

[illegible]

EXE

Mod

EDT

ED

ED
EDED
EDED
EDED
ED

ED

ED

ED

ED

ED
EDSYN
LBA

110

```

LL          IIIIII          SSSSSSSS
LL          IIIIII          SSSSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SSSSSS
LL          II             SSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```



```
0001 0 %TITLE 'EDT$FIXNOTRUN - Fix screen data base in NOTRUNCATE mode'
0002 0 MODULE EDT$FIXNOTRUN ( ! Fix screen data base in NOTRUNCATE mode
0003 0 IDENT = 'V04-000' ! File: FIXNOTRUN.BLI Edit: JBS1018
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 fixes up the screen data base if we are in NOTRUNCATE mode.
0037 1
0038 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0039 1
0040 1 AUTHOR: John Sauter, CREATION DATE: November 1, 1983
0041 1
0042 1 MODIFIED BY:
0043 1
0044 1 1-001 - Original. This is a stub which always forces the data base to be rebuilt.
0045 1 JBS 01-Nov-1982
0046 1 1-002 - Enable this routine. JBS 02-Nov-1982
0047 1 1-003 - Fix a problem in counting records. JBS 09-Nov-1982
0048 1 1-004 - Fix backwards search bug. SMB 23-Nov-1982
0049 1 1-005 - Worry about deleted lines when scanning backwards. JBS 25-Nov-1982
0050 1 1-006 - Revise handling of EDT$$G_SHF. JBS 14-Dec-1982
0051 1 1-007 - Fix tabbing at the end of a line. JBS 15-Dec-1982
0052 1 1-008 - Remove the edit buffer. JBS 27-Dec-1982
0053 1 1-009 - Fix a bug in finding the current screen pointer which caused excessive rebuilding. JBS 30-Dec-198
0054 1 1-010 - Support EDT$$G ANY CHANGES. JBS 05-Apr-1983
0055 1 1-011 - Remove EDT$$G ANY CHANGES; this routine is only called if there have been changes.
0056 1 Also, allocate and deallocate lines in the screen data base in simple cases, to
0057 1 avoid some rebuilds of the screen data base. JBS 19-Apr-1983
```

EDT\$FIXNOTRUN
V04-000

EDT\$FIXNOTRUN - Fix screen data base in NOTRUNC

6 2
16-Sep-1984 00:26:52
14-Sep-1984 12:23:09

VAX-11 Bliss-32 V4.0-742
[EDT.SRC]FIXNOTRUN.BLI;1

Page 2
(1)

```
: 58      0058 1 1-012 - Don't recompute the length of a deleted line. JBS 23-May-1983
: 59      0059 1 1-013 - When reclaiming a deleted screen pointer, mark it no longer deleted. JBS 25-May-1983
: 60      0060 1 1-014 - Fix an error in edit 1-013. JBS 26-May-1983
: 61      0061 1 1-015 - Don't change an inserted line to a deleted line. JBS 27-May-1983
: 62      0062 1 1-016 - If we undelete the current line, adjust the record number of the current line. JBS 03-Jun-1983
: 63      0063 1 1-017 - Fix breaking a line at its wrap point. JBS 23-Jun-1983
: 64      0064 1 1-018 - Rather than failing if the screen data base is bad, just rebuild it. JBS 22-Jul-1983
: 65      0065 1 --
: 66      0066 1
```

EDT
V04

EDT\$FIXNOTRUN
V04-000

EDT\$FIXNOTRUN - Fix screen data base in NOTRUNC
Declarations

H 2
16-Sep-1984 00:26:52
14-Sep-1984 12:23:09

VAX-11 Bliss-32 V4.0-742
[EDT.SRC]FIXNOTRUN.BLI;1

Page 3
(2)

```
: 68      0067 1 %SBTTL 'Declarations'
: 69      0068 1
: 70      0069 1 TABLE OF CONTENTS:
: 71      0070 1
: 72      0071 1
: 73      0072 1 REQUIRE 'EDTSRC:TRAROUNAM';
: 74      0511 1
: 75      0512 1 FORWARD ROUTINE
: 76      0513 1     EDT$FIX_NOTRUNC : NOVALUE;
: 77      0514 1
: 78      0515 1
: 79      0516 1 INCLUDE FILES:
: 80      0517 1
: 81      0518 1
: 82      0519 1 REQUIRE 'EDTSRC:EDTREQ';
: 83      0654 1
: 84      0655 1
: 85      0656 1 MACROS:
: 86      0657 1
: 87      0658 1     NONE
: 88      0659 1
: 89      0660 1 EQUATED SYMBOLS:
: 90      0661 1
: 91      0662 1     NONE
: 92      0663 1
: 93      0664 1 OWN STORAGE:
: 94      0665 1
: 95      0666 1     NONE
: 96      0667 1
: 97      0668 1 EXTERNAL REFERENCES:
: 98      0669 1
: 99      0670 1 In the routine
```

EDT
V04

```
101 0671 1 %SBTTL 'EDT$$FIX_NOTRUNC - fix screen data base in NOTRUNCATE mode'
102 0672 1
103 0673 1 GLOBAL ROUTINE EDT$$FIX_NOTRUNC ! Fix screen data base in NOTRUNCATE mode
104 0674 1 : NOVALUE =
105 0675 1
106 0676 1 ++
107 0677 1 FUNCTIONAL DESCRIPTION:
108 0678 1
109 0679 1 Fix the screen data base if we are in NOTRUNCATE mode. If a character is
110 0680 1 inserted on a line, or deleted from a line, or even replaced (since characters
111 0681 1 have different widths), the limits of this line must be recomputed. If this line's
112 0682 1 limits change the limits of the next line may also change.
113 0683 1
114 0684 1 FORMAL PARAMETERS:
115 0685 1
116 0686 1 NONE
117 0687 1
118 0688 1 IMPLICIT INPUTS:
119 0689 1
120 0690 1 EDT$$A_LN_PTR
121 0691 1 EDT$$T_LN_BUF
122 0692 1 EDT$$A_CUR_SCRPTR
123 0693 1 EDT$$A_FST_SCRPTR
124 0694 1 EDT$$A_FST_AVLN
125 0695 1 EDT$$L_CUR_SCRLN
126 0696 1 EDT$$A_SCR_BUF
127 0697 1 EDT$$A_CUR_BUF
128 0698 1 EDT$$L_CS_CN
129 0699 1 EDT$$L_CUR_SCRLN
130 0700 1 EDT$$A_CSR_SCRPTR
131 0701 1 EDT$$A_WK_CN
132 0702 1 EDT$$G_MEM_CNT
133 0703 1 EDT$$G_SHF
134 0704 1 EDT$$G_TI_WID
135 0705 1
136 0706 1 IMPLICIT OUTPUTS:
137 0707 1
138 0708 1 EDT$$G_SCR_REBUILD
139 0709 1 EDT$$A_FST_AVLN
140 0710 1 EDT$$G_RECS_INSERTED
141 0711 1
142 0712 1 ROUTINE VALUE:
143 0713 1
144 0714 1 NONE
145 0715 1
146 0716 1 SIDE EFFECTS:
147 0717 1
148 0718 1 MANY
149 0719 1
150 0720 1 --
151 0721 1
152 0722 2 BEGIN
153 0723 2
154 0724 2 EXTERNAL ROUTINE
155 0725 2 EDT$$SC_MOVTOLN, ! Move to a record in the work file relative to the current record
156 0726 2 EDT$$FMT_CHWID, ! Compute the width of a character
157 0727 2 EDT$$SC_FNDREC; ! Find the screen pointer for the current record
```



```
158 0728 2
159 0729 2
160 0730 2 EXTERNAL
161 0731 2 EDT$A_LN_PTR, ! Current line character position
162 0732 2 EDT$T_LN_BUF, ! Current line buffer
163 0733 2 EDT$G_SCR_REBUILD, ! Rebuild the screen data base
164 0734 2 EDT$A_FST_SCRPTR : REF SCREEN_LINE, ! First screen line info address
165 0735 2 EDT$A_FST_AVLN : REF SCREEN_LINE, ! List of available lines
166 0736 2 EDT$L_LNO_EMPTY : REF LN_BLOCK,
167 0737 2 EDT$L_CUR_SCRLN : REF LN_BLOCK,
168 0738 2 EDT$A_SCR_BUF : REF TBCB_BLOCK,
169 0739 2 EDT$A_CUR_BUF : REF TBCB_BLOCK,
170 0740 2 EDT$L_CS_LN : REF LN_BLOCK,
171 0741 2 EDT$A_WK_LN : REF LN_BLOCK,
172 0742 2 EDT$G_MEM_CNT,
173 0743 2 EDT$G_SHF, ! Number of columns shifted
174 0744 2 EDT$G_TI_WID, ! Screen width
175 0745 2 EDT$G_RECS_INSERTED; ! Count of records (lines) inserted and deleted.
176 0746 2
177 0747 2 LOCAL
178 0748 2 SCRPTR : REF SCREEN_LINE, ! Address of a current screen line buffer
179 0749 2 NXT_SCRPTR : REF SCREEN_LINE, ! Address of next screen line buffer
180 0750 2 REC_NO, ! Current relative record number
181 0751 2 FST_RECNO, ! Relative record number of first line in the screen data base
182 0752 2 UPDATE_DONE,
183 0753 2 ANOTHER_PASS,
184 0754 2 WIDTH,
185 0755 2 DISP,
186 0756 2 COL,
187 0757 2 LEFT,
188 0758 2 RIGHT,
189 0759 2 DIR,
190 0760 2 TXT,
191 0761 2 CHAR,
192 0762 2 LEN;
193 0763 2
194 0764 2 !+
195 0765 2 ! If we are already going to rebuild the screen data base, just return.
196 0766 2 !-
197 0767 2
198 0768 2 IF .EDT$G_SCR_REBUILD THEN RETURN;
199 0769 2
200 0770 2 !+
201 0771 2 ! Compute the direction of motion.
202 0772 2 !-
203 0773 2
204 0774 2 IF (LINNOEQL (EDT$L_LNO_EMPTY, EDT$L_CUR_SCRLN)) OR (.EDT$A_SCR_BUF NEQA .EDT$A_CUR_BUF)
205 0775 2 THEN
206 0776 2 DIR = 1
207 0777 2 ELSE
208 0778 2 DIR = CMPLNO (EDT$L_CS_LN, EDT$L_CUR_SCRLN);
209 0779 2
210 0780 2 !+
211 0781 2 ! Find the relative record number of the old cursor line.
212 0782 2 ! We must be careful of deleted lines. The convention is that a deleted line
213 0783 2 ! has the record number of the next lower line. This prevents deleted
214 0784 2 ! lines before record zero from having negative absolute record numbers.
```

```
215 0785 2 !-
216 0786 2 SCRPTR = EDT$$SC_FNDREC (.EDT$$A_LN_PTR - EDT$$T_LN_BUF, DISP);
217 0787 2
218 0788 2 IF (.SCRPTR EQLA 0)
219 0789 2 THEN
220 0790 2 BEGIN
221 0791 2 EDT$$G_SCR_REBUILD = 1;
222 0792 2 RETURN;
223 0793 2 END;
224 0794 2
225 0795 2 REC_NO = 0;
226 0796 2
227 0797 2 CASE .DIR FROM -1 TO 1 OF
228 0798 2 SET
229 0799 2
230 0800 2 [1] :
231 0801 2 BEGIN
232 0802 2
233 0803 2 !+ The new line is after the old. We must move back in the work file.
234 0804 2 !-
235 0805 2
236 0806 2 DO
237 0807 2 BEGIN
238 0808 2
239 0809 2 IF ((.SCRPTR [SCR_LINE_IDX] EQL 0) OR ((.SCRPTR [SCR_EDIT_FLAGS] AND SCR_EDIT_DELLN) NEQ 0))
240 0810 2 THEN
241 0811 2 BEGIN
242 0812 2
243 0813 2 LOCAL
244 0814 2 PREV_SCPTR : REF SCREEN_LINE;
245 0815 2
246 0816 2 PREV_SCPTR = .SCRPTR [SCR_PRV_LINE];
247 0817 2
248 0818 2 IF (.PREV_SCPTR NEQA 0)
249 0819 2 THEN
250 0820 2 BEGIN
251 0821 2
252 0822 2 IF ((.PREV_SCPTR [SCR_EDIT_FLAGS] AND SCR_EDIT_DELLN) EQL 0)
253 0823 2 THEN
254 0824 2 REC_NO = .REC_NO - 1;
255 0825 2
256 0826 2 END;
257 0827 2
258 0828 2 END;
259 0829 2
260 0830 2 SCRPTR = .SCRPTR [SCR_PRV_LINE];
261 0831 2 END
262 0832 2 UNTIL ((.SCRPTR EQLA .EDT$$A_CSR_SCPTR) OR (.SCRPTR EQLA 0));
263 0833 2
264 0834 2 END;
265 0835 2
266 0836 2 [0] :
267 0837 2 BEGIN
268 0838 2
269 0839 2 !+ We are positioned correctly in the work file.
270 0840 2 !-
271 0841 2 SCRPTR = .EDT$$A_CSR_SCPTR;
```



```
272 0842 2      END;
273 0843 2
274 0844 2      [-1] :
275 0845 2      BEGIN
276 0846 2      !+
277 0847 2      !- The new line is before the old. We must move forward in the work file.
278 0848 2
279 0849 2
280 0850 2      DO
281 0851 2      BEGIN
282 0852 2
283 0853 2      LOCAL
284 0854 2      NEXT_SCRPTR : REF SCREEN_LINE;
285 0855 2
286 0856 2      NEXT_SCRPTR = .SCRPTR [SCR_NXT_LINE];
287 0857 2
288 0858 2      IF (.NEXT_SCRPTR NEQA 0)
289 0859 2      THEN
290 0860 2      BEGIN
291 0861 2
292 0862 2      IF (((.SCRPTR [SCR_EDIT_FLAGS] AND SCR_EDIT_DELLN) EQL 0) AND      !
293 0863 2      ((.NEXT_SCRPTR [SCR_LINE_IDX] EQL 0) OR      !
294 0864 2      ((.NEXT_SCRPTR [SCR_EDIT_FLAGS] AND SCR_EDIT_DELLN) NEQ 0)))
295 0865 2      THEN
296 0866 2      REC_NO = .REC_NO + 1;
297 0867 2
298 0868 2      END;
299 0869 2
300 0870 2      SCRPTR = .SCRPTR [SCR_NXT_LINE];
301 0871 2      END
302 0872 2      UNTIL ((.SCRPTR EQLA .EDT$$A_CSR_SCRPTR) OR (.SCRPTR EQLA 0));
303 0873 2
304 0874 2      END;
305 0875 2
306 0876 2      [OUTRANGE] :
307 0877 2      ASSERT (0);
308 0878 2      TES;
309 0879 2
310 0880 2      !+
311 0881 2      !- If we couldn't find it, rebuild the screen data base.
312 0882 2
313 0883 2
314 0884 2      IF (.SCRPTR NEQA .EDT$$A_CSR_SCRPTR)
315 0885 2      THEN
316 0886 2      BEGIN
317 0887 2      EDT$$G_SCR_REBUILD = 1;
318 0888 2      RETURN;
319 0889 2      END;
320 0890 2
321 0891 2      !+
322 0892 2      !- Now work backwards to the first line.
323 0893 2
324 0894 2
325 0895 2      WHILE ((.SCRPTR NEQA .EDT$$A_FST_SCRPTR) AND (.SCRPTR NEQA 0)) DO
326 0896 2      BEGIN
327 0897 2
328 0898 2      IF (((.SCRPTR [SCR_LINE_IDX] EQL 0) OR ((.SCRPTR [SCR_EDIT_FLAGS] AND SCR_EDIT_DELLN) NEQ 0))
```

```

329      0899 3      THEN
330      0900 4      BEGIN
331      0901 4
332      0902 4      LOCAL
333      0903 4      PREV_SCRPTR : REF SCREEN_LINE;
334      0904 4
335      0905 4      PREV_SCRPTR = .SCRPTR [SCR_PRV_LINE];
336      0906 4
337      0907 5      IF (.PREV_SCRPTR NEQA 0)
338      0908 4      THEN
339      0909 5      BEGIN
340      0910 5
341      0911 5      IF ((.PREV_SCRPTR [SCR_EDIT_FLAGS] AND SCR_EDIT_DELLN) EQL 0) THEN REC_NO = .REC_NO - 1;
342      0912 5
343      0913 4      END;
344      0914 4
345      0915 3      END;
346      0916 3
347      0917 3      SCRPTR = .SCRPTR [SCR_PRV_LINE];
348      0918 3      END;
349      0919 2
350      0920 2      FST_RECNO = .REC_NO;
351      0921 2      +
352      0922 2      :- If we didn't find it, rebuild the screen data base.
353      0923 2      :-
354      0924 2
355      0925 3      IF (.SCRPTR NEQA .EDT$$A_FST_SCRPTR)
356      0926 3      THEN
357      0927 3      BEGIN
358      0928 3      EDT$$G_SCR_REBUILD = 1;
359      0929 3      RETURN;
360      0930 3      END;
361      0931 2
362      0932 2      +
363      0933 2      :- Recompute the end points of any lines which have changed.
364      0934 2      :-
365      0935 2
366      0936 2      DO
367      0937 3      BEGIN
368      0938 3      ANOTHER_PASS = 0;
369      0939 3      REC_NO = .FST_RECNO;
370      0940 3      SCRPTR = .EDT$$A_FST_SCRPTR;
371      0941 3
372      0942 3      DO
373      0943 4      BEGIN
374      0944 4      UPDATE_DONE = 0;
375      0945 4
376      0946 5      IF (((.SCRPTR [SCR_EDIT_FLAGS] AND (SCR_EDIT_MODIFY OR SCR_EDIT_INSLN)) NEQ 0) AND !
377      0947 5      ((.SCRPTR [SCR_EDIT_FLAGS] AND SCR_EDIT_DELLN) EQL 0))
378      0948 5      THEN
379      0949 5      BEGIN
380      0950 5      +
381      0951 5      :- Compute the width of the current line.
382      0952 5      :-
383      0953 5      LEFT = .SCRPTR [SCR_CHR_FROM];
384      0954 5
385      0955 6      IF (.LEFT EQL 0)
```



```
386 0956 5 THEN
387 0957 6 BEGIN
388 0958 6 WIDTH = .EDT$$G_TI_WID + .EDT$$G_SHF;
389 0959 6 COL = 0;
390 0960 6 END
391 0961 5 ELSE
392 0962 6 BEGIN
393 0963 6 WIDTH = .EDT$$G_TI_WID + .EDT$$G_SHF;
394 0964 6 COL = .EDT$$G_SHF + 2;
395 0965 5 END;
396 0966 5
397 0967 5 !+
398 0968 5 !- We must compute the width of each character to see how many will fit.
399 0969 5 !-
400 0970 5
401 0971 6 IF ( NOT EDT$$SC_MOVTOLN (.REC_NO))
402 0972 5 THEN
403 0973 6 BEGIN
404 0974 6 !+
405 0975 6 !- Something appears to be wrong with the screen data base. Rebuild it.
406 0976 6 !-
407 0977 6 EDT$$SC_MOVTOLN (0);
408 0978 6 EDT$$G_SCR_REBUILD = 1;
409 0979 6 RETURN;
410 0980 5 END;
411 0981 5
412 0982 5 LEN = .EDT$$A_WK_LN [LIN_LENGTH] - .LEFT;
413 0983 5 TXT = CH$PLUS (EDT$$A_WK_LN [LIN_TEXT], .LEFT);
414 0984 5 RIGHT = .LEFT - 1;
415 0985 5 CHAR = CH$RCHAR_A (TXT);
416 0986 5
417 0987 5 WHILE ((.LEN GTR 0) AND ((.COL + EDT$$FMT_CHWID (.CHAR, .COL)) LEQ .WIDTH)) DO
418 0988 6 BEGIN
419 0989 6 LEN = .LEN - 1;
420 0990 6 RIGHT = .RIGHT + 1;
421 0991 6 COL = .COL + EDT$$FMT_CHWID (.CHAR, .COL);
422 0992 6 CHAR = CH$RCHAR_A (TXT);
423 0993 5 END;
424 0994 5
425 0995 5 IF (.LEN EQL 0) THEN RIGHT = 255;
426 0996 5
427 0997 6 IF (.SCRPTR [SCR_CHR_TO] NEQ .RIGHT)
428 0998 5 THEN
429 0999 6 BEGIN
430 1000 6 !+
431 1001 6 !- This line's width has changed. Store the new width and
432 1002 6 !- arrange to recompute the width of the next line.
433 1003 6 !-
434 1004 6 SCRPTR [SCR_CHR_TO] = .RIGHT;
435 1005 6
436 1006 7 IF (.RIGHT NEQ 255)
437 1007 6 THEN
438 1008 7 BEGIN
439 1009 7
440 1010 7 LOCAL
441 1011 7 NEED_NEW_SCRPTR;
442 1012 7
```

```

: 443      1013 7      NEED_NEW_SCRPTR = 0;
: 444      1014 7      NXT_SCRPTR = .SCRPTR [SCR_NXT_LINE];
: 445      1015 7
: 446      1016 8      IF (.NXT_SCRPTR EQLA 0)
: 447      1017 7      THEN
: 448      1018 7          NEED_NEW_SCRPTR = 1
: 449      1019 7      ELSE
: 450      1020 7
: 451      1021 8          IF ((.NXT_SCRPTR [SCR_CHR_FROM] EQL 0) AND !
: 452      1022 8              ((.NXT_SCRPTR [SCR_EDIT_FLAGS] AND SCR_EDIT_DELLN) EQL 0))
: 453      1023 7          THEN
: 454      1024 7              NEED_NEW_SCRPTR = 1;
: 455      1025 7
: 456      1026 7      IF .NEED_NEW_SCRPTR
: 457      1027 7      THEN
: 458      1028 8          BEGIN
: 459      1029 8      !+
: 460      1030 8      ! We need a new screen pointer. If we can get one easily, do so.
: 461      1031 8      ! Otherwise rebuild the screen data base.
: 462      1032 8      !-
: 463      1033 8
: 464      1034 8          LOCAL
: 465      1035 8              TMP_SCRPTR : REF SCREEN_LINE;
: 466      1036 8
: 467      1037 8          NXT_SCRPTR = .EDT$$A_FST_AVLN;
: 468      1038 8
: 469      1039 9          IF (.NXT_SCRPTR EQLA 0)
: 470      1040 8          THEN
: 471      1041 9              BEGIN
: 472      1042 9                  EDT$$$SC MOVTOLN (0);
: 473      1043 9                  EDT$$G_SCR_REBUILD = 1;
: 474      1044 9                  RETURN;
: 475      1045 8                  END;
: 476      1046 8
: 477      1047 8          EDT$$G_MEM_CNT = .EDT$$G_MEM_CNT + 1;
: 478      1048 8          EDT$$G_RECS_INSERTED = .EDT$$G_RECS_INSERTED + 1;
: 479      1049 8          EDT$$A_FST_AVLN = .NXT_SCRPTR [SCR_NXT_LINE];
: 480      1050 8          NXT_SCRPTR [SCR_PRV_LINE] = .SCRPTR;
: 481      1051 8          TMP_SCRPTR = .SCRPTR [SCR_NXT_LINE];
: 482      1052 8          NXT_SCRPTR [SCR_NXT_LINE] = .TMP_SCRPTR;
: 483      1053 8          SCRPTR [SCR_NXT_LINE] = .NXT_SCRPTR;
: 484      1054 8
: 485      1055 8          IF (.TMP_SCRPTR NEQA 0) THEN TMP_SCRPTR [SCR_PRV_LINE] = .NXT_SCRPTR;
: 486      1056 8
: 487      1057 8          NXT_SCRPTR [SCR_EDIT_FLAGS] = SCR_EDIT_INSLN;
: 488      1058 7          END;
: 489      1059 7
: 490      1060 7      NXT_SCRPTR [SCR_LINE_IDX] = .SCRPTR [SCR_LINE_IDX] + 1;
: 491      1061 7
: 492      1062 8      IF (.NXT_SCRPTR [SCR_LINE_IDX] EQL 0)
: 493      1063 7      THEN
: 494      1064 8          BEGIN
: 495      1065 8              EDT$$$SC MOVTOLN (0);
: 496      1066 8              EDT$$G_SCR_REBUILD = 1;
: 497      1067 8              RETURN;
: 498      1068 7              END;
: 499      1069 7
```



```
500 1070 7 !+
501 1071 7 !- If we are undeleting the current line, adjust the record number of the current line.
502 1072 7 !-
503 1073 7
504 1074 8 IF (((.NXT_SCRPTR [SCR_EDIT_FLAGS] AND SCR_EDIT_DELLN) NEQ 0) AND !
505 1075 8 (.NXT_SCRPTR EQLA .EDT$$$A_CSR_SCRPTR))
506 1076 7 THEN
507 1077 7 SUBLINE (NUMBER_ONE, EDT$$L_CUR_SCRLN);
508 1078 7
509 1079 7 NXT_SCRPTR [SCR_CHR_FROM] = .RIGHT + 1;
510 1080 7 NXT_SCRPTR [SCR_CHR_TO] = 255;
511 1081 7 NXT_SCRPTR [SCR_EDIT_MINPOS] = 0;
512 1082 7 NXT_SCRPTR [SCR_EDIT_MAXPOS] = 255;
513 1083 7 NXT_SCRPTR [SCR_EDIT_FLAGS] = (.NXT_SCRPTR [SCR_EDIT_FLAGS] OR SCR_EDIT_MODIFY);
514 1084 7 NXT_SCRPTR [SCR_EDIT_FLAGS] = .NXT_SCRPTR [SCR_EDIT_FLAGS] AND (NOT SCR_EDIT_DELLN);
515 1085 7 END
516 1086 6 ELSE
517 1087 7 BEGIN
518 1088 7 !+
519 1089 7 !- We have reached the end of the record, make sure there aren't any more lines allocated for it.
520 1090 7 !-
521 1091 7
522 1092 7 LOCAL
523 1093 7 RECORD_DONE;
524 1094 7
525 1095 7 DO
526 1096 8 BEGIN
527 1097 8 NXT_SCRPTR = .SCRPTR [SCR_NXT_LINE];
528 1098 8
529 1099 9 IF (.NXT_SCRPTR NEQA 0)
530 1100 8 THEN
531 1101 9 BEGIN
532 1102 9
533 1103 10 IF (.NXT_SCRPTR [SCR_CHR_FROM] NEQ 0)
534 1104 9 THEN
535 1105 10 BEGIN
536 1106 10 !+
537 1107 10 !- If the next line is marked as insert, then we have a strange state: the line
538 1108 10 was apparently inserted long and then truncated before we had a chance to
539 1109 10 update the screen. Rather than trying to handle this case correctly just
540 1110 10 rebuild the screen data base.
541 1111 10 !-
542 1112 10
543 1113 11 IF (((.NXT_SCRPTR [SCR_EDIT_FLAGS] AND SCR_EDIT_INSLN) NEQ 0)
544 1114 10 THEN
545 1115 11 BEGIN
546 1116 11 EDT$$$SC MOVTOLN (0);
547 1117 11 EDT$$$G_SCR_REBUILD = 1;
548 1118 11 RETURN;
549 1119 10 END;
550 1120 10
551 1121 10 !+
552 1122 10 !- Mark this as an independent, deleted line. The block will be discarded by the time
553 1123 10 the screen updater completes.
554 1124 10 !-
555 1125 10
556 1126 10 NXT_SCRPTR [SCR_EDIT_FLAGS] = SCR_EDIT_DELLN;
NXT_SCRPTR [SCR_LINE_IDX] = 0;
```

```

: 557      1127 10      NXT_SCRPTR [SCR_CHR_FROM] = 0;
: 558      1128 10      NXT_SCRPTR [SCR_CHR_TO] = 255;
: 559      1129 10
: 560      1130 10      !+ Since a deleted line has the number of the next following non-deleted line,
: 561      1131 10      !- if we are deleting the current line add one to its line number.
: 562      1132 10
: 563      1133 10
: 564      1134 11      IF (.NXT_SCRPTR EQLA .EDT$$A_CSR_SCRPTR)      !
: 565      1135 10      THEN
: 566      1136 10          ADDLINE (NUMBER_ONE, EDT$$L_CUR_SCRLN);
: 567      1137 10
: 568      1138 10      EDT$$G_RECS_INSERTED = .EDT$$G_RECS_INSERTED + 1;
: 569      1139 10      SCRPTR = .NXT_SCRPTR;
: 570      1140 10      RECORD_DONE = 0;
: 571      1141 10      END
: 572      1142 9        ELSE
: 573      1143 9          RECORD_DONE = 1;
: 574      1144 9
: 575      1145 9        END
: 576      1146 8        ELSE
: 577      1147 8          RECORD_DONE = 1;
: 578      1148 8
: 579      1149 8        END
: 580      1150 7        UNTIL .RECORD_DONE;
: 581      1151 7
: 582      1152 6        END;
: 583      1153 6
: 584      1154 5        END;
: 585      1155 5
: 586      1156 4        END;
: 587      1157 4
: 588      1158 5        IF ( NOT .UPDATE_DONE)
: 589      1159 4        THEN
: 590      1160 5            BEGIN
: 591      1161 5
: 592      1162 6            IF (.SCRPTR [SCR_NXT_LINE] EQLA 0)
: 593      1163 5            THEN
: 594      1164 5                UPDATE_DONE = 1
: 595      1165 5            ELSE
: 596      1166 6                BEGIN
: 597      1167 6
: 598      1168 6                LOCAL
: 599      1169 6                    NEXT_SCRPTR : REF SCREEN_LINE;
: 600      1170 6
: 601      1171 6                    NEXT_SCRPTR = .SCRPTR [SCR_NXT_LINE];
: 602      1172 6
: 603      1173 7                    IF (((.SCRPTR [SCR_EDIT_FLAGS] AND SCR_EDIT_DELLN) EQL 0) AND      !
: 604      1174 8                        ((.NEXT_SCRPTR [SCR_LINE_IDX] EQL 0) OR      !
: 605      1175 7                        ((.NEXT_SCRPTR [SCR_EDIT_FLAGS] AND SCR_EDIT_DELLN) NEQ 0)))
: 606      1176 6                    THEN
: 607      1177 6                        REC_NO = .REC_NO + 1;
: 608      1178 6
: 609      1179 6                    SCRPTR = .SCRPTR [SCR_NXT_LINE];
: 610      1180 5                    END;
: 611      1181 5
: 612      1182 4        END;
: 613      1183 4        END;
```


EDT\$FIXNOTRUN
V04-000

EDT\$FIXNOTRUN - Fix screen data base in NOTRUNC
EDT\$\$\$FIX_NOTRUNC - fix screen data base in NOTR

E 3

16-Sep-1984 00:26:52
14-Sep-1984 12:23:09

VAX-11 Bliss-32 V4.0-742
[EDT.SRC]FIXNOTRUN.BLI;1

Page 13
(3)

```
: 614      1184  4      END
: 615      1185  4      UNTIL .UPDATE_DONE;
: 616      1186  4
: 617      1187  4      END
: 618      1188  4      UNTIL ( NOT .ANOTHER_PASS);
: 619      1189  4
: 620      1190  2      EDT$$$SC_MOVTOLN (0);
: 621      1191  1      END;
```

! of routine EDT\$\$\$FIX_NOTRUNC

.TITLE EDT\$FIXNOTRUN EDT\$FIXNOTRUN - Fix screen data base in NOTRUNC

.IDENT \V04-000\

.EXTRN EDT\$\$\$SC_MOVTOLN
.EXTRN EDT\$\$\$FMT_CHWID, EDT\$\$\$SC_FNDREC
.EXTRN EDT\$\$\$A_LN_PTR, EDT\$\$\$T_LN_BUF
.EXTRN EDT\$\$\$G_SCR_REBUILD
.EXTRN EDT\$\$\$A_FST_SCRPTR
.EXTRN EDT\$\$\$A_FST_AVLN
.EXTRN EDT\$\$\$L_LNO_EMPTY
.EXTRN EDT\$\$\$L_CUR_SCRLN
.EXTRN EDT\$\$\$A_SCR_BUF, EDT\$\$\$A_CUR_BUF
.EXTRN EDT\$\$\$L_CS_CN, EDT\$\$\$A_CSR_SCRPTR
.EXTRN EDT\$\$\$A_WK_LN, EDT\$\$\$G_MEM_CNT
.EXTRN EDT\$\$\$G_SHF, EDT\$\$\$G_TI_WID
.EXTRN EDT\$\$\$G_RECS_INSERTED
.EXTRN EDT\$\$\$INTER_ERR

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

.ENTRY EDT\$\$\$FIX_NOTRUNC, Save R2,R3,R4,R5,R6,R7,- R8,R9,R10,R11 : 0673

SUBL2 #20, SP
BLBC EDT\$\$\$G_SCR_REBUILD, 1\$: 0768

RET
MOVL LOW_2, R3 : 0774

CMPL LOW_1, R3
BNEQ 2\$

CMPL HIGH_1, HIGH_2
BEQL 5\$

CMPL EDT\$\$\$A_SCR_BUF, EDT\$\$\$A_CUR_BUF : 0778

BNEQ 5\$
MOVZWL HIGH_1, R1

MOVZWL HIGH_2, R0
CMPL R1, R0

BLSSU 3\$
BNEQ 5\$

MOVL LOW_1, R0
CMPL R0, R3

BGEQU 4\$
MNEGL #1, DIR

BRB 6\$
BNEQ 5\$

CLRL DIR
BRB 6\$

MOVL #1, DIR

```
OFFC 00000
5E 14 C2 00002
01 00 0000000G 00 E9 00005
04 0000C
53 00 0000000G 00 D0 0000D 1$:
53 00 0000000G 00 D1 00014
0D 12 0001B
00000000G 00 0000000G 00 B1 0001D
39 13 00028
00000000G 00 0000000G 00 D1 0002A 2$:
2C 12 00035
51 00 0000000G 00 3C 00037
50 00 0000000G 00 3C 0003E
50 51 D1 00045
0E 1F 00048
17 12 0004A
50 00 0000000G 00 D0 0004C
53 50 D1 00053
05 1E 00056
52 01 CE 00058 3$:
09 11 0005B
04 12 0005D 4$:
52 D4 0005F
03 11 00061
52 01 D0 00063 5$:
```

			10	AE	9F	00066	6\$:	PUSHAB	DISP		0786
			00	00	9E	00069		MOVAB	EDT\$ST LN BUF, R0		
7E	00000000G	50	00	50	C3	00070		SUBL3	R0, EDT\$SA LN PTR, -(SP)		
	00000000G	00		02	FB	00078		CALLS	#2, EDT\$ST_FNDREC		
		53		50	D0	0007F		MOVL	R0, SCRPTTR		
				03	12	00082		BNEQ	8\$		0788
				021B	31	00084	7\$:	BRW	42\$		
				56	D4	00087	8\$:	CLRL	REC NO		0795
02	FFFFFFFF	8F		52	CF	00089		CASEL	DIR, #-1, #2		0797
000F		0037		0040		00091	9\$:	.WORD	14\$-9\$,-		
									13\$-9\$,-		
									10\$-9\$		
	00000000G	00		00	FB	00097		CALLS	#0, EDT\$INTER_ERR		0877
				59	11	0009E		BRB	17\$		0797
			08	A3	95	000A0	10\$:	TSTB	8(SCRPTTR)		0809
				05	13	000A3		BEQL	11\$		
0C	0D	A3		02	E1	000A5		BBC	#2, 13(SCRPTTR), 12\$		
		50		63	D0	000AA	11\$:	MOVL	(SCRPTTR), PREV_SCRPTTR		0816
				07	13	000AD		BEQL	12\$		0818
02	0D	A0		02	E0	000AF		BBS	#2, 13(PREV_SCRPTTR), 12\$		0822
				56	D7	000B4		DECL	REC NO		0824
		53		63	D0	000B6	12\$:	MOVL	(SCRPTTR), SCRPTTR		0830
00000000G	00			53	D1	000B9		CMPL	SCRPTTR, EDT\$SA_CSR_SCRPTTR		0832
				37	13	000C0		BEQL	17\$		
				53	D5	000C2		TSTL	SCRPTTR		
				DA	12	000C4		BNEQ	10\$		
				31	11	000C6		BRB	17\$		0797
		53	00000000G	00	D0	000C8	13\$:	MOVL	EDT\$SA_CSR_SCRPTTR, SCRPTTR		0841
				28	11	000CF		BRB	17\$		0797
		50	04	A3	D0	000D1	14\$:	MOVL	4(SCRPTTR), NEXT_SCRPTTR		0856
				11	13	000D5		BEQL	16\$		0858
0C	0D	A3		02	E0	000D7		BBS	#2, 13(SCRPTTR), 16\$		0862
			08	A0	95	000DC		TSTB	8(NEXT_SCRPTTR)		0863
				05	13	000DF		BEQL	15\$		
02	0D	A0		02	E1	000E1		BBC	#2, 13(NEXT_SCRPTTR), 16\$		0864
				56	D6	000E6	15\$:	INCL	REC NO		0866
		53	04	A3	D0	000E8	16\$:	MOVL	4(SCRPTTR), SCRPTTR		0870
00000000G	00			53	D1	000EC		CMPL	SCRPTTR, EDT\$SA_CSR_SCRPTTR		0872
				04	13	000F3		BEQL	17\$		
				53	D5	000F5		TSTL	SCRPTTR		
				D8	12	000F7		BNEQ	14\$		
	00000000G	00		53	D1	000F9	17\$:	CMPL	SCRPTTR, EDT\$SA_CSR_SCRPTTR		0884
				82	12	00100		BNEQ	7\$		
		51	00000000G	00	D0	00102		MOVL	EDT\$SA_FST_SCRPTTR, R1		0895
		51		53	D1	00109	18\$:	CMPL	SCRPTTR, R1		
				1F	13	0010C		BEQL	21\$		
				53	D5	0010E		TSTL	SCRPTTR		
				1B	13	00110		BEQL	21\$		
			08	A3	95	00112		TSTB	8(SCRPTTR)		0898
				05	13	00115		BEQL	19\$		
0C	0D	A3		02	E1	00117		BBC	#2, 13(SCRPTTR), 20\$		
		50		63	D0	0011C	19\$:	MOVL	(SCRPTTR), PREV_SCRPTTR		0905
				07	13	0011F		BEQL	20\$		0907
02	0D	A0		02	E0	00121		BBS	#2, 13(PREV_SCRPTTR), 20\$		0911
				56	D7	00126		DECL	REC NO		
		53		63	D0	00128	20\$:	MOVL	(SCRPTTR), SCRPTTR		0917
				DC	11	0012B		BRB	18\$		0895

OC	AE	56	D0	0012D	21\$:	MOVL	REC NO, FST_RECNO	0920	
	51	53	D1	00131		CMPL	SCRPTR, R1	0925	
		03	13	00134		BEQL	22\$		
		0169	31	00136		BRW	42\$		
	08	AE	D4	00139	22\$:	CLRL	ANOTHER PASS	0938	
	56	OC	AE	D0	0013C	MOVL	FST_RECNO, REC NO	0939	
	53	00000000G	00	D0	00140	MOVL	EDT\$A_FST_SCRPTR, SCRPTR	0940	
		5A	D4	00147	23\$:	CLRL	UPDATE-DONE	0944	
	03	0D	A3	93	00149	BITB	13(SCRPTR), #3	0946	
		03	12	0014D		BNEQ	25\$		
		018D	31	0014F	24\$:	BRW	47\$		
F8	OD	A3	02	E0	00152	25\$:	BBS	#2, 13(SCRPTR), 24\$	0947
		55	A3	9A	00157	MOVZBL	9(SCRPTR), LEFT	0953	
04	AE	00000000G	00	D0	0015B	MOVL	EDT\$G_SHF, RO	0958	
		50	C1	00162		ADDL3	RO, EDT\$G_TI_WID, WIDTH		
		55	D5	0016B		TSTL	LEFT	0955	
		04	12	0016D		BNEQ	26\$		
		58	D4	0016F		CLRL	COL	0959	
		04	11	00171		BRB	27\$	0955	
	58	02	A0	9E	00173	26\$:	MOVAB	2(RO), COL	0964
		56	DD	00177	27\$:	PUSHL	REC_NO	0971	
00000000G	00	01	FB	00179		CALLS	#1, EDT\$SC_MOVTOLN		
	03	50	E8	00180		BLBS	RO, 28\$		
		0113	31	00183		BRW	41\$		
	50	00000000G	00	D0	00186	28\$:	MOVL	EDT\$A_WK_LN, RO	0982
	59	60	9A	0018D		MOVZBL	(RO), LEN		
	59	55	C2	00190		SUBL2	LEFT, LEN		
	5B	07	A5	40	9E	00193	MOVAB	7(LEFT)(RO), TXT	0983
	57	FF	A5	9E	00198	MOVAB	-1(R5), RIGHT	0984	
	6E	8B	9A	0019C	29\$:	MOVZBL	(TXT)+, CHAR	0985	
		59	D5	0019F		TSTL	LEN	0987	
		2A	15	001A1		BLEQ	30\$		
		58	DD	001A3		PUSHL	COL		
		04	AE	DD	001A5	PUSHL	CHAR		
00000000G	00	02	FB	001A8		CALLS	#2, EDT\$FMT_CHWID		
	50	58	C0	001AF		ADDL2	COL, RO		
04	AE	50	D1	001B2		CMPL	RO, WIDTH		
		15	14	001B6		BGTR	30\$		
		59	D7	001B8		DECL	LEN	0989	
		57	D6	001BA		INCL	RIGHT	0990	
		58	DD	001BC		PUSHL	COL	0991	
		04	AE	DD	001BE	PUSHL	CHAR		
00000000G	00	02	FB	001C1		CALLS	#2, EDT\$FMT_CHWID		
	58	50	C0	001C8		ADDL2	RO, COL		
		CF	11	001CB		BRB	29\$	0992	
		59	D5	001CD	30\$:	TSTL	LEN	0995	
		04	12	001CF		BNEQ	31\$		
	57	FF	8F	9A	001D1	MOVZBL	#255, RIGHT		
57	OA	A3	00	ED	001D5	31\$:	CMPZV	#0, #8, 10(SCRPTR), RIGHT	0997
			03	12	001DB	BNEQ	32\$		
		00FF	31	001DD		BRW	47\$		
		57	90	001E0	32\$:	MOVB	RIGHT, 10(SCRPTR)	1004	
	OA	A3	57	D1	001E4	CMPL	RIGHT, #255	1006	
000000FF	8F	03	12	001EB		BNEQ	33\$		
		0099	31	001ED		BRW	40\$		
		50	D4	001F0	33\$:	CLRL	NEED NEW SCRPTR	1013	
	52	04	A3	D0	001F2	MOVL	4(SCRPTR), NXT_SCRPTR	1014	

			09	0A	13	001F6	BEQL	34\$	:	1016
				A2	95	001F8	TSTB	9(NXT_SCRPTR)	:	1021
				08	12	001FB	BNEQ	35\$	:	
03	0D	A2		02	E0	001FD	BBS	#2, 13(NXT_SCRPTR), 35\$	:	1022
		50		01	D0	00202	34\$:	MOV	:	1024
		37		50	E9	00205	35\$:	BLBC	:	1026
		52	00000000G	00	D0	00208		MOV	:	1037
				34	13	0020F		BEQL	:	1039
			00000000G	00	D6	00211		INCL	:	1047
			00000000G	00	D6	00217		INCL	:	1048
	00000000G	00	04	A2	D0	0021D		MOV	:	1049
		62		53	D0	00225		MOV	:	1050
		50	04	A3	D0	00228		MOV	:	1051
	04	A2		50	D0	0022C		MOV	:	1052
	04	A3		52	D0	00230		MOV	:	1053
				50	D5	00234		TSTL	:	1055
				03	13	00236		BEQL	:	
		60		52	D0	00238		MOV	:	
08	A2	0D		02	90	0023B	36\$:	MOV	:	1057
		08		01	81	0023F	37\$:	ADDB3	:	1060
				52	13	00245	38\$:	BEQL	:	1062
25	0D	A2		02	E1	00247		BBC	:	1074
	00000000G	00		52	D1	0024C		CMPL	:	1075
				1C	12	00253		BNEQ	:	
		50	00000000G	00	D0	00255		MOV	:	1077
				00	D7	0025C		DECL	:	
		50	00000000G	00	D1	00262		CMPL	:	
				06	1B	00269		BLEQU	:	
			00000000G	00	B7	0026B		DECW	:	
09	A2	57		01	81	00271	39\$:	ADDB3	:	1079
		A2	FF	8F	9B	00276		MOVZBW	:	1080
	0C	A2		01	8E	0027B		MNEGB	:	1082
	0D	A2		01	88	0027F		BISB2	:	1083
	0D	A2		04	8A	00283		BICB2	:	1084
				56	11	00287		BRB	:	1006
		52	04	A3	D0	00289	40\$:	MOV	:	1097
				4A	13	0028D		BEQL	:	1099
			09	A2	95	0028F		TSYB	:	1103
				45	13	00292		BEQL	:	
11	0D	A2		01	E1	00294		BBC	:	1113
				7E	D4	00299	41\$:	CLRL	:	1116
	00000000G	00		01	FB	0029B		CALLS	:	
	00000000G	00		01	D0	002A2	42\$:	MOV	:	1117
				04	002A9			RET	:	1115
	0D	A2		04	90	002AA	43\$:	MOV	:	1125
			08	A2	B4	002AE		CLRW	:	1126
	0A	A2		01	8E	002B1		MNEGB	:	1128
	00000000G	00		52	D1	002B5		CMPL	:	1134
				0E	12	002BC		BNEQ	:	
		00000000G		00	D6	002BE		INCL	:	1136
				06	12	002C4		BNEQ	:	
		00000000G		00	B6	002C6		INCL	:	
		00000000G		00	D6	002CC	44\$:	INCL	:	1138
		53		52	D0	002D2		MOV	:	1139
				54	D4	002D5		CLRL	:	1140
				03	11	002D7		BRB	:	1103
		54		01	D0	002D9	45\$:	MOV	:	1147

EDT\$FIXNOTRUN
V04-000

EDT\$FIXNOTRUN - fix screen data base in NOTRUNC 16-Sep-1984 00:26:52
EDT\$\$FIX_NOTRUNC - fix screen data base in NOTR 14-Sep-1984 12:23:09

VAX-11 Bliss-32 V4.0-742
[EDT.SRC]FIXNOTRUN.BLI;1

Page 17
(3)

	AA		54	E9	002DC	46\$:	BLBC	RECORD_DONE, 40\$	:	1150
	28		5A	E8	002DF	47\$:	BLBS	UPDATE_DONE, 52\$	:	1158
	51	04	A3	D0	002E2		MOVL	4(SCRPTR), R1	:	1162
			05	12	002E6		BNEQ	48\$	:	
	5A		01	D0	002E8		MOVL	#1, UPDATE_DONE	:	1164
			17	11	002EB		BRB	51\$	:	
	50		51	D0	002ED	48\$:	MOVL	R1, NEXT_SCRPTR	:	1171
0C	0D	A3	02	E0	002F0		BBS	#2, 13(SCRPTR), 50\$	:	1173
			08	A0	95 002F5		TSTB	8(NEXT_SCRPTR)	:	1174
			05	13	002F8		BEQL	49\$	:	
02	0D	A0	02	E1	002FA		BBC	#2, 13(NEXT_SCRPTR), 50\$	:	1175
			56	D6	002FF	49\$:	INCL	REC_NO	:	1177
			51	D0	00301	50\$:	MOVL	R1, -SCRPTR	:	1179
			5A	E8	00304	51\$:	BLBS	UPDATE_DONE, 52\$	:	1185
			FE3D	31	00307		BRW	23\$	:	
		03	08	AE	E9 0030A	52\$:	BLBC	ANOTHER_PASS, 53\$	:	1188
			FE28	31	0030E		BRW	22\$	:	
			7E	D4	00311	53\$:	CLRL	-(SP)	:	1190
00000000G	00		01	FB	00313		CALLS	#1, EDT\$\$SC_MOVTOLN	:	
			04	0031A			RET		:	1191

; Routine Size: 795 bytes, Routine Base: _EDT\$CODE + 0000

; 622 1192 1
; 623 1193 1 !<BLF/PAGE>

EDT\$FIXNOTRUN
V04-000

EDT\$FIXNOTRUN - Fix screen data base in NOTRUNC 16-Sep-1984 00:26:52
EDT\$FIX_NOTRUNC - fix screen data base in NOTR 14-Sep-1984 12:23:09

VAX-11 Bliss-32 V4.0-742
[EDT.SRC]FIXNOTRUN.BLI;1

Page 18
(4)

: 625 1194 1 END
: 626 1195 1
: 627 1196 0 ELUDOM

! of module EDT\$FIXNOTRUN

PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	795	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	57	15	40	00:00.2
_\$255\$DUA28:[EDT.SRC]PSECTS.L32;1	2	1	50	7	00:00.1

COMMAND QUALIFIERS

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

: Size: 795 code + 0 data bytes
: Run Time: 00:43.2
: Elapsed Time: 00:49.7
: Lines/CPU Min: 1659
: Lexemes/CPU-Min: 9113
: Memory Used: 323 pages
: Compilation Complete

0134 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY