```
EEEEEEEEEEEEEEE  DDDDDDDDDDDD     TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD     TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD     TTTTTTTTTTTTTTT
EEE              DDD         DDD        TTT
EEE              DDD         DDD        TTT
EEE              DDD         DDD        TTT
EEE              DDD         DDD        TTT
EEE              DDD         DDD        TTT
EEE              DDD         DDD        TTT
EEEEEEEEEEEE     DDD         DDD        TTT
EEEEEEEEEEEE     DDD         DDD        TTT
EEEEEEEEEEEE     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
EEEEEEEEEEEEEEE  DDDDDDDDDDDD           TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD           TTT
EEEEEEEEEEEEEEE  DDDDDDDDDDDD           TTT
```

```
**FILE**ID**SCRCOMCUR                                J 12
```

```
  SSSSSSSS    CCCCCCCC  RRRRRRRR      CLCCCCCC    OOOOOO    MM      MM    CCCCCCCC  UU      UU  RRRRRRRR
  SSSSSSSS    CCCCCCCC  RRRRRRRR      CCCCCCCC    OOOOOO    MM      MM    CCCCCCCC  UU      UU  RRRRRRRR
SS          CC          RR      RR  CC          OO      OO  MMMM  MMMM  CC          UU      UU  RR      RR
SS          CC          RR      RR  CC          OO      OO  MMMM  MMMM  CC          UU      UU  RR      RR
SS          CC          RR      RR  CC          OO      OO  MM  MM  MM  CC          UU      UU  RR      RR
SS          CC          RR      RR  CC          OO      OO  MM  MM  MM  CC          UU      UU  RR      RR
  SSSSSS    CC          RRRRRRRR    CC          OO      OO  MM      MM  CC          UU      UU  RRRRRRRR
  SSSSSS    CC          RRRRRRRR    CC          OO      OO  MM      MM  CC          UU      UU  RRRRRRRR
        SS  CC          RR  RR      CC          OO      OO  MM      MM  CC          UU      UU  RR  RR
        SS  CC          RR  RR      CC          OO      OO  MM      MM  CC          UU      UU  RR  RR
        SS  CC          RR    RR    CC          OO      OO  MM      MM  CC          UU      UU  RR    RR         ....
        SS  CC          RR    RR    CC          OO      OO  MM      MM  CC          UU      UU  RR    RR         ....
SSSSSSSS      CCCCCCCC  RR      RR    CCCCCCCC    OOOOOO    MM      MM    CCCCCCCC  UUUUUUUUUU  RR      RR       ....
SSSSSSSS      CCCCCCCC  RR      RR    CCCCCCCC    OOOOOO    MM      MM    CCCCCCCC  UUUUUUUUUU  RR      RR       ....

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
```

EDT
V04

```
0 %TITLE 'EDT$SCRCOMCUR - compute cursor position'
0 MODULE EDT$SCRCOMCUR (                                ! Compute cursor position
0                 IDENT = 'V04-000'                             ! File: SCRCOMCUR.BLI Edit: JBS1010
0                 ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:     EDT -- The DEC Standard Editor
1 !
1 ! ABSTRACT:
1 !
1 !       This module computes the current cursor position.
1 !
1 ! ENVIRONMENT:  Runs at any access mode - AST reentrant
1 !
1 ! AUTHOR: Bob Kushlis, CREATION DATE: September 8, 1979
1 !
1 ! MODIFIED BY:
1 !
1 ! 1-001 - Original.  DJS 12-Feb-1981.  This module was created by
1 !       extracting the routine EDT$$SC_CPUCSPOS  from module SCREEN.
1 ! 1-002 - Regularize headers.  JBS 13-Mar-1981
1 ! 1-003 - Rewrite for new screen logic.  JBS 12-Oct-1982
1 ! 1-004 - Fix a couple of minor bugs.  JBS 13-Oct-1982
1 ! 1-005 - Fix call to EDT$$FMT_CHWID.  JBS 13-Oct-1982
1 ! 1-006 - Fix problem with SHL.  JBS 27-Oct-1982
1 ! 1-007 - Fix the cursor position in NOTRUNCATE mode.  JBS 09-Nov-1982
1 ! 1-008 - Fix the cursor positioning again.  JBS 10-Nov-1982
1 ! 1-009 - Change the handling of EDT$$G_SHF.  JBS 14-Dec-1982
1 ! 1-010 - Correct tab at front of continued line.  JBS 15-Dec-1982
1 !--
1
```

```
0057  1 %SBTTL 'Declarations'
0058  1 !
0059  1 ! TABLE OF CONTENTS:
0060  1 !
0061  1
0062  1 REQUIRE 'EDTSRC:TRAROUNAM';
0501  1
0502  1 FORWARD ROUTINE
0503  1     EDT$$SC_CPUCSPOS : NOVALUE;
0504  1
0505  1 !
0506  1 ! INCLUDE FILES:
0507  1 !
0508  1
0509  1 REQUIRE 'EDTSRC:EDTREQ';
0644  1
0645  1 !
0646  1 ! MACROS:
0647  1 !
0648  1 !       NONE
0649  1 !
0650  1 ! EQUATED SYMBOLS:
0651  1 !
0652  1 !       NONE
0653  1 !
0654  1 ! OWN STORAGE:
0655  1 !
0656  1 !       NONE
0657  1 !
0658  1 ! EXTERNAL REFERENCES:
0659  1 !
0660  1 !       In the routine
```

```
1 %SBTTL 'EDT$$SC_CPUCSPOS  - compute cursor position'
1
1 GLOBAL ROUTINE EDT$$SC_CPUCSPOS (                    ! Compute cursor position
1     LINE,                                            ! Where to return line number
1     COLUMN                                           ! Where to return column number
1     ) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine computes the current cursor position returning the line
1 !       and column numbers in the ref parameters LINE and COLUMN.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !  LINE                 Cursor's relative line number
1 !
1 !  COLUMN                       Cursor's column number
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       EDT$$G_SHF
1 !       EDT$$G_TI_WID
1 !       EDT$$G_TRUN
1 !       EDT$$T_LN_BUF
1 !       EDT$$A_LN_PTR
1 !       EDT$$A_LN_END
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2     BEGIN
2
2     EXTERNAL ROUTINE
2       EDT$$FMT_CHWID;                                ! Compute the width of a character
2
2     EXTERNAL
2       EDT$$G_SHF,                                    ! The number of columns shifted.
2       EDT$$G_TI_WID,                                 ! Width of terminal line.
2       EDT$$G_TRUN,                                   ! Truncate or wrap long lines.
2       EDT$$T_LN_BUF,                                 ! Current line buffer.
2       EDT$$A_LN_PTR,                                 ! Current character pointer.
2       EDT$$A_LN_END;                                 ! End of line
2
2     LOCAL
2       CP,                                            ! Character pointer into the current record
```

```
2           LIN,                                        ! Relative line number (first line = 0)
2           COL,                                        ! Column number (first column = 0), unshifted
2           CHAR,                                       ! Current character
2           CHAR_WID,                                   ! Width of the current character
2           LINE_DONE;                                  ! 1 = we are done with this line
2
2       LIN = 0;
2       COL = 0;
2       CP = EDT$$T_LN_BUF;
2       LINE_DONE = 0;
2
2       WHILE ((.CP LSSA .EDT$$A_LN_PTR) AND ( NOT .LINE_DONE)) DO
3           BEGIN
3           CHAR = CH$RCHAR_A (CP);
3           CHAR_WID = EDT$$FMT_CHWID (.CHAR, .COL);
3
4           IF ((.COL + .CHAR_WID) LEQ (.EDT$$G_TI_WID + .EDT$$G_SHF))
3           THEN
3 !+
3 ! The character fits on this line, count it and go on to the next.
3 !-
3               COL = .COL + .CHAR_WID
3           ELSE
3 !+
3 ! The character does not fit on this line.
3 !-
3
3               IF .EDT$$G_TRUN
3               THEN
4                   BEGIN
4 !+
4 ! In TRUNCATE mode, just position to the last column and terminate.
4 !-
4                   COL = .EDT$$G_TI_WID + .EDT$$G_SHF - 1;
4                   LINE_DONE = 1;
4                   END
3               ELSE
4                   BEGIN
4 !+
4 ! In NOTRUNCATE mode, try fitting it on the next line.  Don't produce too many lines.
4 !-
4                   LIN = .LIN + 1;
4                   COL = .EDT$$G_SHF + 2;
4 !+
4 ! We can't use .CHAR_WID in the next statement because the width of a tab may be
4 ! different on the new line since it is in a new position.
4 !-
4
4                   IF (.LIN GEQ 255) THEN LINE_DONE = 1 ELSE COL = .COL + EDT$$FMT_CHWID (.CHAR, .COL);
4
3                   END;
3
2           END;
2
2 !+
2 ! In NOTRUNCATE mode, make sure the current character will fit on this line.  If it will not,
2 ! move the cursor to the beginning of the next line.
```

```
:  205          0775  2 !-
:  206          0776  2
:  207          0777  3       IF (( NOT .EDT$$G_TRUN) AND ( NOT .LINE_DONE) AND (.EDT$$A_LN_PTR NEQA .EDT$$A_LN_END))
:  208          0778  2       THEN
:  209          0779  3           BEGIN
:  210          0780  3           CHAR = CH$RCHAR_A (CP);
:  211          0781  3           CHAR_WID = EDT$$FMT_CHWID (.CHAR, .COL);
:  212          0782  3
:  213          0783  4           IF ((.COL + .CHAR_WID) GTR (.EDT$$G_TI_WID + .EDT$$G_SHF))
:  214          0784  3           THEN
:  215          0785  4               BEGIN
:  216          0786  4               LIN = .LIN + 1;
:  217          0787  4               COL = .EDT$$G_SHF + 2;
:  218          0788  3               END;
:  219          0789  3
:  220          0790  2           END;
:  221          0791  2
:  222          0792  2       .LINE = .LIN;
:  223          0793  2       .COLUMN = MAX (0, MIN (.COL - .EDT$$G_SHF, .EDT$$G_TI_WID - 1));
:  224          0794  1       END;                                              ! of routine EDT$$SC_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

                                            0FFC 00000         .ENTRY  EDT$$SC_CPUCSPOS, Save R2,R3,R4,R5,R6,R7,-   ; 0663
                                                                       R8,R9,R10,R11                                 ;
                         5B 00000000G   00   9E 00002         MOVAB   EDT$$G_TI_WID, R11                            ;
                         5A 00000000G   00   9E 00009         MOVAB   EDT$$G_SHF, R10                               ;
                         59 00000000G   00   9E 00010         MOVAB   EDT$$FMT_CHWID, R9                            ;
                                        58   D4 00017         CLRL    LIN                                           ; 0724
                                        52   D4 00019         CLRL    COL                                           ; 0725
                         55 00000000G   00   9E 0001B         MOVAB   EDT$$T_LN_BUF, CP                             ; 0726
                                        54   D4 00022         CLRL    LINE_DONE                                     ; 0727
              00000000G  00             55   D1 00024 1$:     CMPL    CP, EDT$$A_LN_PTR                             ; 0729
                                        52   1E 0002B         BGEQU   6$                                            ;
                         4F             54   E8 0002D         BLBS    LINE_DONE, 6$                                 ;
                         57             85   9A 00030         MOVZBL  (CP)+, CHAR                                   ; 0731
                                        52   DD 00033         PUSHL   COL                                           ; 0732
                                        57   DD 00035         PUSHL   CHAR                                          ;
                         69             02   FB 00037         CALLS   #2, EDT$$FMT_CHWID                            ;
                         56             50   D0 0003A         MOVL    R0, CHAR_WID                                  ;
              53         52             56   C1 0003D         ADDL3   CHAR_WID, COL, R3                             ; 0734
                         51             6A   D0 00041         MOVL    EDT$$G_SHF, R1                                ;
              50         6B             51   C1 00044         ADDL3   R1, EDT$$G_TI_WID, R0                         ;
                         50             53   D1 00048         CMPL    R3, R0                                        ;
                                        05   14 0004B         BGTR    2$                                            ;
                         52             56   C0 0004D         ADDL2   CHAR_WID, COL                                 ; 0739
                                        D2   11 00050         BRB     1$                                            ;
```

```
                           06 00000000G  00  E9 00052 2$:    BLBC    EDT$$G_TRUN, 3$                       ; 0745
                           52        FF  A0  9E 00059        MOVAB   -1(R0), COL                           ; 0751
                                     0F  11 0005D            BRB     4$                                    ; 0752
                                     58  D6 0005F 3$:        INCL    LIN                                   ; 0759
                           52        02  A1  9E 00061        MOVAB   2(R1), COL                            ; 0760
               000000FF    8F        58  D1 00065            CMPL    LIN, #255                             ; 0766
                                     05  19 0006C            BLSS    5$                                    ;
                           54        01  D0 0006E 4$:        MOVL    #1, LINE_DONE                         ;
                                     B1  11 00071            BRB     1$                                    ;
                                     52  DD 00073 5$:        PUSHL   COL                                   ;
                                     57  DD 00075            PUSHL   CHAR                                  ;
                           69        02  FB 00077            CALLS   #2, EDT$$FMT_CHWID                    ;
                           52        50  C0 0007A            ADDL2   R0, COL                               ;
                                     A5  11 0007D            BRB     1$                                    ; 0729
                           33 00000000G  00  E8 0007F 6$:    BLBS    EDT$$G_TRUN, 7$                       ; 0777
                           30        54  E8 00086            BLBS    LINE_DONE, 7$                         ;
               00000000G   00 00000000G  00  D1 00089        CMPL    EDT$$A_LN_PTR, EDT$$A_LN_END          ;
                                     23  13 00094            BEQL    7$                                    ;
                           57        85  9A 00096            MOVZBL  (CP)+, CHAR                           ; 0780
                                     52  DD 00099            PUSHL   COL                                   ; 0781
                                     57  DD 0009B            PUSHL   CHAR                                  ;
                           69        02  FB 0009D            CALLS   #2, EDT$$FMT_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    EDT$$G_SHF, R1                        ;
                  50       6B        51  C1 000AA            ADDL3   R1, EDT$$G_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    BC        58  D0 000B9 7$:        MOVL    LIN, @LINE                            ; 0792
                           52        6A  C2 000BD            SUBL2   EDT$$G_SHF, R2                        ; 0793
                  50       6B        01  C3 000C0            SUBL3   #1, EDT$$G_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                           ;
                                         04 000D6            RET                                           ; 0794

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


;  225           0795  1
;  226           0796  1 !<BLF/PAGE>
;  228          0797  1 END                                                    ! of module EDT$SCRCOMCUR
;  229          0798  1
;  230          0799  0 ELUDOM



;                                            PSECT SUMMARY
;
;       Name                            Bytes                           Attributes
;
;  _EDT$CODE                                  215  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)



;                                    Library Statistics
;
;                                                 -------- Symbols --------        Pages        Processing
;       File                                      Total    Loaded   Percent       Mapped        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
;       CRCOMCUR)

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY