

```

EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDDD TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDDD TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDDD TTTTTTTTTTTTTTTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEEEEEEEEEEEEE DDD DDD TTT
EEEEEEEEEEEEEE DDD DDD TTT
EEEEEEEEEEEEEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDDD TTT
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDDD TTT
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDDD TTT

```

EXE

Mod

EDT

ED

ED
EDED
EDED
EDED
ED

ED

ED

ED

ED

ED
EDSYN
LBA

110

```
LL      SSSSSSSS  EEEEEEEEE  TTTTTTTTTT
LL      SSSSSSSS  EEEEEEEEE  TTTTTTTTTT
LL      SS        EE          TT
LL      SS        EE          TT
LL      SS        EE          TT
LL      SS        EE          TT
LL      SSSSSS    EEEEEEEE    TT
LL      SSSSSS    EEEEEEEE    TT
LL              SS        EE    TT
LL              SS        EE    TT
LL              SS        EE    TT
LL              SS        EE    TT
LLLLLLLLLL SSSSSSSS EEEEEEEEE TT
LLLLLLLLLL SSSSSSSS EEEEEEEEE TT
               ....
```

```
LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS
```

```
0001 0 %TITLE 'EDT$LSET - SET line-mode command'
0002 0 MODULE EDT$LSET (
0003 0 IDENT = 'V04-000'
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1
0031 1 ++
0032 1 FACILITY: EDT -- The DEC Standard Editor
0033 1
0034 1 ABSTRACT:
0035 1
0036 1 This module executes the line mode SET command.
0037 1
0038 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0039 1
0040 1 AUTHOR: Bob Kushlis, CREATION DATE: February 3, 1978
0041 1
0042 1 MODIFIED BY:
0043 1
0044 1 1-001 - Original. DJS 28-JAN-1981. This module was created by
0045 1 extracting the routine EDT$$SET_CMD from the module EXEC.BLI.
0046 1 1-002 - Regularize headers. JBS 20-Mar-1981
0047 1 1-003 - Use the ASSERT macro. JBS 01-Jun-1981
0048 1 1-004 - Implement virtual memory deallocation TMV 6-Aug-81
0049 1 1-005 - Use the new message codes. JBS 06-Aug-1981
0050 1 1-006 - Add set command for repeat/norepeat. STS 26-Aug-1981
0051 1 1-007 - Add set command for fnf/nofnf TMV 10-Sept-1981
0052 1 1-008 - Correct SET SEARCH command. JBS 29-Sep-1981
0053 1 1-009 - Add SET [NO]SUMMARY command, the SET SEARCH WPS command and put
0054 1 in a stub for SET PROMPT. STS 01-Oct-1981
0055 1 1-010 - Set up proper search routine with set search WPS. STS 02-Oct-1981
0056 1 1-011 - Add set up of text for page and end. STS 06-Oct-1981
0057 1 1-012 - Don't allow escape or control chars for set text string. STS 20-Oct-1981
```

```
58 0058 1 1-013 - Don't allow characters with ascii rep > delete either. STS 20-Oct-1981
59 0059 1 1-014 - Implement SET PROMPT. JBS 21-Oct-1981
60 0060 1 1-015 - Add set word and Set para. STS 22-Oct-1981
61 0061 1 1-016 - Add four more prompts and increase their lengths. JBS 23-Oct-1981
62 0062 1 1-017 - Remove check to see if we have the original strings when allocating
63 0063 1 memory for text and entity strings. STS 06-Nov-1981
64 0064 1 1-018 - Add setting and clearing of EDT$G_ENB_AUTRPT. JBS 10-Feb-1982
65 0065 1 1-019 - Add more range checking. JBS 10-Feb-1982
66 0066 1 1-020 - Correct range checks -- MAX and MIN confusion. JBS 13-Feb-1982
67 0067 1 1-021 - Perform aux keypad enable/disable on SET [NO]KEYPAD. SMB 23-Feb-1982
68 0068 1 1-022 - Only enable/disable numeric keypad if an EXT command. SMB 26-Feb-1982
69 0069 1 1-023 - Add range checks to some SET commands. JBS 10-Mar-1982
70 0070 1 1-024 - Correct the reversed test is SET CURSOR. JBS 11-Mar-1982
71 0071 1 1-025 - Add SET COMMAND. JBS 04-May-1982
72 0072 1 1-026 - Respond to error return from EDT$SET_HLPFNAM. JBS 04-May-1982
73 0073 1 1-027 - Take out setting of EDT$G_HELP SET. SMB 27-May-1982
74 0074 1 1-028 - Call EDT$SET_COMFNAM on SET COMMAND. JBS 07-Jun-1982
75 0075 1 1-029 - Don't allow SET COMMAND with no argument. JBS 08-Jun-1982
76 0076 1 1-030 - Remove prompt PRTC. STS 07-Jul-1982
77 0077 1 1-031 - Force CR,LF into first 2 prompt character positions. SMB 15-Jul-1982
78 0078 1 1-032 - Add new string search options. JBS 19-Jul-1982
79 0079 1 1-033 - Call a routine to set screen width. SMB 29-Jul-1982
80 0080 1 1-034 - Change the interface to EDT$SET_COMFNAM. JBS 23-AUG-1982
81 0081 1 1-035 - Add more SET TERM commands. JBS 02-Sep-1982
82 0082 1 1-036 - Conditionalize screen changed settings. SMB 11-Sep-1982
83 0083 1 1-037 - New screen update logic. JBS 13-Sep-1982
84 0084 1 1-038 - Change EDT$G_SCR_CHGD to EDT$G_SCR_REBUILD in a few places. JBS 09-Oct-1982
85 0085 1 1-039 - Repaint the screen if any terminal parameter is changed. JBS 01-Dec-1982
86 0086 1 1-040 - Don't allow changing of terminal type from change mode. STS 13-Dec-1982
87 0087 1 1-041 - Rebuild the screen data base on SET SCREEN. JBS 15-Dec-1982
88 0088 1 1-042 - Remove unused reference to EDT$ERA_MSGLN. JBS 20-Jan-1983
89 0089 1 1-043 - Add conditionals for WPS and VT220 support. JBS 10-Feb-1983
90 0090 1 1-044 - Don't let SET TEXT PAGE be longer than 26 characters, else it is hard
91 0091 1 to display a record containing 255 form feeds in nottruncate mode on
92 0092 1 80-character terminals. JBS 14-Jul-1983
93 0093 1 1-045 - Don't let SET SCREEN be 0. JBS 19-Sep-1983
94 0094 1 1-046 - Set EDT$G_DEC CRT when user does a SET TERMINAL VT100, VT52,
95 0095 1 HCPY, or EIGHTBIT. REM 19-Apr-1983
96 0096 1 --
97 0097 1
```

EDT\$LSET
V04-000

EDT\$LSET - SET line-mode command
Declarations

C 9
16-Sep-1984 00:57:38
14-Sep-1984 12:23:40

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

Page 3
(2)

```

: 99      0098 1 %SBTTL 'Declarations'
: 100     0099 1
: 101     0100 1 : TABLE OF CONTENTS:
: 102     0101 1 :
: 103     0102 1 :
: 104     0103 1 REQUIRE 'EDTSRC:TRAROUNAM';
: 105     0542 1
: 106     0543 1 FORWARD ROUTINE
: 107     0544 1     EDT$$SET_CMD : NOVALUE;
: 108     0545 1
: 109     0546 1
: 110     0547 1 : INCLUDE FILES:
: 111     0548 1 :
: 112     0549 1 :
: 113     0550 1 REQUIRE 'EDTSRC:EDTREQ';
: 114     0685 1
: 115     0686 1 LIBRARY 'EDTSRC:SUPPORTS';
: 116     0687 1
: 117     0688 1
: 118     0689 1 : MACROS:
: 119     0690 1
: 120     0691 1 : NONE
: 121     0692 1 :
: 122     0693 1 : EQUATED SYMBOLS:
: 123     0694 1 :
: 124     0695 1
: 125     0696 1 BIND
: 126     0697 1     KEYPAD_MODE = UPLIT (%STRING (%CHAR (ASC_K_ESC), '=')),
: 127     0698 1     NOKEYPAD_MODE = UPLIT (%STRING (%CHAR (ASC_K_ESC), '>'));
: 128     0699 1
: 129     0700 1 LITERAL
: 130     0701 1     KEYPAD_MODE_LEN = 2,
: 131     0702 1     NOKEYPAD_MODE_LEN = 2;
: 132     0703 1
: 133     0704 1
: 134     0705 1 : OWN STORAGE:
: 135     0706 1
: 136     0707 1 : NONE
: 137     0708 1
: 138     0709 1 : EXTERNAL REFERENCES:
: 139     0710 1
: 140     0711 1 : In the routine
```

EDT
V04

EDT\$LSET
V04-000

EDT\$SET - SET line-mode command
EDT\$\$SET_CMD - SET line-mode command

D 9
16-Sep-1984 00:57:38
14-Sep-1984 12:23:40

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

Page 4
(3)

```

: 142 0712 1 %SBTTL 'EDT$$SET_CMD - SET line-mode command'
: 143 0713 1
: 144 0714 1 GLOBAL ROUTINE EDT$$SET_CMD ! SET line-mode command
: 145 0715 1 : NOVALUE =
: 146 0716 1
: 147 0717 1 !++
: 148 0718 1 FUNCTIONAL DESCRIPTION:
: 149 0719 1
: 150 0720 1 Command processing routine for SET. The SET TYPE field
: 151 0721 1 contains an index identifying the type of SET command;
: 152 0722 1 case on it and handle the particular command.
: 153 0723 1
: 154 0724 1 FORMAL PARAMETERS:
: 155 0725 1
: 156 0726 1 NONE
: 157 0727 1
: 158 0728 1 IMPLICIT INPUTS:
: 159 0729 1
: 160 0730 1 EDT$$A_US_ENT
: 161 0731 1 EDT$$A_US_TXT
: 162 0732 1 EDT$$G_SCR_LNS
: 163 0733 1 EDT$$A_EXE_CURCMD
: 164 0734 1
: 165 0735 1 IMPLICIT OUTPUTS:
: 166 0736 1
: 167 0737 1 EDT$$G_NOS
: 168 0738 1 EDT$$G_CAS_FLG
: 169 0739 1 EDT$$G_EXCT_MATCH
: 170 0740 1 EDT$$G_SEA_BEG
: 171 0741 1 EDT$$G_SEA_BNDD
: 172 0742 1 EDT$$G_TI_TYP
: 173 0743 1 EDT$$G_VFY
: 174 0744 1 EDT$$G_TRUN
: 175 0745 1 EDT$$G_KPAD
: 176 0746 1 EDT$$G_SCR_CHGD
: 177 0747 1 EDT$$G_SCR_REBUILD
: 178 0748 1 EDT$$G_WD_WRAP
: 179 0749 1 EDT$$G_SCLL_BOT
: 180 0750 1 EDT$$G_SCLL_TOP
: 181 0751 1 EDT$$G_TI_WID
: 182 0752 1 EDT$$G_EDIT_DFLTMOD
: 183 0753 1 EDT$$G_SCR_LNS
: 184 0754 1 EDT$$A_US_ENT
: 185 0755 1 EDT$$A_US_TXT
: 186 0756 1 EDT$$G_QUIET
: 187 0757 1 EDT$$G_RPT
: 188 0758 1 EDT$$G_FNF_MSGFLG
: 189 0759 1 EDT$$G_TAB_SIZ
: 190 0760 1 EDT$$G_TAB_LVL
: 191 0761 1 EDT$$G_SUMRY
: 192 0762 1 EDT$$G_ENB_AUTRPT
: 193 0763 1 EDT$$T_PMT_LINE
: 194 0764 1 EDT$$T_PMT_KPD
: 195 0765 1 EDT$$T_PMT_NOKPD
: 196 0766 1 EDT$$T_PMT_HCCHG
: 197 0767 1 EDT$$T_PMT_INS
: 198 0768 1 EDT$$T_PMT_INSN
```

EDT
V04

EDT\$LSET
V04-000

EDT\$SET - SET line-mode command
EDT\$\$SET_CMD - SET line-mode command

E 9
16-Sep-1984 00:57:38
14-Sep-1984 12:23:40

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

Page 5
(3)

EDT
V04

```
.. 199 0769 1 | EDT$ST_PMT_QUERY
.. 200 0770 1 | EDT$SG_WRD_TYP
.. 201 0771 1 | EDT$SG_PARTYP
.. 202 0772 1 | EDT$SG_DEC_CRT
.. 203 0773 1 |
.. 204 0774 1 | ROUTINE VALUE:
.. 205 0775 1 |
.. 206 0776 1 | NONE
.. 207 0777 1 |
.. 208 0778 1 | SIDE EFFECTS:
.. 209 0779 1 |
.. 210 0780 1 | NONE
.. 211 0781 1 |
.. 212 0782 1 | --
.. 213 0783 1 |
.. 214 0784 2 | BEGIN
.. 215 0785 2 |
.. 216 0786 2 | EXTERNAL ROUTINE
.. 217 0787 2 | EDT$$SC_SETWID,
.. 218 0788 2 | EDT$FMT_LIT,
.. 219 0789 2 | EDT$SOUT_FMTBUF,
.. 220 0790 2 | EDT$FMT_MSG,
.. 221 0791 2 | EDT$SET_HLPFNAM,
.. 222 0792 2 | EDT$SET_COMFNAM,
.. 223 0793 2 | EDT$ALO_HEAP,
.. 224 0794 2 | EDT$DEA_HEAP : NOVALUE,
.. 225 0795 2 | EDT$CNV_UPC;
.. 226 0796 2 |
.. 227 0797 2 | EXTERNAL
.. 228 0798 2 | EDT$SG_EXT_MOD,
.. 229 0799 2 | EDT$SG_CAS_FLG,
.. 230 0800 2 | EDT$SG_EDIT_DFLTMOD,
.. 231 0801 2 | EDT$SG_FNF_MSGFLG,
.. 232 0802 2 | EDT$SA_US_ENT : VECTOR,
.. 233 0803 2 | EDT$SG_RPT,
.. 234 0804 2 | EDT$SA_US_TXT : VECTOR,
.. 235 0805 2 | EDT$SG_EXCT_MATCH,
.. 236 0806 2 | EDT$SG_KPAD,
.. 237 0807 2 | EDT$SG_NOS,
.. 238 0808 2 | EDT$SG_SEA_BEG,
.. 239 0809 2 | EDT$SG_SEA_BNDD,
.. 240 0810 2 | EDT$SG_SCR_CHGD,
.. 241 0811 2 | EDT$SG_SCR_REBUILD,
.. 242 0812 2 | EDT$SG_SCR_LNS,
.. 243 0813 2 | EDT$SG_SCLC_BOT,
.. 244 0814 2 | EDT$SG_SCLL_TOP,
.. 245 0815 2 | EDT$SG_QUIET,
.. 246 0816 2 | EDT$SG_TAB_SIZ,
.. 247 0817 2 | EDT$SG_TAB_LVL,
.. 248 0818 2 | EDT$SG_TRUN,
.. 249 0819 2 | EDT$SG_TI_TYP,
.. 250 0820 2 | EDT$SG_DEC_CRI,
.. 251 0821 2 | EDT$SG_TI_QID,
.. 252 0822 2 | EDT$SG_VFY,
.. 253 0823 2 | EDT$SG_WD_WRAP,
.. 254 0824 2 |
.. 255 L 0825 2 %IF SUPPORT_WPS
```

```
! Set the command file name
! Allocate heap storage
! Deallocate heap storage
! Convert characters to upper case
```

```
! The screen has been mangled, it must be repainted from scratch
! The text area of the screen must be rebuilt from the work file
```

```
! Terminal service class specifier
```

```
256 0826 2 %THEN
257 0827 EDT$$G_WRDTP, ! flag indicating word with delimiter or not
258 0828 EDT$$G_PARTYP, ! flag indicating wps para or not
259 0829 EDT$$G_SUMRY, ! output summary on exit flag
260 0830 %FI
261 0831
262 0832 EDT$$G_ENB_AUTRPT, ! 1 = manipulate auto-repeat on VT100, 0 = don't
263 0833 EDT$$G_TI_SCROLL, ! 1 = terminal has scrolling regions
264 0834
265 L 0835 %IF SUPPORT_VT220
266 0836 %THEN
267 0837 EDT$$G_EIGHT_BIT, ! 1 = this is an eight-bit terminal
268 0838 %FI
269 0839
270 0840 EDT$$G_TI_EDIT, ! 1 = this terminal has editing features (ICM, DCH, IL, DL)
271 0841 EDT$$A_EXE_CURCMD : REF NODE_BLOCK, ! Pointer to the current command.
272 0842 EDT$$T_PMT_LINE : VECTOR [32, BYTE], ! Counted ASCII string of line-mode prompt
273 0843 EDT$$T_PMT_KPD : VECTOR [32, BYTE], ! Counted ASCII string of keypad prompt
274 0844 EDT$$T_PMT_NOKPD : VECTOR [32, BYTE], ! Counted ASCII string of nokeypad prompt
275 0845 EDT$$T_PMT_HCHG : VECTOR [32, BYTE], ! Counted ASCII string of hard copy change mode prompt
276 0846 EDT$$T_PMT_INS : VECTOR [32, BYTE], ! Counted ASCII string of line-mode insert prompt
277 0847 EDT$$T_PMT_INSN : VECTOR [32, BYTE], ! Counted ASCII string of line-mode insert nonumbers prompt
278 0848 EDT$$T_PMT_QUERY : VECTOR [32, BYTE], ! Counted ASCII string of /QUERY prompt
279 0849 EDT$$T_CMD_NAM_DEF4; ! Default name for command file
280 0850
281 0851 !
282 0852 !
283 0853 !
284 0854 CASE .EDT$$A_EXE_CURCMD [SET_TYPE] FROM 1 TO 36 OF
285 0855 SET
286 0856
287 0857 [1] : ! Set numbers
288 0858 EDT$$G_NOS = 1;
289 0859
290 0860 [2] : ! Set nonumbers
291 0861 EDT$$G_NOS = 0;
292 0862
293 0863 [3] : ! Set case
294 0864 EDT$$G_CAS_FLG = .EDT$$A_EXE_CURCMD [SET_VAL] - 1;
295 0865
296 0866 [4] : ! Set search
297 0867
298 0868 CASE .EDT$$A_EXE_CURCMD [SET_VAL] FROM 1 TO 11 OF
299 0869 SET
300 0870
301 0871 [1] : ! General
302 0872 EDT$$G_EXCT_MATCH = 0;
303 0873
304 0874 [2] : ! Exact
305 0875 EDT$$G_EXCT_MATCH = 1;
306 0876
307 0877 [3, 4] : ! Begin/End
308 0878 EDT$$G_SEA_BEG = .EDT$$A_EXE_CURCMD [SET_VAL] - 3;
309 0879
310 0880 [5, 6] : ! Bounded/Unbounded
311 0881 EDT$$G_SEA_BNDD = .EDT$$A_EXE_CURCMD [SET_VAL] - 5;
312 0882
```

EDT\$LSET
V04-000

EDT\$SET - SET line-mode command
EDT\$\$SET_CMD - SET line-mode command

G 9
16-Sep-1984 00:57:38
14-Sep-1984 12:23:40

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

Page 7
(3)

EDT
V04

```

313 0883 2      [7] :                               ! WPS type search
314 0884 2      EDT$$G_EXCT_MATCH = 2;
315 0885 2
316 0886 2      [8, 10] :                             ! CI or CASE INSENSITIVE
317 0887 2      EDT$$G_EXCT_MATCH = 3;
318 0888 2
319 0889 2      [9, 11] :                             ! DI or DIACRITICAL INSENSITIVE
320 0890 2      EDT$$G_EXCT_MATCH = 4;
321 0891 2
322 0892 2      [OUTRANGE] :
323 0893 2      ASSERT (0);
324 0894 2      TES;
325 0895 2
326 0896 2      [5] :                               ! Terminal
327 0897 2      BEGIN
328 0898 2
329 0899 2      IF (.EDT$$G_EXT_MOD AND (.EDT$$A_EXE_CURCMD [SET_VAL] LSS 4))
330 0900 2      THEN
331 0901 2      EDT$$FMT_MSG (EDT$_NOSETTRM)
332 0902 2      ELSE
333 0903 2      BEGIN
334 0904 2
335 0905 2      CASE .EDT$$A_EXE_CURCMD [SET_VAL] FROM 1 TO 9 OF
336 0906 2      SET
337 0907 2
338 0908 2      [1] :                               ! VT52
339 0909 2      BEGIN
340 0910 2      EDT$$G_DEC_CRT = 0;           ! indicate service class 0
341 0911 2      EDT$$G_TI_TYP = TERM_VT52;
342 0912 2      EDT$$G_TI_SCROLL = 0
343 0913 2      END;
344 0914 2
345 0915 2      [2] :                               ! VT100
346 0916 2      BEGIN
347 0917 2      EDT$$G_DEC_CRT = 1;           ! indicate service class 1
348 0918 2      EDT$$G_TI_TYP = TERM_VT100;
349 0919 2      EDT$$G_TI_SCROLL = 1
350 0920 2      END;
351 0921 2
352 0922 2      [3] :                               ! HCPY
353 0923 2      BEGIN
354 0924 2      EDT$$G_DEC_CRT = 0;           ! indicate service class 0
355 0925 2      EDT$$G_TI_TYP = TERM_HCPY
356 0926 2      END;
357 0927 2
358 0928 2      [4] :                               ! SCROLL
359 0929 2      EDT$$G_TI_SCROLL = 1;
360 0930 2
361 0931 2      [5] :                               ! NOSCROLL
362 0932 2      EDT$$G_TI_SCROLL = 0;
363 0933 2
364 0934 2      [6] :                               ! EIGHTBIT
365 0935 2      BEGIN
366 0936 2
367 0937 2      L %IF SUPPORT_VT220
368 0938 2      %THEN
369 0939 2      EDT$$G_DEC_CRT = 2;           ! indicate service class 2
```

```
: 370          0940 5          EDT$$G_EIGHT_BIT = 1;
: 371          U 0941 5 %ELSE
: 372          U 0942 5          0
: 373          0943 5 %FI
: 374          0944 5
: 375          0945 4          END;
: 376          0946 4
: 377          0947 4          [7] :          ! NOEIGHTBIT
: 378          0948 5          BEGIN
: 379          0949 5
: 380          L 0950 5 %IF SUPPORT_VT220
: 381          0951 5 %THEN
: 382          0952 5          EDT$$G_EIGHT_BIT = 0;
: 383          U 0953 5 %ELSE
: 384          U 0954 5          0
: 385          0955 5 %FI
: 386          0956 5
: 387          0957 4          END;
: 388          0958 4
: 389          0959 4          [8] :          ! EDIT
: 390          0960 4          EDT$$G_TI_EDIT = 1;
: 391          0961 4
: 392          0962 4          [9] :          ! NOEDIT
: 393          0963 4          EDT$$G_TI_EDIT = 0;
: 394          0964 4
: 395          0965 4          [OUTRANGE] :
: 396          0966 4          ASSERT (0);
: 397          0967 4          TES;
: 398          0968 4
: 399          0969 4          EDT$$G_SCR_CHGD = 1;
: 400          0970 3          END;
: 401          0971 3
: 402          0972 3          END;
: 403          0973 3
: 404          0974 3          [6] :          ! Verify
: 405          0975 3          EDT$$G_VFY = 1;
: 406          0976 3
: 407          0977 3          [7] :          ! Noverify
: 408          0978 3          EDT$$G_VFY = 0;
: 409          0979 3
: 410          0980 3          [8] :          ! Truncate
: 411          0981 3
: 412          0982 3          IF (.EDT$$G_TRUN NEQ 1)
: 413          0983 3          THEN
: 414          0984 3          BEGIN
: 415          0985 3          EDT$$G_TRUN = 1;
: 416          0986 3          EDT$$G_SCR_REBUILD = 1;
: 417          0987 3          END;
: 418          0988 3
: 419          0989 3          [9] :          ! Notruncate
: 420          0990 3
: 421          0991 3          IF (.EDT$$G_TRUN NEQ 0)
: 422          0992 3          THEN
: 423          0993 3          BEGIN
: 424          0994 3          EDT$$G_TRUN = 0;
: 425          0995 3          EDT$$G_SCR_REBUILD = 1;
: 426          0996 3          END;
```

```

427 0997 2
428 0998 2 [10] : ! Keypad
429 0999 2 BEGIN
430 1000 2
431 1001 4 IF (((.EDT$$G_TI_TYP EQL TERM_VT52) OR (.EDT$$G_TI_TYP EQL TERM_VT100)) AND (.EDT$$G_EXT_MOD))
432 1002 3 THEN
433 1003 4 BEGIN
434 1004 4 EDT$$FMT_LIT (KEYPAD_MODE, KEYPAD_MODE_LEN);
435 1005 4 EDT$$OUT_FMTBUF ();
436 1006 4 EDT$$G_SCR_REBUILD = 1;
437 1007 4 END;
438 1008 2
439 1009 2 EDT$$G_KPAD = 1;
440 1010 2 END;
441 1011 2
442 1012 2 [11] : ! Nokeypad
443 1013 2 BEGIN
444 1014 2
445 1015 4 IF (((.EDT$$G_TI_TYP EQL TERM_VT52) OR (.EDT$$G_TI_TYP EQL TERM_VT100)) AND (.EDT$$G_EXT_MOD))
446 1016 3 THEN
447 1017 4 BEGIN
448 1018 4 EDT$$FMT_LIT (NOKEYPAD_MODE, NOKEYPAD_MODE_LEN);
449 1019 4 EDT$$OUT_FMTBUF ();
450 1020 4 EDT$$G_SCR_REBUILD = 1;
451 1021 4 END;
452 1022 2
453 1023 2 EDT$$G_KPAD = 0;
454 1024 2 END;
455 1025 2
456 1026 2 [12] : ! Wrap
457 1027 2 BEGIN
458 1028 2
459 1029 4 IF (.EDT$$A_EXE_CURCMD [SET_VAL] GTRU 255)
460 1030 2 THEN
461 1031 2 EDT$$FMT_MSG (EDT$_NUMVALILL)
462 1032 2 ELSE
463 1033 2 EDT$$G_WD_WRAP = .EDT$$A_EXE_CURCMD [SET_VAL];
464 1034 2
465 1035 2 END;
466 1036 2
467 1037 2 [13] : ! Nowrap
468 1038 2 EDT$$G_WD_WRAP = 256;
469 1039 2
470 1040 2 [14] : ! Cursor
471 1041 2 BEGIN
472 1042 2
473 1043 2 !+ Set top and bottom margin, making sure neither exceeds the
474 1044 2 number of lines on the screen.
475 1045 2 -
476 1046 2
477 1047 4 IF ((.EDT$$A_EXE_CURCMD [SET_VAL] GEQU .EDT$$G_SCR_LNS) OR !
478 1048 4 (.EDT$$A_EXE_CURCMD [SET_VAL] GEQU .EDT$$G_SCR_LNS) OR !
479 1049 4 (.EDT$$A_EXE_CURCMD [SET_VAL] GTR .EDT$$A_EXE_CURCMD [SET_VAL]))
480 1050 2 THEN
481 1051 2 EDT$$FMT_MSG (EDT$_NUMVALILL)
482 1052 2 ELSE
483 1053 2 BEGIN
```

```
484 1054 4
485 1055 5
486 1056 5
487 1057 4
488 1058 5
489 1059 5
490 1060 5
491 1061 5
492 1062 4
493 1063 4
494 1064 4
495 1065 4
496 1066 4
497 1067 4
498 1068 4
499 1069 4
500 1070 4
501 1071 4
502 1072 4
503 1073 4
504 1074 4
505 1075 4
506 1076 4
507 1077 4
508 1078 4
509 1079 4
510 1080 4
511 1081 4
512 1082 4
513 1083 4
514 1084 4
515 1085 4
516 1086 4
517 1087 4
518 1088 4
519 1089 4
520 1090 4
521 1091 4
522 1092 4
523 1093 4
524 1094 4
525 1095 4
526 1096 4
527 1097 4
528 1098 4
529 1099 4
530 1100 4
531 1101 4
532 1102 4
533 1103 4
534 1104 4
535 1105 4
536 1106 4
537 1107 4
538 1108 4
539 1109 4
540 1110 2

      IF ((.EDT$$G_SCLL_TOP NEQ .EDT$$A_EXE_CURCMD [SET_VAL1]) OR
          (.EDT$$G_SCLL_BOT NEQ .EDT$$A_EXE_CURCMD [SET_VAL]))
      THEN
      BEGIN
        EDT$$G_SCLL_TOP = .EDT$$A_EXE_CURCMD [SET_VAL1];
        EDT$$G_SCLL_BOT = .EDT$$A_EXE_CURCMD [SET_VAL];
        EDT$$G_SCR_REBUILD = 1;
      END;
    END;
  END;
[15] :                               ! Screen
      BEGIN
      IF ((.EDT$$A_EXE_CURCMD [SET_VAL] GTRU 255) OR (.EDT$$A_EXE_CURCMD [SET_VAL] EQLU 0))
      THEN
        EDT$$FMT_MSG (EDT$_NUMVALILL)
      ELSE
        IF (EDT$$$SC_SETWID (.EDT$$A_EXE_CURCMD [SET_VAL])) THEN EDT$$G_SCR_CHGD = 1;
      END;
      !+ Rebuild the screen data base, since in nottruncate mode the records being displayed may
      !- occupy a different number of screen lines.
      EDT$$G_SCR_REBUILD = 1;
    END;
[16] :                               ! Mode
      EDT$$G_EDIT_DFLTMOD = .EDT$$A_EXE_CURCMD [SET_VAL] - 1;
[17] :                               ! Lines
      BEGIN
      IF (.EDT$$A_EXE_CURCMD [SET_VAL] GTRU 22)
      THEN
        EDT$$FMT_MSG (EDT$_NUMVALILL)
      ELSE
      BEGIN
        EDT$$G_SCR_LNS = .EDT$$A_EXE_CURCMD [SET_VAL];
        !+ Re-adjust the top and bottom lines if necessary.
        !-
        IF (.EDT$$G_SCLL_TOP GEQ .EDT$$G_SCR_LNS) THEN EDT$$G_SCLL_TOP = .EDT$$G_SCR_LNS - 1;
        IF (.EDT$$G_SCLL_BOT GEQ .EDT$$G_SCR_LNS) THEN EDT$$G_SCLL_BOT = .EDT$$G_SCR_LNS - 1;
        EDT$$G_SCR_REBUILD = 1;
      END;
    END;
[18] :                               ! Entity
```

```

541      1111      3      BEGIN
542      1112      3
543      1113      3      LOCAL
544      1114      3      LEN,
545      1115      3      LEN_PRV : BYTE,
546      1116      3      ENT_NUM;
547      1117      3
548      1118      3      EDT$$CNV_UPC (.EDT$$A_EXE_CURCMD [AS_STR], .EDT$$A_EXE_CURCMD [AS_LEN]);
549      1119      3      LEN = .EDT$$A_EXE_CURCMD [AS_LEN] + 1;
550      1120      3      ENT_NUM = .EDT$$A_EXE_CURCMD [SET_VAL] - 1;
551      1121      3      +
552      1122      3      Get the length of the previous entity
553      1123      3      -
554      1124      3      LEN_PRV = CH$RCHAR (.EDT$$A_US_ENT [.ENT_NUM]);
555      1125      3      +
556      1126      3      There was virtual memory allocated for entity so deallocate it
557      1127      3      -
558      1128      3      EDT$$DEA_HEAP (%REF (.LEN_PRV + 1), EDT$$A_US_ENT [.ENT_NUM]);
559      1129      3      +
560      1130      3      And allocate a new chunk no matter what
561      1131      3      -
562      1132      3
563      1133      3      IF EDT$$ALO_HEAP (LEN, EDT$$A_US_ENT [.ENT_NUM])
564      1134      3      THEN
565      1135      3      BEGIN
566      1136      3      CH$WCHAR (.LEN - 1, CH$PTR (.EDT$$A_US_ENT [.ENT_NUM]));
567      1137      3      EDT$$CPY_MEM (.LEN - 1, .EDT$$A_EXE_CURCMD [AS_STR], CH$PTR (.EDT$$A_US_ENT [.ENT_NUM], 1));
568      1138      3      END
569      1139      3      ELSE
570      1140      3      EDT$$FMT_MSG (EDT$_INSMEM);
571      1141      3
572      1142      3      END;
573      1143      3
574      1144      3      [19] : ! Quiet
575      1145      3      EDT$$G_QUIET = 1;
576      1146      3
577      1147      3      [20] : ! Noquiet
578      1148      3      EDT$$G_QUIET = 0;
579      1149      3
580      1150      3      [21] : ! Tab
581      1151      3      BEGIN
582      1152      3
583      1153      3      IF (.EDT$$A_EXE_CURCMD [SET_VAL] GTRU 255)
584      1154      3      THEN
585      1155      3      EDT$$FMT_MSG (EDT$_NUMVALILL)
586      1156      3      ELSE
587      1157      3      BEGIN
588      1158      3      EDT$$G_TAB_SIZ = .EDT$$A_EXE_CURCMD [SET_VAL];
589      1159      3      EDT$$G_TAB_LVL = 1;
590      1160      3      END;
591      1161      3
592      1162      3      END;
593      1163      3
594      1164      3      [22] : ! Notab
595      1165      3      EDT$$G_TAB_SIZ = 0;
596      1166      3
597      1167      3      [23] : ! Allow repeat counts
```

EDT\$SET
V04-000

EDT\$SET - SET line-mode command
EDT\$\$SET_CMD - SET line-mode command

L 9
16-Sep-1984 00:57:38
14-Sep-1984 12:23:40

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

Page 12
(3)

**F

```

598      1168      2      EDT$$G_RPT = 1;
599      1169      2
600      1170      2      [24] :      ! Don't allow repeat counts
601      1171      2      EDT$$G_RPT = 0;
602      1172      2
603      1173      2      [25] :      ! Allow file_not_found_msg
604      1174      2      EDT$$G_FNF_MSGFLG = 1;
605      1175      2
606      1176      2      [26] :      ! Don't allow file_not_found_msg
607      1177      2      EDT$$G_FNF_MSGFLG = 0;
608      1178      2
609      1179      2      [27] :      ! summary
610      1180      2      BEGIN
611      1181      2
612      L 1182      2      %IF SUPPORT_WPS
613      1183      2      %THEN
614      1184      2      EDT$$G_SUMRY = 1;      ! Type out summary when exiting
615      U 1185      2      %ELSE
616      U 1186      2      0
617      1187      2      %FI
618      1188      2
619      1189      2      END;
620      1190      2
621      1191      2      [28] :      ! nosummary
622      1192      2      BEGIN
623      1193      2
624      L 1194      2      %IF SUPPORT_WPS
625      1195      2      %THEN
626      1196      2      EDT$$G_SUMRY = 0;      ! suppress summary when exiting
627      U 1197      2      %ELSE
628      U 1198      2      0
629      1199      2      %FI
630      1200      2
631      1201      2      END;
632      1202      2
633      1203      2      [29] :      ! Set prompt
634      1204      2      BEGIN
635      1205      2
636      1206      2      LOCAL
637      1207      2      I,      ! Index into prompt
638      1208      2      CP,      ! Character pointer of string
639      1209      2      LEN,      ! Length of the prompt string
640      1210      2      PROMPT_NUM,      ! Number corresponding to which prompt
641      1211      2      PROMPT_ADDR : REF VECTOR [32, BYTE];      ! Address of prompt string
642      1212      2
643      1213      2      LEN = .EDT$$A_EXE_CURCMD [AS LEN];
644      1214      2      PROMPT_NUM = .EDT$$A_EXE_CURCMD [SET_VAL];
645      1215      2
646      1216      2      IF (.LEN GTR 31)
647      1217      2      THEN
648      1218      2      EDT$$FMT_MSG (EDT$_INVSTR)
649      1219      2      ELSE
650      1220      2      BEGIN
651      1221      2      PROMPT_ADDR = (CASE .PROMPT_NUM FROM 1 TO 7 OF
652      1222      2      SET
653      1223      2      [1] : EDT$$T_PMT_LINE;
654      1224      2      [2] : EDT$$T_PMT_KPD;
```

```

: 655      1225 5      [3] : EDT$$T_PMT_NOKPD;
: 656      1226 5      [4] : EDT$$T_PMT_HCCHG;
: 657      1227 5      [5] : EDT$$T_PMT_INS;
: 658      1228 5      [6] : EDT$$T_PMT_INSN;
: 659      1229 5      [7] : EDT$$T_PMT_QUERY;
: 660      1230 5      [OUTRANGE] :
: 661      1231 6      BEGIN
: 662      1232 6      ASSERT (0);
: 663      1233 6      0
: 664      1234 5      END;
: 665      1235 4      TES);
: 666      1236 4
: 667      1237 4      !+ Now copy the specified string into the global prompt string.
: 668      1238 4      !- If a CR,LF does not exist where required then force one into the string.
: 669      1239 4
: 670      1240 4      CP = CH$PTR (.EDT$$A_EXE_CURCMD [AS_STR]);
: 671      1241 4      I = 1;
: 672      1242 4
: 673      1243 5      IF (.PROMPT_NUM NEQ 2) AND (.PROMPT_NUM NEQ 3)
: 674      1244 4      THEN
: 675      1245 4
: 676      1246 5      IF (CH$RCHAR (.CP) NEQ 13) OR (CH$RCHAR (.CP + 1) NEQ 10)
: 677      1247 4      THEN
: 678      1248 5      BEGIN
: 679      1249 5      CH$MOVE (2, UPLIT BYTE(13, 10), PROMPT_ADDR [.I]);
: 680      1250 5      I = 3;
: 681      1251 5      LEN = .LEN + 2;
: 682      1252 4      END;
: 683      1253 4
: 684      1254 4      PROMPT_ADDR [0] = .LEN;
: 685      1255 4      CH$MOVE (.LEN, .CP, PROMPT_ADDR [.I]);
: 686      1256 4      END
: 687      1257 4
: 688      1258 2      END;
: 689      1259 2
: 690      1260 2      [30] : SET TEXT
: 691      1261 3      BEGIN
: 692      1262 3
: 693      1263 3      LOCAL
: 694      1264 3      LEN,
: 695      1265 3      LEN_PTV : BYTE,
: 696      1266 3      CHAR_PTR,
: 697      1267 3      ESTATUS,
: 698      1268 3      CHAR,
: 699      1269 3      TEXT_NUM;
: 700      1270 3
: 701      1271 3      LEN = .EDT$$A_EXE_CURCMD [AS_LEN] + 1;
: 702      1272 3      TEXT_NUM = .EDT$$A_EXE_CURCMD [SET_VAL] - 1;
: 703      1273 3
: 704      1274 3      !+ Don't let the PAGE string be longer than 26 characters, otherwise we won't be
: 705      1275 3      !- able to display a record containing 255 form feeds on an 80-column screen in
: 706      1276 3      !- nottruncate mode. If the PAGE string could be 27 characters long then one
: 707      1277 3      !- record would require more than 110 lines to display, which is all that the
: 708      1278 3      !- screen data base can hold. The screen updater assumes that the screen data
: 709      1279 3      !- base can hold one record, even if not all of that record is required to fill
: 710      1280 3      !- the screen.
: 711      1281 3      !- Note that LEN is the length plus 1, since it includes the length byte.
```

```

712 1282 3  !-
713 1283 3
714 1284 3      IF (.TEXT_NUM NEQ 0) THEN LEN = MIN (.LEN, 27);
715 1285 3
716 1286 3      ESTATUS = 1;
717 1287 3
718 1288 3  !+ Make sure that there are no escape or control chars in the string
719 1289 3  !-
720 1290 3      CHAR_PTR = CH$PTR (.EDT$$A_EXE_CURCMD [AS_STR]);
721 1291 3
722 1292 3      INCR I FROM 1 TO .LEN - 1 DO
723 1293 4          BEGIN
724 1294 4              CHAR = CH$RCHAR_A (CHAR_PTR);
725 1295 4
726 1296 5              IF ((.CHAR GEQ 127) OR (.CHAR LSS 32))
727 1297 4                  THEN
728 1298 5                      BEGIN
729 1299 5                          EDT$$FMT_MSG (EDT$_INVSTR); ! output error msg.
730 1300 5                          ESTATUS = 0;
731 1301 5                          EXITLOOP; ! exit--no reason to check rest
732 1302 4                          END;
733 1303 4
734 1304 3          END;
735 1305 3
736 1306 3  !+ Get the length of the previous string
737 1307 3  !-
738 1308 3
739 1309 3      IF (.ESTATUS NEQ 0)
740 1310 4          THEN
741 1311 3              BEGIN
742 1312 4                  LEN_PRV = CH$RCHAR (.EDT$$A_US_TXT [.TEXT_NUM]);
743 1313 4
744 1314 4  !+ There was virtual memory allocated for it so deallocate it
745 1315 4  !-
746 1316 4                  EDT$$DEA_HEAP (%REF (.LEN_PRV + 1), EDT$$A_US_TXT [.TEXT_NUM]);
747 1317 4
748 1318 4  !+ And allocate a new chunk
749 1319 4  !-
750 1320 4
751 1321 4      IF EDT$$ALO_HEAP (LEN, EDT$$A_US_TXT [.TEXT_NUM])
752 1322 4          THEN
753 1323 4          BEGIN
754 1324 5              CH$WCHAR (.LEN - 1, CH$PTR (.EDT$$A_US_TXT [.TEXT_NUM]));
755 1325 5              EDT$$CPY_MEM (.LEN - 1, .EDT$$A_EXE_CURCMD [AS_STR], !
756 1326 5                  CH$PTR (.EDT$$A_US_TXT [.TEXT_NUM], 1));
757 1327 5              EDT$$G_SCR_REBUILD = 1;
758 1328 5          END
759 1329 5      ELSE
760 1330 4          EDT$$FMT_MSG (EDT$_INSMEM);
761 1331 4
762 1332 4      END;
763 1333 3
764 1334 3      END;
765 1335 2
766 1336 2      [31] :
767 1337 2      BEGIN
768 1338 3
```

```
: 769      1339 3
: 770      L 1340 3 %IF SUPPORT_WPS
: 771      1341 3 %THEN
: 772      1342 3 EDT$$G_WRDTP = .EDT$$A_EXE_CURCMD [SET_VAL] - 1;
: 773      U 1343 3 %ELSE
: 774      U 1344 3 0
: 775      1345 3 %FI
: 776      1346 3
: 777      1347 3 END;
: 778      1348 3
: 779      1349 3 [32] :
: 780      1350 3 BEGIN
: 781      1351 3
: 782      L 1352 3 %IF SUPPORT_WPS
: 783      1353 3 %THEN
: 784      1354 3 EDT$$G_PARTYP = .EDT$$A_EXE_CURCMD [SET_VAL] - 1;
: 785      U 1355 3 %ELSE
: 786      U 1356 3 0
: 787      1357 3 %FI
: 788      1358 3
: 789      1359 3 END;
: 790      1360 3
: 791      1361 3 [33] : ! Set Help File Name
: 792      1362 3 EDT$$SET_HLPFNAM (.EDT$$A_EXE_CURCMD [FILSPEC], .EDT$$A_EXE_CURCMD [FSPCLEN]);
: 793      1363 3
: 794      1364 3 [34] : ! Set Autorepeat
: 795      1365 3 EDT$$G_ENB_AUTRPT = 1;
: 796      1366 3
: 797      1367 3 [35] : ! Set Noautorepeat
: 798      1368 3 EDT$$G_ENB_AUTRPT = 0;
: 799      1369 3
: 800      1370 3 [36] : ! Set Command
: 801      1371 3 BEGIN
: 802      1372 3
: 803      1373 4 IF (.EDT$$A_EXE_CURCMD [FSPCLEN] EQL 0)
: 804      1374 3 THEN
: 805      1375 4 BEGIN
: 806      1376 4 EDT$$FMT_MSG (EDT$_INVSTR);
: 807      1377 4 END
: 808      1378 3 ELSE
: 809      1379 4 BEGIN
: 810      1380 4 EDT$$SET COMFNAM (
: 811      1381 4 .EDT$$A_EXE_CURCMD [FILSPEC], ! Name of file
: 812      1382 4 .EDT$$A_EXE_CURCMD [FSPCLEN], ! Length of name
: 813      1383 4 CH$PTR (EDT$$T_CMD_NAM_DEF4, 1), ! Default name of file
: 814      1384 4 CH$RCHAR (EDT$$T_CMD_NAM_DEF4) ! Length of default name of file
: 815      1385 4 );
: 816      1386 3 END;
: 817      1387 3
: 818      1388 3 END;
: 819      1389 3
: 820      1390 3 [OUTRANGE] :
: 821      1391 3 ASSERT (0);
: 822      1392 3 TES;
: 823      1393 3
: 824      1394 1 END; ! of routine EDT$$$SET_CMD
```

```
.TITLE EDT$LSET EDT$LSET - SET line-mode command
.IDENT \V04-000\

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

00 00 3D 1B 00000 P.AAA: .ASCII <27>\=\<0><0>
00 00 3E 1B 00004 P.AAB: .ASCII <27>\>\<0><0>
0A 0D 00008 P.AAC: .BYTE 13, 10

KEYPAD_MODE= P.AAA
NOKEYPAD_MODE= P.AAB

.EXTRN EDT$$SC_SETWID, EDT$$FMT_LIT
.EXTRN EDT$$OUT_FMTBUF
.EXTRN EDT$$FMT_MSG, EDT$$SET_HLPFNAM
.EXTRN EDT$$SET_COMFNAM
.EXTRN EDT$$ALO_HEAP, EDT$$DEA_HEAP
.EXTRN EDT$$CNV_UPC, EDT$$G_EXT_MOD
.EXTRN EDT$$G_CAS_FLG, EDT$$G_EDIT_DFLTMOD
.EXTRN EDT$$G_FNF_MSGFLG
.EXTRN EDT$$A_US_ENT, EDT$$G_RPT
.EXTRN EDT$$A_US_TXT, EDT$$G_EXCT_MATCH
.EXTRN EDT$$G_KPAD, EDT$$G_NOS
.EXTRN EDT$$G_SEA_BEG, EDT$$G_SEA_BNDD
.EXTRN EDT$$G_SCR_CHGD
.EXTRN EDT$$G_SCR_REBUILD
.EXTRN EDT$$G_SCR_LNS, EDT$$G_SCLL_BOT
.EXTRN EDT$$G_SCLL_TOP
.EXTRN EDT$$G_QUIET, EDT$$G_TAB_SIZ
.EXTRN EDT$$G_TAB_LVL, EDT$$G_TRUN
.EXTRN EDT$$G_TI_TYP, EDT$$G_DEC_CRT
.EXTRN EDT$$G_TI_WID, EDT$$G_VFY
.EXTRN EDT$$G_WD_WRAP, EDT$$G_WRD_TYP
.EXTRN EDT$$G_PARTYP, EDT$$G_SUMRY
.EXTRN EDT$$G_ENB_AUTRPT
.EXTRN EDT$$G_TI_SCROLL
.EXTRN EDT$$G_EIGHT_BIT
.EXTRN EDT$$G_TI_EDIT, EDT$$A_EXE_CURCMD
.EXTRN EDT$$T_PMT_LINE
.EXTRN EDT$$T_PMT_KPD, EDT$$T_PMT_NOKPD
.EXTRN EDT$$T_PMT_HCCHG
.EXTRN EDT$$T_PMT_INS, EDT$$T_PMT_INSN
.EXTRN EDT$$T_PMT_QUERY
.EXTRN EDT$$T_CMD_NAM_DEF4
.EXTRN EDT$_INSMEM, EDT$_INVSTR
.EXTRN EDT$_NOSETTRM, EDT$_NUMVALILL
.EXTRN EDT$$INTER_ERR

.OFFC 00000

5B 00000000G 00 9E 00002 MOVAB EDT$$G_TRUN, R11
5A 00000000G 00 9E 00009 MOVAB EDT$$G_DEC_CRT, R10
59 00000000G 00 9E 00010 MOVAB EDT$$A_EXE_CURCMD, R9
58 00000000G 00 9E 00017 MOVAB EDT$$G_TI_TYP, R8
57 00000000G 00 9E 0001E MOVAB EDT$$G_EXCT_MATCH, R7
5E 0C C2 00025 SUBL2 #12, SP
52 69 D0 00028 MOVL EDT$$A_EXE_CURCMD, R2
```

EDT\$LSET
V04-000

EDT\$LSET - SET line-mode command
EDT\$\$\$SET_CMD - SET line-mode command

D 10
16-Sep-1984 00:57:38
14-Sep-1984 12:23:40

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

Page 17
(3)

0063	23	01	04	A2	CF	0002B	CASEL	4(R2), #1, #35	:
0144	0059	0052		004A		00030	.WORD	2\$-1\$,-	:
01C5	013D	0135		00AD		00038		3\$-1\$,-	:
0257	018F	0158		014E		00040		4\$-1\$,-	:
0311	0223	01E2		01D8		00048		5\$-1\$,-	:
0344	0309	02A4		0261		00050		15\$-1\$,-	:
0362	033C	0335		0318		00058		28\$-1\$,-	:
04B4	035A	0353		034B		00060		29\$-1\$,-	:
04D9	04AA	0409		0369		00068		30\$-1\$,-	:
	04D2	04CA		04BE		00070		31\$-1\$,-	:
								35\$-1\$,-	:
								38\$-1\$,-	:
								41\$-1\$,-	:
								42\$-1\$,-	:
								43\$-1\$,-	:
								46\$-1\$,-	:
								49\$-1\$,-	:
								50\$-1\$,-	:
								55\$-1\$,-	:
								57\$-1\$,-	:
								58\$-1\$,-	:
								59\$-1\$,-	:
								61\$-1\$,-	:
								62\$-1\$,-	:
								63\$-1\$,-	:
								64\$-1\$,-	:
								65\$-1\$,-	:
								66\$-1\$,-	:
								67\$-1\$,-	:
								68\$-1\$,-	:
								81\$-1\$,-	:
								91\$-1\$,-	:
								92\$-1\$,-	:
								93\$-1\$,-	:
								94\$-1\$,-	:
								95\$-1\$,-	:
								96\$-1\$	:
								7\$	1391
	00000000G	00		34	11	00078	BRB		0858
				01	D0	0007A	MOVL	#1, EDT\$\$G_NOS	
				04	04	00081	RET		
				00	D4	00082	CLRL	EDT\$\$G_NOS	0861
				04	04	00088	RET		
	00000000G	00	10	A2	01	C3	SUBL3	#1, 16(R2), EDT\$\$G_CAS_FLG	0864
					04	00092	RET		
					CF	00093	CASEL	16(R2), #1, #10	0868
0025	0A	01	10	A2	001E	00098	.WORD	8\$-6\$,-	
003D	0025	0021		002F	002F	000A0		9\$-6\$,-	
	0039	002F		0041	000A8			10\$-6\$,-	
	0041	003D						10\$-6\$,-	
								11\$-6\$,-	
								11\$-6\$,-	
								12\$-6\$,-	
								13\$-6\$,-	
								14\$-6\$,-	
								13\$-6\$,-	
								14\$-6\$	
	00000000G	00		00	FB	000AE	CALLS	#0, EDT\$\$INTER_ERR	0893

EDT\$LSET
V04-000

EDT\$SET - SET line-mode command
EDT\$\$SET_CMD - SET line-mode command

E 10
16-Sep-1984 00:57:38
14-Sep-1984 12:23:40

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

Page 18
(3)

				67	04	000B5	RET		0868
					D4	000B6	8\$: CLRL	EDT\$\$G_EXCT_MATCH	0872
					04	000B8	RET		
		67		01	D0	000B9	9\$: MOVL	#1, EDT\$\$G_EXCT_MATCH	0875
					04	000BC	RET		
00000000G	00	10	A2	03	C3	000BD	10\$: SUBL3	#3, 16(R2), EDT\$\$G_SEA_BEG	0878
					04	000C6	RET		
00000000G	00	10	A2	05	C3	000C7	11\$: SUBL3	#5, 16(R2), EDT\$\$G_SEA_BNDD	0881
					04	000D0	RET		
		67		02	D0	000D1	12\$: MOVL	#2, EDT\$\$G_EXCT_MATCH	0884
					04	000D4	RET		
		67		03	D0	000D5	13\$: MOVL	#3, EDT\$\$G_EXCT_MATCH	0887
					04	000D8	RET		
		67		04	D0	000D9	14\$: MOVL	#4, EDT\$\$G_EXCT_MATCH	0890
					04	000DC	RET		0868
		0F	00000000G	00	E9	000DD	15\$: BLBC	EDT\$\$G_EXT_MOD, 16\$	0899
		04	10	A2	D1	000E4	CMPL	16(R2), #4	
				09	18	000E8	BGEQ	16\$	
			00000000G	8F	DD	000EA	PUSHL	#EDT\$_NOSETTRM	0901
				04	21	31	000F0	BRW	98\$
		08		01	A2	CF	000F3	16\$: CASEL	16(R2), #1, #8
0031		002A		001B			000F8	17\$: .WORD	18\$-17\$,-
0056		004E		003A			00100		19\$-17\$,-
				005F			00108		20\$-17\$,-
									21\$-17\$,-
									22\$-17\$,-
									23\$-17\$,-
									24\$-17\$,-
									25\$-17\$,-
									26\$-17\$
			00000000G	00	00	FB	0010A	CALLS	#0, EDT\$\$INTER_ERR
					4A	11	00111	BRB	27\$
					6A	D4	00113	18\$: CLRL	EDT\$\$G_DEC_CRT
			68		01	D0	00115	MOVL	#1, EDT\$\$G_TI_TYP
					18	11	00118	BRB	22\$
			6A		01	D0	0011A	19\$: MOVL	#1, EDT\$\$G_DEC_CRT
			68		02	D0	0011D	MOVL	#2, EDT\$\$G_TI_TYP
					07	11	00120	BRB	21\$
					6A	D4	00122	20\$: CLRL	EDT\$\$G_DEC_CRT
			68		03	D0	00124	MOVL	#3, EDT\$\$G_TI_TYP
					34	11	00127	BRB	27\$
			00000000G	00	01	D0	00129	21\$: MOVL	#1, EDT\$\$G_TI_SCROLL
					2B	11	00130	BRB	27\$
					00	D4	00132	22\$: CLRL	EDT\$\$G_TI_SCROLL
					23	11	00138	BRB	27\$
			6A		02	D0	0013A	23\$: MOVL	#2, EDT\$\$G_DEC_CRT
			00000000G	00	01	D0	0013D	MOVL	#1, EDT\$\$G_EIGHT_BIT
					17	11	00144	BRB	27\$
					00	D4	00146	24\$: CLRL	EDT\$\$G_EIGHT_BIT
					0F	11	0014C	BRB	27\$
			00000000G	00	01	D0	0014E	25\$: MOVL	#1, EDT\$\$G_TI_EDIT
					06	11	00155	BRB	27\$
					00	D4	00157	26\$: CLRL	EDT\$\$G_TI_EDIT
			00000000G	00	01	D0	0015D	27\$: MOVL	#1, EDT\$\$G_SCR_CHGD
						04	00164	RET	
			00000000G	00	01	D0	00165	28\$: MOVL	#1, EDT\$\$G_VFY
					04	0016C	RET		

00000000G	00	D4	0016D	29\$:	CLRL	EDT\$\$G_VFY	0978
		04	00173		RET		
01	6B	D1	00174	30\$:	CMPL	EDT\$\$G_TRUN, #1	0982
	07	13	00177		BEQL	32\$	
6B	01	D0	00179		MOVL	#1, EDT\$\$G_TRUN	0985
	07	11	0017C		BRB	34\$	0986
	6B	D5	0017E	31\$:	TSTL	EDT\$\$G_TRUN	0991
	01	12	00180	32\$:	BNEQ	33\$	
		04	00182		RET		
	6B	D4	00183	33\$:	CLRL	EDT\$\$G_TRUN	0994
	034A	31	00185	34\$:	BRW	90\$	0995
50	68	D0	00188	35\$:	MOVL	EDT\$\$G_TI_TYP, R0	1001
01	50	D1	0018B		CMPL	R0, #1	
	05	13	0018E		BEQL	36\$	
02	50	D1	00190		CMPL	R0, #2	
	22	12	00193		BNEQ	37\$	
1B	00000000G	00	E9	00195	36\$:	BLBC	EDT\$\$G_EXT_MOD, 37\$
		02	DD	0019C		PUSHL	#2
	FE54	CF	9F	0019E		PUSHAB	KEYPAD MODE
00000000G	00	02	FB	001A2		CALLS	#2, EDT\$\$FMT_LIT
00000000G	00	00	FB	001A9		CALLS	#0, EDT\$\$OUT_FMTBUF
00000000G	00	01	D0	001B0		MOVL	#1, EDT\$\$G_SCR_REBUILD
00000000G	00	01	D0	001B7	37\$:	MOVL	#1, EDT\$\$G_KPAD
		04	001BE		RET		
50	68	D0	001BF	38\$:	MOVL	EDT\$\$G_TI_TYP, R0	
01	50	D1	001C2		CMPL	R0, #1	
	05	13	001C5		BEQL	39\$	
02	50	D1	001C7		CMPL	R0, #2	
	22	12	001CA		BNEQ	40\$	
1B	00000000G	00	E9	001CC	39\$:	BLBC	EDT\$\$G_EXT_MOD, 40\$
		02	DD	001D3		PUSHL	#2
	FE21	CF	9F	001D5		PUSHAB	NOKEYPAD MODE
00000000G	00	02	FB	001D9		CALLS	#2, EDT\$\$FMT_LIT
00000000G	00	00	FB	001E0		CALLS	#0, EDT\$\$OUT_FMTBUF
00000000G	00	01	D0	001E7		MOVL	#1, EDT\$\$G_SCR_REBUILD
	00000000G	00	D4	001EE	40\$:	CLRL	EDT\$\$G_KPAD
		04	001F4		RET		
000000FF	8F	10	A2	D1	001F5	41\$:	CMPL
		24	1A	001FD		BGTRU	16(R2), #255
00000000G	00	10	A2	D0	001FF		44\$
		04	00207		MOVL	16(R2), EDT\$\$G_WD_WRAP	
00000000G	00	0100	8F	3C	00208	42\$:	RET
		04	00211		MOVZWL	#256, EDT\$\$G_WD_WRAP	
	50	00000000G	00	D0	00212	43\$:	RET
	50	0C	A2	D1	00219		MOVL
		78	1E	0021D		CMPL	EDT\$\$G_SCR_LNS, R0
	50	10	A2	D1	0021F		12(R2), R0
		72	1E	00223	44\$:	BGEQU	51\$
10	A2	0C	A2	D1	00225		CMPL
		6B	14	0022A		BGTR	16(R2), R0
0C	A2	00000000G	00	D1	0022C		CMPL
		0B	12	00234		BNEQ	12(R2), 16(R2)
10	A2	00000000G	00	D1	00236		CMPL
		01	12	0023E		BNEQ	EDT\$\$G_SCLL_TOP, 12(R2)
		04	00240		RET	45\$	
00000000G	00	0C	A2	D0	00241	45\$:	CMPL
00000000G	00	10	A2	D0	00249		EDT\$\$G_SCLL_BOT, 16(R2)
					MOVL	12(R2), EDT\$\$G_SCLL_TOP	
					MOVL	16(R2), EDT\$\$G_SCLL_BOT	

EDT\$LSET
V04-000

EDT\$SET - SET line-mode command
EDT\$\$\$SET_CMD - SET line-mode command

G 10
16-Sep-1984 00:57:38
14-Sep-1984 12:23:40

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

Page 20
(3)

000000FF	8F	10	7E	11	00251	BRB	54\$	1061
			A2	D1	00253	CMPL	16(R2), #255	1071
		10	05	1A	0025B	BGTRU	47\$	
			A2	D5	0025D	TSTL	16(R2)	
			0F	12	00260	BNEQ	48\$	
00000000G	00	00000000G	8F	DD	00262	PUSHL	#EDT\$ NUMVALILL	1073
			01	FB	00268	CALLS	#1, EDT\$FMT_MSG	
		10	60	11	0026F	BRB	54\$	
00000000G	00		A2	DD	00271	PUSHL	16(R2)	1076
	53		01	FB	00274	CALLS	#1, EDT\$\$\$SC_SETWID	
00000000G	00		50	E9	0027B	BLBC	R0, 54\$	
			01	D0	0027E	MOVL	#1, EDT\$\$G_SCR_CHGD	
			4A	11	00285	BRB	54\$	1082
00000000G	00	10	A2	C3	00287	SUBL3	#1, 16(R2), EDT\$\$G_EDIT_DFLTMOD	1086
		16		04	00290	RET		
			A2	D1	00291	CMPL	16(R2), #22	1091
			09	1B	00295	BLEQU	52\$	
		00000000G	8F	DD	00297	PUSHL	#EDT\$ NUMVALILL	1093
			0274	31	0029D	BRW	98\$	
00000000G	00	10	A2	D0	002A0	MOVL	16(R2), EDT\$\$G_SCR_LNS	1096
	50	00000000G	00	D0	002A8	MOVL	EDT\$\$G_SCR_LNS, R0	1101
	50	00000000G	00	D1	002AF	CMPL	EDT\$\$G_SCLC_TOP, R0	
			08	19	002B6	BLSS	53\$	
00000000G	00	FF	A0	9E	002B8	MOVAB	-1(R0), EDT\$\$G_SCLL_TOP	
	50	00000000G	00	D1	002C0	CMPL	EDT\$\$G_SCLL_BOT, R0	1103
			08	19	002C7	BLSS	54\$	
00000000G	00	FF	A0	9E	002C9	MOVAB	-1(R0), EDT\$\$G_SCLL_BOT	
			01FE	31	002D1	BRW	90\$	1105
	7E	08	A2	7D	002D4	MOVQ	8(R2), -(SP)	1118
00000000G	00		02	FB	002D8	CALLS	#2, EDT\$CNV_UPC	
	50		69	D0	002DF	MOVL	EDT\$A_EXE_CURCMD, R0	1119
04	AE	0C	A0	01	C1	ADDL3	#1, 12(R0), LEN	
	50	10	A0	01	C3	SUBL3	#1, 16(R0), ENT_NUM	1120
			53	00000000G	0040	MOVAL	EDT\$A_US_ENT[ENT_NUM], R3	1124
			51	00	B3	MOVB	@0(R3), LEN_PRV	
				53	DD	PUSHL	R3	1128
	04	AE	51	9A	002FB	MOVZBL	LEN_PRV, 4(SP)	
			04	AE	D6	INCL	4(SP)	
			04	AE	9F	PUSHAB	4(SP)	
00000000G	00		02	FB	00305	CALLS	#2, EDT\$DEA_HEAP	
			53	DD	0030C	PUSHL	R3	1133
			08	AE	9F	PUSHAB	LEN	
00000000G	00		02	FB	00311	CALLS	#2, EDT\$ALO_HEAP	
	15		50	E9	00318	BLBC	R0, 56\$	
	51		63	D0	0031B	MOVL	(R3), R1	1136
	52	04	AE	01	C3	SUBL3	#1, LEN, R2	
			61	52	90	MOVB	R2, (R1)	
			50	69	D0	MOVL	EDT\$A_EXE_CURCMD, R0	1137
01	A1	08	B0	52	28	MOVC3	R2, @8(R0), 1(R1)	
				04	0032F	RET		1133
		00000000G	8F	DD	00330	PUSHL	#EDT\$ INSMEM	1140
			01DB	31	00336	BRW	98\$	
00000000G	00		01	D0	00339	MOVL	#1, EDT\$\$G_QUIET	1145
				04	00340	RET		
		00000000G	00	D4	00341	CLRL	EDT\$\$G_QUIET	1148
				04	00347	RET		
000000FF	8F	10	A2	D1	00348	CMPL	16(R2), #255	1153

EDT
V04

EDT\$LSET
V04-000

EDT\$SET - SET line-mode command
EDT\$\$SET_CMD - SET line-mode command

H 10
16-Sep-1984 00:57:38
14-Sep-1984 12:23:40

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

Page 21
(3)

00000000G	00	10	FF42	03 1B 00350	BLEQU	60\$	1158
00000000G	00		A2	31 00352	BRW	51\$	1159
			01	D0 00355	MOVL	16(R2), EDT\$\$G_TAB_SIZ	0854
				D0 0035D	MOVL	#1, EDT\$\$G_TAB_LVL	1165
			04	00364	RET		
00000000G	00		00	D4 00365	CLRL	EDT\$\$G_TAB_SIZ	1168
				04 0036B	RET		
00000000G	00		01	D0 0036C	MOVL	#1, EDT\$\$G_RPT	1171
				04 00373	RET		
			00	D4 00374	CLRL	EDT\$\$G_RPT	1174
				04 0037A	RET		
00000000G	00		01	D0 0037B	MOVL	#1, EDT\$\$G_FNF_MSGFLG	1177
				04 00382	RET		
			00	D4 00383	CLRL	EDT\$\$G_FNF_MSGFLG	1184
				04 00389	RET		0854
00000000G	00		01	D0 0038A	MOVL	#1, EDT\$\$G_SUMRY	1196
				04 00391	RET		0854
			00	D4 00392	CLRL	EDT\$\$G_SUMRY	1213
				04 00398	RET		1214
	54	0C	A2	D0 00399	MOVL	12(R2), LEN	1216
	53	10	A2	D0 0039D	MOVL	16(R2), PROMPT_NUM	
	1F		54	D1 003A1	CMPL	LEN, #31	
			03	15 003A4	BLEQ	69\$	
			0165	31 003A6	BRW	97\$	
0034	06	01	53	CF 003A9	CASEL	PROMPT_NUM, #1, #6	1221
	002B	0022	0019	003AD	.WORD	71\$-70\$,-	
	004F	0046	003D	003B5		72\$-70\$,-	
						73\$-70\$,-	
						74\$-70\$,-	
						75\$-70\$,-	
						76\$-70\$,-	
						77\$-70\$	
00000000G	00		00	FB 003BB	CALLS	#0, EDT\$\$INTER_ERR	1232
			51	D4 003C2	CLRL	PROMPT_ADDR	1231
			3D	11 003C4	BRB	78\$	
	51	00000000G	00	9E 003C6	MOVAB	EDT\$\$T_PMT_LINE, PROMPT_ADDR	1221
			34	11 003CD	BRB	78\$	
	51	00000000G	00	9E 003CF	MOVAB	EDT\$\$T_PMT_KPD, PROMPT_ADDR	
			2B	11 003D6	BRB	78\$	
	51	00000000G	00	9E 003D8	MOVAB	EDT\$\$T_PMT_NOKPD, PROMPT_ADDR	
			22	11 003DF	BRB	78\$	
	51	00000000G	00	9E 003E1	MOVAB	EDT\$\$T_PMT_HCCHG, PROMPT_ADDR	
			19	11 003E8	BRB	78\$	
	51	00000000G	00	9E 003EA	MOVAB	EDT\$\$T_PMT_INS, PROMPT_ADDR	
			10	11 003F1	BRB	78\$	
	51	00000000G	00	9E 003F3	MOVAB	EDT\$\$T_PMT_INSN, PROMPT_ADDR	
			07	11 003FA	BRB	78\$	
	51	00000000G	00	9E 003FC	MOVAB	EDT\$\$T_PMT_QUERY, PROMPT_ADDR	
	50		69	D0 00403	MOVL	EDT\$\$A_EXE_CURCMD, R0	1240
	50	08	A0	D0 00406	MOVL	8(R0), CP	
	52		01	D0 0040A	MOVL	#1, I	1241
	02		53	D1 0040D	CMPL	PROMPT_NUM, #2	1243
			1E	13 00410	BEQL	80\$	
	03		53	D1 00412	CMPL	PROMPT_NUM, #3	
			19	13 00415	BEQL	80\$	
	0D		60	91 00417	CMPB	(CP), #13	1246
			06	12 0041A	BNEQ	79\$	

0A	01	A0	91	0041C	CMPB	1(CP), #10	
		0E	13	00420	BEQL	80\$	
		6241	9F	00422	PUSHAB	(I)[PROMPT_ADDR]	1249
		9E	CF	00425	MOVW	P.AAC, @ (SP)+	
		52	03	0042A	MOVL	#3, I	1250
		54	02	0042D	ADDL2	#2, LEN	1251
		61	54	00430	MOVB	LEN, (PROMPT_ADDR)	1254
6241		60	54	00433	MOVC3	LEN, (CP), (I)[PROMPT_ADDR]	1255
			04	00438	RET		1216
08	AE	OC	01	C1 00439	ADDL3	#1, 12(R2), LEN	1271
53	10	A2	01	C3 0043F	SUBL3	#1, 16(R2), TEXT_NUM	1272
			10	13 00444	BEQL	83\$	1284
		50	AE	D0 00446	MOVL	LEN, R0	
		1B	50	D1 0044A	CMPL	R0, #27	
			03	15 0044D	BLEQ	82\$	
		50	1B	D0 0044F	MOVL	#27, R0	
	08	AE	50	D0 00452	MOVL	R0, LEN	
		55	01	D0 00456	MOVL	#1, ESTATUS	1286
		56	08	A2 D0 00459	MOVL	8(R2), CHAR_PTR	1290
			52	D4 0045D	CLRL	I	1292
			22	11 0045F	BRB	86\$	
		54	86	9A 00461	MOVZBL	(CHAR_PTR)+, CHAR	1294
0000007F		8F	54	D1 00464	CMPL	CHAR, #127	1296
		20	05	18 0046B	BGEQ	85\$	
			54	D1 0046D	CMPL	CHAR, #32	
			11	18 00470	BGEQ	86\$	
		00000000G	8F	DD 00472	PUSHL	#EDT\$ INVSTR	1299
00000000G	00		01	FB 00478	CALLS	#1, EDT\$\$FMT_MSG	
			55	D4 0047F	CLRL	ESTATUS	1300
			05	11 00481	BRB	87\$	1298
D9		52	08	AE F2 00483	AOBLSS	LEN, I, 84\$	1292
			55	D5 00488	TSTL	ESTATUS	1310
			01	12 0048A	BNEQ	88\$	
			04	0048C	RET		
		52	00000000G00	43 DE 0048D	MOVAL	EDT\$\$A_US TXT[TEXT_NUM], R2	1313
		50	00	B2 90 00495	MOVB	@0(R2), LEN_PRV	
			52	DD 00499	PUSHL	R2	1317
	04	AE	50	9A 0049B	MOVZBL	LEN PRV, 4(SP)	
			04	AE D6 0049F	INCL	4(SP)	
			04	AE 9F 004A2	PUSHAB	4(SP)	
00000000G	00		02	FB 004A5	CALLS	#2, EDT\$\$DEA_HEAP	
			52	DD 004AC	PUSHL	R2	1322
			0C	AE 9F 004AE	PUSHAB	LEN	
00000000G	00		02	FB 004B1	CALLS	#2, EDT\$\$ALO_HEAP	
		03	50	E8 004B8	BLBS	R0, 89\$	
			FE72	31 004BB	BRW	56\$	
		51	62	D0 004BE	MOVL	(R2), R1	1325
	52	08	AE	01 C3 004C1	SUBL3	#1, LEN, R2	
			61	52 90 004C6	MOVB	R2, (R1)	
		50	69	D0 004C9	MOVL	EDT\$\$A_EXE_CURCMD, R0	1327
01	A1	08	52	28 004CC	MOVC3	R2, @8(R0), 1(R1)	
		00000000G	00	01 D0 004D2	MOVL	#1, EDT\$\$G_SCR_REBUILD	1328
				04 004D9	RET		1322
00000000G	00	10	A2	01 C3 004DA	SUBL3	#1, 16(R2), EDT\$\$G_WRDTP	1342
				04 004E3	RET		0854
00000000G	00	10	A2	01 C3 004E4	SUBL3	#1, 16(R2), EDT\$\$G_PARTYP	1354
				04 004ED	RET		0854

EDT\$LSET
V04-000

EDT\$LSET - SET line-mode command
EDT\$\$\$SET_CMD - SET line-mode command

J 10
16-Sep-1984 00:57:38
14-Sep-1984 12:23:40

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

Page 23
(3)

00000000G	7E 00	08	A2 02	7D 004EE	93\$:	MOVQ	8(R2), -(SP)	:	1362
				FB 004F2		CALLS	#2, EDT\$\$\$SET_HLPFNAM	:	
00000000G	00		01	04 004F9		RET		:	
				D0 004FA	94\$:	MOVL	#1, EDT\$\$G_ENB_AUTRPT	:	1365
				04 00501		RET		:	
		00000000G	00	D4 00502	95\$:	CLRL	EDT\$\$G_ENB_AUTRPT	:	1368
				04 00508		RET		:	
		0C	A2	D5 00509	96\$:	TSTL	12(R2)	:	1373
			0E	12 0050C		BNEQ	99\$	:	
		00000000G	8F	DD 0050E	97\$:	PUSHL	#EDT\$ INVSTR	:	1376
00000000G	00		01	FB 00514	98\$:	CALLS	#1, EDT\$\$FMT_MSG	:	
				04 0051B		RET		:	1373
	7E	00000000G	00	9A 0051C	99\$:	MOVZBL	EDT\$\$T_CMD_NAM_DEF4, -(SP)	:	1384
		00000000G	00	9F 00523		PUSHAB	EDT\$\$T_CMD_NAM_DEF4+1	:	1383
	7E	08	A2	7D 00529		MOVQ	8(R2), -(SP)	:	1384
00000000G	00		04	FB 0052D		CALLS	#4, EDT\$\$\$SET_COMFNAM	:	
				04 00534		RET		:	1394

; Routine Size: 1333 bytes, Routine Base: _EDT\$CODE + 000A

: 825 1395 1
: 826 1396 1 !<BLF/PAGE>

EDT\$LSET
VO4-000

EDT\$LSET - SET line-mode command
EDT\$\$\$SET_CMD - SET line-mode command

K 10
16-Sep-1984 00:57:38 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:23:40 DISK\$VMSMASTER:[EDT.SRC]LSET.BLI;1

Page 24
(4)

: 828
: 829
: 830

1397 1 END
1398 1
1399 0 ELUDOM

! of module EDT\$LSET

PSECT SUMMARY

Name Bytes Attributes

_EDT\$CODE 1343 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	55	14	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

COMMAND QUALIFIERS

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

: Size: 1333 code + 10 data bytes
: Run Time: 00:51.9
: Elapsed Time: 01:10.9
: Lines/CPU Min: 1617
: Lexemes/CPU-Min: 7317
: Memory Used: 309 pages
: Compilation Complete

0136

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY