


```
SSSSSSSS  CCCCCCCC  RRRRRRRR  CCCCCCCC  000000  MM  MM  CCCCCCCC  UU  UU  RRRRRRRR
SSSSSSSS  CCCCCCCC  RRRRRRRR  CCCCCCCC  000000  MM  MM  CCCCCCCC  UU  UU  RRRRRRRR
SS  SS  CC  CC  RR  RR  CC  CC  00  00  MMMM  MMMM  CC  CC  UU  UU  RR  RR
SS  SS  CC  CC  RR  RR  CC  CC  00  00  MMMM  MMMM  CC  CC  UU  UU  RR  RR
SS  SS  CC  CC  RR  RR  CC  CC  00  00  MM  MM  CC  CC  UU  UU  RR  RR
SSSSSS  SS  CC  CC  RRRRRRRR  CCCCCCCC  00  00  MM  MM  CC  CC  UU  UU  RRRRRRRR
SSSSSS  SS  CC  CC  RRRRRRRR  CCCCCCCC  00  00  MM  MM  CC  CC  UU  UU  RRRRRRRR
SS  SS  CC  CC  RR  RR  CC  CC  00  00  MM  MM  CC  CC  UU  UU  RR  RR
SS  SS  CC  CC  RR  RR  CC  CC  00  00  MM  MM  CC  CC  UU  UU  RR  RR
SSSSSSSS  CCCCCCCC  RR  RR  CCCCCCCC  000000  MM  MM  CCCCCCCC  UUUUUUUUUU  RR  RR
SSSSSSSS  CCCCCCCC  RR  RR  CCCCCCCC  000000  MM  MM  CCCCCCCC  UUUUUUUUUU  RR  RR

LL  LL  I I I I I  SSSSSSSS
LL  LL  I I I I I  SSSSSSSS
LL  LL  I  I  SS
LL  LL  I  I  SS
LL  LL  I  I  SS
LL  LL  I  I  SSSSSS
LL  LL  I  I  SSSSSS
LL  LL  I  I  SS
LL  LL  I  I  SS
LL  LL  I  I  SS
LLLLLLLLLLLL  I I I I I  SSSSSSSS
LLLLLLLLLLLL  I I I I I  SSSSSSSS
```

```
0001 0 %TITLE 'EDT$SCRCOMCUR - compute cursor position'
0002 0 MODULE EDT$SCRCOMCUR ( ! Compute cursor position
0003 0 IDENT = 'V04-000' ! File: SCRCOMCUR.BLI Edit: JBS1010
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 computes the current cursor position.
0037 1
0038 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0039 1
0040 1 AUTHOR: Bob Kushlis, CREATION DATE: September 8, 1979
0041 1
0042 1 MODIFIED BY:
0043 1
0044 1 1-001 - Original. DJS 12-Feb-1981. This module was created by
0045 1 extracting the routine EDT$SC CPU$CPOS from module SCREEN.
0046 1 1-002 - Regularize headers. JBS 13-Mar-1981
0047 1 1-003 - Rewrite for new screen logic. JBS 12-Oct-1982
0048 1 1-004 - Fix a couple of minor bugs. JBS 13-Oct-1982
0049 1 1-005 - Fix call to EDT$FMT_CHWID. JBS 13-Oct-1982
0050 1 1-006 - Fix problem with SHL. JBS 27-Oct-1982
0051 1 1-007 - Fix the cursor position in NOTRUNCATE mode. JBS 09-Nov-1982
0052 1 1-008 - Fix the cursor positioning again. JBS 10-Nov-1982
0053 1 1-009 - Change the handling of EDT$SG SHF. JBS 14-Dec-1982
0054 1 1-010 - Correct tab at front of continued line. JBS 15-Dec-1982
0055 1
0056 1
```

EDTSSRCOMCUR
V04-000

EDTSSRCOMCUR - compute cursor position
Declarations

L 12
16-Sep-1984 01:30:17
14-Sep-1984 12:24:22

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

Page 2
(2)

```
: 58      0057 1 %SBTTL 'Declarations'
: 59      0058 1
: 60      0059 1 | TABLE OF CONTENTS:
: 61      0060 1 |
: 62      0061 1 |
: 63      0062 1 REQUIRE 'EDTSRC:TRAROUNAM';
: 64      0501 1
: 65      0502 1 FORWARD ROUTINE
: 66      0503 1     EDTSSC_CPUCSPOS : NOVALUE;
: 67      0504 1
: 68      0505 1 |
: 69      0506 1 | INCLUDE FILES:
: 70      0507 1 |
: 71      0508 1
: 72      0509 1 REQUIRE 'EDTSRC:EDTREQ';
: 73      0644 1
: 74      0645 1 |
: 75      0646 1 | MACROS:
: 76      0647 1 |
: 77      0648 1 |     NONE
: 78      0649 1 |
: 79      0650 1 | EQUATED SYMBOLS:
: 80      0651 1 |
: 81      0652 1 |     NONE
: 82      0653 1 |
: 83      0654 1 | OWN STORAGE:
: 84      0655 1 |
: 85      0656 1 |     NONE
: 86      0657 1 |
: 87      0658 1 | EXTERNAL REFERENCES:
: 88      0659 1 |
: 89      0660 1 |     In the routine
```

```

: 91      0661 1 %SBTTL 'EDTSSC_CPUCSPOS - compute cursor position'
: 92      0662 1
: 93      0663 1 GLOBAL ROUTINE EDTSSC_CPUCSPOS (
: 94      0664 1     LINE,
: 95      0665 1     COLUMN
: 96      0666 1     ) : NOVALUE =
: 97      0667 1
: 98      0668 1
: 99      0669 1 ++
100      0670 1 FUNCTIONAL DESCRIPTION:
101      0671 1     This routine computes the current cursor position returning the line
102      0672 1     and column numbers in the ref parameters LINE and COLUMN.
103      0673 1
104      0674 1 FORMAL PARAMETERS:
105      0675 1
106      0676 1     LINE                Cursor's relative line number
107      0677 1
108      0678 1     COLUMN                Cursor's column number
109      0679 1
110      0680 1 IMPLICIT INPUTS:
111      0681 1
112      0682 1     EDTSSG_SHF
113      0683 1     EDTSSG_TI_WID
114      0684 1     EDTSSG_TRON
115      0685 1     EDTST_LN_BUF
116      0686 1     EDTSSA_LN_PTR
117      0687 1     EDTSSA_LN_END
118      0688 1
119      0689 1 IMPLICIT OUTPUTS:
120      0690 1
121      0691 1     NONE
122      0692 1
123      0693 1 ROUTINE VALUE:
124      0694 1
125      0695 1     NONE
126      0696 1
127      0697 1 SIDE EFFECTS:
128      0698 1
129      0699 1     NONE
130      0700 1
131      0701 1 --
132      0702 1
133      0703 2 BEGIN
134      0704 2
135      0705 2 EXTERNAL ROUTINE
136      0706 2     EDTSSFMT_CHWID;
137      0707 2
138      0708 2 EXTERNAL
139      0709 2     EDTSSG_SHF,
140      0710 2     EDTSSG_TI_WID,
141      0711 2     EDTSSG_TRON,
142      0712 2     EDTST_LN_BUF,
143      0713 2     EDTSSA_LN_PTR,
144      0714 2     EDTSSA_LN_END;
145      0715 2
146      0716 2 LOCAL
147      0717 2     CP,

```

! Compute cursor position
! Where to return line number
! Where to return column number

! Compute the width of a character

! The number of columns shifted.
! Width of terminal line.
! Truncate or wrap long lines.
! Current line buffer.
! Current character pointer.
! End of line

! Character pointer into the current record

```
148 0718 2 LIN,
149 0719 2 COL,
150 0720 2 CHAR,
151 0721 2 CHAR_WID,
152 0722 2 LINE_DONE;
153 0723 2
154 0724 2 LIN = 0;
155 0725 2 COL = 0;
156 0726 2 CP = EDT$T_LN_BUF;
157 0727 2 LINE_DONE = 0;
158 0728 2
159 0729 2 WHILE ((.CP LSSA .EDT$A_LN_PTR) AND ( NOT .LINE_DONE)) DO
160 0730 2 BEGIN
161 0731 2 CHAR = CH$RCHAR A (CP);
162 0732 2 CHAR_WID = EDT$FMT_CHWID (.CHAR, .COL);
163 0733 2
164 0734 2 IF ((.COL + .CHAR_WID) LEQ (.EDT$G_TI_WID + .EDT$G_SHF))
165 0735 2 THEN
166 0736 2 +
167 0737 2 | The character fits on this line, count it and go on to the next.
168 0738 2 | -
169 0739 2 | COL = .COL + .CHAR_WID
170 0740 2 | ELSE
171 0741 2 | +
172 0742 2 | The character does not fit on this line.
173 0743 2 | -
174 0744 2 |
175 0745 2 | IF .EDT$G_TRUN
176 0746 2 | THEN
177 0747 2 | BEGIN
178 0748 2 | +
179 0749 2 | In TRUNCATE mode, just position to the last column and terminate.
180 0750 2 | -
181 0751 2 | COL = .EDT$G_TI_WID + .EDT$G_SHF - 1;
182 0752 2 | LINE_DONE = 1;
183 0753 2 | END
184 0754 2 | ELSE
185 0755 2 | BEGIN
186 0756 2 | +
187 0757 2 | In NOTRUNCATE mode, try fitting it on the next line. Don't produce too many lines.
188 0758 2 | -
189 0759 2 | LIN = .LIN + 1;
190 0760 2 | COL = .EDT$G_SHF + 2;
191 0761 2 | +
192 0762 2 | We can't use .CHAR_WID in the next statement because the width of a tab may be
193 0763 2 | different on the new line since it is in a new position.
194 0764 2 | -
195 0765 2 |
196 0766 2 | IF (.LIN GEQ 255) THEN LINE_DONE = 1 ELSE COL = .COL + EDT$FMT_CHWID (.CHAR, .COL);
197 0767 2 |
198 0768 2 | END;
199 0769 2 |
200 0770 2 | END;
201 0771 2 |
202 0772 2 | +
203 0773 2 | In NOTRUNCATE mode, make sure the current character will fit on this line. If it will not,
204 0774 2 | move the cursor to the beginning of the next line.
```

```
205 0775 2 !-
206 0776 2
207 0777 2
208 0778 2
209 0779 2
210 0780 2
211 0781 2
212 0782 2
213 0783 2
214 0784 2
215 0785 2
216 0786 2
217 0787 2
218 0788 2
219 0789 2
220 0790 2
221 0791 2
222 0792 2
223 0793 2
224 0794 1

IF (( NOT .EDT$G_TRUN) AND ( NOT .LINE_DONE) AND (.EDT$A_LN_PTR NEQA .EDT$A_LN_END))
THEN
  BEGIN
    CHAR = CH$RCHAR A (CP);
    CHAR_WID = EDT$FMT_CHWID (.CHAR, .COL);
    IF ((.COL + .CHAR_WID) GTR (.EDT$G_TI_WID + .EDT$G_SHF))
    THEN
      BEGIN
        LIN = .LIN + 1;
        COL = .EDT$G_SHF + 2;
      END;
    END;
    .LINE = .LIN;
    .COLUMN = MAX (0, MIN (.COL - .EDT$G_SHF, .EDT$G_TI_WID - 1));
  END;
  ! of routine EDT$SSC_CPUCSPOS
```

.TITLE EDT\$SCRCOMCUR EDT\$SCRCOMCUR - compute cursor po
sition

.IDENT \V04-000\

.EXTRN EDT\$FMT_CHWID, EDT\$G_SHF
.EXTRN EDT\$G_TI_WID, EDT\$G_TRUN
.EXTRN EDT\$T_LN_BUF, EDT\$A_LN_PTR
.EXTRN EDT\$A_LN_END

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

.ENTRY EDT\$SSC_CPUCSPOS, Save R2,R3,R4,R5,R6,R7,- : 0663
R8,R9,R10,R11

```
MOVAB EDT$G_TI_WID, R11
MOVAB EDT$G_SHF, R10
MOVAB EDT$FMT_CHWID, R9
CLRL LIN
CLRL COL
MOVAB EDT$T_LN_BUF, CP
CLRL LINE_DONE
CMPL CP, EDT$A_LN_PTR
BGEQU 6$
BLBS LINE_DONE, 6$
MOVZBL (CP)+, CHAR
PUSHL COL
PUSHL CHAR
CALLS #2, EDT$FMT_CHWID
MOVL R0, CHAR_WID
ADDL3 CHAR_WID, COL, R3
MOVL EDT$G_SHF, R1
ADDL3 R1, EDT$G_TI_WID, R0
CMPL R3, R0
BCTR 2$
ADDL2 CHAR_WID, COL
BRB 1$
```

0724
0725
0726
0727
0729
0731
0732
0734
0739

OFFC 00000

```
5B 00000000G 00 9E 00002
5A 00000000G 00 9E 00009
59 00000000G 00 9E 00010
58 D4 00017
52 D4 00019
55 00000000G 00 9E 0001B
54 D4 00022
00000000G 00 55 D1 00024 1$:
52 1E 0002B
4F 54 E8 0002D
57 85 9A 00030
52 DD 00033
57 DD 00035
69 02 FB 00037
56 50 D0 0003A
53 52 C1 0003D
51 6A D0 00041
50 6B 51 C1 00044
50 53 D1 00048
52 05 14 0004B
56 C0 0004D
D2 11 00050
```

53

50

	06	00000000G	00	E9	00052	2\$:	BLBC	EDTSSG_TRUN, 3\$	:	0745
	52	FF	A0	9E	00059		MOVAB	-1(R0), COL	:	0751
			0F	11	0005D		BRB	4\$	:	0752
			58	D6	0005F	3\$:	INCL	LIN	:	0759
	52	02	A1	9E	00061		MOVAB	2(R1), COL	:	0760
000000FF	8F		58	D1	00065		CMPL	LIN, #255	:	0766
			05	19	0006C		BLSS	5\$	:	
	54		01	D0	0006E	4\$:	MOVL	#1, LINE_DONE	:	
			B1	11	00071		BRB	1\$	:	
			52	DD	00073	5\$:	PUSHL	COL	:	
			57	DD	00075		PUSHL	CHAR	:	
	69		02	FB	00077		CALLS	#2, EDTSSFMT_CHWID	:	
	52		50	C0	0007A		ADDL2	R0, COL	:	
			A5	11	0007D		BRB	1\$	:	0729
	33	00000000G	00	E8	0007F	6\$:	BLBS	EDTSSG_TRUN, 7\$	:	0777
	30		54	E8	00086		BLBS	LINE_DONE, 7\$	:	
00000000G	00	00000000G	00	D1	00089		CMPL	EDTSSA_LN_PTR, EDTSSA_LN_END	:	
			23	13	00094		BEQL	7\$	:	
	57		85	9A	00096		MOVZBL	(CP)+, CHAR	:	0780
			52	DD	00099		PUSHL	COL	:	0781
			57	DD	0009B		PUSHL	CHAR	:	
	69		02	FB	0009D		CALLS	#2, EDTSSFMT_CHWID	:	
	56		50	D0	000A0		MOVL	R0, CHAR_WID	:	
53	52		56	C1	000A3		ADDL3	CHAR_WID, COL, R3	:	0783
	51		6A	D0	000A7		MOVL	EDTSSG_SHF, R1	:	
50	6B		51	C1	000AA		ADDL3	R1, EDTSSG_TI_WID, R0	:	
	50		53	D1	000AE		CMPL	R3, R0	:	
			06	15	000B1		BLEQ	7\$	:	
			58	D6	000B3		INCL	LIN	:	0786
	52	02	A1	9E	000B5		MOVAB	2(R1), COL	:	0787
	04		58	D0	000B9	7\$:	MOVL	LIN, @LINE	:	0792
	52		6A	C2	000BD		SUBL2	EDTSSG_SHF, R2	:	0793
50	6B		01	C3	000C0		SUBL3	#1, EDTSSG_TI_WID, R0	:	
	50		52	D1	000C4		CMPL	R2, R0	:	
			03	15	000C7		BLEQ	8\$	:	
	52		50	D0	000C9		MOVL	R0, R2	:	
			52	D5	000CC	8\$:	TSTL	R2	:	
			02	18	000CE		BGEQ	9\$	:	
			52	D4	000D0		CLRL	R2	:	
08	BC		52	D0	000D2	9\$:	MOVL	R2, @COLUMN	:	0794
			04	00	000D6		RET		:	

; Routine Size: 215 bytes, Routine Base: _EDT\$CODE + 0000

; 225 0795 1
; 226 0796 1 !<BLF/PAGE>

EDT\$SCRCOMCUR
V04-000

EDT\$SCRCOMCUR - compute cursor position
EDT\$SSC_CPUCSPOS - compute cursor position

D 13
16-Sep-1984 01:30:17
14-Sep-1984 12:24:22

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

Page 7
(4)

: 228 0797 1 END
: 229 0798 1
: 230 0799 0 ELUDOM

! of module EDT\$SCRCOMCUR

PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	215	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	0	0	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\$:SCRCOMCUR/OBJ=OBJ\$:SCRCOMCUR MSRC\$:SCRCOMCUR.BLI/UPDATE=(ENH\$:S
RCOMCUR)

: Size: 215 code + 0 data bytes
: Run Time: 00:14.6
: Elapsed Time: 00:31.0
: Lines/CPU Min: 3274
: Lexemes/CPU-Min: 9127
: Memory Used: 95 pages
: Compilation Complete

0138

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY