

EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTTTTTTTTTTTTTTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEEEEEEEEEEEEE	DDD	TTT
EEEEEEEEEEEEEE	DDD	TTT
EEEEEEEEEEEEEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEE	DDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTT
EEEEEEEEEEEEEEEE	DDDDDDDDDDDDDD	TTT

```
EEEEEEEEEE XX XX TTTTTTTTTT EEEEEEEEEE NN NN DDDDDDDD
EEEEEEEEEE XX XX TTTTTTTTTT EEEEEEEEEE NN NN DDDDDDDD
EE XX XX TTT DD DD
EE XX XX TTT DD DD
EE XX XX TTT DD DD
EEEEEEEEE XX XX TTT DD DD
EEEEEEEEE XX XX TTT DD DD
EE XX XX TTT DD DD
EE XX XX TTT DD DD
EE XX XX TTT DD DD
EEEEEEEEEE XX XX TTT DD DD
EEEEEEEEEE XX XX TTT DD DD
.....
.....
.....
.....
```

```
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
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS
```



```
0001 0 %TITLE 'EDT$EXTEND - EXTEND change-mode command'
0002 0 MODULE EDT$EXTEND ( ! EXTEND change-mode command
0003 0 IDENT = 'V04-000' ! File: EXTEND.BLI Edit: REM1024
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 Do the EXTEND change-mode command, which allows a line-mode
0037 1 command to be executed from change mode.
0038 1
0039 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0040 1
0041 1 AUTHOR: Bob Kushlis, CREATION DATE: 6-AUG-1979
0042 1
0043 1 MODIFIED BY:
0044 1
0045 1 1-001 - Original. DJS 18-FEB-1981. This module was created by
0046 1 extracting routine EDT$SEXT_CMD from module EDTCTR.
0047 1 1-002 - Regularize headers. JBS 04-Mar-1981
0048 1 1-003 - Do prompting from a global and remove LLINE and LCHANGE. JBS 23-Oct-1981
0049 1 1-004 - Replace L_LINE and L_CHANGE 7-Dec-81
0050 1 1-005 - Add EDT$LOAD_EXTEND, so that EDT$SLNM_CMD can reload this routine into
0051 1 memory prior to returning to it. Also, use a different symbol
0052 1 for EDT$SLNM_CMD, so the overlay analyzer can be told to ignore it.
0053 1 JBS 12-Mar-1982
0054 1 1-006 - Do the same for EDT$SPA_CMD. JBS 16-Mar-1982
0055 1 1-007 - Remove L_LINE and L_CHANGE. JBS 03-Jun-1982
0056 1 1-008 - Allow any key to terminate the "Continue" message. SMB 10-Jun-1982
0057 1 1-009 - Fix bug in /RECOVER associated with above change. SMB 16-Jun-1982
```



```

: 58      0058 1 1-010 - Position the cursor better when waiting for a command to execute.
: 59      0059 1      SMB 28-Jun-1982
: 60      0060 1 1-011- We can't use TI WRSTR unless this is a VT100/52.  SMB 02-Jul-1982
: 61      0061 1 1-012 - Use new formatting flags to control output at the bottom
: 62      0062 1      of the screen.  JBS 05-Jul-1982
: 63      0063 1 1-013 - Fix up response to PRTC when reading the journal file.  JBS 06-Jul-1982
: 64      0064 1 1-014 - Check for control C.  STS 15-Jul-1982
: 65      0065 1 1-015 - Write out the journal buffer before asking for input.  JBS 03-Aug-1982
: 66      0066 1 1-016 - Force a cursor positioning sequence and reverse video.  SMB 30-Aug-1982
: 67      0067 1 1-017 - Output the format buffer after turning off reverse video.  SMB 01-Sept-1982
: 68      0068 1 1-018 - Don't destroy EDT$SG_PRV_COL, it is now being maintained accurately.  JBS 05-Oct-1982
: 69      0069 1 1-019 - Force a write to the journal file after return is pressed.  STS 07-Oct-1982
: 70      0070 1 1-020 - Don't clear EDT$SG_SCR_CHGD.  JBS 09-Oct-1982
: 71      0071 1 1-021 - Check for hardcopy terminal.  STS 02-Dec-1982
: 72      0072 1 1-022 - Also check for terminal type unknown.  STS 03-Dec-1982
: 73      0073 1 1-023 - Don't permit a long response to destroy the stack.  JBS 13-Dec-1982
: 74      0074 1 1-024 - Added logic to maintain EDT$SG_TIN_OBUFPOS during /RECOVERY mode.
: 75      0075 1      REM 10-Oct-1983
: 76      0076 1  --
: 77      0077 1
```

EDT\$EXTEND
V04-000

EDT\$EXTEND - EXTEND change-mode command
Declarations

E 8
16-Sep-1984 00:18:24
14-Sep-1984 12:23:03

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

```

: 79      0078 1 %SBTTL 'Declarations'
: 80      0079 1
: 81      0080 1  TABLE OF CONTENTS:
: 82      0081 1
: 83      0082 1
: 84      0083 1  REQUIRE 'EDTSRC:TRAROUNAM';
: 85      0522 1
: 86      0523 1  FORWARD ROUTINE
: 87      0524 1      EDT$EXT_CMD : NOVALUE,
: 88      0525 1      EDT$LOAD_EXTEND : NOVALUE;
: 89      0526 1
: 90      0527 1
: 91      0528 1  INCLUDE FILES:
: 92      0529 1
: 93      0530 1
: 94      0531 1  REQUIRE 'EDTSRC:EDTREQ';
: 95      0666 1
: 96      0667 1
: 97      0668 1  MACROS:
: 98      0669 1
: 99      0670 1      NONE
: 100     0671 1
: 101     0672 1  EQUATED SYMBOLS:
: 102     0673 1
: 103     0674 1      NONE
: 104     0675 1
: 105     0676 1  OWN STORAGE:
: 106     0677 1
: 107     0678 1      NONE
: 108     0679 1
: 109     0680 1  EXTERNAL REFERENCES:
: 110     0681 1
: 111     0682 1      In the routine
```

EDT
V04

EDT\$EXTEND
V04-000

EDT\$EXTEND - EXTEND change-mode command
EDT\$\$EXT_CMD - EXTEND change-mode command

F 8
16-Sep-1984 00:18:24
14-Sep-1984 12:23:03

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

```

113 0683 1 %SBTTL 'EDT$$EXT_CMD - EXTEND change-mode command'
114 0684 1
115 0685 1 GLOBAL ROUTINE EDT$$EXT_CMD ! EXTEND change-mode command
116 0686 1 : NOVALUE =
117 0687 1
118 0688 1 ++
119 0689 1 FUNCTIONAL DESCRIPTION:
120 0690 1
121 0691 1 Do the EXTEND change-mode command. This allows a line-mode
122 0692 1 command to be executed from change mode.
123 0693 1
124 0694 1 FORMAL PARAMETERS:
125 0695 1
126 0696 1 NONE
127 0697 1
128 0698 1 IMPLICIT INPUTS:
129 0699 1
130 0700 1 EDT$$G_MESSAGE_LINE
131 0701 1 EDT$$G_EXI
132 0702 1 EDT$$T_CMD_BUF
133 0703 1 EDT$$A_CMD_BUF
134 0704 1 EDT$$A_CMD_END
135 0705 1 EDT$$G_EXITD
136 0706 1 EDT$$G_RCOV_MOD
137 0707 1 EDT$$Z_PA_STK
138 0708 1
139 0709 1 IMPLICIT OUTPUTS:
140 0710 1
141 0711 1 EDT$$G_SCR_CHGD
142 0712 1 EDT$$G_TXT_ONSCR
143 0713 1 EDT$$G_FMT_LNPOS
144 0714 1 EDT$$G_LASTMSG
145 0715 1 EDT$$G_TIN_OBUFPOS
146 0716 1
147 0717 1 ROUTINE VALUE:
148 0718 1
149 0719 1 NONE
150 0720 1
151 0721 1 SIDE EFFECTS:
152 0722 1
153 0723 1 MANY
154 0724 1
155 0725 1 --
156 0726 1
157 0727 2 BEGIN
158 0728 2
159 0729 2 EXTERNAL ROUTINE
160 0730 2 EDT$$SC_REVID,
161 0731 2 EDT$$SC_NONREVID,
162 0732 2 EDT$$SC_POSCSIF,
163 0733 2 EDT$$FMT_CRLF,
164 0734 2 EDT$$STOP_WKINGMSG,
165 0735 2 EDT$$RD_JOUTXT,
166 0736 2 EDT$$TI_WRSTR,
167 0737 2 EDT$$KPAD_INP,
168 0738 2 EDT$$TI_INPCH,
169 0739 2 EDT$$CHR_CC,
```

```
170      0740      2      EDT$MSG_TOSTR,  
171      0741      2      EDT$OUT_FMTBUF,  
172      0742      2      EDT$SPA_CMD_NOOVERLAY,  
173      0743      2      EDT$SLNM_CMD_NOOVERLAY,  
174      0744      2      EDT$TI_FLUSHJOUFI : NOVALUE,  
175      0745      2      EDT$FMT_MSG : NOVALUE,  
176      0746      2      EDT$ALO_HEAP,  
177      0747      2      EDT$DEA_HEAP : NOVALUE;  
178      0748  
179      0749      2      EXTERNAL  
180      0750      2      EDT$G_PUT_JOU,  
181      0751      2      EDT$G_MESSAGE_LINE,  
182      0752      2      EDT$G_EXI,  
183      0753      2      EDT$A_FMT_WRRUT,  
184      0754      2      EDT$T_CMD_BUF,  
185      0755      2      EDT$A_CMD_BUF,  
186      0756      2      EDT$A_CMD_END,  
187      0757      2      EDT$G_EXITD,  
188      0758      2      EDT$G_EDIT_MOD,  
189      0759      2      EDT$G_RCOV_MOD,  
190      0760      2      EDT$G_SCR_CHGD,  
191      0761      2      EDT$G_TXT_ONSCR,  
192      0762      2      EDT$G_CC_DONE,  
193      0763      2      EDT$G_TI_TYP,  
194      0764      2      EDT$Z_PA_STK,  
195      0765      2      EDT$G_FMT_BOT,  
196      0766      2      EDT$G_FMT_LCNT,  
197      0767      2      EDT$G_FMT_LNPOS,  
198      0768      2      EDT$G_TIN_OBUFPOS,  
199      0769      2      EDT$G_LASTMSG;  
200      0770  
201      0771      2      MESSAGES ((PRERETCON, INSMEM));  
202      0772  
203      0773      2      LOCAL  
204      0774      2      CONT_RESP : REF VECTOR [256, BYTE],  
205      0775      2      C,  
206      0776      2      TERM,  
207      0777      2      LEN;  
208      0778  
209      0779      2      CH$WCHAR ('!', EDT$A_CMD_END);  
210      0780      2      EDT$G_TXT_ONSCR = 0;  
211      0781      2      EDT$G_EDIT_MOD = LINE_MODE;  
212      0782      2      !+  
213      0783      2      !- Position the cursor to the message line.  
214      0784  
215      0785  
216      0786      2      IF ((.EDT$G_TI_TYP NEQ TERM_HCPY) AND (.EDT$G_TI_TYP NEQ TERM_UNKNOWN))  
217      0787      2      THEN  
218      0788      2      BEGIN  
219      0789      2      EDT$SC_POSCSIF (.EDT$G_MESSAGE_LINE + 1, 0);  
220      0790      2      EDT$G_FMT_LNPOS = 0;  
221      0791      2      EDT$G_FMT_BOT = 1;  
222      0792      2      EDT$G_FMT_LCNT = 0;  
223      0793      2      EDT$SC_REVID ();  
224      0794      2      END;  
225      0795  
226      0796      2      DO
```


EDT\$EXTEND
V04-000

EDT\$EXTEND - EXTEND change-mode command
EDT\$EXT_CMD - EXTEND change-mode command

H 8
16-Sep-1984 00:18:24
14-Sep-1984 12:23:03

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]EXTEND.BLI;1 Page 6
(3)

```

227 0797 3 BEGIN
228 0798 3
229 0799 3 IF EDT$PA_CMD_NOOVERLAY () THEN EDT$LN_CMD_NOOVERLAY (EDT$Z_PA_STK) ELSE EDT$G_TXT_ONSCR = 1;
230 0800 3
231 0801 3 END
232 0802 2 UNTIL ((CH$RCHAR (.EDT$A_CMD_BUF) EQL %C'!') OR (.EDT$G_EDIT_MOD EQL CHANGE_MODE) OR (EDT$CHK_CC ()))
233 0803 2
234 0804 2 +
235 0805 2 - See if the reason we are exiting is because of a control C seen.
236 0806 2
237 0807 2
238 0808 2 IF ( NOT ((CH$RCHAR (.EDT$A_CMD_BUF) EQL %C'!') OR (.EDT$G_EDIT_MOD EQL CHANGE_MODE)))
239 0809 2 THEN
240 0810 2 EDT$G_CC_DONE = 1;
241 0811 2
242 0812 2 +
243 0813 2 - Text on the screen means we have more than 2 lines of messages and that
244 0814 2 we need to issue the Press return to continue message.
245 0815 2
246 0816 2
247 0817 2 IF (.EDT$G_TXT_ONSCR NEQ 0) AND (.EDT$A_FMT_WRRUT EQL EDT$TI_WRSTR)
248 0818 2 THEN
249 0819 2 BEGIN
250 0820 2 +
251 0821 2 - Allocate heap storage for the response, even though we use so much space only in /RECOVER mode,
252 0822 2 so that if we run out of storage during /RECOVER we will also run out in the original session.
253 0823 2
254 0824 2
255 0825 4 IF ( NOT EDT$ALO_HEAP (%REF (256), CONT_RESP))
256 0826 3 THEN
257 0827 4 BEGIN
258 0828 4 EDT$FMT_MSG (EDT$_INSMEM);
259 0829 4 END
260 0830 3 ELSE
261 0831 4 BEGIN
262 0832 4
263 0833 5 IF (.EDT$G_RCOV_MOD EQL 0)
264 0834 4 THEN
265 0835 5 BEGIN
266 0836 5 EDT$STOP_WKINGMSG ();
267 0837 5 +
268 0838 5 - Output the "Press return to continue"
269 0839 5 message and wait for ANY key to be pressed before refreshing the screen
270 0840 5 and returning to change mode.
271 0841 5
272 0842 5 EDT$FMT_CRLF ();
273 0843 5 +
274 0844 5 - Don't call FMT_MSG because we don't want the bell to ring.
275 0845 5
276 0846 5 EDT$SC_REVID ();
277 0847 5 EDT$MSG_TOSTR (EDT$_PRERETCON);
278 0848 5 EDT$OUT_FMTBUF ();
279 0849 5 +
280 0850 5 - Make sure the journal buffer has been written to the journal file,
281 0851 5 since we are about to wait for terminal input.
282 0852 5
283 0853 5 EDT$TI_FLUSHJOUFI (%C'T');
```


EDT\$EXTEND
V04-000

EDT\$EXTEND - EXTEND change-mode command
EDT\$\$EXT_CMD - EXTEND change-mode command

I 8
16-Sep-1984 00:18:24
14-Sep-1984 12:23:03

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]EXTEND.BLI;1

Page 7
(3)

```

: 284      0854      5  !+
: 285      0855      5  ! Get the next character of input, including any characters generated by
: 286      0856      5  ! striking a keypad key.
: 287      0857      5  !-
: 288      0858      5      EDT$$TI_INPCH (C);
: 289      0859      5      EDT$$KPAD_INP (.C, .C, TERM);
: 290      0860      5      END
: 291      0861      4      ELSE
: 292      0862      5      BEGIN
: 293      0863      5      edt$$g_tin_obufpos = 0;
: 294      0864      5      edt$$rd_joutxt (cont_resp [0], len)
: 295      0865      4      END;
: 296      0866      4      EDT$$DEA_HEAP (%REF (256), CONT_RESP);
: 297      0867      4      END;
: 298      0868      3
: 299      0869      3
: 300      0870      3      EDT$$G_LASTMSG = 1;
: 301      0871      3      EDT$$G_SCR_CHGD = 1;
: 302      0872      3      END;
: 303      0873      2
: 304      0874      2      IF (.EDT$$G_EXITD NEQ 0) THEN EDT$$G_EXI = 1;
: 305      0875      2
: 306      0876      2      EDT$$G_PUT_JOU = 1;
: 307      0877      2      EDT$$SC_NONREVID ();
: 308      0878      2      EDT$$OUT_FMTBUF ();
: 309      0879      2      EDT$$G_FMT_BOT = 0;
: 310      0880      2      EDT$$G_EDIT_MOD = CHANGE_MODE;
: 311      0881      1      END;

```

! of routine EDT\$\$EXT_CMD

.TITLE EDT\$EXTEND EDT\$EXTEND - EXTEND change-mode comm
and

.IDENT \V04-000\

```

.EXTRN EDT$$SC_REVID, EDT$$SC_NONREVID
.EXTRN EDT$$SC_POSCSIF
.EXTRN EDT$$FMT_CRLF, EDT$$STOP_WKINGMSG
.EXTRN EDT$$RD_JOUTXT, EDT$$TI_ORSTR
.EXTRN EDT$$KPAD_INP, EDT$$TI_INPCH
.EXTRN EDT$$CHK_CC, EDT$$MSG_TOSTR
.EXTRN EDT$$OUT_FMTBUF
.EXTRN EDT$$PA_CMD_NOOVERLAY
.EXTRN EDT$$LNM_CMD_NOOVERLAY
.EXTRN EDT$$TI_FLUSHJOUFI
.EXTRN EDT$$FMT_MSG, EDT$$ALO_HEAP
.EXTRN EDT$$DEA_HEAP, EDT$$G_PUT_JOU
.EXTRN EDT$$G_MESSAGE_LINE
.EXTRN EDT$$G_EXI, EDT$$A_FMT_WRRUT
.EXTRN EDT$$T_CMD_BUF, EDT$$A_CMD_BUF
.EXTRN EDT$$A_CMD_END, EDT$$G_EXITD
.EXTRN EDT$$G_EDIT_MOD
.EXTRN EDT$$G_RCOV_MOD
.EXTRN EDT$$G_SCR_CHGD
.EXTRN EDT$$G_TXT_ONSCR
.EXTRN EDT$$G_CC_DONE, EDT$$G_TI_TYP
.EXTRN EDT$$Z_PA_STK, EDT$$G_FMT_BOT
.EXTRN EDT$$G_FMT_LCNT

```

```
.EXTRN EDT$$G_FMT_LNPOS
.EXTRN EDT$$G_TIN_OBUFPOS
.EXTRN EDT$$G_LASTMSG, EDT$_PRERETCON
.EXTRN EDT$_INSMEM

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

.ENTRY EDT$SEXT_CMD, Save R2, R3, R4, R5, R6, R7      : 0685
MOVAB EDT$OUT_FMTBUF, R7
MOVAB EDT$SA_CMD_BUF, R6
MOVAB EDT$SC_REVID, R5
MOVAB EDT$G_FMT_BOT, R4
MOVAB EDT$G_TXT_ONSCR, R3
MOVAB EDT$G_EDIT_MOD, R2
SUBL2 #20, SP
MOVL EDT$SA_CMD_END, R0      : 0779
MOVB #33, (R0)
CLRL EDT$G_TXT_ONSCR      : 0780
MOVL #1, EDT$G_EDIT_MOD   : 0781
MOVL EDT$G_TI_TYP, R0     : 0786
CML R0, #3
BEQL 1$
TSTL R0
BEQL 1$
CLRL -(SP)      : 0789
ADDL3 #1, EDT$G_MESSAGE_LINE, -(SP)
CALLS #2, EDT$SC_POSCSIF
CLRL EDT$G_FMT_LNPOS      : 0790
MOVL #1, EDT$G_FMT_BOT    : 0791
CLRL EDT$G_FMT_LCNT      : 0792
CALLS #0, EDT$SC_REVID     : 0793
CALLS #0, EDT$PA_CMD_NOOVERLAY
BLBC R0, 2$      : 0799
PUSHAB EDT$Z_PA_STK
CALLS #1, EDT$CNM_CMD_NOOVERLAY
BRB 3$
MOVL #1, EDT$G_TXT_ONSCR
MOVL EDT$SA_CMD_BUF, R0      : 0802
CMPB (R0), #33
BEQL 4$
TSTL EDT$G_EDIT_MOD
BEQL 4$
CALLS #0, EDT$CHK_CC
BLBC R0, 1$
MOVL EDT$SA_CMD_BUF, R0      : 0808
CMPB (R0), #33
BEQL 5$
TSTL EDT$G_EDIT_MOD
BEQL 5$
MOVL #1, EDT$G_CC_DONE
TSTL EDT$G_TXT_ONSCR      : 0810
BNEQ 7$      : 0817
BRW 12$
MOVAB EDT$TI_WSTR, R0
CML EDT$SA_FMT_WRRUT, R0
BNEQ 6$
PUSHAB CONT_RESP      : 0825
```


EDT\$EXTEND
V04-000

EDT\$EXTEND - EXTEND change-mode command
EDT\$\$EXT_CMD - EXTEND change-mode command

K 8
16-Sep-1984 00:18:24
14-Sep-1984 12:23:03

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]EXTEND.BLI;1

Page 9
(3)

04	AE	0100	8F	3C	000D0	MOVZWL	#256, 4(SP)	:	
		04	AE	9F	000D6	PUSHAB	4(SP)	:	
00000000G	00		02	FB	000D9	CALLS	#2, EDT\$\$ALO_HEAP	:	
	0F		50	E8	000E0	BLBS	R0, 8\$	:	
00000000G	00	00000000G	8F	DD	000E3	PUSHL	#EDT\$ INSMEM	:	0828
			01	FB	000E9	CALLS	#1, EDT\$\$FMT_MSG	:	
			76	11	000F0	BRB	11\$	:	0825
		00000000G	00	D5	000F2	TSTL	EDT\$\$G_RCOV_MOD	:	0833
			48	12	000F8	BNEQ	9\$	:	
00000000G	00		00	FB	000FA	CALLS	#0, EDT\$\$STOP_WKINGMSG	:	0836
00000000G	00		00	FB	00101	CALLS	#0, EDT\$\$FMT_CRLF	:	0842
	65		00	FB	00108	CALLS	#0, EDT\$\$SC_REVID	:	0846
		00000000G	8F	DD	0010B	PUSHL	#EDT\$ PRERETCON	:	0847
00000000G	00		01	FB	00111	CALLS	#1, EDT\$\$MSG_TOSTR	:	
	67		00	FB	00118	CALLS	#0, EDT\$\$OUT_FMTBUF	:	0848
	7E	54	8F	9A	0011B	MOVZBL	#84, -(SP)	:	0853
00000000G	00		01	FB	0011F	CALLS	#1, EDT\$\$TI_FLUSHJOUFI	:	
		04	AE	9F	00126	PUSHAB	C	:	0858
00000000G	00		01	FB	00129	CALLS	#1, EDT\$\$TI_INPCH	:	
		08	AE	9F	00130	PUSHAB	TERM	:	0859
		08	AE	DD	00133	PUSHL	C	:	
		0C	AE	DD	00136	PUSHL	C	:	
00000000G	00		03	FB	00139	CALLS	#3, EDT\$\$KPAD_INP	:	
			13	11	00140	BRB	10\$	:	0833
		00000000G	00	D4	00142	CLRL	EDT\$\$G_TIN_OBUFPOS	:	0863
		0C	AE	9F	00148	PUSHAB	LEN	:	0864
		14	AE	DD	0014B	PUSHL	CONT_RESP	:	
00000000G	00		02	FB	0014E	CALLS	#2, EDT\$\$RD_JOUTXT	:	
		10	AE	9F	00155	PUSHAB	CONT_RESP	:	0867
04	AE	0100	8F	3C	00158	MOVZWL	#256, 4(SP)	:	
		04	AE	9F	0015E	PUSHAB	4(SP)	:	
00000000G	00		02	FB	00161	CALLS	#2, EDT\$\$DEA_HEAP	:	
00000000G	00		01	D0	00168	MOVL	#1, EDT\$\$G_LASTMSG	:	0870
00000000G	00		01	D0	0016F	MOVL	#1, EDT\$\$G_SCR_CHGD	:	0871
		00000000G	00	D5	00176	TSTL	EDT\$\$G_EXITD	:	0874
			07	13	0017C	BEQL	13\$	:	
00000000G	00		01	D0	0017E	MOVL	#1, EDT\$\$G_EXI	:	
00000000G	00		01	D0	00185	MOVL	#1, EDT\$\$G_PUT_JOU	:	0876
00000000G	00		00	FB	0018C	CALLS	#0, EDT\$\$SC_NONREVID	:	0877
	67		00	FB	00193	CALLS	#0, EDT\$\$OUT_FMTBUF	:	0878
			64	D4	00196	CLRL	EDT\$\$G_FMT_BOT	:	0879
			62	D4	00198	CLRL	EDT\$\$G_EDIT_MOD	:	0880
			04	0019A	RET			:	0881

; Routine Size: 411 bytes, Routine Base: _EDT\$CODE + 0000

; 312 0882 1

EDT\$EXTEND
V04-000

EDT\$EXTEND - EXTEND change-mode command
EDT\$\$LOAD_EXTEND - load this module into memory

L 8
16-Sep-1984 00:18:24
14-Sep-1984 12:23:03

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]EXTEND.BLI;1

Page 10
(4)

```

: 314 0883 1 %SBTTL 'EDT$$LOAD_EXTEND - load this module into memory'
: 315 0884 1
: 316 0885 1 GLOBAL ROUTINE EDT$$LOAD_EXTEND ! Load this module into memory
: 317 0886 1 : NOVALUE =
: 318 0887 1
: 319 0888 1 ++
: 320 0889 1 FUNCTIONAL DESCRIPTION:
: 321 0890 1
: 322 0891 1 This routine has no function. It exists as an entry point so that
: 323 0892 1 EDT$$XXX_NOOVERLAY can call this module back into memory before
: 324 0893 1 returning to it.
: 325 0894 1
: 326 0895 1 FORMAL PARAMETERS:
: 327 0896 1
: 328 0897 1 NONE
: 329 0898 1
: 330 0899 1 IMPLICIT INPUTS:
: 331 0900 1
: 332 0901 1 NONE
: 333 0902 1
: 334 0903 1 IMPLICIT OUTPUTS:
: 335 0904 1
: 336 0905 1 NONE
: 337 0906 1
: 338 0907 1 ROUTINE VALUE:
: 339 0908 1
: 340 0909 1 NONE
: 341 0910 1
: 342 0911 1 SIDE EFFECTS:
: 343 0912 1
: 344 0913 1 NONE
: 345 0914 1
: 346 0915 1 --
: 347 0916 1
: 348 0917 2 BEGIN
: 349 0918 2 0
: 350 0919 1 END; ! of routine EDT$$LOAD_EXTEND
```

0000 00000
04 00002

.ENTRY EDT\$\$LOAD_EXTEND, Save nothing
RET

: 0885
: 0919

; Routine Size: 3 bytes, Routine Base: _EDT\$CODE + 019B

```

: 351 0920 1
: 352 0921 1 !<BLF/PAGE>
```


: 354
: 355
: 356

0922 1 END
0923 1
0924 0 ELUDOM

! of module EDT\$EXTEND

PSECT SUMMARY		
Name	Bytes	Attributes
_EDT\$CODE	414	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics					
File	-----		Symbols		Processing
	Total		Loaded	Percent	
\$255\$DUA28:[EDT.SRC]EDT.L32;1	377		5	1	00:00.2
_\$255\$DUA28:[EDT.SRC]PSECTS.L32;1	2		1	50	00:00.1

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS\$:EXTEND/OBJ=OBJ\$:EXTEND MSRC\$:EXTEND.BLI/UPDATE=(ENH\$:EXTEND)

Size: 414 code + 0 data bytes

Run Time: 00:18.7

Elapsed Time: 00:21.8

Lines/CPU Min: 2959

Lexemes/CPU-Min: 8245

Memory Used: 115 pages

Compilation Complete

0133 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

EXTEND
LIS

FDEC
LIS

FILL
LIS

FINDPARA
LIS

FCRLF
LIS

EDT
LIS

EXEC
LIS

EXECNOO
LIS

FILEIO
LIS

EDTVECTOR
LIS

FINDKEY
LIS

FCOLINC
LIS

FINAL
LIS

FINDHDLR
LIS

DEFKEY
LIS

ERRMSG
LIS

FCHAR
LIS