	CALLS	#0, EDT\$\$TY_CURLN		
		04	0015F 18\$:	RET			0835

; Routine Size: 352 bytes, Routine Base: _EDT\$CODE + 0000

; 266 0836 1

EDT\$LINSE
V04-000

EDT\$LINSE - insert text
EDT\$\$INS_CMD - INSERT line-mode command

J 3
16-Sep-1984 00:52:19
14-Sep-1984 12:23:36

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]LINSE.BLI;1 Page 9
(4)

```

: 268 0837 1 %SBTTL 'EDT$$INS_CMD - INSERT line-mode command'
: 269 0838 1
: 270 0839 1 GLOBAL ROUTINE EDT$$INS_CMD ! INSERT line-mode command
: 271 0840 1 : NOVALUE =
: 272 0841 1
: 273 0842 1 !++
: 274 0843 1 FUNCTIONAL DESCRIPTION:
: 275 0844 1
: 276 0845 1 Command processing routine for the INSERT command. Position to the
: 277 0846 1 specified range and, if it positioned successfully, do the insert.
: 278 0847 1
: 279 0848 1 FORMAL PARAMETERS:
: 280 0849 1
: 281 0850 1 NONE
: 282 0851 1
: 283 0852 1 IMPLICIT INPUTS:
: 284 0853 1
: 285 0854 1 EDT$$A_EXE_CURCMD
: 286 0855 1
: 287 0856 1 IMPLICIT OUTPUTS:
: 288 0857 1
: 289 0858 1 NONE
: 290 0859 1
: 291 0860 1 ROUTINE VALUE:
: 292 0861 1
: 293 0862 1 NONE
: 294 0863 1
: 295 0864 1 SIDE EFFECTS:
: 296 0865 1
: 297 0866 1 NONE
: 298 0867 1
: 299 0868 1 --
: 300 0869 1
: 301 0870 2 BEGIN
: 302 0871 2
: 303 0872 2 EXTERNAL ROUTINE
: 304 0873 2 EDT$$RNG_REPOS;
: 305 0874 2
: 306 0875 2 EXTERNAL
: 307 0876 2 EDT$$G_RNG_FRSTLN,
: 308 0877 2 EDT$$A_CUR_BUF : REF TBCB_BLOCK,
: 309 0878 2 EDT$$Z_RNG_ORIGPOS : POS_BLOCK,
: 310 0879 2 EDT$$A_EXE_CURCMD : REF NODE_BLOCK; ! Pointer to the current command.
: 311 0880 2
: 312 0881 2 EDT$$G_RNG_FRSTLN = 1;
: 313 0882 2 EDT$$CPY_MEM (POS_SIZE, .EDT$$A_CUR_BUF, EDT$$Z_RNG_ORIGPOS);
: 314 0883 2
: 315 0884 2 IF EDT$$RNG_REPOS (.EDT$$A_EXE_CURCMD [RANGE1])
: 316 0885 2 THEN
: 317 0886 3 BEGIN
: 318 0887 3 EDT$$INS_TXT ();
: 319 0888 2 END;
: 320 0889 2
: 321 0890 1 END; ! of routine EDT$$INS_CMD
```

EDT\$LINERT
V04-000

EDT\$LINERT - insert text
EDT\$\$INS_CMD - INSERT line-mode command

K 3
16-Sep-1984 00:52:19
14-Sep-1984 12:23:36

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]LINERT.BLI;1
Page 10
(4)

```

00000000G 00 003C 00000
50 00000000G 01 D0 00002
60 00000000G 00 D0 00009
50 00000000G 0E 28 00010
04 00000000G 00 D0 00018
A0 DD 0001F
01 FB 00022
05 E9 00029
FE6F CF 00 FB 0002C
04 00031 1$:
```

```
.EXTRN EDT$$RNG_REPOS, EDT$$G_RNG_FRSTLN
.EXTRN EDT$$A_CUR_BUF, EDT$$Z_RNG_ORIGPOS
.EXTRN EDT$$A_EXE_CURCMD
```

```
.ENTRY EDT$$INS_CMD, Save R2,R3,R4,R5      : 0839
MOVL #1, EDT$$G_RNG_FRSTLN                  : 0881
MOVL EDT$$A_CUR_BUF, R0                      : 0882
MOVCL #14, (R0), EDT$$Z_RNG_ORIGPOS          :
MOVL EDT$$A_EXE_CURCMD, R0                  : 0884
PUSHL 4(R0)                                  :
CALLS #1, EDT$$RNG_REPOS                     :
BLBC R0, 1$                                  :
CALLS #0, EDT$$INS_TXT                       : 0887
RET                                           : 0890
```

; Routine Size: 50 bytes, Routine Base: _EDT\$CODE + 0160

```
: 322      0891 1
: 323      0892 1 !<BLF/PAGE>
```


EDT\$LINSE
V04-000

EDT\$LINSE - insert text
EDT\$\$INS_CMD - INSERT line-mode command

L 3
16-Sep-1984 00:52:19
14-Sep-1984 12:23:36

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]LINSE.BLI;1 Page 11 (5)

: 325 0893 1 END
: 326 0894 1
: 327 0895 0 ELUDOM

! of module EDT\$LINSE

PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	402	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	91	24	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\$:LINSE/OBJ=OBJ\$:LINSE MSRC\$:LINSE.BLI/UPDATE=(ENH\$:LINSE

: Size: 402 code + 0 data bytes
: Run Time: 00:19.7
: Elapsed Time: 00:24.8
: Lines/CPU Min: 2723
: Lexemes/CPU-Min: 9824
: Memory Used: 133 pages
: Compilation Complete

0136

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY