


```

LL               IIIIII
LL               IIIIII
LL               III
LL               III
LL               III
LL               III
LL               III
LL               III
LL               III
LL               III
LL               III
LL               III
LL               III
LL               III
LLLLLLLLLLLLLL  IIIIII
LLLLLLLLLLLLLL  IIIIII

                SSSSSSSS
                SSSSSSSS
                SS
                SS
                SS
                SS
                SSSSSS
                SSSSSS
                SS
                SS
                SS
                SS
                SSSSSSSS
                SSSSSSSS

```



```

0001 0 XTITLE 'EDTSSCRCURS - conditionally position the cursor'
0002 0 MODULE EDTSSCRCURS ( ! Conditionally position the cursor
0003 0 IDENT = 'V04-000' ! File: SCRCURS.BLI Edit: JBS1012
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 puts out a cursor position sequence to a specified
0037 1 line and column, unless the cursor is already there.
0038 1
0039 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0040 1
0041 1 AUTHOR: Bob Kushlis, CREATION DATE: September 8, 1979
0042 1
0043 1 MODIFIED BY:
0044 1
0045 1 1-001 - Original. DJS 12-Feb-1981. This module was created by
0046 1 extracting the routine EDTSSC_POSCSIF from module SCREEN.
0047 1 1-002 - Regularize headers and don't store into the parameter list. This
0048 1 change saved two words on the PDP-11. JBS 13-Mar-1981
0049 1 1-003 - Always position to the last column, to avoid autowrap problems.
0050 1 JBS 15-Apr-1982
0051 1 1-004 - Remove edit 1-003, since it doesn't help a VT100, and add some
0052 1 cursor positioning optimizations. For now, turn off the cursor
0053 1 positioning optimizations, until EDT passes the QA system. JBS 05-Oct-1982
0054 1 1-005 - Turn the cursor positioning optimizations on, to see if EDT still
0055 1 passes the QA system. JBS 15-Oct-1982
0056 1 1-006 - When positioning below the scrolling region don't use the
0057 1 relative cursor motion commands. JBS 15-Oct-1982

```

EDTSSCRCURS
V04-000

EDTSSCRCURS - conditionally position the cursor

G 13

16-Sep-1984 01:31:04
14-Sep-1984 12:24:23

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

Page 2
(1)

```

: 58      0058 1 | 1-007 - Fix a bug involving BS. JBS 17-Oct-1982
: 59      0059 1 | 1-008 - For dumb terminal drivers, always use absolute cursor positioning.
: 60      0060 1 | JBS 17-Oct-1982
: 61      0061 1 | 1-009 - Preserve EDTSSG_FMT_LNPOS. JBS 28-Oct-1982
: 62      0062 1 | 1-010 - Changed test for being outside scrolling region. JBS for SB 17-Jan-1983
: 63      0063 1 | 1-011 - Add more error checking. JBS 17-Jan-1983
: 64      0064 1 | 1-012 - If we are coming from the right margin, always do an absolute cursor
: 65      0065 1 | positioning sequence, since the right margin may not be where we think
: 66      0066 1 | it is, and the cursor may be misplaced. JBS 08-Apr-1983
: 67      0067 1 | --
: 68      0068 1 |
```

EDT
V04

EDTSSRCURS
V04-000

EDTSSRCURS - conditionally position the cursor
Declarations

H 13
16-Sep-1984 01:31:04
14-Sep-1984 12:24:23

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

```

: 70      0069 1 %SBTTL 'Declarations'
: 71      0070 1
: 72      0071 1 | TABLE OF CONTENTS:
: 73      0072 1 |
: 74      0073 1 |
: 75      0074 1 REQUIRE 'EDTSRC:TRAROUNAM';
: 76      0513 1
: 77      0514 1 FORWARD ROUTINE
: 78      0515 1     EDTSSC_POSCSIF : NOVALUE;
: 79      0516 1
: 80      0517 1 |
: 81      0518 1 | INCLUDE FILES:
: 82      0519 1 |
: 83      0520 1
: 84      0521 1 REQUIRE 'EDTSRC:EDTREQ';
: 85      0656 1
: 86      0657 1 |
: 87      0658 1 | MACROS:
: 88      0659 1 |
: 89      0660 1 |     NONE
: 90      0661 1 |
: 91      0662 1 | EQUATED SYMBOLS:
: 92      0663 1 |
: 93      0664 1 |     NONE
: 94      0665 1 |
: 95      0666 1 | OWN STORAGE:
: 96      0667 1 |
: 97      0668 1 |     NONE
: 98      0669 1 |
: 99      0670 1 | EXTERNAL REFERENCES:
: 100     0671 1 |
: 101     0672 1 |     In the routine
```

EDT
V04

```
103 0673 1 %SBTTL 'EDT$$$SC_POSCSIF - conditionally position the cursor'
104 0674 1
105 0675 1 GLOBAL ROUTINE EDT$$$SC_POSCSIF (           ! Conditionally position the cursor
106 0676 1     LINE,                                   ! Desired line
107 0677 1     POS                                     ! Desired column
108 0678 1     ) : NOVALUE =
109 0679 1
110 0680 1 ++
111 0681 1 FUNCTIONAL DESCRIPTION:
112 0682 1
113 0683 1     This routine positions the cursor to [LINE, POS]. No output is
114 0684 1     produced if the cursor is already positioned correctly.
115 0685 1
116 0686 1 FORMAL PARAMETERS:
117 0687 1
118 0688 1     LINE                                     Desired line
119 0689 1
120 0690 1     POS                                     Desired column
121 0691 1
122 0692 1 IMPLICIT INPUTS:
123 0693 1
124 0694 1     EDT$$G-PRV-LN
125 0695 1     EDT$$G-PRV-COL
126 0696 1     EDT$$G-BOT-SCREG
127 0697 1     EDT$$G-TI-DUMB
128 0698 1     EDT$$G-TI-WID
129 0699 1
130 0700 1 IMPLICIT OUTPUTS:
131 0701 1
132 0702 1     NONE
133 0703 1
134 0704 1 ROUTINE VALUE:
135 0705 1
136 0706 1     NONE
137 0707 1
138 0708 1 SIDE EFFECTS:
139 0709 1
140 0710 1     May position the cursor.
141 0711 1
142 0712 1 --
143 0713 1
144 0714 2 BEGIN
145 0715 2
146 0716 2 EXTERNAL ROUTINE
147 0717 2     EDT$$$SC_POSABS : NOVALUE;           ! Force a cursor position sequence to the output buffer
148 0718 2     EDT$$$STORE_FMTCH : NOVALUE;         ! Store a character in the format buffer
149 0719 2     EDT$$FMT_DCML : NOVALUE;             ! Place a decimal number in the format buffer
150 0720 2
151 0721 2 EXTERNAL
152 0722 2     EDT$$G-BOT-SCREG,                       ! Bottom of current scrolling region
153 0723 2     EDT$$G-PRV-LN,                         ! Previous line number
154 0724 2     EDT$$G-PRV-COL,                       ! Previous column number
155 0725 2     EDT$$G-TI-TYP,                       ! Terminal type, either VT52 or VT100
156 0726 2     EDT$$G-TI-DUMB,                       ! 1 = this is a dumb terminal driver
157 0727 2     EDT$$G-FMT_LNPOS,                     ! Cursor position, for formatting purposes.
158 0728 2     EDT$$G-TI-WID;                         ! Terminal width
159 0729 2
```



```
160 0730 2 LOCAL
161 0731 SAVE_FMTLNPOS;
162 0732
163 0733 ASSERT (.LINE GEQ 0);
164 0734
165 0735 + If we are already positioned correctly, do nothing.
166 0736 -
167 0737
168 0738 IF ((.LINE EQL .EDT$$G_PRV_LN) AND (.POS EQL .EDT$$G_PRV_COL)) THEN RETURN;
169 0739
170 0740 +
171 0741 Preserve EDT$$G_FMT_LNPOS, since the calls to EDT$$FMT_DCML will increment it,
172 0742 but those characters do not in fact move the cursor, since they are in a control sequence.
173 0743 -
174 0744 SAVE_FMTLNPOS = .EDT$$G_FMT_LNPOS;
175 0745 +
176 0746 If this is a dumb terminal driver, we are positioning below the scrolling
177 0747 region, or coming from the right margin, issue an absolute cursor position.
178 0748 A dumb terminal driver is one which does not permit its autowrap logic to be disabled.
179 0749 -
180 0750
181 0751 IF (((.LINE + 1) GEQ .EDT$$G_BOT_SCREG) OR (.EDT$$G_PRV_COL EQL (.EDT$$G_TI_WID - 1)) OR .EDT$$G_TI_DUMB
182 0752 THEN
183 0753 BEGIN
184 0754 EDTSSC_POSABS (.LINE, .POS);
185 0755 EDT$$G_FMT_LNPOS = .SAVE_FMTLNPOS;
186 0756 RETURN;
187 0757 END;
188 0758
189 0759 +
190 0760 If the desired column is zero, but the current column is non-zero, issue a
191 0761 carriage return provided that the rows match, so that we will not be issuing
192 0762 an absolute cursor position later.
193 0763 -
194 0764
195 0765 IF ((.POS EQL 0) AND (.EDT$$G_PRV_COL NEQ 0) AND
196 0766 ((.LINE EQL .EDT$$G_PRV_LN) OR (.LINE EQL (.EDT$$G_PRV_LN + 1))))
197 0767 THEN
198 0768 BEGIN
199 0769 EDT$$STORE_FMTCH (ASC_K_CR);
200 0770 EDT$$G_PRV_COL = 0;
201 0771 END;
202 0772
203 0773 +
204 0774 If the column is correct but the row is incorrect, issue an appropriate
205 0775 sequence.
206 0776 -
207 0777
208 0778 IF ((.POS EQL .EDT$$G_PRV_COL) AND (.LINE NEQ .EDT$$G_PRV_LN))
209 0779 THEN
210 0780 BEGIN
211 0781 +
212 0782 If we are moving down one, issue a line feed.
213 0783 -
214 0784
215 0785 IF (.LINE EQL (.EDT$$G_PRV_LN + 1))
216 0786 THEN
```

EDT\$SCRCURS
V04-000

K 13
EDT\$SCRCURS - conditionally position the cursor 16-Sep-1984 01:31:04
EDT\$SC_POSCSIF - conditionally position the c 14-Sep-1984 12:24:23

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

```

217 0787 4 BEGIN
218 0788 4 EDT$$STORE_FMTCH (ASC_K_LF);
219 0789 4 EDT$$G_PRV_LN = .EDT$$G_PRV_LN + 1;
220 0790 4 END
221 0791 3 ELSE
222 0792 3 +
223 0793 3 Otherwise issue an appropriate relative cursor motion command. Note that the VT52
224 0794 3 can only move by one.
225 0795 3 -
226 0796 3
227 0797 4 IF ((.EDT$$G_TI_TYP EQL TERM_VT100) OR (ABS (.LINE - .EDT$$G_PRV_LN) EQL 1))
228 0798 3 THEN
229 0799 4 BEGIN
230 0800 4 EDT$$STORE_FMTCH (ASC_K_ESC);
231 0801 4
232 0802 4 IF (.EDT$$G_TI_TYP EQL TERM_VT100) THEN EDT$$STORE_FMTCH (%C'L');
233 0803 4
234 0804 5 IF (.LINE LSS .EDT$$G_PRV_LN)
235 0805 4 THEN
236 0806 5 BEGIN
237 0807 5
238 0808 5 IF ((.EDT$$G_PRV_LN - .LINE) NEQ 1) THEN EDT$$FMT_DCML (.EDT$$G_PRV_LN - .LINE);
239 0809 5
240 0810 5 EDT$$STORE_FMTCH (%C'A');
241 0811 5 EDT$$G_PRV_LN = .LINE;
242 0812 5 EDT$$G_PRV_COL = .POS;
243 0813 5 END
244 0814 4 ELSE
245 0815 5 BEGIN
246 0816 5 EDT$$FMT_DCML (.LINE - .EDT$$G_PRV_LN);
247 0817 5 EDT$$STORE_FMTCH (%C'B');
248 0818 5 EDT$$G_PRV_LN = .LINE;
249 0819 5 EDT$$G_PRV_COL = .POS;
250 0820 4 END;
251 0821 4
252 0822 3 END;
253 0823 3
254 0824 2 END;
255 0825 2
256 0826 2 +
257 0827 2 Now check for moving left or right. Note again that the VT52 can only
258 0828 2 move by one.
259 0829 2 -
260 0830 2
261 0831 3 IF ((.POS NEQ .EDT$$G_PRV_COL) AND (.LINE EQL .EDT$$G_PRV_LN))
262 0832 2 THEN
263 0833 3 BEGIN
264 0834 3 +
265 0835 3 If we are just going back one, use BS.
266 0836 3 -
267 0837 3
268 0838 4 IF (.POS EQL (.EDT$$G_PRV_COL - 1))
269 0839 3 THEN
270 0840 4 BEGIN
271 0841 4 EDT$$STORE_FMTCH (ASC_K_BS);
272 0842 4 EDT$$G_PRV_COL = .EDT$$G_PRV_COL - 1;
273 0843 4 END
```



```
274 0844 3 ELSE
275 0845 3
276 0846 4 IF ((.EDT$$G_TI_TYP EQL TERM_VT100) OR (ABS (.POS - .EDT$$G_PRV_COL) EQL 1))
277 0847 3 THEN
278 0848 4 BEGIN
279 0849 4 !+
280 0850 4 Otherwise, use a relative cursor motion command.
281 0851 4 !-
282 0852 4 EDT$$STORE_FMTCH (ASC_K_ESC);
283 0853 4
284 0854 4 IF (.EDT$$G_TI_TYP EQL TERM_VT100) THEN EDT$$STORE_FMTCH (%C'L');
285 0855 4
286 0856 5 IF (.POS LSS .EDT$$G_PRV_COL)
287 0857 4 THEN
288 0858 5 BEGIN
289 0859 5 EDT$$FMT_DCML (.EDT$$G_PRV_COL - .POS);
290 0860 5 EDT$$STORE_FMTCH (%C'D');
291 0861 5 EDT$$G_PRV_COL = .POS;
292 0862 5 END
293 0863 4 ELSE
294 0864 5 BEGIN
295 0865 5
296 0866 5 IF ((.POS - .EDT$$G_PRV_COL) NEQ 1) THEN EDT$$FMT_DCML (.POS - .EDT$$G_PRV_COL);
297 0867 5
298 0868 5 EDT$$STORE_FMTCH (%C'C');
299 0869 5 EDT$$G_PRV_COL = .POS;
300 0870 4 END;
301 0871 4
302 0872 3 END;
303 0873 3
304 0874 2 END;
305 0875 2
306 0876 2 !+
307 0877 2 If the above logic does not succeed in getting the cursor positioned correctly,
308 0878 2 do it using an absolute cursor position sequence.
309 0879 2 !-
310 0880 2
311 0881 3 IF ((.LINE NEQ .EDT$$G_PRV_LN) OR (.POS NEQ .EDT$$G_PRV_COL)) !
312 0882 2 THEN
313 0883 2 EDTSSSC_POSABS (.LINE, .POS);
314 0884 2
315 0885 2 !+
316 0886 2 Restore the format line position.
317 0887 2 !-
318 0888 2 EDT$$G_FMT_LNPOS = .SAVE_FMTLNPOS;
319 0889 2 RETURN;
320 0890 1 END;
```

! of routine EDTSSSC_POSCSIF

.TITLE EDTSSCRCURS EDTSSCRCURS - conditionally positio
n the cursor

.IDENT \V04-000\

.EXTRN EDTSSSC_POSABS, EDT\$\$STORE_FMTCH
.EXTRN EDT\$\$FMT_DCML, EDT\$\$G_BOT_SCREG
.EXTRN EDT\$\$G_PRV_LN, EDT\$\$G_PRV_COL
.EXTRN EDT\$\$G_TI_TYP, EDT\$\$G_TI_DUMB

		OFFC	00000				
5B	00000000G	00	9E	00002	MOVAB	EDTSSG_FMT_LNPOS	
5A	00000000G	00	9E	00009	MOVAB	EDTSSG_TI_WID, EDTSSINTER_ERR	
59	00000000G	00	9E	00010	MOVAB		
58	00000000G	00	9E	00017	MOVAB	EDTSSSTORE_FMTCH, R9	
57	00000000G	00	9E	0001E	MOVAB	EDTSSG_PRV_LN, R8	
54	04	AC	D0	00025	MOVAB	EDTSSG_PRV_COL, R7	
		07	18	00029	MOVL	LINE, R4	0733
00000000G	00	00	FB	0002B	BGEQ	1\$	
68		54	D1	00032	CALLS	#0, EDTSSINTER_ERR	
		07	12	00035	CMPL	R4, EDTSSG_PRV_LN	0738
67	08	AC	D1	00037	BNEQ	2\$	
		01	12	0003B	CMPL	POS, EDTSSG_PRV_COL	
		04	00	0003D	BNEQ	2\$	
56	00000000G	00	D0	0003E	RET		
55	01	A4	9E	00045	MOVL	EDTSSG_FMT_LNPOS, SAVE_FMTLNPOS	0744
00000000G	00	55	D1	00049	MOVAB	1(R4), R5	0751
		14	18	00050	CMPL	R5, EDTSSG_BOT_SCREG	
50 00000000G	00	01	C3	00052	BGEQ	3\$	
	50	67	D1	0005A	SUBL3	#1, EDTSSG_TI_WID, R0	
		07	13	0005D	CMPL	EDTSSG_PRV_COL, R0	
06 00000000G	00	E9	00	0005F	BEQL	3\$	
	08	AC	DD	00066	BLBC	EDTSSG_TI_DUMB, 4\$	
		01	0F	31	PUSHL	POS	0754
53	08	AC	D0	0006C	BRW	24\$	
		1A	12	00070	MOVL	POS, R3	0765
		67	D5	00072	BNEQ	6\$	
		16	13	00074	TSTL	EDTSSG_PRV_COL	
50		68	D0	00076	BEQL	6\$	
50		54	D1	00079	MOVL	EDTSSG_PRV_LN, R0	0766
		07	13	0007C	CMPL	R4, R0	
		50	D6	0007E	BEQL	5\$	
50		54	D1	00080	INCL	R0	
		07	12	00083	CMPL	R4, R0	
		0D	DD	00085	BNEQ	6\$	
69		01	FB	00087	PUSHL	#13	0769
		67	D4	0008A	CALLS	#1, EDTSSSTORE_FMTCH	
67		53	D1	0008C	CLRL	EDTSSG_PRV_COL	0770
		6C	12	0008F	CMPL	R3, EDTSSG_PRV_COL	0778
68		54	D1	00091	BNEQ	14\$	
		67	13	00094	CMPL	R4, EDTSSG_PRV_LN	
52		68	D0	00096	BEQL	14\$	
50	01	A2	9E	00099	MOVL	EDTSSG_PRV_LN, R2	0785
50		54	D1	0009D	MOVAB	1(R2), R0	
		09	12	000A0	CMPL	R4, R0	
		0A	DD	000A2	BNEQ	7\$	
69		01	FB	000A4	PUSHL	#10	0788
		68	D6	000A7	CALLS	#1, EDTSSSTORE_FMTCH	
		52	11	000A9	INCL	EDTSSG_PRV_LN	0789
02		6A	D1	000AB	BRB	14\$	0785
		0E	13	000AE	CMPL	EDTSSG_TI_TYP, #2	0797
					BEQL	9\$	

50	54	52	C3	000B0	SUBL3	R2, R4, R0		
		03	18	000B4	BGEQ	8\$		
	50	50	CE	000B6	MNEGL	R0, R0		
	01	50	D1	000B9	CMPL	R0, #1		
		3F	12	000BC	BNEQ	14\$		
		1B	DD	000BE	PUSHL	#27		0800
	69	01	FB	000C0	CALLS	#1, EDT\$\$\$STORE_FMTCH		
	02	6A	D1	000C3	CMPL	EDT\$G_TI_TYP, #2		0802
		07	12	000C6	BNEQ	10\$		
	7E	5B	8F	9A 000C8	MOVZBL	#91, -(SP)		
	69	01	FB	000CC	CALLS	#1, EDT\$\$\$STORE_FMTCH		
	50	68	D0	000CF	MOVL	EDT\$G_PRV_LN, R0		0804
	50	54	D1	000D2	CMPL	R4, R0		
		12	18	000D5	BGEQ	12\$		
	55	50	D1	000D7	CMPL	R0, R5		0808
		07	13	000DA	BEQL	11\$		
7E	50	54	C3	000DC	SUBL3	R4, R0, -(SP)		
	6B	01	FB	000E0	CALLS	#1, EDT\$FMT_DCML		
	7E	41	8F	9A 000E3	MOVZBL	#65, -(SP)		0810
		0B	11	000E7	BRB	13\$		
7E	54	50	C3	000E9	SUBL3	R0, R4, -(SP)		0816
	6B	01	FB	000ED	CALLS	#1, EDT\$FMT_DCML		
	7E	42	8F	9A 000F0	MOVZBL	#66, -(SP)		0817
	69	01	FB	000F4	CALLS	#1, EDT\$\$\$STORE_FMTCH		
	68	54	D0	000F7	MOVL	R4, EDT\$G_PRV_LN		0811
	67	53	D0	000FA	MOVL	R3, EDT\$G_PRV_COL		0812
	55	67	D0	000FD	MOVL	EDT\$G_PRV_COL, R5		0831
	55	53	D1	00100	CMPL	R3, R5		
		6A	13	00103	BEQL	22\$		
	68	54	D1	00105	CMPL	R4, EDT\$G_PRV_LN		
		65	12	00108	BNEQ	22\$		
	50	FF	A5	9E 0010A	MOVAB	-1(R5), R0		0838
	50	53	D1	0010E	CMPL	R3, R0		
		09	12	00111	BNEQ	15\$		
		08	DD	00113	PUSHL	#8		0841
	69	01	FB	00115	CALLS	#1, EDT\$\$\$STORE_FMTCH		
		67	D7	00118	DECL	EDT\$G_PRV_COL		0842
		53	11	0011A	BRB	22\$		0838
	02	6A	D1	0011C	CMPL	EDT\$G_TI_TYP, #2		0846
		0E	13	0011F	BEQL	17\$		
50	53	55	C3	00121	SUBL3	R5, R3, R0		
		03	18	00125	BGEQ	16\$		
	50	50	CE	00127	MNEGL	R0, R0		
	01	50	D1	0012A	CMPL	R0, #1		
		40	12	0012D	BNEQ	22\$		
		1B	DD	0012F	PUSHL	#27		0852
	69	01	FB	00131	CALLS	#1, EDT\$\$\$STORE_FMTCH		
	02	6A	D1	00134	CMPL	EDT\$G_TI_TYP, #2		0854
		07	12	00137	BNEQ	18\$		
	7E	5B	8F	9A 00139	MOVZBL	#91, -(SP)		
	69	01	FB	0013D	CALLS	#1, EDT\$\$\$STORE_FMTCH		
	55	67	D0	00140	MOVL	EDT\$G_PRV_COL, R5		0856
	55	53	D1	00143	CMPL	R3, R5		
		0D	18	00146	BGEQ	19\$		
7E	55	53	C3	00148	SUBL3	R3, R5, -(SP)		0859
	6B	01	FB	0014C	CALLS	#1, EDT\$FMT_DCML		
	7E	44	8F	9A 0014F	MOVZBL	#68, -(SP)		0860

EDT\$SCRCURS
V04-000

EDT\$SCRCURS - conditionally position the cursor
EDT\$\$\$SC_POSCSIF - conditionally position the c

B 14

16-Sep-1984 01:31:04
14-Sep-1984 12:24:23

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

Page 10
(3)

	50	01	14	11	00153	BRB	21\$	:	
	50		A5	9E	00155	MOVAB	1(R5), R0	:	0866
			53	D1	00159	CMPL	R3, R0	:	
			07	13	0015C	BEQL	20\$	:	
7E	53		55	C3	0015E	SUBL3	R5, R3, -(SP)	:	
6B			01	FB	00162	CALLS	#1, EDT\$FMT_DCML	:	
7E		43	8F	9A	00165	MOVZBL	#67, -(SP)	:	0868
69			01	FB	00169	CALLS	#1, EDT\$STORE_FMTCH	:	
67			53	D0	0016C	MOVL	R3, EDT\$G_PRV_COL	:	0861
68			54	D1	0016F	CMPL	R4, EDT\$G_PRV_LN	:	0881
			05	12	00172	BNEQ	23\$	:	
67			53	D1	00174	CMPL	R3, EDT\$G_PRV_COL	:	
			0B	13	00177	BEQL	25\$	:	
			53	DD	00179	PUSHL	R3	:	0883
			54	DD	0017B	PUSHL	R4	:	
00000000G	00		02	FB	0017D	CALLS	#2, EDT\$\$\$SC_POSABS	:	
00000000G	00		56	D0	00184	MOVL	SAVE_FMTLNPOS, EDT\$G_FMT_LNPOS	:	0888
			04	0018B	RET			:	0890

; Routine Size: 396 bytes, Routine Base: _EDT\$CODE + 0000

: 321 0891 1
: 322 0892 1 !<BLF/PAGE>

: 324
: 325
: 326

0893 1 END
0894 1
0895 0 ELUDOM

! of module EDT\$SCRCURS

PSECT SUMMARY		
Name	Bytes	Attributes
_EDT\$CODE	396	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	6	1	40	00:00.2
_\$255\$DUA28:[EDT.SRC]PSECTS.L32;1	2	1	50	7	00:00.1

COMMAND QUALIFIERS

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

Size: 396 code + 0 data bytes

Run Time: 00:20.2

Elapsed Time: 00:23.6

Lines/CPU Min: 2653

Lexemes/CPU-Min: 7929

Memory Used: 138 pages

Compilation Complete

0138

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY