


```

LL          SSSSSSSS HH      HH      000000  WW      WW
LL          SSSSSSSS HH      HH      000000  WW      WW
LL          SS          HH      HH      00      00  WW      WW
LL          SS          HH      HH      00      00  WW      WW
LL          SS          HH      HH      00      00  WW      WW
LL          SS          HH      HH      00      00  WW      WW
LL          SSSSSS    HHHHHHHHHH 00      00  WW      WW
LL          SSSSSS    HHHHHHHHHH 00      00  WW      WW
LL          SS          HH      HH      00      00  WW      WW
LL          SS          HH      HH      00      00  WW      WW
LL          SS          HH      HH      00      00  WWW     WWW
LL          SS          HH      HH      00      00  WWW     WWW
LLLLLLLLLL SSSSSSSS HH      HH      000000  WW      WW
LLLLLLLLLL SSSSSSSS HH      HH      000000  WW      WW

```

```

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
LLLLLLLLLL IIIIII  SSSSSSSS
LLLLLLLLLL IIIIII  SSSSSSSS

```

.....

```

1 0001 0 %TITLE 'EDT$LSHOW - SHOW line-mode command'
2 0002 0 MODULE EDT$LSHOW ( ! SHOW line-mode command
3 0003 0 IDENT = 'V04-000' ! File: LSHOW.BLI Edit: JBS1041
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 SHOW 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 30-JAN-1981. This module was created by
45 0045 1 extracting the routine EDT$$SHW_CMD from the module EXEC.BLI.
46 0046 1 1-002 - Regularize headers. JBS 20-Mar-1981
47 0047 1 1-003 - Change E_VERSION to EDT$$T_VER. JBS 31-Mar-1981
48 0048 1 1-004 - Use the ASSERT macro. JBS 01-Jun-1981
49 0049 1 1-005 - Use new message codes. JBS 06-Aug-1981
50 0050 1 1-006 - Add show command for REPEAT. STS 26-Aug-1981
51 0051 1 1-007 - Add show command for FNF. TMV 10-Sept-1981
52 0052 1 1-008 - Add show command for WPS search type and SUMMARY.
53 0053 1 Also a stub for PROMPT. STS 01-Oct-1981
54 0054 1 1-009 - Add show command for the text of page and end. STS 06-Oct-1981
55 0055 1 1-010 - Implement SHOW PROMPT. JBS 21-Oct-1981
56 0056 1 1-011 - Add show word and show para. STS 22-Oct-1981
57 0057 1 1-012 - Add four more prompts and don't document their lengths. JBS 23-Oct-1981

```

```
58 0058 1 1-013 - remove external reference to EDT$$A_INI_TXT. It was never used here
59 0059 1          anyway. STS 06-Nov-1981
60 0060 1 1-014 - Add SHOW HELP command. SMB 18-Dec-1981
61 0061 1 1-015 - Change SHO BUF to call 48-bit count display module. SMB 5-Feb-1982
62 0062 1 1-016 - Add SHOW AUTOREPEAT. JBS 09-Feb-1982
63 0063 1 1-017 - Use VERSION.REQ to define the version number of EDT. JBS 09-Mar-1982
64 0064 1 1-018 - Add SHOW COMMAND. JBS 04-May-1982
65 0065 1 1-019 - Change format of SHOW HELP. SMB 25-May-1982
66 0066 1 1-020 - Change SHOW HELP for the 1i's. SMB 2-Jun-1982
67 0067 1 1-021 - Track changes to SET COMMAND in SHOW COMMAND. JBS 07-Jun-1982
68 0068 1 1-022 - Make all the show messages look alike. STS 08-Jun-1982
69 0069 1 1-023 - Correct counts in SHOW WORD and SHOW PARA messages. JBS 09-Jun-1982
70 0070 1 1-024 - Modify for new message processing. SMB 29-Jun-1982
71 0071 1 1-025 - Add CRLF to SHOW BUFFER if using TI WRSTR. SMB 02-Jun-1982
72 0072 1 1-026 - Use EDT$$FMT CRLF instead of EDT$$OOT_FMTBUF. JBS 05-Jul-1982
73 0073 1 1-027 - Remove prompt PRTC. 07-Jul-1982
74 0074 1 1-028 - Add two new SHOW SEARCH possibilities. JBS 20-Jul-1982
75 0075 1 1-029 - Add the SHOW FILES command. SMB 04-Aug-1982
76 0076 1 1-030 - Add a check for READONLY output files to edit 1-029. SMB 09-Aug-1982
77 0077 1 1-031 - New design for defined keys. JBS 13-Aug-1982
78 0078 1 1-032 - Make use of the bit masks for the options word. 17-Aug-1982
79 0079 1 1-033 - Fix up references to the EDT$M symbols. JBS 23-Aug-1982
80 0080 1 1-034 - Add information to SHOW TERM. JBS 02-Sep-1982
81 0081 1 1-035 - Improve SHOW TERM printout and add copyright statement to SHOW VERSION. JBS 17-Sep-1982
82 0082 1 1-036 - More improvements to SHOW TERMINAL. JBS 29-Oct-1982
83 0083 1 1-037 - Change placement of colons in SHOW FILES. SMB 20-Dec-1982
84 0084 1 1-038 - Add conditionals for WPS support and VT220 support. JBS 10-Feb-1983
85 0085 1 1-039 - Correct the sense of SHOW WORD. JBS 11-Feb-1983
86 0086 1 1-040 - Print key definitions in segments. JBS 07-Mar-1983
87 0087 1 1-041 - Also print user-specified text, the command file name and prompts in segments. JBS 14-Jul-1983
88 0088 1 --
89 0089 1
```

```
0090 1 %SBTTL 'Declarations'  
0091 1  
0092 1 : TABLE OF CONTENTS:  
0093 1 :  
0094 1  
0095 1  
0096 1 REQUIRE 'EDTSRC:TRAROUNAM';  
0534 1  
0535 1 FORWARD ROUTINE  
0536 1 EDT$SSH_CMD : NOVALUE;  
0537 1  
0538 1 :  
0539 1 : INCLUDE FILES:  
0540 1 :  
0541 1  
0542 1 REQUIRE 'EDTSRC:EDTREQ';  
0677 1  
0678 1 LIBRARY 'EDTSRC:SUPPORTS';  
0679 1  
0680 1 LIBRARY 'EDTSRC:KEYPADDEF';  
0681 1  
0682 1 REQUIRE 'EDTSRC:VERSION';  
0750 1  
0751 1 :  
0752 1 : MACROS:  
0753 1 :  
0754 1 : NONE  
0755 1 :  
0756 1 : EQUATED SYMBOLS:  
0757 1 :  
0758 1 : NONE  
0759 1 :  
0760 1 : OWN STORAGE:  
0761 1 :  
0762 1 : NONE  
0763 1 :  
0764 1 : EXTERNAL REFERENCES:  
0765 1 :  
0766 1 : In the routine
```

! Process the SHOW command

```
130 0767 1 %SBTTL 'EDT$$$SHW_CMD - SHOW line-mode command'
131 0768 1
132 0769 1 GLOBAL ROUTINE EDT$$$SHW_CMD ! SHOW line-mode command
133 0770 1 : NOVALUE =
134 0771 1
135 0772 1 !**
136 0773 1 FUNCTIONAL DESCRIPTION:
137 0774 1
138 0775 1 Command processing routine for SHOW. The current command node
139 0776 1 field SET_TYPE gives an index for the type of SHOW which is
140 0777 1 being done. Case on this and put the stuff out.
141 0778 1
142 0779 1 FORMAL PARAMETERS:
143 0780 1
144 0781 1 NONE
145 0782 1
146 0783 1 IMPLICIT INPUTS:
147 0784 1
148 0785 1 EDT$$G_TRN_TBLINIT
149 0786 1 EDT$$A_BUF_LST
150 0787 1 EDT$$G_CAS_FLG
151 0788 1 EDT$$G_EDIT_DFLTMOD
152 0789 1 EDT$$A_US_ENT
153 0790 1 EDT$$G_CMD_NAMLEN
154 0791 1 EDT$$A_CMD_NAM
155 0792 1 EDT$$G_EXCT_MATCH
156 0793 1 EDT$$G_FNF_MSGFLG
157 0794 1 EDT$$A_US_TXT
158 0795 1 EDT$$G_KPAD
159 0796 1 EDT$$G_NOS
160 0797 1 EDT$$G_QUIET
161 0798 1 EDT$$G_SEA_BEG
162 0799 1 EDT$$G_RPT
163 0800 1 EDT$$G_SEA_BNDD
164 0801 1 EDT$$G_SCR_LNS
165 0802 1 EDT$$G_SCLE_BOT
166 0803 1 EDT$$G_SCLL_TOP
167 0804 1 EDT$$G_TAB_SIZ
168 0805 1 EDT$$A_CUR_BUF
169 0806 1 EDT$$G_TAB_LVL
170 0807 1 EDT$$G_TRUN
171 0808 1 EDT$$G_TI_TYP
172 0809 1 EDT$$G_TI_WID
173 0810 1 EDT$$G_VFY
174 0811 1 EDT$$G_WD_WRAP
175 0812 1 EDT$$A_EXE_CURCMD
176 0813 1 EDT$$G_SUMRY
177 0814 1 EDT$$G_ENB_AUTRPT
178 0815 1 EDT$$G_PARTYP
179 0816 1 EDT$$G_WRD TYP
180 0817 1 EDT$$T_PMT_LINE
181 0818 1 EDT$$T_PMT_KPD
182 0819 1 EDT$$T_PMT_NOKPD
183 0820 1 EDT$$T_PMT_HCCHG
184 0821 1 EDT$$T_PMT_INS
185 0822 1 EDT$$T_PMT_INSN
186 0823 1 EDT$$T_PMT_QUERY
```

187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243

```

0824 1 | EDT$$G_FMT_LNPOS
0825 1 |
0826 1 | IMPLICIT OUTPUTS:
0827 1 |
0828 1 | EDT$$G_TXT_ONSCR
0829 1 | EDT$$G_FMT_LNPOS
0830 1 |
0831 1 | ROUTINE VALUE:
0832 1 |
0833 1 | NONE
0834 1 |
0835 1 | SIDE EFFECTS:
0836 1 |
0837 1 | NONE
0838 1 |
0839 1 | --
0840 1 |
0841 2 | BEGIN
0842 2 |
0843 2 | EXTERNAL ROUTINE
0844 2 | EDT$$FMT_CRLF,          ! Terminate an output line
0845 2 | EDT$$TI_ORSTR,
0846 2 | EDT$$SET_HLPFNAM,
0847 2 | EDT$$FMT_STRCNT,
0848 2 | EDT$$TST_BADK,
0849 2 | EDT$$FMT_CH,
0850 2 | EDT$$FMT_DCML,
0851 2 | EDT$$FMT_STR,
0852 2 | EDT$$MSG_TOSTR,
0853 2 | EDT$$FIND_KEY,         ! Find a key's definition
0854 2 | EDT$$DEF_DFLTK,       ! Define the default keypad
0855 2 | EDT$$FMT_CHWID,       ! Compute the width of a character
0856 2 |
0857 2 | %IF %BLISS (BLISS16)
0858 2 | %THEN
0859 2 |
0860 2 | EXTERNAL ROUTINE
0861 2 | EDT$$GET_FNAM;
0862 2 |
0863 2 | %FI
0864 2 |
0865 2 | EXTERNAL LITERAL
0866 2 | EDT$M_NOOUTPUT;
0867 2 |
0868 2 | EXTERNAL
0869 2 | EDT$$V_OPTIONS : BITVECTOR [32], ! bit 3 = /NOOUTPUT
0870 2 | EDT$$A_INP_NAM,         ! Input file name
0871 2 | EDT$$G_INP_NAMLEN,     ! Length of input file name
0872 2 | EDT$$A_OUT_NAM,        ! Output file name
0873 2 | EDT$$G_OUT_NAMLEN,     ! Output file name length
0874 2 | EDT$$A_FMT_WRRUT,      ! Format output routine
0875 2 | EDT$$G_TRN_TBLINIT,    ! True if translation table has been initialized
0876 2 | EDT$$A_BUF_LST,
0877 2 | EDT$$G_CAS_FLG,
0878 2 | EDT$$G_EDIT_DFLTMOD,
0879 2 | EDT$$A_US_ENT : VECTOR,
0880 2 | EDT$$G_EXCT_MATCH,

```

76

69

20

49

40

4E

4E

45

```

244 0881 2 EDT$$A_US_TXT : VECTOR, ! user defined text
245 0882 2 EDT$$G_FNF_MSGFLG,
246 0883 2 EDT$$G_KPAD,
247 0884 2 EDT$$G_NOS,
248 0885 2 EDT$$G_QUIÉT,
249 0886 2 EDT$$G_SEA_BÉG,
250 0887 2 EDT$$G_RPT,
251 0888 2 EDT$$G_SEA_BNDD,
252 0889 2 EDT$$G_SCR_LNS,
253 0890 2 EDT$$G_SCLC_BOI,
254 0891 2 EDT$$G_SCLL_TOP,
255 0892 2 EDT$$G_TAB_SIZ,
256 0893 2 EDT$$A_CUR_BUF : REF TBCB_BLOCK,
257 0894 2 EDT$$G_TAB_LVL,
258 0895 2 EDT$$G_TRUN,
259 0896 2 EDT$$G_TXT_ONSCR,
260 0897 2 EDT$$G_TI_TYP,
261 0898 2 EDT$$G_TI_WID,
262 0899 2 EDT$$G_VFY,
263 0900 2 EDT$$G_WD_WRAP,
264 0901 2 EDT$$T_HELP_NAM, ! help file name
265 0902 2 EDT$$G_HELP_NAMLEN, ! help file name length
266 0903 2 EDT$$G_HELP_SET, ! flag for help file name set
267 0904 2 EDT$$A_CMD_NAM, ! Name of command file
268 0905 2 EDT$$G_CMD_NAMLEN, ! Length of the name of the command file
269 0906 2 EDT$$G_TI_SCROLL, ! 1 = terminal has scrolling regions
270 0907 2
271 L 0908 2 %IF SUPPORT_VT220
272 0909 2 %THEN
273 0910 2 EDT$$G_EIGHT_BIT, ! 1 = this is an eight-bit terminal
274 0911 2 %FI
275 0912 2
276 0913 2 EDT$$G_TI_EDIT, ! 1 = this terminal has 'edit' features (ICM, DCH, IL, DL)
277 0914 2 EDT$$G_ENB_AUTRPT, ! 1 = manipulate autorepeat in VT100, 0 = don't
278 0915 2
279 L 0916 2 %IF SUPPORT_WPS
280 0917 2 %THEN
281 0918 2 EDT$$G_SUMRY, ! summary flag
282 0919 2 EDT$$G_WRD_TYP, ! type of word flag
283 0920 2 EDT$$G_PARTYP, ! type of para flag
284 0921 2 %FI
285 0922 2
286 0923 2 EDT$$A_EXE_CURCMD : REF NODE_BLOCK, ! Pointer to the current command.
287 0924 2 EDT$$G_FMT_LNPOS, ! Current line position
288 0925 2 EDT$$T_PMT_LINE : VECTOR [, BYTE], ! Counted ASCII string of line-mode prompt
289 0926 2 EDT$$T_PMT_KPD : VECTOR [, BYTE], ! Counted ASCII string of keypad prompt
290 0927 2 EDT$$T_PMT_NOKPD : VECTOR [, BYTE], ! Counted ASCII string of nokeypad prompt
291 0928 2 EDT$$T_PMT_HCCHG : VECTOR [, BYTE], ! Counted ASCII string of hard copy change mode prompt
292 0929 2 EDT$$T_PMT_INS : VECTOR [, BYTE], ! Counted ASCII string of line-mode insert prompt
293 0930 2 EDT$$T_PMT_INSN : VECTOR [, BYTE], ! Counted ASCII string of line-mode insert nonumbers prompt
294 0931 2 EDT$$T_PMT_QUERY : VECTOR [, BYTE]; ! Counted ASCII string of /QUERY prompt
295 0932 2
296 0933 2 MESSAGES ((KEYNOTDEF, NODEFN, INVPARFOR));
297 0934 2
298 0935 2 CASE .EDT$$A_EXE_CURCMD [SET_TYPE] FROM 1 TO 29 OF
299 0936 2 SET
300 0937 2

```



```
0938 [1] : ! Show buffers
0939 BEGIN
0940
0941 LOCAL
0942 BUF : REF TBCB_BLOCK;
0943
0944 !+
0945 !- Point to first buffer.
0946
0947 BUF = .EDT$$A_BUF_LST;
0948
0949 !+
0950 !- Loop through all buffers, displaying the number of lines
0951 !- in each.
0952
0953 WHILE (.BUF NEQA 0) DO
0954 BEGIN
0955
0956 !+
0957 !- Put an equal sign before the current buffer.
0958
0959 IF (.BUF EQL .EDT$$A_CUR_BUF) THEN EDT$$FMT_CH (%C'=') ELSE EDT$$FMT_CH (%C' ');
0960
0961 EDT$$FMT_STR (BUF [TBCB_NAME], .BUF [TBCB_NAME_LEN]);
0962 EDT$$FMT_CH (ASC_K_TAB);
0963 EDT$$FMT_STRCNT (BUF [TBCB_LINE_COUNT], UPLIT (%C' '), 0);
0964
0965 !+
0966 !- If the buffer still has a file attached to it, then print
0967 !- an asterisk after the count to indicate there may be more
0968 !- lines than we have indicated.
0969
0970 IF (.BUF [TBCB_INPUT_RAB] NEQ 0) THEN EDT$$FMT_CH (%C'*');
0971
0972 EDT$$FMT_STR (UPLIT (BYTE (ASC_K_TAB, 'lines')), 6);
0973 EDT$$FMT_CRLF ();
0974 BUF = .BUF [TBCB_NEXT_BUF];
0975 END;
0976
0977 END;
0978
0979 [2] : ! Show case
0980 BEGIN
0981
0982 CASE .EDT$$G_CAS_FLG FROM 0 TO 2 OF
0983 SET
0984
0985 [0] :
0986 EDT$$FMT_STR (UPLIT (BYTE ('None')), 4);
0987
0988 [1] :
0989 EDT$$FMT_STR (UPLIT (BYTE ('Upper')), 5);
0990
0991 [2] :
0992 EDT$$FMT_STR (UPLIT (BYTE ('Lower')), 5);
0993
0994 [OUTRANGE] :
```

```

0995          ASSERT (0);
0996          TES;
0997
0998          EDT$$FMT_CRLF ();
0999          END;
1000
1001 [3] :                               ! Show search
1002 BEGIN
1003
1004 CASE .EDT$$G_EXCT_MATCH FROM 0 TO 4 OF
1005 SET
1006
1007 [0] :
1008     EDT$$FMT_STR (UPLIT (BYTE ('general')), 7);
1009
1010 [1] :
1011     EDT$$FMT_STR (UPLIT (BYTE ('exact')), 5);
1012
1013 [2] :
1014     EDT$$FMT_STR (UPLIT (BYTE ('wps')), 3);
1015
1016 [3] :
1017     EDT$$FMT_STR (UPLIT (BYTE ('case insensitive')), 16);
1018
1019 [4] :
1020     EDT$$FMT_STR (UPLIT (BYTE ('diacritical insensitive')), 25);
1021
1022 [OUTRANGE] :
1023     ASSERT (0);
1024     TES;
1025
1026 IF .EDT$$G_SEA_BEG
1027 THEN
1028     EDT$$FMT_STR (UPLIT BYTE(' begin '), 7)
1029 ELSE
1030     EDT$$FMT_STR (UPLIT BYTE(' end '), 5);
1031
1032 IF (.EDT$$G_SEA_BNDD EQL 0) THEN EDT$$FMT_STR (UPLIT BYTE('un'), 2);
1033
1034 EDT$$FMT_STR (UPLIT BYTE('bounded'), 7);
1035 EDT$$FMT_CRLF ();
1036 END;
1037
1038 [4] :                               ! Show terminal
1039 BEGIN
1040
1041 CASE .EDT$$G_TI_TYP FROM 0 TO 3 OF
1042 SET
1043
1044 [0] :
1045     EDT$$FMT_STR (UPLIT (BYTE ('Unknown')), 7);
1046
1047 [TERM VT52] :
1048     EDT$$FMT_STR (UPLIT (BYTE ('VT52')), 4);
1049
1050 [TERM VT100] :
1051     EDT$$FMT_STR (UPLIT (BYTE ('VT100')), 5);

```

```
415 1052
416 1053 [TERM HCPY] :
417 1054 EDT$$FMT_STR (UPLIT (BYTE ('Hardcopy')), 8);
418 1055
419 1056 [OUTRANGE] :
420 1057 ASSERT (0);
421 1058 TES;
422 1059
423 1060 IF .EDT$$G_TI_SCROLL
424 1061 THEN
425 1062 EDT$$FMT_STR (UPLIT (BYTE (' scroll')), 8)
426 1063 ELSE
427 1064 EDT$$FMT_STR (UPLIT (BYTE (' noscroll')), 10);
428 1065
429 L 1066 %IF SUPPORT_VT220
430 1067 %THEN
431 1068
432 1069 IF .EDT$$G_EIGHT_BIT
433 1070 THEN
434 1071 EDT$$FMT_STR (UPLIT (BYTE (' eightbit')), 10)
435 1072 ELSE
436 1073 %FI
437 1074
438 1075 EDT$$FMT_STR (UPLIT (BYTE (' noeightbit')), 12);
439 1076
440 1077 IF .EDT$$G_TI_EDIT
441 1078 THEN
442 1079 EDT$$FMT_STR (UPLIT (BYTE (' edit')), 6)
443 1080 ELSE
444 1081 EDT$$FMT_STR (UPLIT (BYTE (' noedit')), 8);
445 1082
446 1083 EDT$$FMT_CRLF ();
447 1084 END;
448 1085
449 1086 [5] : ! Show version
450 1087 BEGIN
451 1088 !+
452 1089 ! Show the version number.
453 1090 !-
454 1091 EDT$$FMT_STR (VERSION_TEXT, VERSION_TEXT_LEN);
455 1092 !+
456 1093 ! Show the copyright statement. If we have an eight-bit terminal
457 1094 ! we use the international copyright symbol, otherwise we simulate
458 1095 ! it using (C). The copyright statement goes on the same line as
459 1096 ! the version number.
460 1097 !-
461 1098
462 L 1099 %IF SUPPORT_VT220
463 1100 %THEN
464 1101
465 1102 IF .EDT$$G_EIGHT_BIT
466 1103 THEN
467 1104 EDT$$FMT_STR (COPYRIGHT_TEXT_8, COPYRIGHT_TEXT_8_LEN)
468 1105 ELSE
469 1106 %FI
470 1107
471 1108 EDT$$FMT_STR (COPYRIGHT_TEXT_7, COPYRIGHT_TEXT_7_LEN);
```

```

472 1109
473 1110      EDT$$FMT_CRLF ();
474 1111      END;
475 1112
476 1113      [6] :                               ! Show ent
477 1114      BEGIN
478 1115
479 1116      LOCAL
480 1117          LEN,
481 1118          ENT;
482 1119
483 1120      ENT = .EDT$$A_US_ENT [.EDT$$A_EXE_CURCMD [SET_VAL] - 1];
484 1121      LEN = CH$RCHAR_A (ENT);
485 1122      EDT$$FMT_STR (.ENT, .LEN);
486 1123      EDT$$FMT_CRLF ();
487 1124      END;
488 1125
489 1126      [7] :                               ! Show key
490 1127      BEGIN
491 1128
492 1129      LOCAL
493 1130          KEY_PTR : REF BLOCK [, BYTE] FIELD (KEY_DEF_FIELD),
494 1131          STATUS;
495 1132
496 1133      IF ( NOT .EDT$$G_TRN_TBLINIT) THEN STATUS = EDT$$DEF_DFLTK () ELSE STATUS = 1;
497 1134
498 1135      IF .STATUS
499 1136      THEN
500 1137          BEGIN
501 1138
502 1139          IF EDT$$ST_BADK (.EDT$$A_EXE_CURCMD [KEY_VAL])
503 1140          THEN
504 1141              EDT$$MSG_TOSTR (EDT$_KEYNOTDEF)
505 1142          ELSE
506 1143
507 1144              IF EDT$$FIND_KEY (.EDT$$A_EXE_CURCMD [KEY_VAL], KEY_PTR)
508 1145              THEN
509 1146                  BEGIN
510 1147
511 1148                  !+ If the key defintion is long, show it in pieces.
512 1149                  -
513 1150
514 1151                  LOCAL
515 1152                      KEY_P,
516 1153                      KEY_L,
517 1154                      CH;
518 1155
519 1156                      KEY_P = KEY_PTR [KEY_DEF_TEXT];
520 1157                      KEY_L = .KEY_PTR [KEY_DEF_LEN];
521 1158                      EDT$$G_FMT_LNPOS = 0;
522 1159                      CH = CH$RCHAR_A (KEY_P);
523 1160                      KEY_L = .KEY_L - 1;
524 1161
525 1162                      WHILE (.KEY_L GEQ 0) DO
526 1163                          BEGIN
527 1164
528 1165                              IF ((.EDT$$G_FMT_LNPOS + EDT$$FMT_CHWID (.CH, .EDT$$G_FMT_LNPOS)) GEQ

```

```

529      1166      7
530      1167      6
531      1168      6
532      1169      6
533      1170      6
534      1171      6
535      1172      6
536      1173      5
537      1174      5
538      1175      5
539      1176      4
540      1177      4
541      1178      4
542      1179      4
543      1180      3
544      1181      3
545      1182      3
546      1183      3
547      1184      2
548      1185      2
549      1186      2
550      1187      3
551      1188      3
552      1189      3
553      1190      3
554      1191      3
555      1192      2
556      1193      2
557      1194      2
558      1195      3
559      1196      3
560      1197      3
561      1198      2
562      1199      2
563      1200      2
564      1201      3
565      1202      3
566      1203      3
567      1204      3
568      1205      3
569      1206      3
570      1207      2
571      1208      2
572      1209      2
573      1210      3
574      1211      3
575      1212      3
576      1213      3
577      1214      3
578      1215      3
579      1216      2
580      1217      2
581      1218      2
582      1219      3
583      1220      3
584      1221      3
585      1222      3

```

```

      .EDT$$$G_TI_WID)
      THEN
      EDT$$$FMT_CRLF ();

      EDT$$$FMT_CH (.CH);
      CH = CH$RCHAR A (KEY_P);
      KEY_L = .KEY_C - 1;
      END;

      END
      ELSE
      EDT$$$MSG_TOSTR (EDT$_NODEFN);

      END
      ELSE
      EDT$$$MSG_TOSTR (.STATUS);

      EDT$$$FMT_CRLF ();
      END;

[8] : ! Show cursor
      BEGIN
      EDT$$$FMT_DCML (.EDT$$$G_SCLL_TOP);
      EDT$$$FMT_CH (XC:');
      EDT$$$FMT_DCML (.EDT$$$G_SCLL_BOT);
      EDT$$$FMT_CRLF ();
      END;

[9] : ! Show screen.
      BEGIN
      EDT$$$FMT_DCML (.EDT$$$G_TI_WID);
      EDT$$$FMT_CRLF ();
      END;

[10] : ! Show numbers.
      BEGIN
      IF (.EDT$$$G_NOS EQL 0) THEN EDT$$$FMT_STR (UPLIT BYTE('no'), 2);

      EDT$$$FMT_STR (UPLIT BYTE('numbers'), 7);
      EDT$$$FMT_CRLF ();
      END;

[11] : ! Show verify
      BEGIN
      IF (.EDT$$$G_VFY EQL 0) THEN EDT$$$FMT_STR (UPLIT BYTE('no'), 2);

      EDT$$$FMT_STR (UPLIT BYTE('verify'), 6);
      EDT$$$FMT_CRLF ();
      END;

[12] : ! Show truncate
      BEGIN
      IF (.EDT$$$G_TRUN EQL 0) THEN EDT$$$FMT_STR (UPLIT BYTE('no'), 2);

```

```
586      1223      EDT$$FMT_STR (UPLIT BYTE('truncate'), 8);
587      1224      EDT$$FMT_CRLF ();
588      1225      END;
589      1226
590      1227      [13] :                ! Show keypad
591      1228      BEGIN
592      1229
593      1230      IF (.EDT$$G_KPAD EQL 0) THEN EDT$$FMT_STR (UPLIT BYTE('no'), 2);
594      1231
595      1232      EDT$$FMT_STR (UPLIT BYTE('keypad'), 6);
596      1233      EDT$$FMT_CRLF ();
597      1234      END;
598      1235
599      1236      [14] :                ! Show wrap
600      1237      BEGIN
601      1238
602      1239      IF (.EDT$$G_WD_WRAP EQL 256)
603      1240      THEN
604      1241      EDT$$FMT_STR (UPLIT BYTE('nowrap'), 6)
605      1242      ELSE
606      1243      EDT$$FMT_DCML (.EDT$$G_WD_WRAP);
607      1244
608      1245      EDT$$FMT_CRLF ();
609      1246      END;
610      1247
611      1248      [15] :                ! Show mode
612      1249      BEGIN
613      1250
614      1251      IF (.EDT$$G_EDIT_DFLTMOD EQL CHANGE_MODE)
615      1252      THEN
616      1253      EDT$$FMT_STR (UPLIT (BYTE ('Change')), 6)
617      1254      ELSE
618      1255      EDT$$FMT_STR (UPLIT (BYTE ('Line')), 4);
619      1256
620      1257      EDT$$FMT_CRLF ();
621      1258      END;
622      1259
623      1260      [16] :                ! Show lines
624      1261      BEGIN
625      1262      EDT$$FMT_DCML (.EDT$$G_SCR_LNS);
626      1263      EDT$$FMT_CRLF ();
627      1264      END;
628      1265
629      1266      [17] :                ! Show quiet
630      1267      BEGIN
631      1268
632      1269      IF (.EDT$$G_QUIET EQL 0) THEN EDT$$FMT_STR (UPLIT BYTE('no'), 2);
633      1270
634      1271      EDT$$FMT_STR (UPLIT BYTE('quiet'), 5);
635      1272      EDT$$FMT_CRLF ();
636      1273      END;
637      1274
638      1275      [18] :                ! Show tab
639      1276      BEGIN
640      1277
641      1278      IF (.EDT$$G_TAB_SIZ EQL 0)
642      1279      THEN
```

```

: 643      1280      4          BEGIN
: 644      1281      4          EDT$$FMT_STR (UPLIT BYTE('notab'), 5);
: 645      1282      4          EDT$$FMT_CRLF ();
: 646      1283      4          END
: 647      1284      3          ELSE
: 648      1285      4          BEGIN
: 649      1286      4          EDT$$FMT_STR (UPLIT BYTE('tab size '), 9);
: 650      1287      4          EDT$$FMT_DCML (.EDT$$G_TAB_SIZ);
: 651      1288      4          EDT$$FMT_STR (UPLIT BYTE('; tab level '), 12);
: 652      1289      4          EDT$$FMT_DCML (.EDT$$G_TAB_LVL);
: 653      1290      4          EDT$$FMT_CRLF ();
: 654      1291      4          END;
: 655      1292      3
: 656      1293      2          END;
: 657      1294      2
: 658      1295      2          [19] :                ! Show repeat
: 659      1296      2          BEGIN
: 660      1297      2
: 661      1298      2          IF (.EDT$$G_RPT EQL 0) THEN EDT$$FMT_STR (UPLIT BYTE('no'), 2);
: 662      1299      2
: 663      1300      2          EDT$$FMT_STR (UPLIT BYTE('repeat'), 6);
: 664      1301      2          EDT$$FMT_CRLF ();
: 665      1302      2          END;
: 666      1303      2
: 667      1304      2          [20] :                ! Show file_not_found
: 668      1305      2          BEGIN
: 669      1306      2
: 670      1307      2          IF (.EDT$$G_FNF_MSGFLG EQL 0) THEN EDT$$FMT_STR (UPLIT BYTE('no'), 2);
: 671      1308      2
: 672      1309      2          EDT$$FMT_STR (UPLIT BYTE('fnf'), 3);
: 673      1310      2          EDT$$FMT_CRLF ();
: 674      1311      2          END;
: 675      1312      2
: 676      1313      2          [21] :                ! show summary
: 677      1314      2          BEGIN
: 678      1315      2
: 679      1316      2          L %IF SUPPORT_WPS
: 680      1317      2          %THEN
: 681      1318      2
: 682      1319      2          IF ( NOT .EDT$$G_SUMRY)
: 683      1320      2          THEN
: 684      1321      2          EDT$$FMT_STR (UPLIT BYTE('nosummary'), 9)
: 685      1322      2          ELSE
: 686      1323      2          %FI
: 687      1324      2
: 688      1325      2          EDT$$FMT_STR (UPLIT BYTE('summary'), 7);
: 689      1326      2
: 690      1327      2          EDT$$FMT_CRLF ();
: 691      1328      2          END;
: 692      1329      2
: 693      1330      2          [22] :                ! Show Prompt
: 694      1331      2          BEGIN
: 695      1332      2
: 696      1333      2          LOCAL
: 697      1334      2          LEN,                ! Length of the string
: 698      1335      2          TXT,                ! Address of string contents
: 699      1336      2          PROMPT_NUM,            ! Which prompt string

```

```

700      1337      3          CH;
701      1338      3
702      1339      3          PROMPT_NUM = .EDT$$A_EXE_CURCMD [SET_VAL];
703      1340      4          TXT = TCASE .PROMPT_NUM FROM 1 TO 7 OF
704      1341      4          SET
705      1342      4          [1] : EDT$$T_PMT_LINE;
706      1343      4          [2] : EDT$$T_PMT_KPD;
707      1344      4          [3] : EDT$$T_PMT_NOKPD;
708      1345      4          [4] : EDT$$T_PMT_HCCHG;
709      1346      4          [5] : EDT$$T_PMT_INS;
710      1347      4          [6] : EDT$$T_PMT_INSN;
711      1348      4          [7] : EDT$$T_PMT_QUERY;
712      1349      4          [OUTRANGE] :
713      1350      5          BEGIN
714      1351      5          ASSERT (0);
715      1352      5          0
716      1353      4          END;
717      1354      4          TES);
718      1355      3          LEN = CH$RCHAR_A (TXT);
719      1356      3          !+
720      1357      3          !- If the prompt is long, show it in pieces.
721      1358      3
722      1359      3          EDT$$G_FMT_LNPOS = 0;
723      1360      3          CH = CH$RCHAR_A (TXT);
724      1361      3          LEN = .LEN - T;
725      1362      3
726      1363      3          WHILE (.LEN GEQ 0) DO
727      1364      4          BEGIN
728      1365      4
729      1366      5          IF ((.EDT$$G_FMT_LNPOS + EDT$$FMT_CHWID (.CH, .EDT$$G_FMT_LNPOS)) GEQ .EDT$$G_TI_WID)
730      1367      4          THEN
731      1368      4          EDT$$FMT_CRLF ();
732      1369      4
733      1370      4          EDT$$FMT_CH (.CH);
734      1371      4          CH = CH$RCHAR_A (TXT);
735      1372      4          LEN = .LEN - T;
736      1373      3          END;
737      1374      3
738      1375      3          EDT$$FMT_CRLF ();
739      1376      2          END;
740      1377      2
741      1378      2          [23] : ! show text
742      1379      3          BEGIN
743      1380      3
744      1381      3          LOCAL
745      1382      3          LEN,
746      1383      3          TXT,
747      1384      3          CH;
748      1385      3
749      1386      3          TXT = .EDT$$A_US_TXT [.EDT$$A_EXE_CURCMD [SET_VAL] - 1];
750      1387      3          LEN = CH$RCHAR_A (TXT);
751      1388      3
752      1389      3          !+
753      1390      3          !- If the text is long, show it in pieces.
754      1391      3
755      1392      3          EDT$$G_FMT_LNPOS = 0;
756      1393      3          CH = CH$RCHAR_A (TXT);
          LEN = .LEN - T;

```



```

: 757      1394      3
: 758      1395      3
: 759      1396      4
: 760      1397      4
: 761      1398      5
: 762      1399      4
: 763      1400      4
: 764      1401      4
: 765      1402      4
: 766      1403      4
: 767      1404      4
: 768      1405      4
: 769      1406      4
: 770      1407      4
: 771      1408      4
: 772      1409      4
: 773      1410      4
: 774      1411      4
: 775      1412      4
: 776      L 1413      4
: 777      1414      4
: 778      1415      4
: 779      1416      4
: 780      1417      4
: 781      1418      4
: 782      1419      4
: 783      1420      4
: 784      1421      4
: 785      1422      4
: 786      1423      4
: 787      1424      4
: 788      1425      4
: 789      1426      4
: 790      1427      4
: 791      1428      4
: 792      1429      4
: 793      L 1430      4
: 794      1431      4
: 795      1432      4
: 796      1433      4
: 797      1434      4
: 798      1435      4
: 799      1436      4
: 800      1437      4
: 801      1438      4
: 802      1439      4
: 803      1440      4
: 804      1441      4
: 805      1442      4
: 806      1443      4
: 807      1444      4
: 808      1445      4
: 809      1446      4
: 810      1447      4
: 811      1448      4
: 812      1449      4
: 813      1450      4

```

```

      WHILE (.LEN GEQ 0) DO
      BEGIN
      IF ((.EDT$$$G_FMT_LNPOS + EDT$$$FMT_CHWID (.CH, .EDT$$$G_FMT_LNPOS)) GEQ .EDT$$$G_TI_WID)
      THEN
      EDT$$$FMT_CRLF ();

      EDT$$$FMT_CH (.CH);
      CH = CH$RCHAR_A (TXT);
      LEN = .LEN - T;
      END;

      EDT$$$FMT_CRLF ();
      END;

      [24] :                               ! show word type
      BEGIN
      %IF SUPPORT_WPS
      %THEN
      IF .EDT$$$G_WRD TYP
      THEN
      EDT$$$FMT_STR (UPLIT BYTE('nodelimiter '), 12)
      ELSE
      %FI
      EDT$$$FMT_STR (UPLIT BYTE('delimiter '), 10);

      EDT$$$FMT_CRLF ();
      END;

      [25] :                               ! SHOW WORD
      BEGIN
      %IF SUPPORT_WPS
      %THEN
      IF .EDT$$$G_PARTYP
      THEN
      EDT$$$FMT_STR (UPLIT BYTE('nowps '), 6)
      ELSE
      %FI
      EDT$$$FMT_STR (UPLIT BYTE('wps '), 4);

      EDT$$$FMT_CRLF ();
      END;

      [26] :                               ! Show Help File Name
      BEGIN
      LOCAL
      STATUS;

```

```

: 814      1451 3 | If the help file name has not been parsed, call SET HELP so
: 815      1452 3 | that the resultant file name will appear in the message.
: 816      1453 3 | For the PDP-11's, EDT$$G_HELP_SET is always zero.
: 817      1454 3 |
: 818      1455 3 |     STATUS = 1;
: 819      1456 3 |
: 820      1457 4 |     IF ( NOT .EDT$$G_HELP_SET)
: 821      1458 3 |     THEN
: 822      1459 3 |         STATUS = EDT$$SET_HLPFNAM (CH$PTR (EDT$$T_HELP_NAM),
: 823      1460 3 |             .EDT$$G_HELP_NAMLEN);
: 824      1461 3 |
: 825      1462 4 |     IF (.STATUS)
: 826      1463 3 |     THEN
: 827      1464 4 |         BEGIN
: 828      1465 4 |             EDT$$FMT_STR (UPLIT BYTE('Help file name: '), 16);
: 829      1466 4 |
: 830      L 1467 4 | %IF %BLISS (BLISS32)
: 831      1468 4 | %THEN
: 832      1469 4 |             EDT$$FMT_STR (CH$PTR (EDT$$T_HELP_NAM), .EDT$$G_HELP_NAMLEN);
: 833      U 1470 4 | %ELSE
: 834      U 1471 4 |             EDT$$GET_FNAM ();
: 835      1472 4 | %FI
: 836      1473 4 |
: 837      1474 4 |             EDT$$FMT_CRLF ();
: 838      1475 3 |             END;
: 839      1476 3 |
: 840      1477 2 |         END;
: 841      1478 2 |
: 842      1479 2 | [27] :             ! show autorepeat
: 843      1480 3 |     BEGIN
: 844      1481 3 |
: 845      1482 4 |     IF (.EDT$$G_ENB_AUTRPT EQL 1)
: 846      1483 3 |     THEN
: 847      1484 3 |         EDT$$FMT_STR (UPLIT BYTE('autorepeat'), 10)
: 848      1485 3 |     ELSE
: 849      1486 3 |         EDT$$FMT_STR (UPLIT BYTE('noautorepeat'), 12);
: 850      1487 3 |
: 851      1488 3 |     EDT$$FMT_CRLF ();
: 852      1489 2 |     END;
: 853      1490 2 |
: 854      1491 2 | [28] :             ! show command
: 855      1492 3 |     BEGIN
: 856      1493 3 |
: 857      1494 3 |     LOCAL
: 858      1495 3 |         LEN,
: 859      1496 3 |         TXT,
: 860      1497 3 |         CH;
: 861      1498 3 |
: 862      1499 3 |     TXT = .EDT$$A_CMD_NAM;
: 863      1500 3 |     LEN = .EDT$$G_CMD_NAMLEN;
: 864      1501 3 |
: 865      1502 4 |     IF (.TXT NEQA 0)
: 866      1503 3 |     THEN
: 867      1504 4 |         BEGIN
: 868      1505 4 |
: 869      1506 4 |     !+ If the name is long, show it in pieces.
: 870      1507 4 |     !-

```

```

: 871      1508 4      EDT$$G_FMT_LNPOS = 0;
872      1509 4      CH = CHRCHAR A (TXT);
873      1510 4      LEN = .LEN - T;
874      1511 4
875      1512 4      WHILE (.LEN GEQ 0) DO
876      1513 5      BEGIN
877      1514 5
878      1515 6      IF ((.EDT$$G_FMT_LNPOS + EDT$$FMT_CHWID (.CH, .EDT$$G_FMT_LNPOS)) GEQ .EDT$$G_TI_WID)
879      1516 5      THEN
880      1517 5      EDT$$FMT_CRLF ();
881      1518 5
882      1519 5      EDT$$FMT_CH (.CH);
883      1520 5      CH = CHRCHAR A (TXT);
884      1521 5      LEN = .LEN - T;
885      1522 4      END;
886      1523 4
887      1524 3      END;
888      1525 3
889      1526 3      EDT$$FMT_CRLF ();
890      1527 2      END;
891      1528 2
892      1529 2      [29] : ! show input/output files
893      1530 3      BEGIN
894      1531 3
895      1532 3      LOCAL
896      1533 3      LEN,
897      1534 3      TXT;
898      1535 3
899      1536 3      TXT = .EDT$$A_INP_NAM;
900      1537 3      LEN = .EDT$$G_INP_NAMLEN;
901      1538 3      EDT$$FMT_STR (UPLIT BYTE('Input File: '), 13);
902      1539 3      EDT$$FMT_STR (.TXT, .LEN);
903      1540 3      EDT$$FMT_CRLF ();
904      1541 3      EDT$$FMT_STR (UPLIT BYTE('Output File: '), 13);
905      1542 3
906      1543 4      IF (.EDT$$G_OUT_NAMLEN EQL 0)
907      1544 3      THEN
908      1545 3
909      1546 4      IF ((.EDT$$V_OPTIONS AND EDT$M_NOOUTPUT) NEQ 0)
910      1547 3      THEN
911      1548 3      EDT$$FMT_STR (UPLIT BYTE('**Readonly**'), 12)
912      1549 3      ELSE
913      1550 3      EDT$$FMT_STR (.TXT, .LEN)
914      1551 3
915      1552 3      ELSE
916      1553 3      EDT$$FMT_STR (.EDT$$A_OUT_NAM, .EDT$$G_OUT_NAMLEN);
917      1554 3
918      1555 3      EDT$$FMT_CRLF ();
919      1556 2      END;
920      1557 2
921      1558 2      [OUTRANGE] :
922      1559 2      ASSERT (0);
923      1560 2      TES;
924      1561 2
925      1562 1      END; ! of routine EDT$$SHW_CMD

```

														.TITLE	EDT\$LSHOW EDT\$LSHOW - SHOW line-mode command																			
														.IDENT	\V04-000\																			
														.PSECT	_EDT\$CODE,NOWRT, SHR, PIC,2																			
														00000020	00000	P.AAA:	.LONG	32	:															
														09	00004	P.AAB:	.BYTE	9	:															
73	65	6E	69	6C										00005		P.AAC:	.ASCII	\lines\	:															
														0000A		P.AAC:	.ASCII	\None\	:															
72	65	6E	6F	4E	70	70	55								00010		P.AAD:	.ASCII	\Upper\	:														
														00015		P.AAD:	.ASCII	\Upper\	:															
72	65	77	6F	4C										00018		P.AAE:	.ASCII	\Lower\	:															
														0001D		P.AAE:	.ASCII	\Lower\	:															
6C	61	72	65	6E	65	67								00020		P.AAF:	.ASCII	\general\	:															
														00027		P.AAF:	.ASCII	\general\	:															
74	63	61	78	65										00028		P.AAG:	.ASCII	\exact\	:															
														0002D		P.AAG:	.ASCII	\exact\	:															
73	70	77												00030		P.AAH:	.ASCII	\wps\	:															
														00033		P.AAH:	.ASCII	\wps\	:															
76	69	74	69	73	6E	65	73	6E	69	20	65	73	61	63			00034		P.AAI:	.ASCII	\case insensitive\	:												
														65	00043		P.AAI:	.ASCII	\case insensitive\	:														
69	20	6C	61	63	69	63	69	74	69	72	63	61	69	64			00044		P.AAJ:	.ASCII	\diacritical insensitive\	:												
														65	76	69	74	69	73	6E	65	73	6E	00053		P.AAJ:	.ASCII	\diacritical insensitive\	:					
														20	6E	69	67	65	62	20								0005D		P.AAK:	.ASCII	\ begin \	:	
														20	64	6E	65	20								00064		P.AAL:	.ASCII	\ end \	:			
														64	65	64	6E	75	6F	62								00069		P.AAM:	.ASCII	\un\	:	
														00072		P.AAM:	.ASCII	\un\	:															
6E	77	6F	6E	6B	6F	55								00074		P.AAO:	.ASCII	\Unknown\	:															
														0007B		P.AAO:	.ASCII	\Unknown\	:															
														32	35	54	56										0007C		P.AAP:	.ASCII	\VT52\	:		
														30	30	31	54	56										00080		P.AAQ:	.ASCII	\VT100\	:	
														00085		P.AAQ:	.ASCII	\VT100\	:															
														79	70	6F	63	64	72	61	48								00088		P.AAR:	.ASCII	\Hardcopy\	:
														6C	6C	6F	72	63	73	20	2C								00090		P.AAS:	.ASCII	\, scroll\	:
6C	6C	6F	72	63	73	6F	6E	20	2C								00098		P.AAT:	.ASCII	\, noscroll\	:												
														000A2		P.AAT:	.ASCII	\, noscroll\	:															
74	69	62	74	68	67	69	65	20	2C								000A4		P.AAU:	.ASCII	\, eightbit\	:												
														000AE		P.AAU:	.ASCII	\, eightbit\	:															
74	69	62	74	68	67	69	65	6F	6E	20	2C								000B0		P.AAV:	.ASCII	\, noeightbit\	:										
														74	69	64	65	20	2C								000BC		P.AAW:	.ASCII	\, edit\	:		
														000C2		P.AAW:	.ASCII	\, edit\	:															
20	20	20	20	20	20	20	74	69	64	65	6F	6E	20	2C			000C4		P.AAX:	.ASCII	\, noedit\	:												
														30	30	30	2D	34	30	56								000CC		P.AAY:	.ASCII	\V04-000	:	
														20	20	20												000DB		P.AAY:	.ASCII	\V04-000	:	
														000DE		P.AAZ:	.ASCII	\	:															
49	52	59	50	4F	43	20	20	20	20	20	20	20	20	20	20	20	20	000E0		P.AAZ:	.ASCII	\ COPYRIGHT \	:											
														20	54	48	47								000EF		P.AAZ:	.ASCII	\ COPYRIGHT \	:				
														A9	000F3		P.AAZ:	.ASCII	\ COPYRIGHT \	:														
4D	50	49	55	51	45	20	4C	41	54	49	47	49	44	20			000F4		P.AAZ:	.ASCII	\ DIGITAL EQUIPMENT CORPORATION 1980 & 19\	:												
4E	4F	49	54	41	52	4F	50	52	4F	43	20	54	4E	45			00103		P.AAZ:	.ASCII	\ DIGITAL EQUIPMENT CORPORATION 1980 & 19\	:												
														39	31	20	26	20	30	38	39	31	20			00112		P.AAZ:	.ASCII	\ DIGITAL EQUIPMENT CORPORATION 1980 & 19\	:			
														34	38	0011C		P.AAZ:	.ASCII	\84\	:													
														0011E		P.AAZ:	.ASCII	\84\	:															
48	47	49	52	59	50	4F	43	20	20	20	20	20	20	20	20	20	20	00120		P.ABA:	.ASCII	\ COPYRIGHT (C) DIGITAL EQUIPMENT C\	:											
45	20	4C	41	54	49	47	49	44	20	29	43	28	20	54			0012F		P.ABA:	.ASCII	\ COPYRIGHT (C) DIGITAL EQUIPMENT C\	:												

```
30 38 39 31 20 43 20 54 4E 45 4D 50 49 55 51  
4E 4F 49 54 41 52 4F 50 52 4F 0013E  
34 38 39 31 20 26 20 00148  
73 72 65 62 6D 75 6E 00157  
79 66 69 72 65 76 6E 0015E P.ABB: .ASCII  
65 74 61 63 6E 75 72 74 00160 P.ABC: .ASCII  
64 61 70 79 65 68 00167 P.ABD: .ASCII  
70 61 72 77 6F 6E 00169 P.ABE: .ASCII  
65 67 6E 61 68 43 0016F P.ABF: .ASCII  
65 6E 69 4C 00171 P.ABG: .ASCII  
65 6E 69 75 71 00179 P.ABH: .ASCII  
74 65 69 75 71 0017B P.ABI: .ASCII  
74 61 65 70 79 65 64 00181 P.ABJ: .ASCII  
74 61 65 70 79 65 64 00187 .BLKB  
74 61 65 70 79 65 64 00188 P.ABK: .ASCII  
74 61 65 70 79 65 64 0018E .BLKB  
74 61 65 70 79 65 64 00190 P.ABL: .ASCII  
74 61 65 70 79 65 64 00194 P.ABM: .ASCII  
74 61 65 70 79 65 64 00196 P.ABN: .ASCII  
74 61 65 70 79 65 64 00198 P.ABO: .ASCII  
74 61 65 70 79 65 64 001A0 P.ABP: .ASCII  
74 61 65 70 79 65 64 001A9 P.ABQ: .ASCII  
74 61 65 70 79 65 64 001B5 P.ABR: .ASCII  
74 61 65 70 79 65 64 001B7 P.ABS: .ASCII  
74 61 65 70 79 65 64 001BD P.ABT: .ASCII  
74 61 65 70 79 65 64 001BF P.ABU: .ASCII  
74 61 65 70 79 65 64 001C2 P.ABV: .ASCII  
74 61 65 70 79 65 64 001CB P.ABW: .ASCII  
74 61 65 70 79 65 64 001D2 P.ABX: .ASCII  
74 61 65 70 79 65 64 001DE P.ABY: .ASCII  
74 61 65 70 79 65 64 001E8 P.ABZ: .ASCII  
74 61 65 70 79 65 64 001EE P.ACA: .ASCII  
74 61 65 70 79 65 64 001F2 P.ACB: .ASCII  
74 61 65 70 79 65 64 00201  
74 61 65 70 79 65 64 00202 P.ACC: .ASCII  
74 61 65 70 79 65 64 0020C P.ACD: .ASCII  
74 61 65 70 79 65 64 00218 P.ACE: .ASCII  
74 61 65 70 79 65 64 00225 P.ACF: .ASCII  
74 61 65 70 79 65 64 00232 P.ACG: .ASCII
```

```
.ASCII \CORPORATION 1980 & 1984\  
.ASCII \no\  
.ASCII \numbers\  
.ASCII \no\  
.ASCII \verify\  
.ASCII \no\  
.ASCII \truncate\  
.ASCII \no\  
.ASCII \keypad\  
.ASCII \nowrap\  
.BLKB 1  
.ASCII \change\  
.BLKB 2  
.ASCII \line\  
.ASCII \no\  
.ASCII \quiet\  
.ASCII \notab\  
.ASCII \tab size \  
.ASCII \; tab level \  
.ASCII \no\  
.ASCII \repeat\  
.ASCII \no\  
.ASCII \fnf\  
.ASCII \nosummary\  
.ASCII \summary\  
.ASCII \nodelimiter \  
.ASCII \delimiter \  
.ASCII \nows \  
.ASCII \wps \  
.ASCII \Help file name: \  
P.ACC: .ASCII \autorepeat\  
P.ACD: .ASCII \noautorepeat\  
P.ACE: .ASCII \Input File: \  
P.ACF: .ASCII \Output File: \  
P.ACG: .ASCII \**Readonly**\  
.EXTRN EDT$FMT_CRLF, EDT$STI_WRSTR  
.EXTRN EDT$SET_HLPFNAM  
.EXTRN EDT$FMT_STRCNT  
.EXTRN EDT$ST_BADK, EDT$FMT_CH  
.EXTRN EDT$FMT_DCML, EDT$FMT_STR  
.EXTRN EDT$MSG_TOSTR, EDT$FIND_KEY  
.EXTRN EDT$DEF_DFLTK, EDT$FMT_CHWID  
.EXTRN EDT$M_NOOUTPUT, EDT$V_OPTIONS  
.EXTRN EDT$A_INP_NAM, EDT$G_INP_NAMLEN  
.EXTRN EDT$A_OUT_NAM, EDT$G_OUT_NAMLEN  
.EXTRN EDT$A_FMT_WRRUT  
.EXTRN EDT$G_TRN_TBLINIT  
.EXTRN EDT$A_BUF_LST, EDT$G_CAS_FLG  
.EXTRN EDT$G_EDIT_DFLTMOD  
.EXTRN EDT$A_US_ENT, EDT$G_EXCT_MATCH  
.EXTRN EDT$A_US_TXT, EDT$G_FNF_MSGFLG  
.EXTRN EDT$G_KPXD, EDT$G_NOS  
.EXTRN EDT$G_QUIET, EDT$G_SEA_BEG
```

```

.EXTRN EDT$$G_RPT, EDT$$G_SEA_BNDD
.EXTRN EDT$$G_SCR_LNS, EDT$$G_SCLL_BOT
.EXTRN EDT$$G_SCLL_TOP
.EXTRN EDT$$G_TAB_SIZ, EDT$$A_CUR_BUF
.EXTRN EDT$$G_TAB_LVL, EDT$$G_TRUN
.EXTRN EDT$$G_TXT_ONSCR
.EXTRN EDT$$G_TI_TYP, EDT$$G_TI_WID
.EXTRN EDT$$G_VFY, EDT$$G_WD_WRAP
.EXTRN EDT$$G_HELP_NAM
.EXTRN EDT$$G_HELP_NAMLEN
.EXTRN EDT$$G_HELP_SET
.EXTRN EDT$$A_CMD_NAM, EDT$$G_CMD_NAMLEN
.EXTRN EDT$$G_TI_SCROLL
.EXTRN EDT$$G_EIGHT_BIT
.EXTRN EDT$$G_TI_EDIT, EDT$$G_ENB_AUTRPT
.EXTRN EDT$$G_SUMRY, EDT$$G_WRDTYP
.EXTRN EDT$$G_PARTYP, EDT$$A_EXE_CURCMD
.EXTRN EDT$$G_FMT_LNPOS
.EXTRN EDT$$G_PMT_LINE
.EXTRN EDT$$G_PMT_KPD, EDT$$G_PMT_NOKPD
.EXTRN EDT$$G_PMT_HCCHG
.EXTRN EDT$$G_PMT_INS, EDT$$G_PMT_INSN
.EXTRN EDT$$G_PMT_QUERY
.EXTRN EDT$_KEYNOTDEF, EDT$_NODEFN
.EXTRN EDT$_INVPARFOR, EDT$$INTER_ERR

```

OFFC 00000

```

.ENTRY EDT$$$SHW_CMD, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 0769
R10,R11
MOVAB EDT$$INTER_ERR, R11
MOVAB EDT$$FMT_CRLF, R10
MOVAB EDT$$G_FMT_LNPOS, R9
MOVAB EDT$$FMT_CR, R8
MOVAB EDT$$FMT_STR, R7
MOVAB P.AAA, R6
SUBL2 #4, SP
MOVL EDT$$A_EXE_CURCMD, R2
CASEL 4(R2), #1, #28
.WORD 2$-1$, -
      8$-1$, -
      13$-1$, -
      26$-1$, -
      40$-1$, -
      42$-1$, -
      44$-1$, -
      53$-1$, -
      54$-1$, -
      57$-1$, -
      59$-1$, -
      61$-1$, -
      63$-1$, -
      65$-1$, -
      67$-1$, -
      70$-1$, -
      71$-1$, -
      73$-1$, -
      77$-1$, -
      80$-1$, -

```

```

0123 008B 0092 003E 00039 1$:
0265 01D2 01BE 019E 00041
02C1 02A8 028F 027F 00049
0327 030F 02F3 02DA 00051
0399 0380 0348 032F 00059
0493 0454 03CA 03B2 00061
0513 04F9 04C1 04AA 00069
          0555 00071

```

0935

					17\$-14\$,-		
					18\$-14\$,-		
					19\$-14\$		
	68	00	FB	00106	CALLS	#0, EDT\$\$INTER_ERR	1023
		24	11	00109	BRB	21\$	1004
		07	DD	0010B	PUSHL	#7	1008
		20	A6	9F 0010D	PUSHAB	P.AAF	
			1A	11 00110	BRB	20\$	
		05	DD	00112	PUSHL	#5	1011
		28	A6	9F 00114	PUSHAB	P.AAG	
			13	11 00117	BRB	20\$	
		03	DD	00119	PUSHL	#3	1014
		30	A6	9F 0011B	PUSHAB	P.AAH	
			0C	11 0011E	BRB	20\$	
		10	DD	00120	PUSHL	#16	1017
		34	A6	9F 00122	PUSHAB	P.AAI	
			05	11 00125	BRB	20\$	
		19	DD	00127	PUSHL	#25	1020
		44	A6	9F 00129	PUSHAB	P.AAJ	
	67	02	FB	0012C	CALLS	#2, EDT\$\$FMT_STR	
	07	00000000G	00	E9 0012F	BLBC	EDT\$\$G_SEA_BEG, 22\$	1026
			07	DD 00136	PUSHL	#7	1028
		5D	A6	9F 00138	PUSHAB	P.AAK	
			05	11 0013B	BRB	23\$	
		05	DD	0013D	PUSHL	#5	1030
		64	A6	9F 0013F	PUSHAB	P.AAL	
	67	00000000G	02	FB 00142	CALLS	#2, EDT\$\$FMT_STR	
			00	D5 00145	TSTL	EDT\$\$G_SEA_BRDD	1032
			08	12 0014B	BNEQ	24\$	
			02	DD 0014D	PUSHL	#2	
		69	A6	9F 0014F	PUSHAB	P.AAM	
	67	02	FB	00152	CALLS	#2, EDT\$\$FMT_STR	
		07	DD	00155	PUSHL	#7	1034
		68	A6	9F 00157	PUSHAB	P.AAN	
			79	11 0015A	BRB	39\$	
	00	00000000G	00	CF 0015C	CASEL	EDT\$\$G_TI_TYP, #0, #3	1041
0023		03					
	001B						
	0014	00UD		00164	.WORD	28\$-27\$,-	
						29\$-27\$,-	
						30\$-27\$,-	
						31\$-27\$	
	68	00	FB	0016C	CALLS	#0, EDT\$\$INTER_ERR	1057
		1F	11	0016F	BRB	33\$	1041
		07	DD	00171	PUSHL	#7	1045
		74	A6	9F 00173	PUSHAB	P.AAO	
			15	11 00176	BRB	32\$	
		04	DD	00178	PUSHL	#4	1048
		7C	A6	9F 0017A	PUSHAB	P.AAP	
			0E	11 0017D	BRB	32\$	
		05	DD	0017F	PUSHL	#5	1051
		0080	C6	9F 00181	PUSHAB	P.AAQ	
			06	11 00185	BRB	32\$	
		08	DD	00187	PUSHL	#8	1054
		0088	C6	9F 00189	PUSHAB	P.AAR	
	67	02	FB	0018D	CALLS	#2, EDT\$\$FMT_STR	
	08	00000000G	00	E9 00190	BLBC	EDT\$\$G_TI_SCROLL, 34\$	1060
			08	DD 00197	PUSHL	#8	1062
		0090	C6	9F 00199	PUSHAB	P.AAS	

		06	11	0019D		BRB	35\$		
		0A	DD	0019F	34\$:	PUSHL	#10		1064
	0098	C6	9F	001A1		PUSHAB	P.AAT		
67		02	FB	001A5	35\$:	CALLS	#2, EDT\$\$FMT_STR		
08	00000000G	00	E9	001A8		BLBC	EDT\$\$G_EIGHT_BIT, 36\$		1069
		0A	DD	001AF		PUSHL	#10		1071
	00A4	C6	9F	001B1		PUSHAB	P.AAU		
		06	11	001B5		BRB	37\$		
		0C	DD	001B7	36\$:	PUSHL	#12		1075
	00B0	C6	9F	001B9		PUSHAB	P.AAV		
67		02	FB	001BD	37\$:	CALLS	#2, EDT\$\$FMT_STR		
08	00000000G	00	E9	001C0		BLBC	EDT\$\$G_TI_EDIT, 38\$		1077
		06	DD	001C7		PUSHL	#6		1079
	00BC	C6	9F	001C9		PUSHAB	P.AAW		
		39	11	001CD		BRB	43\$		
		08	DD	001CF	38\$:	PUSHL	#8		1081
	00C4	C6	9F	001D1		PUSHAB	P.AAX		
		31	11	001D5	39\$:	BRB	43\$		
		12	DD	001D7	40\$:	PUSHL	#18		1091
	00CC	C6	9F	001D9		PUSHAB	P.AAY		
67		02	FB	001DD		CALLS	#2, EDT\$\$FMT_STR		
08	00000000G	00	E9	001E0		BLBC	EDT\$\$G_EIGHT_BIT, 41\$		1102
		3E	DD	001E7		PUSHL	#62		1104
	00E0	C6	9F	001E9		PUSHAB	P.AAZ		
		19	11	001ED		BRB	43\$		
		3E	DD	001EF	41\$:	PUSHL	#62		1108
	0120	C6	9F	001F1		PUSHAB	P.ABA		
		11	11	001F5		BRB	43\$		
50	10	A2	DO	001F7	42\$:	MOVL	16(R2), R0		1120
50	00000000G	00	DO	001FB		MOVL	EDT\$\$A_US_ENT-4[R0], ENT		
51		80	9A	00203		MOVZBL	(ENT)+, LEN		1121
		03	BB	00206		PUSHR	#*M<R0,R1>		1122
		03D5	31	00208	43\$:	BRW	120\$		
09	00000000G	00	E8	0020B	44\$:	BLBS	EDT\$\$G_TRN_TBLINIT, 45\$		1133
00000000G	00	00	FB	00212		CALLS	#0, EDT\$\$DEF_DFLTK		
		03	11	00219		BRB	46\$		
50		01	DO	0021B	45\$:	MOVL	#1, STATUS		
72		50	E9	0021E	46\$:	BLBC	STATUS, 51\$		1135
50	00000000G	00	DO	00221		MOVL	EDT\$\$A_EXE_CURCMD, R0		1139
	10	A0	DD	00228		PUSHL	16(R0)		
00000000G	00	01	FB	0022B		CALLS	#1, EDT\$\$TST_BADK		
08		50	E9	00232		BLBC	R0, 47\$		
	00000000G	8F	DD	00235		PUSHL	#EDT\$_KEYNOTDEF		1141
		58	11	0023B		BRB	52\$		
		5E	DD	0023D	47\$:	PUSHL	SP		1144
50	00000000G	00	DO	0023F		MOVL	EDT\$\$A_EXE_CURCMD, R0		
	10	A0	DD	00246		PUSHL	16(R0)		
00000000G	00	02	FB	00249		CALLS	#2, EDT\$\$FIND_KEY		
38		50	E9	00250		BLBC	R0, 50\$		
50		6E	DO	00253		MOVL	KEY_PTR, R0		1156
52	07	A0	9E	00256		MOVAB	7(R0), KEY_P		
54	06	A0	9A	0025A		MOVZBL	6(R0), KEY_L		1157
		69	D4	0025E		CLRL	EDT\$\$G_FMT_LNPOS		1158
55		82	9A	00260	48\$:	MOVZBL	(KEY_PT)+, CH		1159
		54	D7	00263		DECL	KEY_C		1160
		5E	19	00265		BLSS	56\$		1162
53		69	DO	00267		MOVL	EDT\$\$G_FMT_LNPOS, R3		1165

.....

		53	DD	0026A	PUSHL	R3		
		55	DD	0026C	PUSHL	CH		
00000000G	00	02	FB	0026E	CALLS	#2, EDT\$\$\$FMT_CHWID		
	50	53	CO	00275	ADDL2	R3, RO		
00000000G	00	50	D1	00278	CMPL	RO, EDT\$\$\$G_TI_WID		1166
		03	19	0027F	BLSS	49\$		
	6A	00	FB	00281	CALLS	#0, EDT\$\$\$FMT_CRLF		1168
		55	DD	00284	PUSHL	CH		1170
	68	01	FB	00286	CALLS	#1, EDT\$\$\$FMT_CH		
		D5	11	00289	BRB	48\$		1171
		00000000G	8F	DD	0028B	#EDT\$_NODEFN		1177
			02	11	00291	BRB	52\$	
			50	DD	00293	PUSHL	STATUS	1181
00000000G	00	01	FB	00295	CALLS	#1, EDT\$\$\$MSG_TOSTR		
		27	11	0029C	BRB	56\$		1183
		00000000G	00	DD	0029E	PUSHL	EDT\$\$\$G_SCLL_TOP	1188
00000000G	00	01	FB	002A4	CALLS	#1, EDT\$\$\$FMT_DCML		
		3A	DD	002AB	PUSHL	#58		1189
	68	01	FB	002AD	CALLS	#1, EDT\$\$\$FMT_CH		
		00000000G	00	DD	002B0	PUSHL	EDT\$\$\$G_SCLL_BOT	1190
			06	11	002B6	BRB	55\$	
		00000000G	00	DD	002B8	PUSHL	EDT\$\$\$G TI WID	1196
00000000G	00	01	FB	002BE	CALLS	#1, EDT\$\$\$FMT_DCML		
		031B	31	002C5	BRW	121\$		1197
		00000000G	00	D5	002C8	TSTL	EDT\$\$\$G_NOS	1203
			09	12	002CE	BNEQ	58\$	
			02	DD	002D0	PUSHL	#2	
	015E	C6	9F	002D2	PUSHAB	P.ABB		
		67	02	FB	002D6	CALLS	#2, EDT\$\$\$FMT_STR	
			07	DD	002D9	PUSHL	#7	1205
	0160	C6	9F	002DB	PUSHAB	P.ABC		
			7D	11	002DF	BRB	69\$	
		00000000G	00	D5	002E1	TSTL	EDT\$\$\$G_VFY	1212
			09	12	002E7	BNEQ	60\$	
			02	DD	002E9	PUSHL	#2	
	0167	C6	9F	002EB	PUSHAB	P.ABD		
		67	02	FB	002EF	CALLS	#2, EDT\$\$\$FMT_STR	
			06	DD	002F2	PUSHL	#6	1214
	0169	C6	9F	002F4	PUSHAB	P.ABE		
			64	11	002F8	BRB	69\$	
		00000000G	00	D5	002FA	TSTL	EDT\$\$\$G_TRUN	1221
			09	12	00300	BNEQ	62\$	
			02	DD	00302	PUSHL	#2	
	016F	C6	9F	00304	PUSHAB	P.ABF		
		67	02	FB	00308	CALLS	#2, EDT\$\$\$FMT_STR	
			08	DD	0030B	PUSHL	#8	1223
	0171	C6	9F	0030D	PUSHAB	P.ABG		
			7C	11	00311	BRB	74\$	
		00000000G	00	D5	00313	TSTL	EDT\$\$\$G_KPAD	1230
			09	12	00319	BNEQ	64\$	
			02	DD	0031B	PUSHL	#2	
	0179	C6	9F	0031D	PUSHAB	P.ABH		
		67	02	FB	00321	CALLS	#2, EDT\$\$\$FMT_STR	
			06	DD	00324	PUSHL	#6	1232
	017B	C6	9F	00326	PUSHAB	P.ABI		
			63	11	0032A	BRB	74\$	
		50 00000000G	00	D0	0032C	MOVL	EDT\$\$\$G_WD_WRAP, RO	1239

00000100	8F	50	D1	00333	CMPL	R0, #256	
		08	12	0033A	BNEQ	66\$	
		05	DD	0033C	PUSHL	#6	1241
	0181	C6	9F	0033E	PUSHAB	P.ABJ	
		4B	11	00342	BRB	74\$	
		50	DD	00344	PUSHL	R0	1243
		6E	11	00346	BRB	76\$	
	00000000G	00	D5	00348	TSTL	EDT\$\$G_EDIT_DFLTMOD	1251
		08	12	0034E	BNEQ	68\$	
		06	DD	00350	PUSHL	#6	1253
	0188	C6	9F	00352	PUSHAB	P.ABK	
		78	11	00356	BRB	79\$	
		04	DD	00358	PUSHL	#4	1255
	0190	C6	9F	0035A	PUSHAB	P.ABL	
		70	11	0035E	BRB	79\$	
	00000000G	00	DD	00360	PUSHL	EDT\$\$G_SCR_LNS	1262
		4E	11	00366	BRB	76\$	
	00000000G	00	D5	00368	TSTL	EDT\$\$G_QUIET	1269
		09	12	0036E	BNEQ	72\$	
		02	DD	00370	PUSHL	#2	
	0194	C6	9F	00372	PUSHAB	P.ABM	
		02	FB	00376	CALLS	#2, EDT\$\$FMT_STR	
		05	DD	00379	PUSHL	#5	1271
	0196	C6	9F	0037B	PUSHAB	P.ABN	
		7F	11	0037F	BRB	84\$	
	00000000G	00	D5	00381	TSTL	EDT\$\$G_TAB_SIZ	1278
		08	12	00387	BNEQ	75\$	
		05	DD	00389	PUSHL	#5	1281
	0198	C6	9F	0038B	PUSHAB	P.ABO	
		6F	11	0038F	BRB	84\$	
		09	DD	00391	PUSHL	#9	1286
	01A0	C6	9F	00393	PUSHAB	P.ABP	
		02	FB	00397	CALLS	#2, EDT\$\$FMT_STR	
		00	DD	0039A	PUSHL	EDT\$\$G_TAB_SIZ	1287
	00000000G	00	FB	003A0	CALLS	#1, EDT\$\$FMT_DCML	
		0C	DD	003A7	PUSHL	#12	1288
	01A9	C6	9F	003A9	PUSHAB	P.ABQ	
		02	FB	003AD	CALLS	#2, EDT\$\$FMT_STR	
		00	DD	003B0	PUSHL	EDT\$\$G_TAB_LVL	1289
		FF05	31	003B6	BRW	55\$	
	00000000G	00	D5	003B9	TSTL	EDT\$\$G_RPT	1298
		09	12	003BF	BNEQ	78\$	
		02	DD	003C1	PUSHL	#2	
	01B5	C6	9F	003C3	PUSHAB	P.ABR	
		02	FB	003C7	CALLS	#2, EDT\$\$FMT_STR	
		06	DD	003CA	PUSHL	#6	1300
	01B7	C6	9F	003CC	PUSHAB	P.ABS	
		2E	11	003D0	BRB	84\$	
	00000000G	00	D5	003D2	TSTL	EDT\$\$G_FNF_MSGFLG	1307
		09	12	003D8	BNEQ	81\$	
		02	DD	003DA	PUSHL	#2	
	01BD	C6	9F	003DC	PUSHAB	P.ABT	
		02	FB	003E0	CALLS	#2, EDT\$\$FMT_STR	
		03	DD	003E3	PUSHL	#3	1309
	01BF	C6	9F	003E5	PUSHAB	P.ABU	
		15	11	003E9	BRB	84\$	
	08 00000000G	00	E8	003EB	BLBS	EDT\$\$G_SUMRY, 83\$	1319

			09	DD	003F2		PUSHL	#9		1321
		01C2	C6	9F	003F4		PUSHAB	P.ABV		
			06	11	003F8		BRB	84\$		
			07	DD	003FA	83\$:	PUSHL	#7		1325
		01CB	C6	9F	003FC		PUSHAB	P.ABV		
			01DD	31	00400	84\$:	BRW	120\$		
		50	10	A2	00403	85\$:	MOVL	16(R2), PROMPT_NUM		1339
	06	01	50	CF	00407		CASEL	PROMPT_NUM, #1, #6		1340
0030	0027	001E	0015		0040B	86\$:	.WORD	87\$-86\$,-		
	004B	0042	0039		00413			88\$-86\$,-		
								89\$-86\$,-		
								90\$-86\$,-		
								91\$-86\$,-		
								92\$-86\$,-		
								93\$-86\$,-		
		6B	00	FB	00419		CALLS	#0, EDT\$\$INTER_ERR		1351
			53	D4	0041C		CLRL	TXT		1350
			3D	11	0041E		BRB	94\$		
		53	00000000G	00	9E	00420	87\$:	MOVAB	EDT\$\$T_PMT_LINE, TXT	1340
				34	11	00427		BRB	94\$	
		53	00000000G	00	9E	00429	88\$:	MOVAB	EDT\$\$T_PMT_KPD, TXT	
				2B	11	00430		BRB	94\$	
		53	00000000G	00	9E	00432	89\$:	MOVAB	EDT\$\$T_PMT_NOKPD, TXT	
				22	11	00439		BRB	94\$	
		53	00000000G	00	9E	0043B	90\$:	MOVAB	EDT\$\$T_PMT_HCCHG, TXT	
				19	11	00442		BRB	94\$	
		53	00000000G	00	9E	00444	91\$:	MOVAB	EDT\$\$T_PMT_INS, TXT	
				10	11	0044B		BRB	94\$	
		53	00000000G	00	9E	0044D	92\$:	MOVAB	EDT\$\$T_PMT_INSN, TXT	
				07	11	00454		BRB	94\$	
		53	00000000G	00	9E	00456	93\$:	MOVAB	EDT\$\$T_PMT_QUERY, TXT	
		55		83	9A	0045D	94\$:	MOVZBL	(TXT)+, LEN	1355
				69	D4	00460		CLRL	EDT\$\$G_FMT_LNPOS	1359
		54		83	9A	00462	95\$:	MOVZBL	(TXT)+, CH	1360
				55	D7	00465		DECL	LEN	1361
				3A	19	00467		BLSS	99\$	1363
		52		69	D0	00469		MOVL	EDT\$\$G_FMT_LNPOS, R2	1366
				52	DD	0046C		PUSHL	R2	
				54	DD	0046E		PUSHL	CH	
	00000000G	00	02	FB	00470		CALLS	#2, EDT\$\$FMT_CHWID		
		50		52	C0	00477		ADDL2	R2, R0	
	00000000G	00	50	D1	0047A		CMPL	R0, EDT\$\$G_TI_WID		
				03	19	00481		BLSS	96\$	
		6A		00	FB	00483		CALLS	#0, EDT\$\$FMT_CRLF	1368
				54	DD	00486	96\$:	PUSHL	CH	1370
		68		01	FB	00488		CALLS	#1, EDT\$\$FMT_CH	
				D5	11	0048B		BRB	95\$	1371
		50	10	A2	D0	0048D	97\$:	MOVL	16(R2), R0	1386
		54	00000000G	0040	D0	00491		MOVL	EDT\$\$A_US TXT-4[R0], TXT	
		55		84	9A	00499		MOVZBL	(TXT)+, LEN	1387
				69	D4	0049C		CLRL	EDT\$\$G_FMT_LNPOS	1391
		53		84	9A	0049E	98\$:	MOVZBL	(TXT)+, CH	1392
				55	D7	004A1		DECL	LEN	1393
				03	18	004A3	99\$:	BGEQ	100\$	1395
				013B	31	004A5		BRW	121\$	
		52		69	D0	004AB	100\$:	MOVL	EDT\$\$G_FMT_LNPOS, R2	1398
				52	DD	004AB		PUSHL	R2	

00000000G	00	53	DD	004AD	PUSHL	CH				
	50	02	FB	004AF	CALLS	#2,	EDT\$\$\$FMT_CHWID			
00000000G	00	52	CO	004B6	ADDL2	R2,	R0			
		50	D1	004B9	CMPL	R0,	EDT\$\$\$G_TI_WID			
		03	19	004C0	BLSS	101\$				
	6A	00	FB	004C2	CALLS	#0,	EDT\$\$\$FMT_CRLF		1400	
		53	DD	004C5	PUSHL	CH			1402	
	68	01	FB	004C7	CALLS	#1,	EDT\$\$\$FMT_CH			
		D2	11	004CA	BRB	98\$			1403	
	08	00000000G	00	E9	004CC	102\$	BLBC	EDT\$\$\$G_WRDTYP,	103\$	1416
			0C	DD	004D3		PUSHL	#12		1418
		01D2	C6	9F	004D5		PUSHAB	P.ABX		
			6E	11	004D9		BRB	111\$		
			0A	DD	004DB	103\$	PUSHL	#10		1422
		01DE	C6	9F	004DD		PUSHAB	P.ABY		
			66	11	004E1		BRB	111\$		
	08	00000000G	00	E9	004E3	104\$	BLBC	EDT\$\$\$G_PARTYP,	105\$	1433
			06	DD	004EA		PUSHL	#6		1435
		01E8	C6	9F	004EC		PUSHAB	P.ABZ		
			57	11	004F0		BRB	111\$		
			04	DD	004F2	105\$	PUSHL	#4		1439
		01EE	C6	9F	004F4		PUSHAB	P.ACA		
			4F	11	004F8		BRB	111\$		
	50		01	D0	004FA	106\$	MOVL	#1,	STATUS	1455
	13	00000000G	00	E8	004FD		BLBS	EDT\$\$\$G_HELP_SET,	107\$	1457
		00000000G	00	DD	00504		PUSHL	EDT\$\$\$G_HELP_NAMLEN		1460
		00000000G	00	9F	0050A		PUSHAB	EDT\$\$\$T_HELP_NAM		1459
00000000G	00		02	FB	00510		CALLS	#2,	EDT\$\$\$SET_HLPFNAM	
	01		50	E8	00517	107\$	BLBS	STATUS,	108\$	1462
				04	0051A		RET			
			10	DD	0051B	108\$	PUSHL	#16		1465
		01F2	C6	9F	0051D		PUSHAB	P.ACB		
	67		02	FB	00521		CALLS	#2,	EDT\$\$\$FMT_STR	
		00000000G	00	DD	00524		PUSHL	EDT\$\$\$G_HELP_NAMLEN		1469
		00000000G	00	9F	0052A		PUSHAB	EDT\$\$\$T_HELP_NAM		
			17	11	00530		BRB	111\$		
	01	00000000G	00	D1	00532	109\$	CMPL	EDT\$\$\$G_ENB_AURPT,	#1	1482
			08	12	00539		BNEQ	110\$		
			0A	DD	0053B		PUSHL	#10		1484
		0202	C6	9F	0053D		PUSHAB	P.ACC		
			06	11	00541		BRB	111\$		
			0C	DC	00543	110\$	PUSHL	#12		1486
		020C	C6	9F	00545		PUSHAB	P.ACD		
			0086	31	00549	111\$	BRW	117\$		
	54	00000000G	00	D0	0054C	112\$	MOVL	EDT\$\$\$A_CMD_NAM,	TXT	1499
	55	00000000G	00	D0	00553		MOVL	EDT\$\$\$G_CMD_NAMLEN,	LEN	1500
			54	D5	0055A		TSTL	TXT		1502
			03	12	0055C		BNEQ	113\$		
			0082	31	0055E		BRW	121\$		
			69	D4	00561	113\$	CLRL	EDT\$\$\$G_FMT_LNPOS		1508
	53		84	9A	00563	114\$	MOVZBL	(TXT)+,	CH	1509
			55	D7	00566		DECL	LEN		1510
			79	19	00568		BLSS	121\$		1512
	52		69	D0	0056A		MOVL	EDT\$\$\$G_FMT_LNPOS,	R2	1515
			52	DD	0056D		PUSHL	R2		
			53	DD	0056F		PUSHL	CH		
00000000G	00		02	FB	00571		CALLS	#2,	EDT\$\$\$FMT_CHWID	

	50		52	C0	00578		ADDL2	R2, R0		
00000000G	00		50	D1	0057B		CMPL	R0, EDT\$\$G_TI_WID		
			03	19	00582		BLSS	115\$		
	6A		00	FB	00584		CALLS	#0, EDT\$\$FMT_CRLF		1517
			53	DD	00587	115\$:	PUSHL	CH		1519
	68		01	FB	00589		CALLS	#1, EDT\$\$FMT_CH		
			D5	11	0058C		BRB	114\$		1520
52	00000000G		00	D0	0058E	116\$:	MOVL	EDT\$\$A_INP_NAM, TXT		1536
53	00000000G		00	D0	00595		MOVL	EDT\$\$G_INP_NAMLEN, LEN		1537
			0D	DD	0059C		PUSHL	#13		1538
		0218	C6	9F	0059E		PUSHAB	P.ACE		
	67		02	FB	005A2		CALLS	#2, EDT\$\$FMT_STR		
			0C	BB	005A5		PUSHR	#*M<R2,R3>		1539
	67		02	FB	005A7		CALLS	#2, EDT\$\$FMT_STR		
	6A		00	FB	005AA		CALLS	#0, EDT\$\$FMT_CRLF		1540
			0D	DD	005AD		PUSHL	#13		1541
		0225	C6	9F	005AF		PUSHAB	P.ACF		
	67		02	FB	005B3		CALLS	#2, EDT\$\$FMT_STR		
50	00000000G		00	D0	005B6		MOVL	EDT\$\$G_OUT_NAMLEN, R0		1543
			19	12	005BD		BNEQ	119\$		
00000000G	8F	00000000G	00	D3	005BF		BITL	EDT\$\$V_OPTIONS, #EDT\$M_NOOUTPUT		1546
			08	13	005CA		BEQL	118\$		
			0C	DD	005CC		PUSHL	#12		1548
		0232	C6	9F	005CE		PUSHAB	P.ACG		
			0C	11	005D2	117\$:	BRB	120\$		
			0C	BB	005D4	118\$:	PUSHR	#*M<R2,R3>		1550
			08	11	005D6		BRB	120\$		
			50	DD	005D8	119\$:	PUSHL	R0		1553
		00000000G	00	DD	005DA		PUSHL	EDT\$\$A_OUT_NAM		
	67		02	FB	005E0	120\$:	CALLS	#2, EDT\$\$FMT_STR		
	6A		00	FB	005E3	121\$:	CALLS	#0, EDT\$\$FMT_CRLF		1555
			04	005E6			RET			1562

; Routine Size: 1511 bytes, Routine Base: _EDT\$CODE + 023E

; 926 1563 1
; 927 1564 1 !<BLF/PAGE>

EDT\$LSHOW
V04-000

EDT\$LSHOW - SHOW line-mode command
EDT\$\$SHW_CMD - SHOW line-mode command

B 13
16-Sep-1984 00:59:07
14-Sep-1984 12:23:40

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

Page 29
(4)

: 929 1565 1 END
: 930 1566 1
: 931 1567 0 ELUDOM

! of module EDT\$LSHOW

PSECT SUMMARY

Name Bytes Attributes
_EDT\$CODE 2085 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
-\$255\$DUA28:[EDT.SRC]SUPPORTS.L32;1	2	2	100	5	00:00.1
-\$255\$DUA28:[EDT.SRC]KEYPADDEF.L32;1	34	5	14	7	00:00.2

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS\$:LSHOW/OBJ=OBJ\$:LSHOW MSRC\$:LSHOW.BLI/UPDATE=(ENHS:LSHOW)
: Size: 1511 code + 574 data bytes
: Run Time: 01:09.7
: Elapsed Time: 01:25.6
: Lines/CPU Min: 1349
: Lexemes/CPU-Min: 5325
: Memory Used: 471 pages
: Compilation Complete

LNCL
LIS

LTYPE
LIS

LQUERY
LIS

LSUB
LIS

LRES
LIS

LTADJ
LIS

LPRINT
LIS

LGETSTR
LIS

LPUTCHR
LIS

LSUBSN
LIS

LSHOW
LIS

LMOVE
LIS

LSET
LIS

LWRITE
LIS

LINSERT
LIS

LNONCTG
LIS

LSUBS
LIS