


```

EEEEEEEEEE XX XX TTTTTTTTTT EEEEEEEEEE NN NN DDDDDDDD
EEEEEEEEEE XX XX TTTTTTTTTT EEEEEEEEEE NN NN DDDDDDDD
EE XX XX TT DD DD
EE XX XX TT DD DD
EE XX XX TT DD DD
EE XX XX TT DD DD
EEEEEEEEE XX XX TT DD DD
EEEEEEEEE XX XX TT DD DD
EE XX XX TT DD DD
EE XX XX TT DD DD
EE XX XX TT DD DD
EEEEEEEEEE XX XX TT DD DD
EEEEEEEEEE XX XX TT 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

```



```

1 0001 0 %TITLE 'EDT$EXTEND - EXTEND change-mode command'
2 0002 0 MODULE EDT$EXTEND ( ! EXTEND change-mode command
3 0003 0 IDENT = 'V04-000' . File: EXTEND.BLI Edit: REM1024
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 Do the EXTEND change-mode command, which allows a line-mode
37 0037 1 command to be executed from change mode.
38 0038 1
39 0039 1 ENVIRONMENT: Runs at any access mode - AST reentrant
40 0040 1
41 0041 1 AUTHOR: Bob Kushlis, CREATION DATE: 6-AUG-1979
42 0042 1
43 0043 1 MODIFIED BY:
44 0044 1
45 0045 1 1-001 - Original. DJS 18-FEB-1981. This module was created by
46 0046 1 extracting routine EDT$$EXT_CMD from module EDTCTR.
47 0047 1 1-002 - Regularize headers. JBS 04-Mar-1981
48 0048 1 1-003 - Do prompting from a global and remove LLINE and LCHANGE. JBS 23-Oct-1981
49 0049 1 1-004 - Replace L_LINE and L_CHANGE 7-Dec-81
50 0050 1 1-005 - Add EDT$$COAD_EXTEND, so that EDT$SLNM_CMD can reload this routine into
51 0051 1 memory prior to returning to it. Also, use a different symbol
52 0052 1 for EDT$SLNM_CMD, so the overlay analyzer can be told to ignore it.
53 0053 1 JBS 12-Mar-1982
54 0054 1 1-006 - Do the same for EDT$$PA_CMD. JBS 16-Mar-1982
55 0055 1 1-007 - Remove L_LINE and L_CHANGE. JBS 03-Jun-1982
56 0056 1 1-008 - Allow any key to terminate the "Continue" message. SMB 10-Jun-1982
57 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 destory EDT$$G_PRV_COL, it is now being maintained accurately. JBS 05-Oct-1982
69 0069 1 : 1-019 - Force a write to the journal file after return is pressed. STS 07-Oct-1982
70 0070 1 : 1-020 - Don't clear EDT$$G_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$$G_TIN_OBUFPOS durring /RECOVERY mode.
75 0075 1 : REM 10-Oct-1983
76 0076 1 : --
77 0077 1 : --
```

```
.. 79      0078 1 %SBTTL 'Declarations'  
.. 80      0079 1  
.. 81      0080 1 : TABLE OF CONTENTS:  
.. 82      0081 1 :  
.. 83      0082 1  
.. 84      0083 1 REQUIRE 'EDT$SRC:TRAROUNAM';  
.. 85      0522 1  
.. 86      0523 1 FORWARD ROUTINE  
.. 87      0524 1     EDT$EXT_CMD : NOVALUE,  
.. 88      0525 1     EDT$LOAD_EXTEND : NOVALUE;  
.. 89      0526 1  
.. 90      0527 1 :  
.. 91      0528 1 : INCLUDE FILES:  
.. 92      0529 1 :  
.. 93      0530 1  
.. 94      0531 1 REQUIRE 'EDT$SRC: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
```

```
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$$PA_CMD_NOOVERLAY,
173 0743 2 EDT$$LNM_CMD_NOOVERLAY,
174 0744 2 EDT$$TI_FLUSRJOUFI : NOVALUE,
175 0745 2 EDT$$FMT_MSG : NOVALUE,
176 0746 2 EDT$$ALO_HEAP,
177 0747 2 EDT$$DEA_HEAP : NOVALUE;
178 0748
179 0749
180 0750 2 EXTERNAL
181 0751 2 EDT$$G_PUT_JOU,
182 0752 2 EDT$$G_MESSAGE_LINE,
183 0753 2 EDT$$G_EXI,
184 0754 2 EDT$$A_FMT_WRRUT,
185 0755 2 EDT$$T_CMD_BUF,
186 0756 2 EDT$$A_CMD_BUF,
187 0757 2 EDT$$A_CMD_END,
188 0758 2 EDT$$G_EXITD,
189 0759 2 EDT$$G_EDIT_MOD,
190 0760 2 EDT$$G_RCOV_MOD,
191 0761 2 EDT$$G_SCR_CHGD,
192 0762 2 EDT$$G_TXT_ONSCR,
193 0763 2 EDT$$G_CC_DONE,
194 0764 2 EDT$$G_TI_TYP,
195 0765 2 EDT$$Z_PA_STK,
196 0766 2 EDT$$G_FMT_BOT,
197 0767 2 EDT$$G_FMT_LCNT,
198 0768 2 EDT$$G_FMT_LNPOS,
199 0769 2 EDT$$G_TIN_OBUFPOS,
200 0770 2 EDT$$G_LASTMSG;
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$$G_EDIT_MOD = LINE_MODE;
212 0782 2
213 0783 2 !+ Position the cursor to the message line.
214 0784 2 !-
215 0785
216 0786 2 IF ((.EDT$$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

```

```

! Write out the journal file's buffer
! Output a message
! Allocate heap storage
! Deallocate heap storage

! force a write to journal file
! Message lines are this + 1
! Change mode has been exited
! Output format routine
! Command line buffer
! Pointer into command line buffer
! End of current command
! Did we exit with the last command?
! Current editing mode
! Are we in recovery mode?
! Screen must be completely rebuilt
! Text was written to screen
! Control c acted upon
! kind of terminal
! Parser semantic stack
! 1 = messages at bottom of screen
! Number of lines printed at the bottom of the screen
! Position on the output line
! Position in journal output buffer
! The last message printed, 1 = none

```

```

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

```



```

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 |
289 0859 5 |     EDT$$TI_INPCH (C);
290 0860 5 |     EDT$$KPAD_INP (.C, .C, TERM);
291 0861 4 |     END
292 0862 5 | ELSE
293 0863 5 |     BEGIN
294 0864 5 |     edt$$g_tin_obufpos = 0;
295 0865 4 |     edt$$rd_joutxt (cont_resp [0], len)
296 0866 4 |     END;
297 0867 4 |
298 0868 3 |     EDT$$DEA_HEAP (%REF (256), CONT_RESP);
299 0869 3 |     END;
300 0870 3 |
301 0871 3 |     EDT$$G_LASTMSG = 1;
302 0872 2 |     EDT$$G_SCR_CHGD = 1;
303 0873 2 |     END;
304 0874 2 | IF (.EDT$$G_EXITD NEQ 0) THEN EDT$$G_EXI = 1;
305 0875 2 |
306 0876 2 | EDT$$G_PUT_JOU = 1;           ! make sure the response gets written out alone
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$$PA_CMD_NOOVERLAY
.EXTRN EDT$$LNA_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$$CMD_BUF, EDT$$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$$EXT_CMD, Save R2,R3,R4,R5,R6,R7      : 0685
57 00000000G 00 00FC 00000      MOVAB EDT$$OUT_FMTBUF, R7
56 00000000G 00 9E 00002      MOVAB EDT$$A_CMD_BUF, R6
55 00000000G 00 9E 00009      MOVAB EDT$$SC_REVID, R5
54 00000000G 00 9E 00017      MOVAB EDT$$G_FMT_BOT, R4
53 00000000G 00 9E 0001E      MOVAB EDT$$G_TXT_ONSCR, R3
52 00000000G 00 9E 00025      MOVAB EDT$$G_EDIT_MOD, R2
5E          14 C2 0002C      SUBL2 #20, SP
50 00000000G 00 D0 0002F      MOVL EDT$$A_CMD_END, R0
60          21 90 00036      MOVB #33, (R0)
          63 D4 00039      CLRL EDT$$G_TXT_ONSCR
62          01 D0 0003B      MOVL #1, EDT$$G_EDIT_MOD
50 00000000G 00 D0 0003E      MOVL EDT$$G_TI_TYP, R0
03          50 D1 00045      CML  R0, #3
          27 13 00048      BEQL 1$
          50 D5 0004A      TSTL R0
          23 13 0004C      BEQL 1$
          7E D4 0004E      CLRL -(SP)
7E 00000000G 00 01 C1 00050      ADDL3 #1, EDT$$G_MESSAGE_LINE, -(SP)
      00000000G 00 02 FB 00058      CAL'S #2, EDT$$SC_POSCSIF
          00000000G 00 D4 0005F      CLRL EDT$$G_FMT_LNPOS
64          01 D0 00065      MOVL #1, EDT$$G_FMT_BOT
          00000000G 00 D4 00068      CLRL EDT$$G_FMT_LCNT
65          00 FB 0006E      CALLS #0, EDT$$SC_REVID
      00000000G 00 00 FB 00071 1$: CALLS #0, EDT$$PA_CMD_NOOVERLAY
      OF          50 E9 00078      BLBC R0, 2$
          00000000G 00 00 9F 0007B      PUSHAB EDT$$Z_PA_STK
      00000000G 00 01 FB 00081      CALLS #1, EDT$$ENM_CMD_NOOVERLAY
          03 11 00088      BRB 3$
63          01 D0 0008A 2$: MOVL #1, EDT$$G_TXT_ONSCR
50          66 D0 0008D 3$: MOVL EDT$$A_CMD_BUF, R0
21          60 91 00090      CMPB (R0), #33
          0E 13 00093      BEQL 4$
          62 D5 00095      TSTL EDT$$G_EDIT_MOD
          0A 13 00097      BEQL 4$
      00000000G 00 00 FB 00099      CALLS #0, EDT$$CHK_CC
      CE          50 E9 000A0      BLBC R0, 1$
50          66 D0 000A3 4$: MOVL EDT$$A_CMD_BUF, R0
21          60 91 000A6      CMPB (R0), #33
          0B 13 000A9      BEQL 5$
          62 D5 000AB      TSTL EDT$$G_EDIT_MOD
          07 13 000AD      BEQL 5$
      00000000G 00 01 D0 000AF      MOVL #1, EDT$$G_CC_DONE
          63 D5 000B6 5$: TSTL EDT$$G_TXT_ONSCR
          03 12 000B8      BNEQ 7$
          00B9 31 000BA 6$: BRW 12$
50 00000000G 00 9E 000BD 7$: MOVAB EDT$$TI_WSTR, R0
50 00000000G 00 D1 000C4      CML  EDT$$A_FMT_WRRUT, R0
          ED 12 000CB      BNEQ 6$
          10 AE 9F 000CD      PUSHAB CONT_RESP

```

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	8\$: 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	00	00000000G	8F	DD	0010B	PUSHL	#EDT\$ PRERETCON		0847
	67		01	FB	00111	CALLS	#1, EDT\$\$MSG_TOSTR		
	7E	54	00	FB	00118	CALLS	#0, EDT\$\$OUT_FMTBUF		0848
00000000G	00		8F	9A	0011B	MOVZBL	#84, -(SP)		0853
		04	01	FB	0011F	CALLS	#1, EDT\$\$TI_FLUSHJOUFI		
00000000G	00		AE	9F	00126	PUSHAB	C		0858
		08	01	FB	00129	CALLS	#1, EDT\$\$TI_INPCH		
		08	AE	9F	00130	PUSHAB	TERM		0859
		0C	AE	DD	00133	PUSHL	C		
00000000G	00		AE	DD	00136	PUSHL	C		
			03	FB	00139	CALLS	#3, EDT\$\$KPAD_INP		
			13	11	00140	BRB	10\$		0833
		00000000G	00	D4	00142	9\$: 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	10\$: 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	11\$: MOVL	#1, EDT\$\$G_LASTMSG		0870
00000000G	00		01	D0	0016F	MOVL	#1, EDT\$\$G_SCR_CHGD		0871
		00000000G	00	D5	00176	12\$: 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	13\$: 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

S
R
E
L
L
M
C

```

: 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 0000
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>

```

EDT\$EXTEND
V04-000

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

M 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 11
(5)

: 354 0922 1 END
: 355 0923 1
: 356 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	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DU'28:[EDT.SRC]EDT.L32;1	377	5	1	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\$:EXTEND/OBJ=OBJS\$:EXTEND MSRCS\$:EXTEND.BLI/UPDATE=(ENHS\$: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

EXTEND
LIS

FDEC
LIS

FILL
LIS

FINDPARA
LIS

FCRLF
LIS

EDT
LIS

EXEC
LIS

EXECNOO
LIS

FILEIO
LIS

EDTVECTOR
LIS

FINDKEY
LIS

FCO,INC
LIS

FINAL
LIS

FINDHDR
LIS

DEFKEY
LIS

ERRMSG
LIS

FCHAR
LIS