C 12

```
   SSSSSSSSSSSS  MMM         MMM     GGGGGGGGGGG   RRRRRRRRRRRR    TTTTTTTTTTTTTTT  LLL
   SSSSSSSSSSSS  MMM         MMM     GGGGGGGGGGG   RRRRRRRRRRRR    TTTTTTTTTTTTTTT  LLL
   SSSSSSSSSSSS  MMM         MMM     GGGGGGGGGGG   RRRRRRRRRRRR    TTTTTTTTTTTTTTT  LLL
SSS              MMMMMM   MMMMMM  GGG              RRR        RRR        TTT        LLL
SSS              MMMMMM   MMMMMM  GGG              RRR        RRR        TTT        LLL
SSS              MMMMMM   MMMMMM  GGG              RRR        RRR        TTT        LLL
SSS              MMM   MMM   MMM  GGG              RRR        RRR        TTT        LLL
SSS              MMM   MMM   MMM  GGG              RRR        RRR        TTT        LLL
SSS              MMM   MMM   MMM  GGG              RRR        RRR        TTT        LLL
   SSSSSSSSS     MMM         MMM  GGG              RRRRRRRRRRRR          TTT        LLL
   SSSSSSSSS     MMM         MMM  GGG              RRRRRRRRRRRR          TTT        LLL
   SSSSSSSSS     MMM         MMM  GGG              RRRRRRRRRRRR          TTT        LLL
            SSS  MMM         MMM  GGG   GGGGGGGGG  RRR   RRR             TTT        LLL
            SSS  MMM         MMM  GGG   GGGGGGGGG  RRR   RRR             TTT        LLL
            SSS  MMM         MMM  GGG   GGGGGGGGG  RRR   RRR             TTT        LLL
            SSS  MMM         MMM  GGG         GGG  RRR      RRR          TTT        LLL
            SSS  MMM         MMM  GGG         GGG  RRR      RRR          TTT        LLL
            SSS  MMM         MMM  GGG         GGG  RRR      RRR          TTT        LLL
SSSSSSSSSSSS     MMM         MMM     GGGGGGGGG     RRR        RRR        TTT        LLLLLLLLLLLLLLL
SSSSSSSSSSSS     MMM         MMM     GGGGGGGGG     RRR        RRR        TTT        LLLLLLLLLLLLLLL
SSSSSSSSSSSS     MMM         MMM     GGGGGGGGG     RRR        RRR        TTT        LLLLLLLLLLLLLLL
```

FILEID**SMGDISCHA

```
  SSSSSSSS  MM        MM  GGGGGGGG  DDDDDDDD       IIIIII     SSSSSSSS    CCCCCCCC  HH        HH    AAAAAA
  SSSSSSSS  MM        MM  GGGGGGGG  DDDDDDDD       IIIIII     SSSSSSSS    CCCCCCCC  HH        HH    AAAAAA
SS          MMMM    MMMM  GG        DD      DD       II     SS          CC          HH        HH  AA      AA
SS          MMMM    MMMM  GG        DD      DD       II     SS          CC          HH        HH  AA      AA
SS          MM  MM  MM    GG        DD      DD       II     SS          CC          HH        HH  AA      AA
SS          MM  MM  MM    GG        DD      DD       II     SS          CC          HH        HH  AA      AA
  SSSSSS    MM        MM  GG        DD      DD       II       SSSSSS    CC          HHHHHHHHHH    AA      AA
  SSSSSS    MM        MM  GG        DD      DD       II       SSSSSS    CC          HHHHHHHHHH    AA      AA
        SS  MM        MM  GG  GGGGGG DD      DD       II             SS  CC          HH        HH  AAAAAAAAAA
        SS  MM        MM  GG  GGGGGG DD      DD       II             SS  CC          HH        HH  AAAAAAAAAA
        SS  MM        MM  GG      GG DD      DD       II             SS  CC          HH        HH  AA      AA
        SS  MM        MM  GG      GG DD      DD       II             SS  CC          HH        HH  AA      AA    ....
SSSSSSSS    MM        MM    GGGGGG   DDDDDDDD       IIIIII   SSSSSSSS      CCCCCCCC  HH        HH  AA      AA    ....
SSSSSSSS    MM        MM    GGGGGG   DDDDDDDD       IIIIII   SSSSSSSS      CCCCCCCC  HH        HH  AA      AA    ....
                                                                                                                 ....

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
```

```
0 %TITLE 'SMG$DISPLAY_CHANGE - Virtual Display -- Changes to Displays'
0 MODULE SMG$DISPLAY_CHANGE (
0               IDENT = '1-048' ! File: SMGDISCHA.B32 Edit: PLL1048
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:      Screen Management
1 !
1 ! ABSTRACT:
1 !       The procedures in this module update and maintain the contents
1 ! of the in-memory representations of the virtual displays.  The data
1 ! areas themselves are allocated/deallocated, pasted/upasted, etc.
1 ! by the procedures in module SMG$DISPLAY_LINKS.  Output from these
1 ! virtual displays is handled by procedures in SMG$DISPLAY_OUTPUT.
1 ! The module SMG$DISPLAY_INPUT contains the routines that support
1 ! input operations on displays.
1 !
1 !
1 !
1 ! ENVIRONMENT:  User mode, Shared library routines.
1 !
1 ! AUTHOR: R. Reichert, CREATION DATE: 26-Jan-1983
1 !
1 ! MODIFIED BY:
1 !
1 ! 1-048 - Change error messages by SMG$SET_CURSOR_REL. STAN 3-Jun-1984.
1 ! 1-047 - Turn off the scrolling bit whenever sequential output ends, ie.
1 !         whenever the cursor may be randomly positioned.  PLL 3-May-1984
1 ! 1-001 - Original.  Skeleton for future code.  RKR 26-Jan-1983
1 !--
```

```
;  1 %SBTTL 'Declarations'
;  1 !
;  1 ! SWITCHES:
;  1 !
;  1
;  1 !
;  1 ! LINKAGES:
;  1 !
;  1 !      NONE
;  1 !
;  1 ! TABLE OF CONTENTS:
;  1 !
;  1
;  1 FORWARD ROUTINE
;  1
;  1 ! Public entry points:
;  1
;  1     SMG$CHANGE_RENDITION,                  ! Change video attributes of
;  1                                            ! something already in a virtual
;  1                                            ! display
;  1
;  1     SMG$CURSOR_COLUMN,                     ! Function to return current
;  1                                            ! cursor col. within virt. disp.
;  1
;  1     SMG$CURSOR_ROW,                        ! Function to return current
;  1                                            ! cursor row within virt. disp.
;  1
;  1     SMG$DELETE_CHARS,                      ! Delete characters from within
;  1                                            ! a line in a virtual display.
;  1
;  1     SMG$DELETE_LINE,                       ! Delete line in a display &
;  1                                            ! scroll remaining lines
;  1
;  1     SMG$ERASE_CHARS,                       ! Erase characters in a display
;  1                                            ! w/out moving remaining text
;  1
;  1     SMG$ERASE_DISPLAY,                     ! Erase virtual display
;  1
;  1     SMG$ERASE_LINE,                        ! Erase from position to end
;  1                                            ! of line
;  1
;  1     SMG$HOME_CURSOR,                       ! Move cursor to one of the 4
;  1                                            ! corners of the virtual display
;  1
;  1     SMG$INSERT_CHARS,                      ! Insert characters in a line
;  1                                            ! of a virtual display.
;  1
;  1     SMG$INSERT_LINE,                       ! Insert line in a virtual
;  1                                            ! display
;  1
;  1     SMG$PUT_CHARS,                         ! Put text to display without
;  1                                            ! additonal cursor motion.
;  1
;  1     SMG$PUT_LINE,                          ! Put text line to display with
;  1                                            ! optional additonal cursor
;  1                                            ! motion.
;  1
```

```
        SMG$READ_FROM_DISPLAY,              ! Read text back from virtual
                                            ! display

        SMG$PUT_WITH_SCROLL,                ! Scroll a virtual display and
                                            ! place callers text into opened
                                            ! area.

        SMG$RETURN_CURSOR_POS,              ! Returns current cursor
                                            ! position (row/column).

        SMG$SCROLL_DISPLAY_AREA,            ! Scroll a rectangular area

        SMG$SET_CURSOR_ABS,                 ! Move cursor to absolute
                                            ! position specified.

        SMG$SET_CURSOR_REL;                 ! Move cursor relative to its
                                            ! current position.


! INCLUDE FILES
!

REQUIRE 'RTLIN:SMGPROLOG';                  ! defines psects, macros, tcb,
                                            !  wcb, & terminal symbols
REQUIRE 'RTLIN:STRLNK';                     ! JSB linkage

!
! EXTERNAL REFERENCES
!
EXTERNAL ROUTINE
    LIB$ANALYZE_SDESC_R2 : LIB$ANALYZE_SDESC_JSB_LINK,

    LIB$SCOPY_R_DX6 : STRING_JSB,           ! String copier

    SMG$$CHECK_FOR_OUTPUT_DCB,  ! Force output if now is the time

    SMG$$PUT_TEXT_TO_BUFFER,    ! Move callers text to virtual
                                ! display text buffer checking
                                ! for characters that affect
                                ! where characters land in
                                ! buffer.

    SMG$$SCROLL_AREA;           ! Scroll a rectangular area

EXTERNAL
    SMG$_FATERRLIB,       ! Fatal error in SMG$ -- internal consistancy
                          ! check failed.
    SMG$_INVARG,          ! Invalid argument
    SMG$_INVCOL,          ! Invalid column number
    SMG$_INVDIS_ID,       ! Invalid virtual display id
    SMG$_INVPAS_ID,       ! Invalid pasteboard id
    SMG$_INVROW,          ! Invalid row number
    SMG$_NO_CHADIS,       ! No change to virtual display
    SMG$_WRONUMARG;       ! Wrong number of arguments

!+
! The following macro determines whether scrolling up, down, or neither
```

```
! should occur.
!-
MACRO $SMG$SET_SCROLLING (SWITCH) =
    BEGIN
    SWITCH = 0;          ! init scroll to off
    IF .DCB [DCB_V_FULL] NEQ 0
    THEN                 ! display is full, check other conditions
        BEGIN
        IF .DCB [DCB_W_CURSOR_ROW] EQL .DCB [DCB_W_BOTTOM_OF_SCRREG]
        THEN
            SWITCH = 1   ! scroll up
        ELSE
            IF .DCB [DCB_W_CURSOR_ROW] EQL .DCB [DCB_W_TOP_OF_SCRREG]
            THEN
                SWITCH = 2;  ! scroll down
        END;
    END;%;

!<BLF/PAGE>
```

```
1 %SBTTL 'SMG$CHANGE_RENDITION - Change Rendition'
1 GLOBAL ROUTINE SMG$CHANGE_RENDITION (
1                               DISPLAY_ID,
1                               START_ROW,
1                               START_COL,
1                               NO_ROW,
1                               NO_COL,
1                               RENDITION_SET,
1                               RENDITION_COMPLEMENT
1                                       ) =
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine changes the video rendition of a rectangular block of
1 !       text already in the specified virtual display.
1 !       Use of this routine makes it easy, for example, to redisplay a
1 !       particular column in reverse video.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = SMG$CHANGE_RENDITION (
1 !                                 DISPLAY_ID.rl.r,
1 !                                 START_ROW.rl.r,
1 !                                 START_COL.rl.r,
1 !                                 NO_ROW.rl.r,
1 !                                 NO_COL.rl.r,
1 !                                 [,RENDITION_SET.rl.r]
1 !                                 [,RENDITION_COMPLEMENT.rl.r])
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       DISPLAY_ID.rl.r           Display id of desired virtual display.
1 !
1 !       START_ROW.rl.r            Starting row position to receive new
1 !                                 rendition
1 !
1 !       START_COL.rl.r            Starting column position to receive
1 !                                 new rendition.
1 !
1 !       NO_ROW.rl.r               No. of rows to receive new rendition
1 !
1 !       NO_COL.rl.r               No. of columns to receive new
1 !                                 rendition
1 !
1 !       RENDITION_SET.rl.r        Each 1 bit attribute in this parameter
1 !                                 causes the corresponding attribute to
1 !                                 be set in the display.  (See below for
1 !                                 list of settable attributes.)
1 !
1 !       RENDITION_COMPLEMENT.rl.r
1 !                                 Each 1 bit attribute in this parameter
1 !                                 causes the corresponding attribute to
1 !                                 be complemented in the display.  (See
1 !                                 below for list of complementable
1 !                                 attributes.)
1 !
1 !       If the same bit is specified in both the RENDITION_SET parameter
```

```
1 !          and in the RENDITION_COMPLEMENT parameter, the application is
1 !          RENDITION_SET followed by RENDITION complement.  Using these two
1 !          parameters together the caller can exercise arbitrary and
1 !          independent control over each attribute on a single call.  On an
1 !          attribute by attribute basis he can cause the following
1 !          transformations:
1 !
1 !                  SET     COMPLEMENT      Action
1 !                  ---     ----------      ------
1 !                  0       0               Attribute unchanged.
1 !                  1       0               Attribute set to "on"
1 !                  0       1               Attribute set to complement of
1 !                                          current setting.
1 !                  1       1               Attribute set to "off".
1 !
1 !
1 !          Attributes which can be manipulated in this manner are:
1 !
1 !          SMG$M_BLINK  displays characters blinking.
1 !          SMG$M_BOLD  displays characters in higher-than-normal
1 !                       intensity.
1 !          SMG$M_REVERSE  displays characters in reverse video -- that is,
1 !                       using the opposite default rendition of the
1 !                       virtual display.
1 !          SMG$M_UNDERLINE  displays characters underlined.
1 !
1 !    IMPLICIT INPUTS:
1 !
1 !          NONE
1 !
1 !    IMPLICIT OUTPUTS:
1 !
1 !          NONE
1 !
1 !    COMPLETION STATUS:
1 !
1 !          SS$_NORMAL        Normal successful completion
1 !          SMG$_INVROW       Invalid start row -- outside virtual display.
1 !          SMG$_INVCOL       Invalid start column -- outside virtual display.
1 !          SMG$_INVDIS_ID     Invalid display id.
1 !          SMG$_INVARG       Invalid no. of rows -- outside virtual display
1 !                         or Invalid no. of columns -- outside virtual display.
1 !                         or Unrecognized action code.
1 !                         or Unrecognized rendition code.
1 !          SMG$_WRONUMARG  Wrong number of arguments.
1 !
1 !    SIDE EFFECTS:
1 !
1 !          NONE
1 !--
2      BEGIN
2
2      LOCAL
2          START_INDEX,              ! Byte offset into attribute buffer
2
2          DCB : REF $DCB_DECL,      ! Addr of display control block
2
```

```
                REND_CODE,                      ! Video rendition -- set up by
                                                ! $SMG$SET_REND_CODE

                ATTR_BUF : REF VECTOR [,BYTE];  ! Addr of attribute buffer for
                                                ! this virtual display.

        LITERAL
            K_SET_ARG = 6,
            K_COMP_ARG = 7;

        $SMG$VALIDATE_ARGCOUNT (5,7);

        $SMG$GET_DCB ( .DISPLAY_ID, DCB);       ! Get addr of display control
                                                ! block

    !+
    ! Validate arguments, make sure row & column are within the display.
    !-

        $SMG$VALIDATE_ROW_COL (..START_ROW, ..START_COL);

        IF ..NO_COL LEQ 0 OR
           ..NO_ROW LEQ 0
        THEN
            RETURN (SMG$_NO_CHADIS);

        $SMG$VALIDATE_ROW_COL (..START_ROW + ..NO_ROW - 1,
                               ..START_COL + ..NO_COL - 1);

        ATTR_BUF = .DCB [DCB_A_ATTR_BUF];
                                        ! addr of attribute buffer

        $SMG$SET_REND_CODE (K_SET_ARG, K_COMP_ARG);
                                        ! set rend_code w/default or caller's args

        !+
        ! Set the rendition bytes in the attribute buffer on a line by line
        ! basis, in case the change does not include the entire line.
        ! Be careful to preserve the other bits in the rendition byte that have
        ! nothing to do with rendition.
        !-

        START_INDEX = $SMG$LINEAR (..START_ROW, ..START_COL);
        INCR LINE_COUNT FROM 1 TO ..NO_ROW
        DO
            BEGIN   ! For each row
            INCR COL_COUNT FROM 0 TO (..NO_COL - 1)
            DO
                BEGIN           ! For each column

                ATTR_BUF [.START_INDEX + .COL_COUNT ] =
                  (.ATTR_BUF [.START_INDEX + .COL_COUNT]
                              AND
                ( NOT (SMG$M_BLINK+SMG$M_BOLD+SMG$M_REVERSE+SMG$M_UNDERLINE)) )
                              OR
                      .REND_CODE ;
```

```
0621  3           END;            ! For each column
0622  3
0623  3        START_INDEX = .START_INDEX + .DCB [DCB_W_NO_COLS]; ! To next row start
0624  2        END;     ! For each row
0625  2
0626  2     !+
0627  2     ! See if this change should be reflected on the screen right away.
0628  2     !-
0629  2
0630  3     RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
0631  2                                      SMG$C_CHANGE_RENDITION));
0632  1     END;                       ! End of routine SMG$CHANGE_RENDITION

                                                        .TITLE  SMG$DISPLAY_CHANGE SMG$DISPLAY_CHANGE - Virtual
;                                                                                  Display -- Changes
                                                        .IDENT  \1-048\

                                                        .EXTRN  LIB$ANALYZE_SDESC_R2
                                                        .EXTRN  LIB$SCOPY_R_DX6
                                                        .EXTRN  SMG$$CHECK_FOR_OUTPUT_DCB
                                                        .EXTRN  SMG$$PUT_TEXT_TO_BUFFER
                                                        .EXTRN  SMG$$SCROLL_AREA
                                                        .EXTRN  SMG$_FATERRLIB, SMG$_INVARG
                                                        .EXTRN  SMG$_INVCOL, SMG$_INVDIS_ID
                                                        .EXTRN  SMG$_INVPAS_ID, SMG$_INVROW
                                                        .EXTRN  SMG$_NO_CHADIS, SMG$_WRONUMARG

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

                              07FC 00000                .ENTRY  SMG$CHANGE_RENDITION, Save R2,R3,R4,R5,R6,-  ; 0451
                                                                R7,R8,R9,R10                                 ;
                   5A 00000000G  8F   D0 00002          MOVL    #SMG$_INVCOL, R10                            ;
                   59 00000000G  8F   D0 00009          MOVL    #SMG$_INVROW, R9                             ;
          50       6C            05   83 00010          SUBB3   #5, (AP), DIFF                               ; 0574
                   02            50   91 00014          CMPB    DIFF, #2                                     ;
                                 08   1B 00017          BLEQU   1$                                           ;
                   50 00000000G  8F   D0 00019          MOVL    #SMG$_WRONUMARG, R0                          ;
                                 04      00020          RET                                                  ;
                   50        04  BC   D0 00021 1$:      MOVL    @DISPLAY_ID, R0                              ; 0576
             04    BC        38  A0   D1 00025          CMPL    56(R0), @DISPLAY_ID                          ;
                                 06   12 0002A          BNEQ    2$                                           ;
                   11        44  A0   91 0002C          CMPB    68(R0), #17                                  ;
                                 08   13 00030          BEQL    3$                                           ;
                   50 00000000G  8F   D0 00032 2$:      MOVL    #SMG$_INVDIS_ID, R0                          ;
                                 04      00039          RET                                                  ;
                   52        04  BC   D0 0003A 3$:      MOVL    @DISPLAY_ID, DCB                             ;
                   54        08  BC   D0 0003E          MOVL    @START_ROW, R4                               ; 0583
                                 08   15 00042          BLEQ    4$                                           ;
 54      02  A2    10            00   ED 00044          CMPZV   #0, #16, 2(DCB), R4                          ;
                                 04   18 0004A          BGEQ    5$                                           ;
                   50            59   D0 0004C 4$:      MOVL    R9, R0                                       ;
                                 04      0004F          RET                                                  ;
                   53        0C  BC   D0 00050 5$:      MOVL    @START_COL, R3                               ;
                                 08   15 00054          BLEQ    6$                                           ;
 53      06  A2    10            00   ED 00056          CMPZV   #0, #16, 6(DCB), R3                          ;
                                 04   18 0005C          BGEQ    7$                                           ;
```

```
                                50          5A  D0 0005E 6$:   MOVL    R10, R0                                  ;
                                                04 00061       RET                                              ;
                                58     14   BC  D0 00062 7$:   MOVL    @NO_COL, R8                              ; 0585
                                            05  15 00066       BLEQ    8$                                       ;
                                       10   BC  D5 00068       TSTL    @NO_ROW                                  ; 0586
                                            08  14 0006B       BGTR    9$                                       ;
                        50 00000000G        00  9E 0006D 8$:   MOVAB   SMG$_NO_CHADIS, R0                       ; 0588
                                                04 00074       RET                                              ;
                50              54     10   BC  C1 00075 9$:   ADDL3   @NO_ROW, R4, R0                          ; 0591
                                01          50  D1 0007A       CMPL    R0, #1                                   ;
                                            0A  15 0007D       BLEQ    10$                                      ;
                                            50  D7 0007F       DECL    R0                                       ;
      50       02    A2         10          00  ED 00081       CMPZV   #0, #16, 2(DCB), R0                      ;
                                            04  18 00087       BGEQ    11$                                      ;
                                50          59  D0 00089 10$:  MOVL    R9, R0                                   ;
                                                04 0008C       RET                                              ;
                50              53          58  C1 0008D 11$:  ADDL3   R8, R3, R0                               ;
                                01          50  D1 00091       CMPL    R0, #1                                   ;
                                            0A  15 00094       BLEQ    12$                                      ;
                                            50  D7 00096       DECL    R0                                       ;
      50       06    A2         10          00  ED 00098       CMPZV   #0, #16, 6(DCB), R0                      ;
                                            04  18 0009E       BGEQ    13$                                      ;
                                50          5A  D0 000A0 12$:  MOVL    R10, R0                                  ;
                                                04 000A3       RET                                              ;
                                51     14   A2  D0 000A4 13$:  MOVL    20(DCB), ATTR_BUF                        ; 0593
                                57     2E   A2  9A 000A8       MOVZBL  46(DCB), REND_CODE                       ; 0596
                                06          6C  91 000AC       CMPB    (AP), #6                                 ;
                                            09  1F 000AF       BLSSU   14$                                      ;
                                       18   AC  D5 000B1       TSTL    24(AP)                                   ;
                                            04  13 000B4       BEQL    14$                                      ;
                                57     18   BC  C8 000B6       BISL2   @RENDITION_SET, REND_CODE                ;
                                07          6C  91 000BA 14$:  CMPB    (AP), #7                                 ;
                                            09  1F 000BD       BLSSU   15$                                      ;
                                       1C   AC  D5 000BF       TSTL    28(AP)                                   ;
                                            04  13 000C2       BEQL    15$                                      ;
                                57     1C   BC  CC 000C4       XORL2   @RENDITION_COMPLEMENT, REND_CODE         ;
                                50     FF   A4  9E 000C8 15$:  MOVAB   -1(R4), R0                               ; 0606
                                54     06   A2  3C 000CC       MOVZWL  6(DCB), R4                               ;
                                50          54  C4 000D0       MULL2   R4, R0                                   ;
                                50     FF A340  9E 000D3       MOVAB   -1(R3)[R0], START_INDEX                  ;
                                            56  D4 000D8       CLRL    LINE_COUNT                               ; 0614
                                            20  11 000DA       BRB     19$                                      ;
                                54          01  CE 000DC 16$:  MNEGL   #1, COL_COUNT                            ;
                                            10  11 000DF       BRB     18$                                      ;
                53              50          54  C1 000E1 17$:  ADDL3   COL_COUNT, START_INDEX, R3               ;
                                55        6341  9A 000E5       MOVZBL  (R3)[ATTR_BUF], R5                       ; 0617
                                55          0F  CA 000E9       BICL2   #15, R5                                  ;
              6341              55          57  89 000EC       BISB3   REND_CODE, R5, (R3)[ATTR_BUF]            ; 0619
                EC              54          58  F2 000F1 18$:  AOBLSS  R8, COL_COUNT, 17$                       ; 0610
                                53     06   A2  3C 000F5       MOVZWL  6(DCB), R3                               ; 0623
                                50          53  C0 000F9       ADDL2   R3, START_INDEX                          ;
                DB              56     10   BC  F3 000FC 19$:  AOBLEQ  @NO_ROW, LINE_COUNT, 16$                 ; 0607
                                            0A  DD 00101       PUSHL   #10                                      ; 0630
                                            52  DD 00103       PUSHL   DCB                                      ;
                     00000000G  00          02  FB 00105       CALLS   #2, SMG$$CHECK_FOR_OUTPUT_DCB            ;
                                                04 0010C       RET                                              ; 0632
```

```
; Routine Size:  269 bytes,    Routine Base:  _SMG$CODE + 0000

;  374          0633  1 !<BLF/PAGE>
```

```
1 %SBTTL 'SMG$CURSOR_COLUMN - Return current cursor column position'
1 GLOBAL ROUTINE SMG$CURSOR_COLUMN ( DISPLAY_ID ) =
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This functions returns the current cursor column position in the
1 !       specified virtual display.
1 !
1 !       If the specified virtual display id is invalid, SMG$_INVDIS_ID
1 !       is signaled.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       CURSOR_COLUMN.wl.r = SMG$CURSOR_COLUMN ( DISPLAY_ID.rl.r )
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       DISPLAY_ID.rl.r Display id of desired display.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !
1 !       CURSOR_COLUMN.wl.v      The current cursor column number in the
1 !                               display.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       SMG$_INVDIS_ID  will be signaled if DISPLAY_ID in invalid.
1 !       SMG$_WRONUMARG  will be signaled if DISPLAY_ID is omitted.
1 !--
2     BEGIN
2     BUILTIN
2         NULLPARAMETER;
2
2     LOCAL
2         DCB : REF $DCB_DECL;    ! Addr. of display control block
2
2     IF NULLPARAMETER(DISPLAY_ID)
2     THEN
2         SIGNAL_STOP ( SMG$_WRONUMARG);  ! Display id has to be there
2
2 !+
2 ! The next few lines of code validate the supplied device id.  It
2 ! does precisely what $SMG$GET_DCB  does except it signals where that
2 ! macro would return a status.
2 !-
2     DCB = ..DISPLAY_ID;                     ! assume it is correct
2
2     IF .DCB [DCB_L_DID] NEQ ..DISPLAY_ID
2     THEN
```

```
;  433          0691  2         SIGNAL_STOP (SMG$_INVDIS_ID);   ! Not pointing to one of our
;  434          0692  2                                         ! control blocks
;  435          0693  2
;  436          0694  2     IF .DCB [DCB_B_STRUCT_TYPE] NEQ DCB_K_STRUCT_TYPE
;  437          0695  2     THEN
;  438          0696  2         SIGNAL_STOP (SMG$_INVDIS_ID);   ! Not pointing to a DCB
;  439          0697  2                                         ! block
;  440          0698  2
;  441          0699  2     RETURN (.DCB [DCB_W_CURSOR_COL]);   ! Give them what they asked for
;  442          0700  2
;  443          0701  1     END;                            ! End of routine SMG$CURSOR_COLUMN



                                        001C 00000          .ENTRY  SMG$CURSOR_COLUMN, Save R2,R3,R4               ; 0635
                       54 00000000G  00  9E 00002          MOVAB   SMG$_INVDIS_ID, R4                               ;
                       53 00000000G  00  9E 00009          MOVAB   LIB$STOP, R3                                     ;
                                     6C  95 00010          TSTB    (AP)                                             ; 0678
                                     05  13 00012          BEQL    1$                                               ;
                               04    AC  D5 00014          TSTL    4(AP)                                            ;
                                     09  12 00017          BNEQ    2$                                               ;
                          00000000G  00  9F 00019 1$:      PUSHAB  SMG$_WRONUMARG                                   ; 0680
                       63            01  FB 0001F          CALLS   #1, LIB$STOP                                     ;
                       52      04    BC  D0 00022 2$:      MOVL    @DISPLAY_ID, DCB                                 ; 0687
                04     BC      38    A2  D1 00026          CMPL    56(DCB), @DISPLAY_ID                             ; 0689
                                     05  13 0002B          BEQL    3$                                               ;
                                     54  DD 0002D          PUSHL   R4                                               ; 0691
                       63            01  FB 0002F          CALLS   #1, LIB$STOP                                     ;
                       11      44    A2  91 00032 3$:      CMPB    68(DCB), #17                                     ; 0694
                                     05  13 00036          BEQL    4$                                               ;
                                     54  DD 00038          PUSHL   R4                                               ; 0696
                       63            01  FB 0003A          CALLS   #1, LIB$STOP                                     ;
                       50      2A    A2  3C 0003D 4$:      MOVZWL  42(DCB), R0                                      ; 0699
                                         04 00041          RET                                                      ; 0701

; Routine Size:  66 bytes,    Routine Base:  _SMG$CODE + 010D


;  444          0702  1 !<BLF/PAGE>
```

```
1 %SBTTL 'SMG$CURSOR_ROW - Return current cursor row position'
1 GLOBAL ROUTINE SMG$CURSOR_ROW ( DISPLAY_ID ) =
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This functions returns the current cursor row position in the
1 !       specified virtual display.
1 !
1 !       If the specified virtual display id is invalid, SMG$_INVDIS_ID
1 !       is signaled.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       CURSOR_ROW.wl.r = SMG$CURSOR_ROW ( DISPLAY_ID.rl.r )
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       DISPLAY_ID.rl.r Display id of desired display.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !
1 !       CURSOR_ROW.wl.v The current cursor row number in the display.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       SMG$_INVDIS_ID  will be signaled if DISPLAY_ID in invalid.
1 !       SMG$_WRONUMARG  will be signaled if DISPLAY_ID is omitted.
1 !--
2     BEGIN
2     BUILTIN
2         NULLPARAMETER;
2
2     LOCAL
2         DCB : REF $DCB_DECL;    ! Addr. of display control block
2
2     IF NULLPARAMETER(DISPLAY_ID)
2     THEN
2         SIGNAL_STOP ( SMG$_WRONUMARG);  ! Display id has to be there
2
2 !+
2 ! The next few lines of code validate the supplied device id.  It
2 ! does precisely what $SMG$GET_DCB  does except it signals where that
2 ! macro would return a status.
2 !-
2     DCB = ..DISPLAY_ID;                     ! assume it is correct
2
2     IF .DCB [DCB_L_DID] NEQ ..DISPLAY_ID
2     THEN
2         SIGNAL_STOP (SMG$_INVDIS_ID);   ! Not pointing to one of our
```

```
;  503     0760  2                                          ! control blocks
;  504     0761  2
;  505     0762  2      IF .DCB [DCB_B_STRUCT_TYPE] NEQ DCB_K_STRUCT_TYPE
;  506     0763  2      THEN
;  507     0764  2          SIGNAL_STOP (SMG$_INVDIS_ID);     ! Not pointing to a DCB
;  508     0765  2                                            ! block
;  509     0766  2
;  510     0767  2      RETURN (.DCB [DCB_W_CURSOR_ROW]);     ! Give them what they asked for
;  511     0768  2
;  512     0769  1      END;                              ! End of routine SMG$CURSOR_ROW



                                    001C 00000         .ENTRY  SMG$CURSOR_ROW, Save R2,R3,R4                    ; 0704
              54 00000000G  00    9E 00002         MOVAB   SMG$_INVDIS_ID, R4                               ;
              53 00000000G  00    9E 00009         MOVAB   LIB$STOP, R3                                     ;
                            6C    95 00010         TSTB    (AP)                                             ; 0746
                            05    13 00012         BEQL    1$                                               ;
                     04     AC    D5 00014         TSTL    4(AP)                                            ;
                            09    12 00017         BNEQ    2$                                               ;
                 00000000G  00    9F 00019 1$:     PUSHAB  SMG$_WRONUMARG                                   ; 0748
              63            01    FB 0001F         CALLS   #1, LIB$STOP                                     ;
              52     04     BC    D0 00022 2$:     MOVL    @DISPLAY_ID, DCB                                 ; 0755
         04   BC     38     A2    D1 00026         CMPL    56(DCB), @DISPLAY_ID                             ; 0757
                            05    13 0002B         BEQL    3$                                               ;
                            54    DD 0002D         PUSHL   R4                                               ; 0759
              63            01    FB 0002F         CALLS   #1, LIB$STOP                                     ;
              11     44     A2    91 00032 3$:     CMPB    68(DCB), #17                                     ; 0762
                            05    13 00036         BEQL    4$                                               ;
                            54    DD 00038         PUSHL   R4                                               ; 0764
              63            01    FB 0003A         CALLS   #1, LIB$STOP                                     ;
              50     28     A2    3C 0003D 4$:     MOVZWL  40(DCB), R0                                      ; 0767
                                  04 00041         RET                                                      ; 0769

; Routine Size:  66 bytes,    Routine Base:  _SMG$CODE + 014F


;  513     0770  1 !<BLF/PAGE>
```

```
1 %SBTTL 'SMG$DELETE_CHARS - Delete Characters'
1 GLOBAL ROUTINE SMG$DELETE_CHARS (
1                                   DISPLAY_ID,
1                                   NUM_CHAR,
1                                   ROW,
1                                   COL
1                                  ) =
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine deletes the number of characters specified starting
1 !       at the row/column specified.
1 !
1 !       Remaining characters on the line to the right of the deleted
1 !       characters are shifted to the left to occupy the vacated
1 !       space(s).
1 !
1 !       The internal display cursor position is left at the specified
1 !       start position after the operation.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !
1 !       ret_status.wlc.v = SMG$DELETE_CHARS (
1 !                                       DISPLAY_ID.rl.r,
1 !                                       NUM_CHAR.rl.r,
1 !                                       ROW.rl.r,
1 !                                       COL.rl.r)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       DISPLAY_ID.rl.r         Display id of desired virtual display.
1 !
1 !       NUM_CHAR.rl.r           Number of characters to be deleted
1 !
1 !       ROW.rl.r                Row position at which to start deletion.
1 !
1 !       COL.rl.r                Column position at which to start
1 !                               deletion.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !       SS$_NORMAL      Normal successful completion
1 !       SMG$_INVROW     Invalid row specification-- outside virtual
1 !                                               display.
1 !       SMG$_INVCOL     Invalid column specification -- outside virtual
1 !                                                       display.
1 !       SMG$_INVDIS_ID  Invalid display id.
1 !       SMG$_WRONUMARG  Wrong number of arguments.
```

```
1 !        SMG$_INVARG      Number of characters specified starting at
1 !                         specified row and column extends outside of
1 !                         virtual display.
1 !
1 ! SIDE EFFECTS:
1 !
1 !        NONE
1 !--
2    BEGIN
2    LOCAL
2        NUM_TO_MOVE,                ! No. of char positions to move left
2
2        DCB : REF $DCB_DECL;        ! Addr of display control block
2
2
2    $SMG$VALIDATE_ARGCOUNT (4,4);
2
2    $SMG$GET_DCB ( .DISPLAY_ID, DCB);     ! Get addr of display control
2
2 !+
2 ! Validity check row and column specified
2 !-
2
2    $SMG$VALIDATE_ROW_COL ( ..ROW, ..COL);
2
2 !+
2 ! Call SMG$$SCROLL_AREA to do the work.  It will shift the text left and
2 ! blank fill the vacated space.
2 !-
2
2    NUM_TO_MOVE = ..NUM_CHAR;             ! Assume number is ok
2    IF ..COL + ..NUM_CHAR -1 GTR .DCB[DCB_W_NO_COLS]
2    THEN                                  ! Truncate to what's left on line
2        NUM_TO_MOVE = .DCB [DCB_W_NO_COLS] - ..COL + 1;
2
2    SMG$$SCROLL_AREA (.DCB, ..ROW, ..COL,
2                      1, .DCB [DCB_W_NO_COLS],
2                      SMG$M_LEFT, .NUM_TO_MOVE);
2
2 !+
2 ! Make sure cursor points to first requested delete position.
2 !-
2
2    DCB [DCB_W_CURSOR_ROW] = ..ROW;
2    DCB [DCB_W_CURSOR_COL] = ..COL;
2    DCB [DCB_V_FULL] = 0;                 ! turn off scrolling
2
2 !+
2 ! See if this change should be reflected on the screen.
2 !-
2
3    RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
3                                       SMG$C_DELETE_CHARS,
2                                       ..ROW));
1    END;                          ! End of routine SMG$DELETE_CHARS
```

```
                                     001C 00000        .ENTRY  SMG$DELETE_CHARS, Save R2,R3,R4               : 0772
                  04            6C   91 00002          CMPB    (AP), #4                                      : 0843
                                08   13 00005          BEQL    1$                                            :
                  50 00000000G  8F   D0 00007          MOVL    #SMG$_WRONUMARG, R0                           :
                                     04 0000E          RET                                                   :
                  50       04   BC   D0 0000F 1$:      MOVL    @DISPLAY_ID, R0                               : 0845
              04  BC       38   A0   D1 00013          CMPL    56(R0), @DISPLAY_ID                           :
                                06   12 00018          BNEQ    2$                                            :
                  11       44   A0   91 0001A          CMPB    68(R0), #17                                   :
                                08   13 0001E          BEQL    3$                                            :
                  50 00000000G  8F   D0 00020 2$:      MOVL    #SMG$_INVDIS_ID, R0                           :
                                     04 00027          RET                                                   :
                  52       04   BC   D0 00028 3$:      MOVL    @DISPLAY_ID, DCB                              :
                  54       0C   BC   D0 0002C          MOVL    @ROW, R4                                      : 0851
                                08   15 00030          BLEQ    4$                                            :
    54       02   A2  10        00   ED 00032          CMPZV   #0, #16, 2(DCB), R4                           :
                                08   18 00038          BGEQ    5$                                            :
                  50 00000000G  8F   D0 0003A 4$:      MOVL    #SMG$_INVROW, R0                              :
                                     04 00041          RET                                                   :
                  53       10   BC   D0 00042 5$:      MOVL    @COL, R3                                      :
                                08   15 00046          BLEQ    6$                                            :
    53       06   A2  10        00   ED 00048          CMPZV   #0, #16, 6(DCB), R3                           :
                                08   18 0004E          BGEQ    7$                                            :
                  50 00000000G  8F   D0 00050 6$:      MOVL    #SMG$_INVCOL, R0                              :
                                     04 00057          RET                                                   :
                  51       08   BC   D0 00058 7$:      MOVL    @NUM_CHAR, R1                                 : 0858
                  50            51   D0 0005C          MOVL    R1, NUM_TO_MOVE                               :
                  51       FF A143   9E 0005F          MOVAB   -1(R1)[R3], R1                                : 0859
    51       06   A2  10        00   ED 00064          CMPZV   #0, #16, 6(DCB), R1                           :
                                0B   18 0006A          BGEQ    8$                                            :
                  51       06   A2   3C 0006C          MOVZWL  6(DCB), R1                                    : 0861
                  51            53   C2 00070          SUBL2   R3, R1                                        :
                  50       01   A1   9E 00073          MOVAB   1(R1), NUM_TO_MOVE                            :
                                50   DD 00077 8$:      PUSHL   NUM_TO_MOVE                                   : 0865
                                08   DD 00079          PUSHL   #8                                            : 0863
                  7E       06   A2   3C 0007B          MOVZWL  6(DCB), -(SP)                                 : 0864
                                01   DD 0007F          PUSHL   #1                                            : 0863
                                53   DD 00081          PUSHL   R3                                            :
                                14   BB 00083          PUSHR   #^M<R2,R4>                                    :
          00000000G  00         07   FB 00085          CALLS   #7, SMG$$SCROLL_AREA                          :
                 28  A2         54   B0 0008C          MOVW    R4, 40(DCB)                                   : 0871
                 2A  A2         53   B0 00090          MOVW    R3, 42(DCB)                                   : 0872
                 34  A2         01   8A 00094          BICB2   #1, 52(DCB)                                   : 0873
                                54   DD 00098          PUSHL   R4                                            : 0881
                                0B   DD 0009A          PUSHL   #11                                           : 0879
                                52   DD 0009C          PUSHL   DCB                                           :
          00000000G  00         03   FB 0009E          CALLS   #3, SMG$$CHECK_FOR_OUTPUT_DCB                 :
                                     04 000A5          RET                                                   : 0882

; Routine Size:  166 bytes,    Routine Base:  _SMG$CODE + 0191

;  627        0883  1 !<BLF/PAGE>
```

```
1 %SBTTL 'SMG$DELETE_LINE - Delete Line in Display'
1 GLOBAL ROUTINE SMG$DELETE_LINE (
1                                 DISPLAY_ID,
1                                 START_LINE_NO,
1                                 NUM_LINES
1                               ) =
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine deletes one or more lines in a virtual display
1 !       and scrolls up remaining lines into the created space.
1 !
1 !       The internal display cursor position is left at the first column
1 !       of the start line.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = SMG$DELETE_LINE (DISPLAY_ID.rl.r,
1 !                                           START_LINE_NO.rl.r
1 !                                           [,NUM_LINES.rl.r])
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       DISPLAY_ID      Display id of desired virtual display.
1 !
1 !       START_LINE_NO.rl.r
1 !                       Address of line number where deleting/scrolling
1 !                       starts.
1 !
1 !       NUM_LINES.rl.r  Optional.  Number of lines to delete/scroll.
1 !                       If not specified, defaults to 1.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !       SS$_NORMAL      Normal successful completion
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !--
2     BEGIN
2     BUILTIN
2       NULLPARAMETER;
2
2     LOCAL
2       DCB : REF $DCB_DECL,    ! addr of display control block
2       WORK_NUM_LINES;         ! working number of lines
2
2     $SMG$VALIDATE_ARGCOUNT (2,3);
```

```
2           $SMG$GET_DCB (.DISPLAY_ID, DCB);
2                                       ! get address of display control block
2
2       !+
2       ! Validate arguments and set up working temporaries.
2       !-
2
2           IF ..START_LINE_NO LSS .DCB [DCB_W_ROW_START] OR
2               ..START_LINE_NO GTR .DCB [DCB_W_NO_ROWS]
2           THEN
2               RETURN (SMG$_INVROW);
2
2           IF NOT NULLPARAMETER (NUM_LINES)
2           THEN
3               BEGIN
3               WORK_NUM_LINES = ..NUM_LINES;
3               IF ..START_LINE_NO + .WORK_NUM_LINES -1 GTR .DCB [DCB_W_NO_ROWS]
3               THEN
3                   RETURN (SMG$_INVARG);
3               END
2           ELSE
2               WORK_NUM_LINES = 1;
2
2       !+
2       ! Reset the line characteristics in case the line was previously
2       ! double high or wide.
2       !-
2
3           BEGIN
3           BIND
3               LINE_CHAR = .DCB [DCB_A_LINE_CHAR];
3           MAP
3               LINE_CHAR : VECTOR [,BYTE];
3
3           INCR ROW_NUM FROM 1 TO .WORK_NUM_LINES DO
3               LINE_CHAR [.ROW_NUM] = 0;
2           END;
2
2       !+
2       ! Call SMG$$SCROLL_AREA to do the work.  It will move lines up or down,
2       ! overwriting the lines that are to be deleted.
2       !-
2
2           SMG$$SCROLL_AREA (.DCB,
2                             ..START_LINE_NO,
2                             .DCB [DCB_W_COL_START],
2                             (.DCB [DCB_W_NO_ROWS] - ..START_LINE_NO + 1),
2                             .DCB [DCB_W_NO_COLS],
2                             SMG$M_UP,
2                             .WORK_NUM_LINES);
2
2       !+
2       ! Set cursor position to start of first line that was 'deleted'.
2       !-
2
2           DCB [DCB_W_CURSOR_ROW] = ..START_LINE_NO;
```

```
;  743    0998 2        DCB [DCB_W_CURSOR_COL] = 1;
;  744    0999 2        DCB [DCB_V_FULL] = 0;                  ! mark that display is not full -
;  745    1000 2                                               ! there is at least 1 line available
;  746    1001 2    !+
;  747    1002 2    ! See if this change should be reflected on the screen right away.
;  748    1003 2    !-
;  749    1004 2
;  750    1005 3        RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
;  751    1006 3                                           SMG$C_DELETE_LINE,
;  752    1007 2                                           ..START_LINE_NO));
;  753    1008 2
;  754    1009 1        END;                                   ! end of routine SMG$DELETE_LINE


                                      000C 00000          .ENTRY  SMG$DELETE_LINE, Save R2,R3                 ; 0885
              50        6C         02   83 00002          SUBB3   #2, (AP), DIFF                              ; 0940
                        01         50   91 00006          CMPB    DIFF, #1                                    ;
                                   08   1B 00009          BLEQU   1$                                          ;
                        50 00000000G    8F D0 0000B       MOVL    #SMG$_WRONUMARG, R0                         ;
                                        04 00012          RET                                                 ;
                        50         04   BC D0 00013 1$:   MOVL    @DISPLAY_ID, R0                             ; 0942
                   04   BC         38   A0 D1 00017       CMPL    56(R0), @DISPLAY_ID                         ;
                                   06   12 0001C          BNEQ    2$                                          ;
                        11         44   A0 91 0001E       CMPB    68(R0), #17                                 ;
                                   08   13 00022          BEQL    3$                                          ;
                        50 00000000G    8F D0 00024 2$:   MOVL    #SMG$_INVDIS_ID, R0                         ;
                                        04 0002B          RET                                                 ;
                        52         04   BC D0 0002C 3$:   MOVL    @DISPLAY_ID, DCB                            ;
                        53         08   BC D0 00030       MOVL    @START_LINE_NO, R3                          ; 0949
   53           62      10              00 ED 00034       CMPZV   #0, #16, (DCB), R3                          ;
                                   08   14 00039          BGTR    4$                                          ;
   53      02   A2      10              00 ED 0003B       CMPZV   #0, #16, 2(DCB), R3                         ; 0950
                                   08   18 00041          BGEQ    5$                                          ;
                        50 00000000G    00 9E 00043 4$:   MOVAB   SMG$_INVROW, R0                             ; 0952
                                        04 0004A          RET                                                 ;
                        03         6C   91 0004B 5$:      CMPB    (AP), #3                                    ; 0954
                                   1E   1F 0004E          BLSSU   6$                                          ;
                                   0C   AC D5 00050       TSTL    12(AP)                                      ;
                                   19   13 00053          BEQL    6$                                          ;
                        51         0C   BC D0 00055       MOVL    @NUM_LINES, WORK_NUM_LINES                  ; 0957
                        50      FF A143 9E 00059          MOVAB   -1(WORK_NUM_LINES)[R3], R0                  ; 0958
   50      02   A2      10              00 ED 0005E       CMPZV   #0, #16, 2(DCB), R0                         ;
                                   0B   18 00064          BGEQ    7$                                          ;
                        50 00000000G    00 9E 00066       MOVAB   SMG$_INVARG, R0                             ; 0960
                                        04 0006D          RET                                                 ;
                        51         01   D0 0006E 6$:      MOVL    #1, WORK_NUM_LINES                          ; 0963
                                   50   D4 00071 7$:      CLRL    ROW_NUM                                     ; 0976
                                   04   11 00073          BRB     9$                                          ;
                              4C B240   94 00075 8$:      CLRB    @76(DCB)[ROW_NUM]                           ; 0977
                F8      50         51   F3 00079 9$:      AOBLEQ  WORK_NUM_LINES, ROW_NUM, 8$                 ;
                                   51   DD 0007D          PUSHL   WORK_NUM_LINES                              ; 0991
                                   01   DD 0007F          PUSHL   #1                                          ; 0985
                        7E         06   A2 3C 00081       MOVZWL  6(DCB), -(SP)                               ; 0989
                        50         02   A2 3C 00085       MOVZWL  2(DCB), R0                                  ; 0988
```

```
                 50          53 C2 00089     SUBL2   R3, R0                              ;
                         01  A0 9F 0008C     PUSHAB  1(R0)                               ;
                 7E      04  A2 3C 0008F     MOVZWL  4(DCB), -(SP)                       ; 0987
                             0C BB 00093     PUSHR   #^M<R2,R3>                          ; 0985
          00000000G  00      07 FB 00095     CALLS   #7, SMG$$SCROLL_AREA                ;
                 28  A2      53 B0 0009C     MOVW    R3, 40(DCB)                         ; 0997
                 2A  A2      01 B0 000A0     MOVW    #1, 42(DCB)                         ; 0998
                 34  A2      01 8A 000A4     BICB2   #1, 52(DCB)                         ; 0999
                             53 DD 000A8     PUSHL   R3                                  ; 1007
                             18 DD 000AA     PUSHL   #24                                 ; 1005
                             52 DD 000AC     PUSHL   DCB                                 ;
          00000000G  00      03 FB 000AE     CALLS   #3, SMG$$CHECK_FOR_OUTPUT_DCB       ;
                                04 000B5     RET                                         ; 1009

; Routine Size:  182 bytes,    Routine Base:  _SMG$CODE + 0237

;  755         1010  1 !<BLF/PAGE>
```

```
1 %SBTTL 'SMG$ERASE_CHARS - Erase Characters in Display'
1 GLOBAL ROUTINE SMG$ERASE_CHARS (
1                                   DISPLAY_ID,
1                                   NUM_CHARS,
1                                   LINE_NO,
1                                   COL_NO
1                                 ) =
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine erases characters in a display by replacing them
1 !       with blanks.  Remaining text is not moved over.
1 !
1 !       Erasing is limited to the specified line.  If NUM_CHARS would
1 !       cause characters past the end of line to be erased, it is ignored
1 !       and only to the end of the line is blanked.
1 !
1 !       The internal display cursor position is set to the first free
1 !       position following the erase.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = SMG$ERASE_CHARS (DISPLAY_ID.rl.r,
1 !                                           NUM_CHARS.rl.r
1 !                                           LINE_NO.rl.r,
1 !                                           COL_NO.rl.r)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       DISPLAY_ID.rl.r Display id of desired virtual display.
1 !
1 !       NUM_CHARS.rl.r  Number of characters to erase.
1 !
1 !       LINE_NO.rl.r    Address of line number where erasing starts.
1 !
1 !       COL_NO.rl.r     Address of column number where erasing starts.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !       SS$_NORMAL      Normal successful completion
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !--
1
2   BEGIN
2
2   LOCAL
```

```
2           DCB : REF $DCB_DECL,        ! Addr of virtual display
2                                       ! control block.
2           DEST_COL,                   ! destination column number
2           START_INDEX,                ! starting linear index into buffers
2           WORK_NUM_CHARS;             ! working number of characters
2
2       $SMG$VALIDATE_ARGCOUNT (4, 4);
2
2       $SMG$GET_DCB ( .DISPLAY_ID, DCB);       ! Get address of virtual
2                                               ! display control block.
2   !+
2   ! Validity check the parameters.  Don't try to erase more characters
2   ! than are on the line.
2   !-
2
2       $SMG$VALIDATE_ROW_COL ( ..LINE_NO, ..COL_NO);
2
2       WORK_NUM_CHARS = ..NUM_CHARS;
2       DEST_COL = ..COL_NO + ..NUM_CHARS - 1;
2       IF .DEST_COL GTR .DCB [DCB_W_NO_COLS]
2       THEN
2           WORK_NUM_CHARS = .DCB [DCB_W_NO_COLS] - ..COL_NO + 1;
2
2   !+
2   ! Put blanks in the text buffer, the default attribute byte into the
2   ! attribute buffer, and the default character set byte into the character
2   ! buffer.
2   !-
2
2       START_INDEX = $SMG$LINEAR (..LINE_NO, ..COL_NO);
2
2       $SMG$BLANK_FILL_DCB (.WORK_NUM_CHARS, .START_INDEX);
2
2   !+
2   ! Virtual display buffers have been erased.  Init cursor to first
2   ! free position.
2   !-
2
2       DCB [DCB_W_CURSOR_ROW] = ..LINE_NO;
2       DCB [DCB_W_CURSOR_COL] = ..COL_NO;
2       DCB [DCB_V_FULL] = 0;                   ! turn off scrolling
2
2   !+
2   ! Call the output routine to see if this change in the virtual display
2   ! should be reflected on the screen immediately.
2   !-
2
2       RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
3                                           SMG$C_ERASE_CHARS,
2                                           ..LINE_NO));
1       END;                                ! End of routine SMG$ERASE_CHARS


                        0FFC 00000      .ENTRY  SMG$ERASE_CHARS, Save R2,R3,R4,R5,R6,R7,R8,-; 1012
```

```
                                                           R9,R10,R11                        ;
                     5E           04  C2 00002      SUBL2  #4, SP                            ;
                     04           6C  91 00005      CMPB   (AP), #4                          ; 1074
                                  08  13 00008      BEQL   1$                                ;
                     50 00000000G 8F  D0 0000A      MOVL   #SMG$_WRONUMARG, R0               ;
                                  04  00011         RET                                      ;
                     50      04   BC  D0 00012 1$:  MOVL   @DISPLAY_ID, R0                   ; 1076
                04   BC      38   A0  D1 00016      CMPL   56(R0), @DISPLAY_ID               ;
                                  06  12 0001B      BNEQ   2$                                ;
                     11      44   A0  91 0001D      CMPB   68(R0), #17                       ;
                                  08  13 00021      BEQL   3$                                ;
                     50 00000000G 8F  D0 00023 2$:  MOVL   #SMG$_INVDIS_ID, R0               ;
                                  04  0002A         RET                                      ;
                     56      04   BC  D0 0002B 3$:  MOVL   @DISPLAY_ID, DCB                  ;
                     6E      0C   BC  D0 0002F      MOVL   @LINE_NO, (SP)                    ; 1083
                                  08  15 00033      BLEQ   4$                                ;
 6E      02   A6     10           00  ED 00035      CMPZV  #0, #16, 2(DCB), (SP)             ;
                                  08  18 0003B      BGEQ   5$                                ;
                     50 00000000G 8F  D0 0003D 4$:  MOVL   #SMG$_INVROW, R0                  ;
                                  04  00044         RET                                      ;
                     5A      10   BC  D0 00045 5$:  MOVL   @COL_NO, R10                      ;
                                  08  15 00049      BLEQ   6$                                ;
 5A      06   A6     10           00  ED 0004B      CMPZV  #0, #16, 6(DCB), R10              ;
                                  08  18 00051      BGEQ   7$                                ;
                     50 00000000G 8F  D0 00053 6$:  MOVL   #SMG$_INVCOL, R0                  ;
                                  04  0005A         RET                                      ;
                     50      08   BC  D0 0005B 7$:  MOVL   @NUM_CHARS, R0                    ; 1085
                     57           50  D0 0005F      MOVL   R0, WORK_NUM_CHARS                ;
                     50   FF A04A     9E 00062      MOVAB  -1(R0)[R10], DEST_COL             ; 1086
 50      06   A6     10           00  ED 00067      CMPZV  #0, #16, 6(DCB), DEST_COL         ; 1087
                                  0B  18 0006D      BGEQ   8$                                ;
                     50      06   A6  3C 0006F      MOVZWL 6(DCB), R0                        ; 1089
                     50           5A  C2 00073      SUBL2  R10, R0                           ;
                     57      01   A0  9E 00076      MOVAB  1(R0), WORK_NUM_CHARS             ;
              50     6E           01  C3 0007A 8$:  SUBL3  #1, (SP), R0                      ; 1097
                     51      06   A6  3C 0007E      MOVZWL 6(DCB), R1                        ;
                     50           51  C4 00082      MULL2  R1, R0                            ;
                     5B   FF AA40     9E 00085      MOVAB  -1(R10)[R0], START_INDEX          ;
                     50      10   A6  D0 0008A      MOVL   16(DCB), TEXT_BUF                 ; 1099
                     58      14   A6  7D 0008E      MOVQ   20(DCB), ATTR_BUF                 ;
 57           20     6E           00  2C 00092      MOVC5  #0, (SP), #32, WORK_NUM_CHARS, -  ;
                              6B40       00097             (START_INDEX)[TEXT_BUF]           ;
 57      2E   A6     6E           00  2C 00099      MOVC5  #0, (SP), 46(DCB), WORK_NUM_CHARS, - ;
                              6B48       0009F             (START_INDEX)[ATTR_BUF]           ;
                                  59  D5 000A1      TSTL   CHAR_BUF                          ;
                                  08  13 000A3      BEQL   9$                                ;
 57      30   A6     6E           00  2C 000A5      MOVC5  #0, (SP), 48(DCB), WORK_NUM_CHARS, - ;
                              6B49       000AB             (START_INDEX)[CHAR_BUF]           ;
              28     A6           6E  B0 000AD 9$:  MOVW   (SP), 40(DCB)                     ; 1106
              2A     A6           5A  B0 000B1      MOVW   R10, 42(DCB)                      ; 1107
              34     A6           01  8A 000B5      BICB2  #1, 52(DCB)                       ; 1108
                                  6E  DD 000B9      PUSHL  (SP)                              ; 1117
                                  19  DD 000BB      PUSHL  #25                               ; 1115
                                  56  DD 000BD      PUSHL  DCB                               ;
          00000000G  00           03  FB 000BF      CALLS  #3, SMG$$CHECK_FOR_OUTPUT_DCB     ;
                                  04  000C6         RET                                      ; 1118
```

```
; Routine Size:  199 bytes,    Routine Base:  _SMG$CODE + 02ED

;  865          1119  1 !<BLF/PAGE>
```

```
1 %SBTTL 'SMG$ERASE_DISPLAY - Erase Display'
1 GLOBAL ROUTINE SMG$ERASE_DISPLAY (
1                                       DISPLAY_ID,
1                                       START_LINE_NO,
1                                       START_COL_NO,
1                                       END_LINE_NO,
1                                       END_COL_NO
1                                        ) =
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine causes the specified display to be erased from the
1 !       specified start position to the specified end position.  If omitted,
1 !       the start position defaults to (1,1).  If the end position is omitted,
1 !       it defaults to the last character position in the display.  Thus the
1 !       entire display is erased if only the display_id is passed.
1 !
1 !       The internal display cursor position is set to the first free
1 !       position (1,1) following the erase.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = SMG$ERASE_DISPLAY (DISPLAY_ID.rl.r
1 !                                             [,START_LINE_NO.rl.r,
1 !                                             START_COL_NO.rl.r]
1 !                                             [,END_LINE_NO.rl.r,
1 !                                             END_COL_NO.rl.r])
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       DISPLAY_ID.rl.r         Display id of the desired virtual display.
1 !
1 !       START_LINE_NO.rl.r      Optional.  Address of line number at which to
1 !                               start erasing.  If omitted, START_COL_NO will be
1 !                               ignored as well and whole display will be
1 !                               erased.
1 !
1 !       START_COL_NO.rl.r       Optional.  Address of Column number at which
1 !                               to start erasing.  If omitted, START_LINE_NO
1 !                               will be ignored as well and entire display will
1 !                               be erased.
1 !
1 !       END_LINE_NO.rl.r        Optional.  Address of line number at which to
1 !                               end erasing.  If omitted, END_COL_NO is also
1 !                               ignored and to the end of the display is erased.
1 !
1 !       END_COL_NO.rl.r         Optional.  Address of column number at which to
1 !                               end erasing.  If omitted, END_LINE_NO is also
1 !                               ignored and to the end of the display is erased.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
```

```
1 !
1 ! COMPLETION STATUS:
1 !
1 !       SS$_NORMAL      Normal successful completion
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !--
1
2     BEGIN
2
2     LOCAL
2         DCB : REF $DCB_DECL,    ! Addr of virtual display
2                                 ! control block.
2         LINE,                   ! temp row
2         COL,                    ! temp column
2         END_ROW,                ! temp end row
2         END_COLUMN,             ! temp end col
2         START_INDEX,            ! starting linear index into buffers
2         END_INDEX,              ! ending linear index into buffers
2         NUM_CHARS;              ! number of characters to put into buffers
2
2     BUILTIN
2         ACTUALCOUNT,
2         NULLPARAMETER;
2
2     LITERAL
2         K_START_LINE_ARG = 2,   ! argument number of start_line_no
2         K_END_LINE_ARG   = 4,   ! argument number of end_line_no
2         K_END_COL_ARG    = 5;   ! argument number of end_col_no
2
2 !+
2 ! Validity check the number of parameters.
2 !-
2
2     $SMG$VALIDATE_ARGCOUNT (1, 5);
2
2     $SMG$GET_DCB ( .DISPLAY_ID, DCB);   ! Get address of virtual
2                                         ! display control block.
2 !+
2 ! Calculate the linear start index, based on whether a cursor position
2 ! was specified.
2 !-
2
2     IF ACTUALCOUNT () LSS K_START_LINE_ARG  OR          ! Quick check
2        NULLPARAMETER (START_LINE_NO)        OR          ! More elaborate checks
2        NULLPARAMETER (START_COL_NO)
2     THEN                        ! default start position
3         BEGIN
3         START_INDEX = 0;
3         LINE = 1;
3         COL = 1;
3         END
2     ELSE                        ! use specified position
3         BEGIN
3         $SMG$VALIDATE_ROW_COL ( ..START_LINE_NO, ..START_COL_NO);
```

```
3               START_INDEX = $SMG$LINEAR (..START_LINE_NO, ..START_COL_NO);
3               LINE = ..START_LINE_NO;
3               COL = ..START_COL_NO;
2               END;
2
2           IF ACTUALCOUNT() GEQ K_END_LINE_ARG     AND   ! Last 2 exist and are not null
2           NOT NULLPARAMETER (END_LINE_NO)         AND
2           NOT NULLPARAMETER (END_COL_NO)
2           THEN
3               BEGIN                       ! use specified end position
3               $SMG$VALIDATE_ROW_COL (..END_LINE_NO, ..END_COL_NO);
3               END_ROW = ..END_LINE_NO;
3               END_COLUMN = ..END_COL_NO;
3               END
2           ELSE                            ! use default end position
3               BEGIN
3               END_ROW = .DCB [DCB_W_NO_ROWS];
3               END_COLUMN = .DCB [DCB_W_NO_COLS];
2               END;
2
2           END_INDEX = $SMG$LINEAR (.END_ROW, .END_COLUMN);
2
2
2       !+
2       ! Reset the line characteristics in case the line was previously
2       ! double high or wide.
2       !-
2
3           BEGIN
3           BIND
3               LINE_CHAR = .DCB [DCB_A_LINE_CHAR];
3           MAP
3               LINE_CHAR : VECTOR [,BYTE];
3           INCR ROW_NUM FROM .LINE TO (.END_ROW - .LINE) DO
3               LINE_CHAR [.ROW_NUM] = 0;
2           END;
2
2       !+
2       ! Put blanks in the text buffer, the default attribute byte into the
2       ! attribute buffer, and the default character set byte into the character
2       ! buffer.
2       !-
2
2           NUM_CHARS = .END_INDEX - .START_INDEX + 1;
2
2           $SMG$BLANK_FILL_DCB (.NUM_CHARS, .START_INDEX);
2
2       !+
2       ! Virtual display buffers have been erased.  Init the cursor to the first
2       ! free position.
2       !-
2
2           DCB [DCB_W_CURSOR_ROW] = .LINE;
2           DCB [DCB_W_CURSOR_COL] = .COL;
2
2           DCB [DCB_V_FULL] = 0;           ! flag that some lines are free
2
```

```
; 1038          1291  2 !+
; 1039          1292  2 ! Call the output routine to see if this change in the virtual display
; 1040          1293  2 ! should be reflected on the screen immediately.
; 1041          1294  2 !-
; 1042          1295  2
; 1043          1296  3     RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
; 1044          1297  2                                       SMG$C_ERASE_DISPLAY));
; 1045          1298  1     END;                          ! End of routine SMG$ERASE_DISPLAY



                                          0FFC 00000          .ENTRY   SMG$ERASE_DISPLAY, Save R2,R3,R4,R5,R6,R7,-  ; 1121
                                                                       R8,R9,R10,R11                                  ;
                              5E        04  C2 00002          SUBL2    #4, SP                                         ;
                  50          6C        01  83 00005          SUBB3    #1, (AP), DIFF                                 ; 1213
                              04        50  91 00009          CMPB     DIFF, #4                                       ;
                                        08  1B 0000C          BLEQU    1$                                             ;
                              50 00000000G  8F  D0 0000E       MOVL     #SMG$_WRONUMARG, R0                           ;
                                        04  00015             RET                                                     ;
                              50     04  BC  D0 00016 1$:      MOVL     @DISPLAY_ID, R0                               ; 1215
                       04     BC     38  A0  D1 0001A          CMPL     56(R0), @DISPLAY_ID                           ;
                                        06  12 0001F          BNEQ     2$                                             ;
                              11     44  A0  91 00021          CMPB     68(R0), #17                                   ;
                                        08  13 00025          BEQL     3$                                             ;
                              50 00000000G  8F  D0 00027 2$:   MOVL     #SMG$_INVDIS_ID, R0                           ;
                                        04  0002E             RET                                                     ;
                              58     04  BC  D0 0002F 3$:      MOVL     @DISPLAY_ID, DCB                              ;
                              02        6C  91 00033          CMPB     (AP), #2                                       ; 1222
                                        0F  1F 00036          BLSSU    4$                                             ;
                                     08  AC  D5 00038          TSTL     8(AP)                                         ; 1223
                                        0A  13 0003B          BEQL     4$                                             ;
                              03        6C  91 0003D          CMPB     (AP), #3                                       ; 1224
                                        05  1F 00040          BLSSU    4$                                             ;
                                     0C  AC  D5 00042          TSTL     12(AP)                                        ;
                                        0A  12 00045          BNEQ     5$                                             ;
                                        5B  D4 00047 4$:      CLRL     START_INDEX                                    ; 1227
                              57        01  D0 00049          MOVL     #1, LINE                                       ; 1228
                              6E        01  D0 0004C          MOVL     #1, COL                                        ; 1229
                                        42  11 0004F          BRB      10$                                            ; 1222
                              52     08  BC  D0 00051 5$:      MOVL     @START_LINE_NO, R2                            ; 1233
                                        08  15 00055          BLEQ     6$                                             ;
       52       02     A8     10        00  ED 00057          CMPZV    #0, #16, 2(DCB), R2                            ;
                                        08  18 0005D          BGEQ     7$                                             ;
                              50 00000000G  8F  D0 0005F 6$:   MOVL     #SMG$_INVROW, R0                              ;
                                        04  00066             RET                                                     ;
                              51     0C  BC  D0 00067 7$:      MOVL     @START_COL_NO, R1                             ;
                                        08  15 0006B          BLEQ     8$                                             ;
       51       06     A8     10        00  ED 0006D          CMPZV    #0, #16, 6(DCB), R1                            ;
                                        08  18 00073          BGEQ     9$                                             ;
                              50 00000000G  8F  D0 00075 8$:   MOVL     #SMG$_INVCOL, R0                              ;
                                        04  0007C             RET                                                     ;
                              50     FF  A2  9E 0007D 9$:      MOVAB    -1(R2), R0                                    ; 1234
                              53     06  A8  3C 00081          MOVZWL   6(DCB), R3                                    ;
                              50        53  C4 00085          MULL2    R3, R0                                         ;
                              5B   FF A140  9E 00088          MOVAB    -1(R1)[R0], START_INDEX                        ;
```

```
                            57           52 D0 0008D        MOVL    R2, LINE                                  ; 1235
                            6E           51 D0 00090        MOVL    R1, COL                                   ; 1236
                            04           6C 91 00093 10$:   CMPB    (AP), #4                                  ; 1239
                                         40 1F 00096        BLSSU   15$                                       ;
                                  10     AC D5 00098        TSTL    16(AP)                                    ; 1240
                                         3B 13 0009B        BEQL    15$                                       ;
                            05           6C 91 0009D        CMPB    (AP), #5                                  ; 1241
                                         36 1F 000A0        BLSSU   15$                                       ;
                                  14     AC D5 000A2        TSTL    20(AP)                                    ;
                                         31 13 000A5        BEQL    15$                                       ;
                            50    10     BC D0 000A7        MOVL    @END_LINE_NO, R0                          ; 1244
                                         08 15 000AB        BLEQ    11$                                       ;
        50      02   A8           10     00 ED 000AD        CMPZV   #0, #16, 2(DCB), R0                       ;
                                         08 18 000B3        BGEQ    12$                                       ;
                            50 00000000G 8F D0 000B5 11$:   MOVL    #SMG$_INVROW, R0                          ;
                                         04 000BC           RET                                               ;
                            51    14     BC D0 000BD 12$:   MOVL    @END_COL_NO, R1                           ;
                                         08 15 000C1        BLEQ    13$                                       ;
        51      06   A8           10     00 ED 000C3        CMPZV   #0, #16, 6(DCB), R1                       ;
                                         08 18 000C9        BGEQ    14$                                       ;
                            50 00000000G 8F D0 000CB 13$:   MOVL    #SMG$_INVCOL, R0                          ;
                                         04 000D2           RET                                               ;
                            52           51 D0 000D3 14$:   MOVL    R1, END_COLUMN                            ; 1246
                                         08 11 000D6        BRB     16$                                       ; 1239
                            50    02     A8 3C 000D8 15$:   MOVZWL  2(DCB), END_ROW                           ; 1250
                            52    06     A8 3C 000DC        MOVZWL  6(DCB), END_COLUMN                        ; 1251
                            51    FF     A0 9E 000E0 16$:   MOVAB   -1(R0), R1                                ; 1254
                            53    06     A8 3C 000E4        MOVZWL  6(DCB), R3                                ;
                            51           53 C4 000E8        MULL2   R3, R1                                    ;
                            56    FF A241   9E 000EB        MOVAB   -1(END_COLUMN)[R1], END_INDEX             ;
                            50           57 C2 000F0        SUBL2   LINE, R0                                  ; 1267
                            51    FF     A7 9E 000F3        MOVAB   -1(R7), ROW_NUM                           ;
                                         04 11 000F7        BRB     18$                                       ;
                                  4C B841   94 000F9 17$:   CLRB    @76(DCB)[ROW_NUM]                         ; 1268
                     F8     51           50 F3 000FD 18$:   AOBLEQ  R0, ROW_NUM, 17$                          ;
                            56           5B C2 00101        SUBL2   START_INDEX, R6                           ; 1277
                                         56 D6 00104        INCL    NUM_CHARS                                 ;
                            50    10     A8 D0 00106        MOVL    16(DCB), TEXT_BUF                         ; 1279
                            59    14     A8 7D 0010A        MOVQ    20(DCB), ATTR_BUF                         ;
        56           20     6E           00 2C 0010E        MOVC5   #0, (SP), #32, NUM_CHARS, (START_INDEX)-  ;
                                       6B40    00113                [TEXT_BUF]                                ;
        56      2E   A8     6E           00 2C 00115        MOVC5   #0, (SP), 46(DCB), NUM_CHARS, (START_INDEX)-;
                                       6B49    0011B                [ATTR_BUF]                                ;
                                         5A D5 0011D        TSTL    CHAR_BUF                                  ;
                                         08 13 0011F        BEQL    19$                                       ;
        56      30   A8     6E           00 2C 00121        MOVC5   #0, (SP), 48(DCB), NUM_CHARS, (START_INDEX)-;
                                       6B4A    00127                [CHAR_BUF]                                ;
                     28   A8             57 B0 00129 19$:   MOVW    LINE, 40(DCB)                             ; 1286
                     2A   A8             6E B0 0012D        MOVW    COL, 42(DCB)                              ; 1287
                     34   A8             01 8A 00131        BICB2   #1, 52(DCB)                               ; 1289
                                         0C DD 00135        PUSHL   #12                                       ; 1296
                                         58 DD 00137        PUSHL   DCB                                       ;
              00000000G   00             02 FB 00139        CALLS   #2, SMG$$CHECK_FOR_OUTPUT_DCB             ;
                                         04 00140           RET                                               ; 1298

; Routine Size:  321 bytes,    Routine Base:  _SMG$CODE + 03B4
```

```
; 1046          1299  1 !<BLF/PAGE>
```

```
%SBTTL 'SMG$ERASE_LINE - Erase Line from Display'
GLOBAL ROUTINE SMG$ERASE_LINE (
                                        DISPLAY_ID,
                                        LINE_NO,
                                        COL_NO
                                       ) =
!++
! FUNCTIONAL DESCRIPTION:
!
!       This routine causes the display to be erased from the specified
!       position to the end of the line.  If the position is not
!       specified, the current cursor position on the screen is assumed.
!
!       The internal display cursor position is set to the first free
!       position following the erase.
!
! CALLING SEQUENCE:
!
!       ret_status.wlc.v = SMG$ERASE_LINE (DISPLAY_ID.rl.r,
!                                          [LINE_NO.rl.r,
!                                          COL_NO.rl.r])
!
! FORMAL PARAMETERS:
!
!       DISPLAY_ID.rl.r Display id of desired virtual display.
!
!       LINE_NO.rl.r    Optional.  Address of line number where erasing
!                       starts.  If this argument is omitted, COL_NO is
!                       ignored as well, and current position will be
!                       used.
!
!       COL_NO.rl.r     Optional.  Address of column number where
!                       erasing starts.  If this argument is omitted,
!                       LINE_NO is ignored as well, and current position
!                       will be used.
!
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! COMPLETION STATUS:
!
!       SS$_NORMAL      Normal successful completion
!
! SIDE EFFECTS:
!
!       NONE
!--

    BEGIN

    LOCAL
        DCB : REF $DCB_DECL,    ! Addr of virtual display
```

```
2                                       ! control block.
2           LINE,                       ! temp line
2           COL,                        ! temp column
2           START_INDEX,                ! starting linear index into buffers
2           END_INDEX,                  ! ending linear index into buffers
2           NUM_CHARS;                  ! number of characters to put into buffers
2
2       BUILTIN
2           ACTUALCOUNT,
2           NULLPARAMETER;
2
2   !+
2   ! Validity check the number of parameters.
2   !-
2
2       $SMG$VALIDATE_ARGCOUNT (1, 3);
2
2       $SMG$GET_DCB ( .DISPLAY_ID, DCB);   ! Get address of virtual
2                                           ! display control block.
2
2
2   !+
2   ! Calculate the linear start index, based on whether a cursor position
2   ! was specified.
2   !-
2
2       IF NULLPARAMETER (LINE_NO) OR
2          NULLPARAMETER (COL_NO)
2       THEN                            ! default start position
3           BEGIN
3           LINE = .DCB [DCB_W_CURSOR_ROW];
3           COL = .DCB [DCB_W_CURSOR_COL];
3           END
2       ELSE                            ! use specified position
3           BEGIN
3           $SMG$VALIDATE_ROW_COL ( ..LINE_NO, ..COL_NO);
3           LINE = ..LINE_NO;
3           COL = ..COL_NO;
3           END;
2   !+
2   ! Put blanks in the text buffer, the default attribute byte into the
2   ! attribute buffer, and the default character set byte into the character
2   ! buffer.
2   !-
2
2       START_INDEX = $SMG$LINEAR (.LINE, .COL);
2       END_INDEX = $SMG$LINEAR (.LINE, .DCB [DCB_W_NO_COLS]);
2       NUM_CHARS = .END_INDEX - .START_INDEX + 1;
2
2       $SMG$BLANK_FILL_DCB (.NUM_CHARS, .START_INDEX);
2
2   !+
2   ! Reset the line characteristics in case the line was previously
2   ! double high or wide.
2   !-
2
3       BEGIN
```

```
; 1162      1414  3      BIND
; 1163      1415  3          LINE_CHAR = .DCB [DCB_A_LINE_CHAR];
; 1164      1416  3      MAP
; 1165      1417  3          LINE_CHAR : VECTOR [,BYTE];
; 1166      1418  3      IF .NUM_CHARS EQL .DCB [DCB_W_NO_COLS]
; 1167      1419  3      THEN
; 1168      1420  3          LINE_CHAR [.LINE] = 0;
; 1169      1421  2      END;
; 1170      1422  2
; 1171      1423  2   !+
; 1172      1424  2   ! Virtual display buffers have been erased.  Init cursor to first
; 1173      1425  2   ! free position.
; 1174      1426  2   !-
; 1175      1427  2
; 1176      1428  2      DCB [DCB_W_CURSOR_ROW] = .LINE;
; 1177      1429  2      DCB [DCB_W_CURSOR_COL] = .COL;
; 1178      1430  2      DCB [DCB_V_FULL] = 0;                   ! turn off scrolling
; 1179      1431  2
; 1180      1432  2   !+
; 1181      1433  2   ! Call the output routine to see if this change in the virtual display
; 1182      1434  2   ! should be reflected on the screen immediately.
; 1183      1435  2   !-
; 1184      1436  2
; 1185      1437  3      RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
; 1186      1438  3                                       SMG$C_ERASE_LINE,
; 1187      1439  2                                       .LINE));
; 1188      1440  1      END;                                 ! End of routine SMG$ERASE_LINE



                                  0FFC 00000          .ENTRY  SMG$ERASE_LINE, Save R2,R3,R4,R5,R6,R7,R8,- ; 1301
                                                              R9,R10,R11                                   ;
                   5E         08  C2 00002            SUBL2   #8, SP                                       ;
        50         6C         01  83 00005            SUBB3   #1, (AP), DIFF                               ; 1372
                   02         50  91 00009            CMPB    DIFF, #2                                     ;
                              08  1B 0000C            BLEQU   1$                                           ;
                   50 00000000G  8F  D0 0000E         MOVL    #SMG$_WRONUMARG, R0                          ;
                                  04 00015            RET                                                  ;
                   50         04  BC  D0 00016 1$:    MOVL    @DISPLAY_ID, R0                              ; 1374
              04   BC         38  A0  D1 0001A        CMPL    56(R0), @DISPLAY_ID                          ;
                              06  12 0001F            BNEQ    2$                                           ;
                   11         44  A0  91 00021        CMPB    68(R0), #17                                  ;
                              08  13 00025            BEQL    3$                                           ;
                   50 00000000G  8F  D0 00027 2$:     MOVL    #SMG$_INVDIS_ID, R0                          ;
                                  04 0002E            RET                                                  ;
                   58         04  BC  D0 0002F 3$:    MOVL    @DISPLAY_ID, DCB                             ;
                   02         6C  91 00033            CMPB    (AP), #2                                     ; 1383
                              0F  1F 00036            BLSSU   4$                                           ;
                              08  AC  D5 00038        TSTL    8(AP)                                        ;
                              0A  13 0003B            BEQL    4$                                           ;
                   03         6C  91 0003D            CMPB    (AP), #3                                     ; 1384
                              05  1F 00040            BLSSU   4$                                           ;
                              0C  AC  D5 00042        TSTL    12(AP)                                       ;
                              0B  12 00045            BNEQ    5$                                           ;
                   56         28  A8  3C 00047 4$:    MOVZWL  40(DCB), LINE                                ; 1387
```

```
                          04     AE        2A    A8  3C 0004B           MOVZWL  42(DCB), COL                              ; 1388
                                                 33  11 00050           BRB     10$                                       ; 1383
                                 51        08    BC  D0 00052 5$:       MOVL    @LINE_NO, R1                              ; 1392
                                                 08  15 00056           BLEQ    6$                                        ;
          51        02    A8     10              00  ED 00058           CMPZV   #0, #16, 2(DCB), R1                       ;
                                                 08  18 0005E           BGEQ    7$                                        ;
                                 50 00000000G    8F  D0 00060 6$:       MOVL    #SMG$_INVROW, R0                          ;
                                                 04  00067              RET                                               ;
                                 50        0C    BC  D0 00068 7$:       MOVL    @COL_NO, R0                               ;
                                                 08  15 0006C           BLEQ    8$                                        ;
          50        06    A8     10              00  ED 0006E           CMPZV   #0, #16, 6(DCB), R0                       ;
                                                 08  18 00074           BGEQ    9$                                        ;
                                 50 00000000G    8F  D0 00076 8$:       MOVL    #SMG$_INVCOL, R0                          ;
                                                 04  0007D              RET                                               ;
                                 56              51  D0 0007E 9$:       MOVL    R1, LINE                                  ; 1393
                          04     AE              50  D0 00081           MOVL    R0, COL                                   ; 1394
                                 50        FF    A6  9E 00085 10$:      MOVAB   -1(R6), R0                                ; 1402
                                 5B        06    A8  3C 00089           MOVZWL  6(DCB), R11                               ;
                                 50              5B  C4 0008D           MULL2   R11, R0                                   ;
                    51    04     AE              01  C3 00090           SUBL3   #1, COL, R1                               ;
                    6E           50              51  C1 00095           ADDL3   R1, R0, START_INDEX                       ;
                                 50        FF AB40   9E 00099           MOVAB   -1(R11)[R0], END_INDEX                    ; 1403
                                 50              6E  C2 0009E           SUBL2   START_INDEX, R0                           ; 1404
                                 57        01    A0  9E 000A1           MOVAB   1(R0), NUM_CHARS                          ;
                                 50        10    A8  D0 000A5           MOVL    16(DCB), TEXT_BUF                         ; 1406
                                 59        14    A8  7D 000A9           MOVQ    20(DCB), ATTR_BUF                         ;
          57              20                     6E  00  2C 000AD       MOVC5   #0, (SP), #32, NUM_CHARS, @START_INDEX-   ;
                                       00 BE40           000B2                  [TEXT_BUF]                                ;
          57        2E    A8                     6E  00  2C 000B5       MOVC5   #0, (SP), 46(DCB), NUM_CHARS, @START_INDEX- ;
                                       00 BE49           000BB                  [ATTR_BUF]                                ;
                                                 5A  D5 000BE           TSTL    CHAR_BUF                                  ;
                                                 09  13 000C0           BEQL    11$                                       ;
          57        30    A8                     6E  00  2C 000C2       MOVC5   #0, (SP), 48(DCB), NUM_CHARS, @START_INDEX- ;
                                       00 BE4A           000C8                  [CHAR_BUF]                                ;
                                 5B              57  D1 000CB 11$:      CMPL    NUM_CHARS, R11                            ; 1418
                                                 04  12 000CE           BNEQ    12$                                       ;
                                       4C B846   94  000D0              CLRB    @76(DCB)[LINE]                            ; 1420
                          28     A8              56  B0 000D4 12$:      MOVW    LINE, 40(DCB)                             ; 1428
                          2A     A8        04    AE  B0 000D8           MOVW    COL, 42(DCB)                              ; 1429
                          34     A8              01  8A 000DD           BICB2   #1, 52(DCB)                               ; 1430
                                                 56  DD 000E1           PUSHL   LINE                                      ; 1439
                                                 0D  DD 000E3           PUSHL   #13                                       ; 1437
                                                 58  DD 000E5           PUSHL   DCB                                       ;
                       00000000G 00              03  FB 000E7           CALLS   #3, SMG$$CHECK_FOR_OUTPUT_DCB             ;
                                                 04  000EE              RET                                               ; 1440

; Routine Size:  239 bytes,    Routine Base:  _SMG$CODE + 04F5

; 1189          1441  1 !<BLF/PAGE>
```

```
%SBTTL 'SMG$HOME_CURSOR - Home cursor to a corner position'
GLOBAL ROUTINE SMG$HOME_CURSOR (
                                DISPLAY_ID,
                                POSITION
                               ) =
!++
! FUNCTIONAL DESCRIPTION:
!
!       This routine moves the cursor to the corner row and column
!       of the specified virtual display according to position specified
!       by the POSITION parameter.
!       The caller does not need to know the dimensions of the
!       virtual display.
!
! CALLING SEQUENCE:
!
!       ret_status.wlc.v = SMG$HOME_CURSOR (
!                                           DISPLAY_ID.rl.r
!                                           [,POSITION.rl.r])
!
! FORMAL PARAMETERS:
!
!       DISPLAY_ID.rl.r The display id of the desired virtual display.
!
!       POSITION.rl.r   Optional.  Position to move cursor to according
!                       to following code:
!
!        0 = SMG$C_UPPER_LEFT   Row 1 Column 1 (upper left-hand corner).
!                               This is the default if argument is
!                               omitted.
!
!        1 = SMG$C_LOWER_LEFT   Row n Column 1 (Lower left-hand corner).
!                               This is a useful position for inputting
!                               into an upward scrolling virtual display
!
!        2 = SMG$C_UPPER_RIGHT  Row 1 Column m (Upper right-hand corner)
!
!        3 = SMG$C_LOWER_RIGHT  Row n Column m (Lower right-hand corner)
!
!                       +------------------+
!                       |0                2|
!                       |                  |
!                       |                  |
!                       |1                3|
!                       +------------------+
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! COMPLETION STATUS:
!
!       SS$_NORMAL      Normal successful completion
!       SMG$_INVDIS_ID  Invalid virtual display id.
```

```
1 !             SMG$WRONUMARG   Wrong number of arguments.
1 !
1 ! SIDE EFFECTS:
1 !
1 !             NONE
1 !--
2       BEGIN
2       BUILTIN
2           NULLPARAMETER;
2
2       LOCAL
2           ROW,                            ! Working row number
2           COL,                            ! Working col number
2           DCB : REF $DCB_DECL;            ! Addr of display control block
2
2       $SMG$VALIDATE_ARGCOUNT (1, 2);
2
2       $SMG$GET_DCB ( .DISPLAY_ID, DCB);   ! Get addr of display control
2                                           ! block
2
2       ROW = 1;        ! Assume defaults
2       COL = 1;
2
2       IF NOT NULLPARAMETER (POSITION)
2       THEN
3           BEGIN   ! User-specified
3           IF ..POSITION LSS SMG$C_UPPER_LEFT OR
3              ..POSITION GTR SMG$C_LOWER_RIGHT
3           THEN
3               RETURN (SMG$_INVARG);
3
3           CASE ..POSITION FROM SMG$C_UPPER_LEFT TO SMG$C_LOWER_RIGHT OF
3           SET
3               [SMG$C_UPPER_LEFT]:
3                   ;
3
3               [SMG$C_LOWER_LEFT]:
3                   ROW = .DCB [DCB_W_NO_ROWS];
3
3               [SMG$C_UPPER_RIGHT]:
3                   COL = .DCB [DCB_W_NO_COLS];
3
3               [SMG$C_LOWER_RIGHT]:
4                   BEGIN
4                   ROW = .DCB [DCB_W_NO_ROWS];
4                   COL = .DCB [DCB_W_NO_COLS];
3                   END;
3
3               [INRANGE, OUTRANGE]:
3                   RETURN SMG$_INVARG;
3
3           TES;
3
2           END;    ! User-specified
2
2       DCB [DCB_W_CURSOR_ROW] = .ROW;
2       DCB [DCB_W_CURSOR_COL] = .COL;
```

```
; 1305      1556 2     DCB [DCB_V_FULL] = 0;                    ! turn off scrolling
; 1306      1557 2
; 1307      1558 3     RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
; 1308      1559 2                            SMG$C_HOME_CURSOR));
; 1309      1560 1     END;                            ! End of routine SMG$HOME_CURSOR



                                        001C 00000       .ENTRY  SMG$HOME_CURSOR, Save R2,R3,R4          ; 1443
              50              6C    01    83 00002       SUBB3   #1, (AP), DIFF                          ; 1514
                              01    50    91 00006       CMPB    DIFF, #1                                ;
                                    08    1B 00009       BLEQU   1$                                      ;
                     50 00000000G   8F    D0 0000B       MOVL    #SMG$_WRONUMARG, R0                     ;
                                    04       00012       RET                                             ;
                     50       04    BC    D0 00013 1$:   MOVL    @DISPLAY_ID, R0                         ; 1516
                 04  BC       38    A0    D1 00017       CMPL    56(R0), @DISPLAY_ID                     ;
                              06    12       0001C       BNEQ    2$                                      ;
                     11       44    A0    91 0001E       CMPB    68(R0), #17                             ;
                              08    13       00022       BEQL    3$                                      ;
                     50 00000000G   8F    D0 00024 2$:   MOVL    #SMG$_INVDIS_ID, R0                     ;
                                    04       0002B       RET                                             ;
                     50       04    BC    D0 0002C 3$:   MOVL    @DISPLAY_ID, DCB                        ;
                     54             01    D0 00030       MOVL    #1, ROW                                 ; 1519
                     53             01    D0 00033       MOVL    #1, COL                                 ; 1520
                     02             6C    91 00036       CMPB    (AP), #2                                ; 1522
                                    35    1F 00039       BLSSU   9$                                      ;
                              08    AC    D5 0003B       TSTL    8(AP)                                   ;
                                    30    13 0003E       BEQL    9$                                      ;
                     52       08    BC    D0 00040       MOVL    @POSITION, R2                           ; 1525
                                    11    19 00044       BLSS    5$                                      ;
                     03             52    D1 00046       CMPL    R2, #3                                  ; 1526
                                    0C    14 00049       BGTR    5$                                      ;
              03     00             52    CF 0004B       CASEL   R2, #0, #3                              ; 1530
    0019     001D   0013        0021         0004F 4$:   .WORD   9$-4$,-                                 ;
                                                                 6$-4$,-                                 ;
                                                                 8$-4$,-                                 ;
                                                                 7$-4$                                   ;
                     51 00000000G   00    9E 00057 5$:   MOVAB   SMG$_INVARG, R1                         ; 1548
                     50             51    D0 0005E       MOVL    R1, R0                                  ;
                                    04       00061       RET                                             ;
                     54       02    A0    3C 00062 6$:   MOVZWL  2(DCB), ROW                             ; 1536
                                    08    11 00066       BRB     9$                                      ;
                     54       02    A0    3C 00068 7$:   MOVZWL  2(DCB), ROW                             ; 1543
                     53       06    A0    3C 0006C 8$:   MOVZWL  6(DCB), COL                             ; 1544
                 28  A0             54    B0 00070 9$:   MOVW    ROW, 40(DCB)                            ; 1554
                 2A  A0             53    B0 00074       MOVW    COL, 42(DCB)                            ; 1555
                 34  A0             01    8A 00078       BICB2   #1, 52(DCB)                             ; 1556
                                    0E    DD 0007C       PUSHL   #14                                     ; 1558
                                    50    DD 0007E       PUSHL   DCB                                     ;
              00000000G  00         02    FB 00080       CALLS   #2, SMG$$CHECK_FOR_OUTPUT_DCB           ;
                                    04       00087       RET                                             ; 1560

; Routine Size:  136 bytes,    Routine Base:  _SMG$CODE + 05E4
```

```
; 1310          1561  1 !<BLF/PAGE>
```

```
1 %SBTTL 'SMG$INSERT_CHARS - Insert Characters'
1 GLOBAL ROUTINE SMG$INSERT_CHARS (
1                                   DISPLAY_ID,
1                                   CHAR_STRING,
1                                   ROW,
1                                   COL,
1                                   RENDITION_SET,
1                                   RENDITION_COMPLEMENT,
1                                   CHAR_SET
1                                  ) =
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine inserts the specified character string at the
1 !       row and column position specified.
1 !
1 !       Characters supplied from CHAR_STRING are inserted into the
1 !       specified display at the specified ROW and COL, shifting
1 !       subsequent characters on the line to the right.
1 !
1 !       SMG$INSERT_CHARS is a character-oriented operation - any
1 !       characters which do not fit on the current line are thrown
1 !       away.  Wrapping is not allowed.
1 !
1 !       The internal display cursor position is left at the character
1 !       position following the last character written.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = SMG$INSERT_CHARS (
1 !                               DISPLAY_ID.rl.r,
1 !                               CHAR_STRING.rt.dx,
1 !                               ROW.rl.r,
1 !                               COL.rl.r,
1 !                               [,RENDITION_SET.rl.r]
1 !                               [,RENDITION_COMPLEMENT.rl.r]
1 !                               [,CHAR_SET.rl.r])
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       DISPLAY_ID.rl.r         Display id of desired virtual display.
1 !
1 !       CHAR_STRING.rt.dx       Character string to be inserted
1 !
1 !       ROW.rl.r                Row position at which to start
1 !                               insertion.
1 !
1 !       COL.rl.r                Column position at which to start
1 !                               insertion.
1 !
1 !       RENDITION_SET.rl.r      Each 1 bit attribute in this parameter
1 !                               causes the corresponding attribute to
1 !                               be set in the display.  (See below for
1 !                               list of settable attributes.)
1 !
1 !       RENDITION_COMPLEMENT.rl.r
1 !                               Each 1 bit attribute in this parameter
```

```
!                                       causes the corresponding attribute to
!                                       be complemented in the display.  (See
!                                       below for list of complementable
!                                       attributes.)
!
!           If the same bit is specified in both the RENDITION_SET parameter
!           and in the RENDITION_COMPLEMENT parameter, the application is
!           RENDITION_SET followed by RENDITION complement.  Using these two
!           parameters together the caller can exercise arbitrary and
!           independent control over each attribute on a single call.  On an
!           attribute by attribute basis he can cause the following
!           transformations:
!
!                   SET     COMPLEMENT      Action
!                   ---     ----------      ------
!                   0       0               Attribute unchanged.
!                   1       0               Attribute set to "on"
!                   0       1               Attribute set to complement of
!                                           current setting.
!                   1       1               Attribute set to "off".
!
!
!           Attributes which can be manipulated in this manner are:
!
!           SMG$M_BLINK  displays characters blinking.
!           SMG$M_BOLD  displays characters in higher-than-normal
!                        intensity.
!           SMG$M_REVERSE  displays characters in reverse video -- that is,
!                        using the opposite default rendition of the
!                        virtual display.
!           SMG$M_UNDERLINE  displays characters underlined.
!
!           CHAR_SET.rl.r               Optional.  Character set to use.
!                                       Choices are:
!                                               SMG$C_UNITED_KINGDOM
!                                               SMG$C_ASCII (default)
!                                               SMG$C_SPEC_GRAPHICS
!                                               SMG$C_ALT_CHAR
!                                               SMG$C_ALT_GRAPHICS
!
!       IMPLICIT INPUTS:
!
!           NONE
!
!       IMPLICIT OUTPUTS:
!
!           NONE
!
!       COMPLETION STATUS:
!
!           SS$_NORMAL      Normal successful completion
!           LIB$_INVSTRDES  Invalid string descriptor
!           SMG$_INVROW     Invalid row specification-- outside virtual
!                           display.
!           SMG$_INVCOL     Invalid column specification -- outside virtual
!                           display.
!           SMG$_INVDIS_ID  Invalid virtual display id.
```

```
1 !         SMG$_WRONUMARG  Wrong number or arguments.
1 !         SMG$_INVARG     Unrecognized rendition code.
1 !
1 ! SIDE EFFECTS:
1 !
1 !         NONE
1 !--
2     BEGIN
2
2     BUILTIN
2         NULLPARAMETER;
2
2     LITERAL
2         K_SET_ARG = 5,
2         K_COMP_ARG = 6;
2
2     LOCAL
2         DCB : REF $DCB_DECL,              ! Addr of display control block
2         STR_ADDR,                         ! Addr of text
2         STR_LEN : INITIAL(0),             ! Length of text
2         PRINT_LEN,                        ! Length of text w/tabs, backspaces, etc
2         REND_CODE,                        ! Rendition code
2         STATUS;                           ! Retd by called routine
2
2     $SMG$VALIDATE_ARGCOUNT (4, 7);
2
2     $SMG$GET_DCB ( .DISPLAY_ID, DCB);    ! Get addr of display control
2
2 !+
2 ! Validity check row and column specified.
2 !-
2
2     $SMG$VALIDATE_ROW_COL ( ..ROW, ..COL);
2
2     IF NOT NULLPARAMETER( CHAR_SET )
2     THEN
3         BEGIN
3         CASE ..CHAR_SET FROM SMG$C_UNITED_KINGDOM TO SMG$C_ALT_GRAPHICS OF
3         SET
3
3         [SMG$C_UNITED_KINGDOM, SMG$C_ASCII, SMG$C_SPEC_GRAPHICS,
3          SMG$C_ALT_CHAR, SMG$C_ALT_GRAPHICS]:
3                 ;
3         [INRANGE, OUTRANGE]:
3                 RETURN (SMG$_INVARG);
3         TES;
2         END;
2
3     IF NOT (STATUS = LIB$ANALYZE_SDESC_R2 (.CHAR_STRING;
3                                             STR_LEN,
3                                             STR_ADDR))
2     THEN
2         RETURN (.STATUS);
2
2 !+
2 ! Reset the line characteristics in case the line was previously
2 ! double high or wide.
```

```
!-

    BEGIN
    BIND
        LINE_CHAR = .DCB [DCB_A_LINE_CHAR];
    MAP
        LINE_CHAR : VECTOR [,BYTE];
    IF .LINE_CHAR [..ROW] NEQ 0          ! prev. wide or high chars
    THEN
        BEGIN                            ! blank line
        LOCAL
            START_INDEX;
        START_INDEX = $SMG$LINEAR (..ROW, 1);
        $SMG$BLANK_FILL_DCB (.DCB [DCB_W_NO_COLS], .START_INDEX);
        LINE_CHAR [..ROW] = 0;
        END;
    END;

!+
! Shift existing characters over to make room for new text.  If the
! space is already blank, just write into it.  Don't write over a
! blank if there are characters to the right of it - an embedded blank
! should be treated as a regular character and shifted.
!
! Macro $SMG$FIND_PRINT_LENGTH scans the string for funny characters
! and determines how many printing characters there are.  Extra space
! is allowed for tabs.  If a LF or CR is found, it terminates the string
! since the effect of this routine is limited to a single line.  STR_LEN
! is modified to reflect the number of characters that comprise PRINT_LEN.
!-

    BEGIN
    LOCAL
        INDEX,
        PTR,
        LINE_LEN,
        TEXT_BUF : REF VECTOR [,BYTE];

    $SMG$FIND_PRINT_LENGTH (STR_LEN, .STR_ADDR, PRINT_LEN);

    INDEX = $SMG$LINEAR (..ROW, ..COL);
    TEXT_BUF = .DCB [DCB_A_TEXT_BUF];

    LINE_LEN = .DCB [DCB_W_NO_COLS] - ..COL + 1;

    PTR = CH$FIND_NOT_CH (.LINE_LEN, TEXT_BUF [.INDEX], %C' ');
                                    ! get ptr to 1st non-blank
    IF .PTR NEQ 0
    THEN
        SMG$$SCROLL_AREA (.DCB, ..ROW, ..COL,
                         1, .DCB [DCB_W_NO_COLS],
                         SMG$M_RIGHT, .PRINT_LEN);

    END;

!+
! Write text into created space.  SMG$$PUT_TEXT_TO_BUFFER checks for
```

```
; 1540          1790  2  ! funny characters (tabs, etc) and also updates the cursor position.
; 1541          1791  2  !-
; 1542          1792  2
; 1543          1793  2      $SMG$SET_REND_CODE (K_SET_ARG, K_COMP_ARG);
; 1544          1794  2
; 1545          1795  2
; 1546          1796  2      DCB [DCB_W_CURSOR_ROW] = ..ROW;
; 1547          1797  2      DCB [DCB_W_CURSOR_COL] = ..COL;
; 1548          1798  2                              ! set current position for PUT_TEXT
; 1549          1799  2
; 1550          1800  2      SMG$$PUT_TEXT_TO_BUFFER (.DCB, .REND_CODE,
; 1551          1801  2                               .STR_LEN, .STR_ADDR,
; 1552          1802  3                               (IF NOT NULLPARAMETER (CHAR_SET)
; 1553          1803  3                                   THEN ..CHAR_SET
; 1554          1804  2                                   ELSE SMG$C_ASCII)    );
; 1555          1805  2
; 1556          1806  2  !+
; 1557          1807  2  ! Turn off scrolling.  Sequential output has been interupted by
; 1558          1808  2  ! moving the cursor for insert_chars.
; 1559          1809  2  !-
; 1560          1810  2
; 1561          1811  2      DCB [DCB_V_FULL] = 0;
; 1562          1812  2
; 1563          1813  2  !+
; 1564          1814  2  ! See if this change needs to be reflected on the screen.
; 1565          1815  2  !-
; 1566          1816  2
; 1567          1817  3      RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
; 1568          1818  3                                         SMG$C_INSERT_CHARS,
; 1569          1819  2                                         ..ROW));
; 1570          1820  1      END;                    ! End of routine SMG$INSERT_CHARS


                                                        .EXTRN  CHAR_TABLE

                                       0FFC 00000       .ENTRY  SMG$INSERT_CHARS, Save R2,R3,R4,R5,R6,R7,-   ; 1563
                                                                R8,R9,R10,R11                                 ;
                      5E          04  C2 00002          SUBL2   #4, SP                                        ;
                                  7E  D4 00005          CLRL    STR_LEN                                       ; 1683
                 50   6C          04  83 00007          SUBB3   #4, (AP), DIFF                                ; 1700
                      03          50  91 0000B          CMPB    DIFF, #3                                      ;
                                  08  1B 0000E          BLEQU   1$                                            ;
                      50 00000000G    8F  D0 00010      MOVL    #SMG$_WRONUMARG, R0                           ;
                                      04 00017          RET                                                   ;
                      50          04  BC  D0 00018 1$:  MOVL    @DISPLAY_ID, R0                               ; 1702
            04        BC          38  A0  D1 0001C      CMPL    56(R0), @DISPLAY_ID                           ;
                                  06  12 00021          BNEQ    2$                                            ;
                      11          44  A0  91 00023      CMPB    68(R0), #17                                   ;
                                  08  13 00027          BEQL    3$                                            ;
                      50 00000000G    8F  D0 00029 2$:  MOVL    #SMG$_INVDIS_ID, R0                           ;
                                      04 00030          RET                                                   ;
                      56          04  BC  D0 00031 3$:  MOVL    @DISPLAY_ID, DCB                              ;
                      59          0C  BC  D0 00035      MOVL    @ROW, R9                                      ; 1708
                                  08  15 00039          BLEQ    4$                                            ;
      59      02      A6          10      00  ED 0003B  CMPZV   #0, #16, 2(DCB), R9                           ;
                                  08  18 00041          BGEQ    5$                                            ;
```

```
                          50 00000000G  8F  D0 00043 4$:    MOVL    #SMG$_INVROW, R0                              ;
                                            04 0004A        RET                                                   ;
                          5A       10   BC  D0 0004B 5$:    MOVL    @COL, R10                                     ;
                                        08  15 0004F        BLEQ    6$                                            ;
    5A      06   A6       10            00  ED 00051        CMPZV   #0, #16, 6(DCB), R10                          ;
                                        08  18 00057        BGEQ    7$                                            ;
                          50 00000000G  8F  D0 00059 6$:    MOVL    #SMG$_INVCOL, R0                              ;
                                            04 00060        RET                                                   ;
                          07            6C  91 00061 7$:    CMPB    (AP), #7                                      ; 1710
                                        1C  1F 00064        BLSSU   9$                                            ;
                                   1C   AC  D5 00066        TSTL    28(AP)                                        ;
                                        17  13 00069        BEQL    9$                                            ;
                  04      00       1C   BC  CF 0006B        CASEL   @CHAR_SET, #0, #4                             ; 1713
  0012          0012    0012          0012     00070 8$:    .WORD   9$-8$,-                                       ;
                                      0012     00078                9$-8$,-                                       ;
                                                                    9$-8$,-                                       ;
                                                                    9$-8$,-                                       ;
                                                                    9$-8$                                         ;
                          50 00000000G  00  9E 0007A        MOVAB   SMG$_INVARG, R0                               ; 1720
                                            04 00081        RET                                                   ;
                          50       08   AC  D0 00082 9$:    MOVL    CHAR_STRING, R0                               ; 1724
                             00000000G  00  16 00086        JSB     LIB$ANALYZE_SDESC_R2                          ;
                          6E            51  7D 0008C        MOVQ    R1, (SP)                                      ;
                          01            50  E8 0008F        BLBS    STATUS, 10$                                   ;
                                            04 00092        RET                                                   ;
                                4C B649  95 00093 10$:      TSTB    @76(DCB)[R9]                                  ; 1740
                                        38  13 00097        BEQL    12$                                           ;
                          50       FF   A9  9E 00099        MOVAB   -1(R9), R0                                    ; 1745
                          51       06   A6  3C 0009D        MOVZWL  6(DCB), R1                                    ;
                          50            51  C4 000A1        MULL2   R1, R0                                        ;
                          5B            50  D0 000A4        MOVL    R0, START_INDEX                               ;
                          50       10   A6  D0 000A7        MOVL    16(DCB), TEXT_BUF                             ; 1746
                          57       14   A6  7D 000AB        MOVQ    20(DCB), ATTR_BUF                             ;
  06   A6           20    6E            00  2C 000AF        MOVC5   #0, (SP), #32, 6(DCB), (START_INDEX)-         ;
                                      6B40     000B5                [TEXT_BUF]                                    ;
  06   A6      2E   A6    6E            00  2C 000B7        MOVC5   #0, (SP), 46(DCB), 6(DCB), (START_INDEX)-     ;
                                      6B47     000BE                [ATTR_BUF]                                    ;
                                        58  D5 000C0        TSTL    CHAR_BUF                                      ;
                                        09  13 000C2        BEQL    11$                                           ;
  06   A6      30   A6    6E            00  2C 000C4        MOVC5   #0, (SP), 48(DCB), 6(DCB), (START_INDEX)-     ;
                                      6B48     000CB                [CHAR_BUF]                                    ;
                                4C B649  94 000CD 11$:      CLRB    @76(DCB)[R9]                                  ; 1747
                          58            01  8E 000D1 12$:   MNEGB   #1, ALLONES                                   ; 1771
                                        55  D4 000D4        CLRL    PRINT_LEN                                     ;
                          54            6E  D0 000D6        MOVL    STR_LEN, BYTES_REMAINING                      ;
                          57       04   AE  D0 000D9        MOVL    STR_ADDR, IN_POINTER                          ;
                                        6E  D4 000DD        CLRL    STR_LEN                                       ;
                                        54  D5 000DF 13$:   TSTL    BYTES_REMAINING                               ;
                                        59  13 000E1        BEQL    17$                                           ;
       58 00000000G  00           67    54  2A 000E3        SCANC   BYTES_REMAINING, (IN_POINTER), CHAR_TABLE, -  ;
                                                                    ALLONES                                       ;
                    52    54            50  C3 000EC        SUBL3   NEW_BYTES_REMAINING, BYTES_REMAINING, R2      ;
                          57            52  C0 000F0        ADDL2   R2, IN_POINTER                                ;
                          55            52  C0 000F3        ADDL2   R2, PRINT_LEN                                 ;
                          6E            52  C0 000F6        ADDL2   R2, STR_LEN                                   ;
                          54            50  D0 000F9        MOVL    NEW_BYTES_REMAINING, BYTES_REMAINING          ;
                                        3E  13 000FC        BEQL    17$                                           ;
```

```
                              50          61   9A 000FE          MOVZBL  (ADDR_DIFF), R0                              ;
                     09       01 00000000G0040  8F 00101          CASEB   CHAR_TABLE[R0], #1, #9                       ;
        001C         002C     002C          002C       0010A 14$:  .WORD   16$-14$,-                                    ;
        0032         0032     0032          0032       00112               16$-14$,-                                    ;
                              002C          0032       0011A               16$-14$,-                                    ;
                                                                           15$-14$,-                                    ;
                                                                           17$-14$,-                                    ;
                                                                           17$-14$,-                                    ;
                                                                           17$-14$,-                                    ;
                                                                           17$-14$,-                                    ;
                                                                           17$-14$,-                                    ;
                                                                           16$-14$                                      ;
                              50 00000000G  8F  D0 0011E          MOVL    #SMG$_FATERRLIB, R0                          ;
                                                04 00125          RET                                                  ;
                              50        2A  A6  3C 00126 15$:     MOVZWL  42(DCB), R0                                  ;
                                            50  D7 0012A          DECL    R0                                           ;
                              50            08  C6 0012C          DIVL2   #8, R0                                       ;
                              55      09 A540   7E 0012F          MOVAQ   9(PRINT_LEN)[R0], PRINT_LEN                  ;
                                            6E  D6 00134          INCL    STR_LEN                                      ;
                                            57  D6 00136 16$:     INCL    IN_POINTER                                   ;
                                            54  D7 00138          DECL    BYTES_REMAINING                              ;
                                            A3  11 0013A          BRB     13$                                          ;
                              50        FF  A9  9E 0013C 17$:     MOVAB   -1(R9), R0                                   ; 1773
                              51        06  A6  3C 00140          MOVZWL  6(DCB), R1                                   ;
                              50            51  C4 00144          MULL2   R1, R0                                       ;
                              52     FF AA40    9E 00147          MOVAB   -1(R10)[R0], INDEX                           ;
                              51        10  A6  D0 0014C          MOVL    16(DCB), TEXT_BUF                            ; 1774
                              50        06  A6  3C 00150          MOVZWL  6(DCB), R0                                   ; 1776
                              50            5A  C2 00154          SUBL2   R10, R0                                      ;
                                            50  D6 00157          INCL    LINE_LEN                                     ;
                     6241     50            20  3B 00159          SKPC    #32, LINE_LEN, (INDEX)[TEXT_BUF]             ; 1778
                                            02  12 0015E          BNEQ    18$                                          ;
                                            51  D4 00160          CLRL    R1                                           ;
                                            51  D5 00162 18$:     TSTL    PTR                                          ; 1780
                                            15  13 00164          BEQL    19$                                          ;
                                            55  DD 00166          PUSHL   PRINT_LEN                                    ; 1784
                                            04  DD 00168          PUSHL   #4                                           ; 1782
                              7E        06  A6  3C 0016A          MOVZWL  6(DCB), -(SP)                                ; 1783
                                            01  DD 0016E          PUSHL   #1                                           ; 1782
                                    0640    8F  BB 00170          PUSHR   #^M<R6,R9,R10>                               ;
                  00000000G   00            07  FB 00174          CALLS   #7, SMG$$SCROLL_AREA                         ;
                              50        2E  A6  9A 0017B 19$:     MOVZBL  46(DCB), REND_CODE                           ; 1793
                              05            6C  91 0017F          CMPB    (AP), #5                                     ;
                                            09  1F 00182          BLSSU   20$                                          ;
                                        14  AC  D5 00184          TSTL    20(AP)                                       ;
                                            04  13 00187          BEQL    20$                                          ;
                              50        14  BC  C8 00189          BISL2   @RENDITION_SET, REND_CODE                    ;
                              06            6C  91 0018D 20$:     CMPB    (AP), #6                                     ;
                                            09  1F 00190          BLSSU   21$                                          ;
                                        18  AC  D5 00192          TSTL    24(AP)                                       ;
                                            04  13 00195          BEQL    21$                                          ;
                              50        18  BC  CC 00197          XORL2   @RENDITION_COMPLEMENT, REND_CODE             ;
                          28  A6            59  B0 0019B 21$:     MOVW    R9, 40(DCB)                                  ; 1796
                          2A  A6            5A  B0 0019F          MOVW    R10, 42(DCB)                                 ; 1797
                              07            6C  91 001A3          CMPB    (AP), #7                                     ; 1802
                                            0A  1F 001A6          BLSSU   22$                                          ;
                                        1C  AC  D5 001A8          TSTL    28(AP)                                       ;
```

```
                               05  13 001AB      BEQL    22$                             ;
                        1C     BC  DD 001AD      PUSHL   @CHAR_SET                       ; 1803
                               02  11 001B0      BRB     23$                             ;
                               01  DD 001B2 22$: PUSHL   #1                              ; 1802
                        08     AE  DD 001B4 23$: PUSHL   STR_ADDR                        ; 1801
                        08     AE  DD 001B7      PUSHL   STR_LEN                         ;
                               50  DD 001BA      PUSHL   REND_CODE                       ; 1800
                               56  DD 001BC      PUSHL   DCB                             ;
           00000000G  00       05  FB 001BE      CALLS   #5, SMG$$PUT_TEXT_TO_BUFFER     ;
                  34  A6       01  8A 001C5      BICB2   #1, 52(DCB)                     ; 1811
                               59  DD 001C9      PUSHL   R9                              ; 1819
                               0F  DD 001CB      PUSHL   #15                             ; 1817
                               56  DD 001CD      PUSHL   DCB                             ;
           00000000G  00       03  FB 001CF      CALLS   #3, SMG$$CHECK_FOR_OUTPUT_DCB   ;
                                   04 001D6      RET                                     ; 1820

; Routine Size:  471 bytes,    Routine Base:  _SMG$CODE + 066C


; 1571        1821  1 !<BLF/PAGE>
```

```
1 %SBTTL 'SMG$INSERT_LINE - Insert a line into a virtual display'
1 GLOBAL ROUTINE SMG$INSERT_LINE   ( DISPLAY_ID,
1                                    LINE_NO,
1                                    STRING,
1                                    DIRECTION,
1                                    RENDITION_SET,
1                                    RENDITION_COMPLEMENT,
1                                    WRAP_FLAG,
1                                    CHAR_SET ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !      This routine allows a line to be inserted into a virtual display
1 !      at a location other than the first or last line.  Existing lines
1 !      are scrolled in the specified direction to create an open space.
1 !      If a string is passed, it will be written in the space created;
1 !      otherwise, the new line will remain blank.  The entire line is
1 !      rewritten; if the string does not span the width of the display,
1 !      it is blank padded.
1 !
1 !      If WRAP_FLAG is set, characters which would exceed the rightmost
1 !      boundary of the display cause another line scroll and the overflow
1 !      characters are written in the created space.  If WRAP_FLAG is off,
1 !      any overflow characters are lost.
1 !
1 !      The internal display cursor position is left at the character
1 !      position following the text written.
1 !
1 !      See SMG$PUT_WITH_SCROLL to add lines at the first or last
1 !      line of a virtual display with scrolling.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !      ret_status.wlc.v = SMG$INSERT_LINE ( DISPLAY_ID.rl.r,
1 !                                           LINE_NO.rlu.r,
1 !                                           [,STRING.rt.dx]
1 !                                           [,DIRECTION.rl.r]
1 !                                           [,RENDITION_SET.rl.r]
1 !                                           [,RENDITION_COMPLEMENT.rl.r]
1 !                                           [,WRAP_FLAG.rl.r]
1 !                                           [,CHAR_SET.rl.r])
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !      DISPLAY_ID.rl.r         Display id of desired virtual display.
1 !
1 !      LINE_NO.rlu.r           Line number at which to start scroll.
1 !
1 !      STRING.rt.dx            Optional.  Text to insert into line
1 !                              created.  If not specified, all blanks
1 !                              are used.
1 !
1 !      DIRECTION.rlu.r         Optional.  Direction to scroll text.
1 !                                      SMG$M_UP (Default)
1 !                                      SMG$M_DOWN
1 !
```

```
1 !          RENDITION_SET.rl.r        Each 1 bit attribute in this parameter
1 !                                    causes the corresponding attribute to
1 !                                    be set in the display.  (See below for
1 !                                    list of settable attributes.)
1 !
1 !          RENDITION_COMPLEMENT.rl.r
1 !                                    Each 1 bit attribute in this parameter
1 !                                    causes the corresponding attribute to
1 !                                    be complemented in the display.  (See
1 !                                    below for list of complementable
1 !                                    attributes.)
1 !
1 !          If the same bit is specified in both the RENDITION_SET parameter
1 !          and in the RENDITION_COMPLEMENT parameter, the application is
1 !          RENDITION_SET followed by RENDITION complement.  Using these two
1 !          parameters together the caller can exercise arbitrary and
1 !          independent control over each attribute on a single call.  On an
1 !          attribute by attribute basis he can cause the following
1 !          transformations:
1 !
1 !                  SET     COMPLEMENT        Action
1 !                  ---     ----------        ------
1 !                  0       0                 Attribute unchanged.
1 !                  1       0                 Attribute set to "on"
1 !                  0       1                 Attribute set to complement of
1 !                                            current setting.
1 !                  1       1                 Attribute set to "off".
1 !
1 !          WRAP_FLAG.rl.r            Optional.  Action to take if text does
1 !                                    not fit on the line.
1 !                                    0 = nowrap (Default)
1 !                                    1 = wrap
1 !
1 !          CHAR_SET.rl.r             Optional.  Character set to use.
1 !                                    Choices are:
1 !                                            SMG$C_UNITED_KINGDOM
1 !                                            SMG$C_ASCII (default)
1 !                                            SMG$C_SPEC_GRAPHICS
1 !                                            SMG$C_ALT_CHAR
1 !                                            SMG$C_ALT_GRAPHICS
1 !
1 !  IMPLICIT INPUTS:
1 !
1 !          NONE
1 !
1 !  IMPLICIT OUTPUTS:
1 !
1 !          NONE
1 !
1 !  COMPLETION STATUS:
1 !
1 !          SS$_NORMAL       Normal successful completion
1 !          SMG$_INVDIS_ID   Invalid display id.
1 !          SMG$_WRONUMARG   Wrong number of arguments.
1 !
1 !  SIDE EFFECTS:
1 !
```

```
1 !         NONE
1 !--
2     BEGIN
2
2     BUILTIN
2         NULLPARAMETER;
2
2     LITERAL
2         K_SET_ARG = 5,
2         K_COMP_ARG = 6;
2
2     LOCAL
2         DONE,
2         STATUS,                          ! Status of subroutine calls
2         DCB : REF $DCB_DECL,             ! Addr of display control block
2         STR_LEN : INITIAL (0),           ! Length of text string
2         STR_ADDR,                        ! Addr of text string
2         DIR,                             ! Working direction
2         C_SET,                           ! Working character set
2         WRAP_CHARS : INITIAL (0),        ! No. of overflow characters
2                                          ! Reset by call to
2                                          ! SMG$$PUT_TEXT_TO_BUFFER
2         REND_CODE;                       ! Rendition code from params
2                                          ! Set by $SMG$SET_REND_CODE
2     $SMG$VALIDATE_ARGCOUNT (2, 8);
2
2     $SMG$GET_DCB ( .DISPLAY_ID, DCB);    ! Get address of display control
2                                          ! block
2
2 !+
2 ! Validity check the arguments.
2 !-
2
2     IF ..LINE_NO LSS .DCB [DCB_W_TOP_OF_SCRREG] OR
2        ..LINE_NO GTR .DCB [DCB_W_BOTTOM_OF_SCRREG]
2     THEN
2         RETURN (SMG$_INVROW);
2
2     DIR = SMG$M_UP;                      ! init to default
2
2     IF NOT NULLPARAMETER( DIRECTION )
2     THEN
3         BEGIN
4         IF (..DIRECTION NEQ SMG$M_UP AND
4             ..DIRECTION NEQ SMG$M_DOWN)
3         THEN
4             RETURN (SMG$_INVARG)
3         ELSE
3             DIR = ..DIRECTION;
2         END;
2
2     C_SET = SMG$C_ASCII;                 ! init to default
2
2     IF NOT NULLPARAMETER( CHAR_SET )
2     THEN
3         BEGIN
3         CASE ..CHAR_SET FROM SMG$C_UNITED_KINGDOM TO SMG$C_ALT_GRAPHICS OF
```

```
3           SET
3
3           [SMG$C_UNITED_KINGDOM, SMG$C_ASCII, SMG$C_SPEC_GRAPHICS,
3            SMG$C_ALT_CHAR, SMG$C_ALT_GRAPHICS]:
3                   C_SET = ..CHAR_SET;
3           [INRANGE, OUTRANGE]:
3                   RETURN (SMG$_INVARG);
3           TES;
2           END;
2
2   !+
2   ! Set up string to output, either the caller's text or a blank line.
2   !-
2
2       IF NOT NULLPARAMETER (STRING)
2       THEN
3           BEGIN                               ! use caller's text
4           IF NOT (STATUS = LIB$ANALYZE_SDESC_R2 (.STRING; STR_LEN, STR_ADDR))
3           THEN
3               RETURN (.STATUS);
2           END;                                ! user caller's text
2
2   !+
2   ! Move existing lines up or down to create an empty space.  This space
2   ! will remain empty if the user did not specify a text string.
2   !-
2
2       IF .DIR EQL SMG$M_UP
2       THEN
2           SMG$$SCROLL_AREA (.DCB, .DCB [DCB_W_TOP_OF_SCRREG],
2                             .DCB [DCB_W_COL_START],
2                             (..LINE_NO - .DCB [DCB_W_TOP_OF_SCRREG] + 1),
2                             .DCB [DCB_W_NO_COLS],
2                             SMG$M_UP, 1)
2       ELSE
2           SMG$$SCROLL_AREA (.DCB, ..LINE_NO, .DCB [DCB_W_COL_START],
2                             (.DCB [DCB_W_BOTTOM_OF_SCRREG] - ..LINE_NO + 1),
2                             .DCB [DCB_W_NO_COLS],
2                             SMG$M_DOWN, 1);
2
2   !+
2   ! Reset the line characteristics in case the line was previously
2   ! double high or wide.
2   !-
2
3       BEGIN
3       BIND
3           LINE_CHAR = .DCB [DCB_A_LINE_CHAR];
3       MAP
3           LINE_CHAR : VECTOR [,BYTE];
3       LINE_CHAR [..LINE_NO] = 0;
2       END;
2
2   !+
2   ! Write text into the space just created.  SMG$$PUT_TEXT_TO_BUFFER
2   ! checks for funny characters (tabs, etc) and also updates the cursor
2   ! position.
```

```
!-

    IF .STR_LEN NEQ 0
    THEN
        BEGIN                               ! parse & move caller's text
        $SMG$SET_REND_CODE (K_SET_ARG, K_COMP_ARG);

        DCB [DCB_W_CURSOR_ROW] = ..LINE_NO;
        DCB [DCB_W_CURSOR_COL] = 1;
                                            ! set current position for PUT_TEXT
        DCB [DCB_V_FULL] = 0;

        SMG$$PUT_TEXT_TO_BUFFER (.DCB,
                                .REND_CODE,
                                .STR_LEN,
                                .STR_ADDR,
                                C_SET,
                                WRAP_CHARS);
        END

    ELSE
        BEGIN
        !+
        ! Used the short cut to put out a blank line (we let SMG$$SCROLL_AREA
        ! do it), so we must update the cursor position ourselves (normally
        ! SMG$$PUT_TEXT_TO_BUFFER would do it).
        !-
        DCB [DCB_W_CURSOR_ROW] = ..LINE_NO + 1;
        IF .DCB [DCB_W_CURSOR_ROW] LEQ .DCB [DCB_W_BOTTOM_OF_SCRREG]
        THEN
            BEGIN
            DCB [DCB_V_FULL] = 0;
            DCB [DCB_W_CURSOR_COL] = .DCB [DCB_W_COL_START]
            END
        ELSE
            BEGIN                           ! just wrote last line
            DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_BOTTOM_OF_SCRREG];
            DCB [DCB_W_CURSOR_COL] = .DCB [DCB_W_NO_COLS];
            DCB [DCB_V_FULL] = 1;           ! remember used all lines
            DCB [DCB_V_COL_80] = 1;         ! remember at last col position
            END;
        END;

!+
! Check for overflow characters which should cause us to scroll 2 lines
! in order to output everything (if wrap requested).  Need to pass the
! number of characters (not the printing length) in the descriptor.
!-

    IF .WRAP_CHARS NEQ 0
    THEN
        BEGIN
        IF NOT NULLPARAMETER( WRAP_FLAG ) AND
            ..WRAP_FLAG NEQ 0               ! wrap requested
        THEN
            BEGIN                           ! recurse until whole string output
            LOCAL
```

```
4                  TEMP_DESC : BLOCK [8, BYTE];
4
4              TEMP_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
4              TEMP_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
4              TEMP_DESC [DSC$W_LENGTH] = .STR_LEN - .DCB [DCB_W_NO_COLS];
4              TEMP_DESC [DSC$A_POINTER] = .STR_ADDR +
4                                       .DCB [DCB_W_NO_COLS];
4
4              !+
4              ! Kludge for downscrolling so that pieces of the line appear
4              ! in the correct order.
4              !-
4              IF .DIR EQL SMG$M_DOWN
4              THEN
5                  BEGIN
5                  LOCAL
5                      OVERFLOW_LINE;
5                  OVERFLOW_LINE = ..LINE_NO + 1;
5                  SMG$INSERT_LINE (.DISPLAY_ID, OVERFLOW_LINE,
5                                   TEMP_DESC, %REF (SMG$M_DOWN),
6                                   (IF NOT NULLPARAMETER (RENDITION_SET)
6                                       THEN .RENDITION_SET
5                                       ELSE 0),
6                                   (IF NOT NULLPARAMETER (RENDITION_COMPLEMENT)
6                                       THEN .RENDITION_COMPLEMENT
5                                       ELSE 0),
5                                   .WRAP_FLAG, C_SET);
5                  END
4              ELSE
4                  SMG$INSERT_LINE (.DISPLAY_ID, .LINE_NO,
4                                   TEMP_DESC, %REF (SMG$M_UP),
5                                   (IF NOT NULLPARAMETER (RENDITION_SET)
5                                       THEN .RENDITION_SET
4                                       ELSE 0),
5                                   (IF NOT NULLPARAMETER (RENDITION_COMPLEMENT)
5                                       THEN .RENDITION_COMPLEMENT
4                                       ELSE 0),
4                                   .WRAP_FLAG,
4                                   C_SET);
4              DONE = 0;
4              RETURN 1;                  ! to keep Bliss happy
4              END
3          ELSE
4              BEGIN
4              !+
4              ! Wrap was not requested but there were overflow characters.
4              ! Put out diamond in last position to show truncation.
4              !-
4              IF .DCB [DCB_V_TRUNC_ICON]
4              THEN
5                  BEGIN
5                  DCB [DCB_W_CURSOR_COL] = .DCB [DCB_W_NO_COLS];
5                  SMG$$PUT_TEXT_TO_BUFFER (.DCB,
5                                           .REND_CODE + ATTR_M_USER_GRAPHIC,
5                                           1, UPLIT (BYTE (DIAMOND)),
5                                           SMG$C_ASCII);
4                  END;
```

```
  4              DONE = 1;
  3              END;
  3          END
  2      ELSE
  2          DONE = 1;                          ! no wrap chars
  2
  2  !+
  2  ! See if this change should be reflected on the screen.  Even if
  2  ! we call SMG$INSERT_LINE again, SMG$$CHECK_FOR_OUTPUT_DCB should
  2  ! be executed only once.
  2  !-
  2
  2      IF .DONE
  2      THEN
  3          RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
  3                                            SMG$C_INSERT_LINE,
  3                                            ..LINE_NO))
  2      ELSE
  2          RETURN 1;
  2
  1      END;                                   ! End of SMG$INSERT_LINE


                                   00843          .BLKB   1
                                60 00844 P.AAA:   .BYTE   96                                        ;


                              03FC 00000          .ENTRY  SMG$INSERT_LINE, Save R2,R3,R4,R5,R6,R7,R8,-; 1823
                                                          R9                                        ;
                 59 00000000G  00  9E 00002       MOVAB   SMG$$PUT_TEXT_TO_BUFFER, R9               ;
                 5E            18  C2 00009       SUBL2   #24, SP                                   ;
                               58  D4 0000C       CLRL    STR_LEN                                   ; 1938
                        04     AE  D4 0000E       CLRL    WRAP_CHARS                                ;
       50        6C            02  83 00011       SUBB3   #2, (AP), DIFF                            ; 1960
                 06            50  91 00015       CMPB    DIFF, #6                                  ;
                               08  1B 00018       BLEQU   1$                                        ;
                 50 00000000G  8F  D0 0001A       MOVL    #SMG$_WRONUMARG, R0                       ;
                                   04 00021       RET                                               ;
                 50        04  BC  D0 00022 1$:   MOVL    @DISPLAY_ID, R0                           ; 1962
            04   BC        38  A0  D1 00026       CMPL    56(R0), @DISPLAY_ID                       ;
                               06  12 0002B       BNEQ    2$                                        ;
                 11        44  A0  91 0002D       CMPB    68(R0), #17                               ;
                               08  13 00031       BEQL    3$                                        ;
                 50 00000000G  8F  D0 00033 2$:   MOVL    #SMG$_INVDIS_ID, R0                       ;
                                   04 0003A       RET                                               ;
                 53        04  BC  D0 0003B 3$:   MOVL    @DISPLAY_ID, DCB                          ;
                 54        08  BC  D0 0003F       MOVL    @LINE_NO, R4                              ; 1969
  54     48  A3      10            00  ED 00043       CMPZV   #0, #16, 72(DCB), R4                      ;
                               08  14 00049       BGTR    4$                                        ;
  54     4A  A3      10            00  ED 0004B       CMPZV   #0, #16, 74(DCB), R4                      ; 1970
                               08  18 00051       BGEQ    5$                                        ;
                 50 00000000G  00  9E 00053 4$:   MOVAB   SMG$_INVROW, R0                           ; 1972
                                   04 0005A       RET                                               ;
                 57            01  D0 0005B 5$:   MOVL    #1, DIR                                   ; 1974
                 04            6C  91 0005E       CMPB    (AP), #4                                  ; 1976
```

```
                                    15  1F 00061         BLSSU   7$                              ;
                              10    AC  D5 00063         TSTL    16(AP)                          ;
                                    10  13 00066         BEQL    7$                              ;
                        01    10    BC  D1 00068         CMPL    @DIRECTION, #1                  ; 1979
                                    06  13 0006C         BEQL    6$                              ;
                        02    10    BC  D1 0006E         CMPL    @DIRECTION, #2                  ; 1980
                                    21  12 00072         BNEQ    9$                              ;
                        57    10    BC  D0 00074 6$:     MOVL    @DIRECTION, DIR                 ; 1984
                  0C    AE          01  D0 00078 7$:     MOVL    #1, C_SET                       ; 1987
                        08          6C  91 0007C         CMPB    (AP), #8                        ; 1989
                                    21  1F 0007F         BLSSU   11$                             ;
                              20    AC  D5 00081         TSTL    32(AP)                          ;
                                    1C  13 00084         BEQL    11$                             ;
              04        00    20    BC  CF 00086         CASEL   @CHAR_SET, #0, #4               ; 1992
    0012    0012      0012        0012     0008B 8$:     .WORD   10$-8$,-                        ;
                                  0012     00093                 10$-8$,-                        ;
                                                                 10$-8$,-                        ;
                                                                 10$-8$,-                        ;
                                                                 10$-8$                          ;
                        50 00000000G  00  9E 00095 9$:   MOVAB   SMG$_INVARG, R0                 ; 1999
                                        04 0009C         RET                                     ;
                  0C    AE    20    BC  D0 0009D 10$:    MOVL    @CHAR_SET, C_SET                ; 1997
                        03          6C  91 000A2 11$:    CMPB    (AP), #3                        ; 2007
                                    16  1F 000A5         BLSSU   12$                             ;
                              0C    AC  D5 000A7         TSTL    12(AP)                          ;
                                    11  13 000AA         BEQL    12$                             ;
                        50    0C    AC  D0 000AC         MOVL    STRING, R0                      ; 2010
                           00000000G  00  16 000B0       JSB     LIB$ANALYZE_SDESC_R2            ;
                        58          51  D0 000B6         MOVL    R1, R8                          ;
                        01          50  E8 000B9         BLBS    STATUS, 12$                     ;
                                        04 000BC         RET                                     ;
                        56    06    A3  9E 000BD 12$:    MOVAB   6(DCB), R6                      ; 2025
                        01          57  D1 000C1         CMPL    DIR, #1                         ; 2020
                                    1C  12 000C4         BNEQ    13$                             ;
                                    01  DD 000C6         PUSHL   #1                              ; 2022
                                    01  DD 000C8         PUSHL   #1                              ;
                        7E          66  3C 000CA         MOVZWL  (R6), -(SP)                     ; 2025
                        50    48    A3  3C 000CD         MOVZWL  72(DCB), R0                     ; 2024
              50        54          50  C3 000D1         SUBL3   R0, R4, R0                      ;
                              01    A0  9F 000D5         PUSHAB  1(R0)                           ;
                        7E    04    A3  3C 000D8         MOVZWL  4(DCB), -(SP)                   ; 2023
                        7E    48    A3  3C 000DC         MOVZWL  72(DCB), -(SP)                  ; 2022
                                    17  11 000E0         BRB     14$                             ;
                                    01  DD 000E2 13$:    PUSHL   #1                              ; 2028
                                    02  DD 000E4         PUSHL   #2                              ;
                        7E          66  3C 000E6         MOVZWL  (R6), -(SP)                     ; 2030
                        50    4A    A3  3C 000E9         MOVZWL  74(DCB), R0                     ; 2029
                        50          54  C2 000ED         SUBL2   R4, R0                          ;
                              01    A0  9F 000F0         PUSHAB  1(R0)                           ;
                        7E    04    A3  3C 000F3         MOVZWL  4(DCB), -(SP)                   ; 2028
                                    54  DD 000F7         PUSHL   R4                              ;
                                    53  DD 000F9 14$:    PUSHL   DCB                             ;
               00000000G  00        07  FB 000FB         CALLS   #7, SMG$$SCROLL_AREA            ;
                           4C B344  94 00102             CLRB    @76(DCB)[R4]                    ; 2043
                                    58  D5 00106         TSTL    STR_LEN                         ; 2052
                                    3D  13 00108         BEQL    17$                             ;
                        55    2E    A3  9A 0010A         MOVZBL  46(DCB), REND_CODE              ; 2055
```

```
                05          6C  91 0010E          CMPB    (AP), #5
                            09  1F 00111          BLSSU   15$
                      14    AC  D5 00113          TSTL    20(AP)
                            04  13 00116          BEQL    15$
                55    14    BC  C8 00118          BISL2   @RENDITION_SET, REND_CODE
                06          6C  91 0011C  15$:    CMPB    (AP), #6
                            09  1F 0011F          BLSSU   16$
                      18    AC  D5 00121          TSTL    24(AP)
                            04  13 00124          BEQL    16$
                55    18    BC  CC 00126          XORL2   @RENDITION_COMPLEMENT, REND_CODE
                28  A3      54  B0 0012A  16$:    MOVW    R4, 40(DCB)                      2057
                2A  A3      01  B0 0012E          MOVW    #1, 42(DCB)                      2058
                34  A3      01  8A 00132          BICB2   #1, 52(DCB)                      2060
                      04    AE  9F 00136          PUSHAB  WRAP_CHARS                       2062
                      10    AE  9F 00139          PUSHAB  C_SET
                            52  DD 0013C          PUSHL   STR_ADDR                         2065
                    0128    8F  BB 0013E          PUSHR   #^M<R3,R5,R8>                    2062
                    69      06  FB 00142          CALLS   #6, SMG$$PUT_TEXT_TO_BUFFER
                            24  11 00145          BRB     19$                              2052
        28  A3          54  01  A1 00147  17$:    ADDW3   #1, R4, 40(DCB)                  2077
                4A  A3  28  A3  B1 0014C          CMPW    40(DCB), 74(DCB)                 2078
                            0B  1A 00151          BGTRU   18$
                34  A3      01  8A 00153          BICB2   #1, 52(DCB)                      2081
                2A  A3  04  A3  B0 00157          MOVW    4(DCB), 42(DCB)                  2082
                            0D  11 0015C          BRB     19$
                28  A3  4A  A3  B0 0015E  18$:    MOVW    74(DCB), 40(DCB)                 2086
                2A  A3      66  B0 00163          MOVW    (R6), 42(DCB)                    2087
                34  A3      03  88 00167          BISB2   #3, 52(DCB)                      2089
                      04    AE  D5 0016B  19$:    TSTL    WRAP_CHARS                       2099
                            03  12 0016E          BNEQ    20$
                          00C0  31 00170          BRW     34$
                07          6C  91 00173  20$:    CMPB    (AP), #7                         2102
                            03  1E 00176          BGEQU   22$
                          009F  31 00178  21$:    BRW     33$
                      1C    AC  D5 0017B  22$:    TSTL    28(AP)
                            F8  13 0017E          BEQL    21$
                      1C    BC  D5 00180          TSTL    @WRAP_FLAG                       2103
                            F3  13 00183          BEQL    21$
                12  AE  010E  8F  B0 00185        MOVW    #270, TEMP_DESC+2                2110
        10  AE      58      66  A3 0018B          SUBW3   (R6), STR_LEN, TEMP_DESC         2111
                    50      66  3C 00190          MOVZWL  (R6), R0                         2113
        14  AE      52      50  C1 00193          ADDL3   R0, STR_ADDR, TEMP_DESC+4
                    02      57  D1 00198          CMPL    DIR, #2                          2119
                            3C  12 0019B          BNEQ    27$
                08  AE  01  A4  9E 0019D          MOVAB   1(R4), OVERFLOW_LINE             2124
                        0C  AE  9F 001A2          PUSHAB  C_SET                            2125
                        1C  AC  DD 001A5          PUSHL   WRAP_FLAG                        2133
                06          6C  91 001A8          CMPB    (AP), #6                         2130
                            0A  1F 001AB          BLSSU   23$
                      18    AC  D5 001AD          TSTL    24(AP)
                            05  13 001B0          BEQL    23$
                      18    AC  DD 001B2          PUSHL   RENDITION_COMPLEMENT             2131
                            02  11 001B5          BRB     24$
                            7E  D4 001B7  23$:    CLRL    -(SP)                            2130
                05          6C  91 001B9  24$:    CMPB    (AP), #5                         2127
                            0A  1F 001BC          BLSSU   25$
                      14    AC  D5 001BE          TSTL    20(AP)
```

```
                                05   13 001C1         BEQL    25$                                    ;
                          14    AC   DD 001C3         PUSHL   RENDITION_SET                          ; 2128
                                02   11 001C6         BRB     26$                                    ;
                                7E   D4 001C8 25$:    CLRL    -(SP)                                  ; 2127
               10     AE        02   D0 001CA 26$:    MOVL    #2, 16(SP)                             ; 2126
                          10    AE   9F 001CE         PUSHAB  16(SP)                                 ;
                          24    AE   9F 001D1         PUSHAB  TEMP_DESC                              ; 2125
                          20    AE   9F 001D4         PUSHAB  OVERFLOW_LINE                          ;
                                35   11 001D7         BRB     32$                                    ;
                          0C    AE   9F 001D9 27$:    PUSHAB  C_SET                                  ; 2136
                          1C    AC   DD 001DC         PUSHL   WRAP_FLAG                              ; 2144
                      06        6C   91 001DF         CMPB    (AP), #6                               ; 2141
                                0A   1F 001E2         BLSSU   28$                                    ;
                          18    AC   D5 001E4         TSTL    24(AP)                                 ;
                                05   13 001E7         BEQL    28$                                    ;
                          18    AC   DD 001E9         PUSHL   RENDITION_COMPLEMENT                   ; 2142
                                02   11 001EC         BRB     29$                                    ;
                                7E   D4 001EE 28$:    CLRL    -(SP)                                  ; 2141
                      05        6C   91 001F0 29$:    CMPB    (AP), #5                               ; 2138
                                0A   1F 001F3         BLSSU   30$                                    ;
                          14    AC   D5 001F5         TSTL    20(AP)                                 ;
                                05   13 001F8         BEQL    30$                                    ;
                          14    AC   DD 001FA         PUSHL   RENDITION_SET                          ; 2139
                                02   11 001FD         BRB     31$                                    ;
                                7E   D4 001FF 30$:    CLRL    -(SP)                                  ; 2138
               10     AE        01   D0 00201 31$:    MOVL    #1, 16(SP)                             ; 2137
                          10    AE   9F 00205         PUSHAB  16(SP)                                 ;
                          24    AE   9F 00208         PUSHAB  TEMP_DESC                              ; 2136
                          08    AC   DD 0020B         PUSHL   LINE_NO                                ;
                          04    AC   DD 0020E 32$:    PUSHL   DISPLAY_ID                             ;
             FDEA     CF        08   FB 00211         CALLS   #8, SMG$INSERT_LINE                    ;
                                52   D4 00216         CLRL    DONE                                   ; 2146
                                2E   11 00218         BRB     35$                                    ; 2147
       14      2F     A3        01   E1 0021A 33$:    BBC     #1, 47(DCB), 34$                       ; 2155
               2A     A3        66   B0 0021F         MOVW    (R6), 42(DCB)                          ; 2158
                                01   DD 00223         PUSHL   #1                                     ; 2159
                        FDD6    CF   9F 00225         PUSHAB  P.AAA                                  ; 2161
                                01   DD 00229         PUSHL   #1                                     ; 2159
                          40    A5   9F 0022B         PUSHAB  64(REND_CODE)                          ; 2160
                                53   DD 0022E         PUSHL   DCB                                    ; 2159
                      69        05   FB 00230         CALLS   #5, SMG$$PUT_TEXT_TO_BUFFER            ;
                      52        01   D0 00233 34$:    MOVL    #1, DONE                               ; 2168
                      0F        52   E9 00236         BLBC    DONE, 35$                              ; 2176
                          08    BC   DD 00239         PUSHL   @LINE_NO                               ; 2180
                                10   DD 0023C         PUSHL   #16                                    ; 2178
                                53   DD 0023E         PUSHL   DCB                                    ;
        00000000G     00        03   FB 00240         CALLS   #3, SMG$$CHECK_FOR_OUTPUT_DCB          ;
                                     04 00247         RET                                            ; 2182
                      50        01   D0 00248 35$:    MOVL    #1, R0                                 ;
                                     04 0024B         RET                                            ; 2184

; Routine Size:  588 bytes,    Routine Base:  _SMG$CODE + 0845


; 1936          2185   1 !<BLF/PAGE>
```

```
%SBTTL 'SMG$PUT_CHARS - Put Text to Display'
GLOBAL ROUTINE SMG$PUT_CHARS (
                                  DISPLAY_ID,
                                  TEXT    : REF BLOCK [,BYTE],
                                  LINE_NO,
                                  COL_NO,
                                  ERASE_LINE_FLAG,
                                  RENDITION_SET,
                                  RENDITION_COMPLEMENT,
                                  CHAR_SET
                                ) =
!++
! FUNCTIONAL DESCRIPTION:
!
!       This routine is used to write messages to the virtual display.
!       No additional cursor motion is added.  Existing text is NOT
!       shifted over - SMG$PUT_CHARS will overwrite anything already
!       in the specified positions.  (See SMG$INSERT_CHARS to write new
!       text while preserving existing text.)
!
!       By default, SMG$PUT_CHARS will write out only the text passed to
!       it - other character positions in the line are not modified.
!       Optionally, the caller can request that the line be erased before
!       the new characters are output by setting ERASE_LINE_FLAG.  This
!       option is provided for callers who do not want to preserve the
!       previous contents of the line.
!
!       The internal display cursor position is left at the character position
!       following the text written.
!
! CALLING SEQUENCE:
!
!       ret_status.wlc.v = SMG$PUT_CHARS (DISPLAY_ID.rl.r,
!                                         TEXT.rt.dx
!                                         [,LINE_NO.rl.r, COL_NO.rl.r]
!                                         [,ERASE_LINE_FLAG.r[.r]
!                                         [,RENDITION_SET.rl.r]
!                                         [,RENDITION_COMPLEMENT.rl.r]
!                                         [,CHAR_SET.rl.r])
!
! FORMAL PARAMETERS:
!
!       DISPLAY_ID.rl.r Display id of virtual display
!
!       TEXT.rt.dx      Address of descriptor of output string.
!
!       LINE_NO.rl.r    Optional.  Address of line number at which to
!                       start output.  If omitted (=0), the current
!                       line number is used.
!
!       COL_NO.rl.r     Optional.  Address of column number at which
!                       to start output.  If omitted (=0), the current
!                       line number is used.
!
!       ERASE_LINE_FLAG.rl.r
!                       Optional.  Address of flag which specifies
!                       whether to erase the line before outputting
```

```
!                       the specified text.
!                       1==> Erase
!                       0==> Don't erase
!
!       RENDITION_SET.rl.r        Each 1 bit attribute in this parameter
!                                 causes the corresponding attribute to
!                                 be set in the display.  (See below for
!                                 list of settable attributes.)
!
!       RENDITION_COMPLEMENT.rl.r
!                                 Each 1 bit attribute in this parameter
!                                 causes the corresponding attribute to
!                                 be complemented in the display.  (See
!                                 below for list of complementable
!                                 attributes.)
!
!       If the same bit is specified in both the RENDITION_SET parameter
!       and in the RENDITION_COMPLEMENT parameter, the application is
!       RENDITION_SET followed by RENDITION complement.  Using these two
!       parameters together the caller can exercise arbitrary and
!       independent control over each attribute on a single call.  On an
!       attribute by attribute basis he can cause the following
!       transformations:
!
!               SET     COMPLEMENT      Action
!               ---     ----------      ------
!               0       0               Attribute unchanged.
!               1       0               Attribute set to "on"
!               0       1               Attribute set to complement of
!                                       current setting.
!               1       1               Attribute set to "off".
!
!
!       Attributes which can be manipulated in this manner are:
!
!       SMG$M_BLINK  displays characters blinking.
!       SMG$M_BOLD  displays characters in higher-than-normal
!                   intensity.
!       SMG$M_REVERSE  displays characters in reverse video -- that is,
!                   using the opposite default rendition of the
!                   virtual display.
!       SMG$M_UNDERLINE  displays characters underlined.
!
!       CHAR_SET.rl.r   Optional.  Character set to use.
!                       Choices are:
!                               SMG$C_UNITED_KINGDOM
!                               SMG$C_ASCII (default)
!                               SMG$C_SPEC_GRAPHICS
!                               SMG$C_ALT_CHAR
!                               SMG$C_ALT_GRAPHICS
!
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
```

```
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !       SS$_NORMAL        Normal successful completion
1 !       SMG$_INVCOL       Invalid column specification
1 !       SMG$_INVROW       Invalid row specification
1 !       LIB$_INVSTRDES    Invalid string descriptor
1 !       SMG$_WRONUMARG    Wrong number (or combination of) arguments
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !--
1
2     BEGIN
2
2     BUILTIN
2         NULLPARAMETER;
2
2     LOCAL
2         ROW,                                  ! Row where text goes
2         COL,                                  ! Column where text goes
2         REND_CODE,                            ! RENDITION_CODE code to use.
2         STR_LEN : INITIAL (0),                ! Length of text string
2         STR_ADDR,                             ! Addr. of text string
2         DCB : REF $DCB_DECL,                  ! Address of virtual display
2                                               ! control block
2         STATUS;                               ! Status of subroutine calls
2
2     LITERAL
2         K_SET_ARG = 6,
2         K_COMP_ARG = 7;
2
2     $SMG$VALIDATE_ARGCOUNT (2, 8);
2
2     $SMG$GET_DCB ( .DISPLAY_ID, DCB);     ! Get address of virtual display
2                                           ! control block.
2
2 !+
2 ! Get the length and address of the text string.
2 !-
2
3     IF NOT (STATUS = LIB$ANALYZE_SDESC_R2 ( .TEXT,
3                                             STR_LEN,
3                                             STR_ADDR))
2     THEN
2         RETURN (.STATUS);
2
2 !+
2 ! Check for all optional arguments.  Set local variables to caller's
2 ! values, when available, and defaults when arguments omitted.
2 !-
2
2     IF NOT NULLPARAMETER (LINE_NO) AND
2         ..LINE_NO NEQ 0
2     THEN
```

```
2             ROW = ..LINE_NO
2         ELSE
2             ROW = .DCB [DCB_W_CURSOR_ROW];
2
2         IF NOT NULLPARAMETER (COL_NO) AND
2             ..COL_NO NEQ 0
2         THEN
2             COL = ..COL_NO
2         ELSE
2             COL = .DCB [DCB_W_CURSOR_COL];
2
2         $SMG$VALIDATE_ROW_COL (.ROW, .COL);
2                                             ! verify row & col within display
2
2         $SMG$SET_REND_CODE (K_SET_ARG, K_COMP_ARG);
2                                             ! macro to use caller's args if present
2
2         IF NOT NULLPARAMETER( CHAR_SET )
2         THEN
3             BEGIN
3             CASE ..CHAR_SET FROM SMG$C_UNITED_KINGDOM TO SMG$C_ALT_GRAPHICS OF
3             SET
3
3             [SMG$C_UNITED_KINGDOM, SMG$C_ASCII, SMG$C_SPEC_GRAPHICS,
3               SMG$C_ALT_CHAR, SMG$C_ALT_GRAPHICS]:
3                     ;
3             [INRANGE, OUTRANGE]:
3                     RETURN (SMG$_INVARG);
3             TES;
2             END;
2
2
2     !+
2     ! Reset the line characteristics in case the line was previously
2     ! double high or wide.
2     !-
2
3         BEGIN
3         BIND
3             LINE_CHAR = .DCB [DCB_A_LINE_CHAR];
3         MAP
3             LINE_CHAR : VECTOR [,BYTE];
3         IF .LINE_CHAR [.ROW] NEQ 0          ! prev wide or high chars
3         THEN
4             BEGIN                           ! blank line
4             LOCAL
4                 START_INDEX;
4             START_INDEX = $SMG$LINEAR (.ROW, 1);
4             $SMG$BLANK_FILL_DCB (.DCB [DCB_W_NO_COLS], .START_INDEX);
4             LINE_CHAR [.ROW] = 0;
3             END;
2         END;
2
2     !+
2     ! If erase_line_flag is set, blank out the line before outputting the text.
2     !-
2
```

```
    IF NOT NULLPARAMETER (ERASE_LINE_FLAG) AND
        ..ERASE_LINE_FLAG NEQ 0
    THEN
        BEGIN
        LOCAL
            START_INDEX;
        START_INDEX = $SMG$LINEAR (.ROW, 1);
        $SMG$BLANK_FILL_DCB (.DCB [DCB_W_NO_COLS], .START_INDEX);
        END;

!+
! All local variables are set up.  Call routine to put text into the
! display buffer.
!-

    DCB [DCB_W_CURSOR_ROW] = .ROW;
    DCB [DCB_W_CURSOR_COL] = .COL;
                                  ! set current position for PUT_TEXT

    IF NOT (STATUS = SMG$$PUT_TEXT_TO_BUFFER ( .DCB,
                                               .REND_CODE,
                                               .STR_LEN,
                                               .STR_ADDR,
                                               IF NOT NULLPARAMETER (CHAR_SET)
                                               THEN ..CHAR_SET
                                               ELSE SMG$C_ASCII ))
    THEN
        RETURN (.STATUS);

!+
! Check for non-printable characters, such as LF, that may cause this
! call to affect several lines.  Normally we try to tell the output
! routines exactly what line changed (so it can optimize), but if we
! encountered funny characters which may have changed more than 1 line,
! just pass a 0 (which will turn off optimization).
!-
    BEGIN
    LOCAL
        SAVED_STRING_LEN,
        PRINT_LEN;

        SAVED_STRING_LEN = .STR_LEN;
                                  ! side effect of macro is to change STR_LEN
        $SMG$FIND_PRINT_LENGTH (STR_LEN, .STR_ADDR, PRINT_LEN);
                                  ! macro to do SCANC
        IF .PRINT_LEN NEQ .SAVED_STRING_LEN
        THEN
            ROW = 0;
    END;

!+
! Turn off scrolling.  This change just moved the cursor to some
! random position.
!-
    DCB [DCB_V_FULL] = 0;
```

```
; 2223          2471  2 !+
; 2224          2472  2 ! See if this change should be reflected on the screen.
; 2225          2473  2 !-
; 2226          2474  2
; 2227          2475  3     RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
; 2228          2476  3                                        SMG$C_PUT_CHARS,
; 2229          2477  2                                        .ROW));
; 2230          2478  1     END;                               ! End of routine SMG$PUT_CHARS



                                   0FFC 00000          .ENTRY  SMG$PUT_CHARS, Save R2,R3,R4,R5,R6,R7,R8,-  ; 2187
                                                               R9,R10,R11                                    ;
                   5E           0C  C2 00002           SUBL2   #12, SP                                       ;
                                7E  D4 00005           CLRL    STR_LEN                                       ; 2315
         50        6C           02  83 00007           SUBB3   #2, (AP), DIFF                                ; 2334
                   06           50  91 0000B           CMPB    DIFF, #6                                      ;
                                08  1B 0000E           BLEQU   1$                                            ;
                   50 00000000G 8F  D0 00010           MOVL    #SMG$_WRONUMARG, R0                           ;
                                04  00017              RET                                                   ;
                   50     04    BC  D0 00018 1$:       MOVL    @DISPLAY_ID, R0                               ; 2336
              04   BC     38    A0  D1 0001C           CMPL    56(R0), @DISPLAY_ID                           ;
                                06  12 00021           BNEQ    2$                                            ;
                   11     44    A0  91 00023           CMPB    68(R0), #17                                   ;
                                08  13 00027           BEQL    3$                                            ;
                   50 00000000G 8F  D0 00029 2$:       MOVL    #SMG$_INVDIS_ID, R0                           ;
                                04  00030              RET                                                   ;
                   57     04    BC  D0 00031 3$:       MOVL    @DISPLAY_ID, DCB                              ;
                   50     08    AC  D0 00035           MOVL    TEXT, R0                                      ; 2343
                      00000000G 00  16 00039           JSB     LIB$ANALYZE_SDESC_R2                          ;
              04   AE           50  D0 0003F           MOVL    R0, STATUS                                    ;
                   6E           51  D0 00043           MOVL    R1, (SP)                                      ;
              0C   AE           52  D0 00046           MOVL    R2, 12(SP)                                    ;
                   03     04    AE  E8 0004A           BLBS    STATUS, 4$                                    ;
                              0152  31 0004E           BRW     22$                                           ;
                   03           6C  91 00051 4$:       CMPB    (AP), #3                                      ; 2354
                                10  1F 00054           BLSSU   5$                                            ;
                          0C    AC  D5 00056           TSTL    12(AP)                                        ;
                                0B  13 00059           BEQL    5$                                            ;
                          0C    BC  D5 0005B           TSTL    @LINE_NO                                      ; 2355
                                06  13 0005E           BEQL    5$                                            ;
                   56     0C    BC  D0 00060           MOVL    @LINE_NO, ROW                                 ; 2357
                                04  11 00064           BRB     6$                                            ;
                   56     28    A7  3C 00066 5$:       MOVZWL  40(DCB), ROW                                  ; 2359
                   04           6C  91 0006A 6$:       CMPB    (AP), #4                                      ; 2361
                                10  1F 0006D           BLSSU   7$                                            ;
                          10    AC  D5 0006F           TSTL    16(AP)                                        ;
                                0B  13 00072           BEQL    7$                                            ;
                          10    BC  D5 00074           TSTL    @COL_NO                                       ; 2362
                                06  13 00077           BEQL    7$                                            ;
                   5B     10    BC  D0 00079           MOVL    @COL_NO, COL                                  ; 2364
                                04  11 0007D           BRB     8$                                            ;
                   5B     2A    A7  3C 0007F 7$:       MOVZWL  42(DCB), COL                                  ; 2366
                                56  D5 00083 8$:       TSTL    ROW                                           ; 2368
                                08  15 00085           BLEQ    9$                                            ;
```

```
      56       02   A7          10          00   ED 00087        CMPZV   #0, #16, 2(DCB), ROW                ;
                                            08   18 0008D        BGEQ    10$                                 ;
                                50 00000000G 8F  D0 0008F 9$:    MOVL    #SMG$_INVROW, R0                    ;
                                                 04 00096        RET                                         ;
                                            5B   D5 00097 10$:   TSTL    COL                                 ;
                                            08   15 00099        BLEQ    11$                                 ;
      5B       06   A7          10          00   ED 0009B        CMPZV   #0, #16, 6(DCB), COL                ;
                                            08   18 000A1        BGEQ    12$                                 ;
                                50 00000000G 8F  D0 000A3 11$:   MOVL    #SMG$_INVCOL, R0                    ;
                                                 04 000AA        RET                                         ;
                          08   AE      2E   A7   9A 000AB 12$:   MOVZBL  46(DCB), REND_CODE                  ; 2371
                               06           6C   91 000B0        CMPB    (AP), #6                            ;
                                            0A   1F 000B3        BLSSU   13$                                 ;
                                       18   AC   D5 000B5        TSTL    24(AP)                              ;
                                            05   13 000B8        BEQL    13$                                 ;
                          08   AE      18   BC   C8 000BA        BISL2   @RENDITION_SET, REND_CODE           ;
                               07           6C   91 000BF 13$:   CMPB    (AP), #7                            ;
                                            0A   1F 000C2        BLSSU   14$                                 ;
                                       1C   AC   D5 000C4        TSTL    28(AP)                              ;
                                            05   13 000C7        BEQL    14$                                 ;
                          08   AE      1C   BC   CC 000C9        XORL2   @RENDITION_COMPLEMENT, REND_CODE    ;
                               08           6C   91 000CE 14$:   CMPB    (AP), #8                            ; 2374
                                            1C   1F 000D1        BLSSU   16$                                 ;
                                       20   AC   D5 000D3        TSTL    32(AP)                              ;
                                            17   13 000D6        BEQL    16$                                 ;
                    04         00      20   BC   CF 000D8        CASEL   @CHAR_SET, #0, #4                   ; 2377
   0012           0012       0012         0012      000DD 15$:   .WORD   16$-15$,-                           ;
                                          0012      000E5                16$-15$,-                           ;
                                                                         16$-15$,-                           ;
                                                                         16$-15$,-                           ;
                                                                         16$-15$                             ;
                               50 00000000G 00   9E 000E7        MOVAB   SMG$_INVARG, R0                     ; 2384
                                                 04 000EE        RET                                         ;
                                     4C B746     95 000EF 16$:   TSTB    @76(DCB)[ROW]                       ; 2399
                                            38   13 000F3        BEQL    18$                                 ;
                               50      FF   A6   9E 000F5        MOVAB   -1(R6), R0                          ; 2404
                               51      06   A7   3C 000F9        MOVZWL  6(DCB), R1                          ;
                               50           51   C4 000FD        MULL2   R1, R0                              ;
                               5A           50   D0 00100        MOVL    R0, START_INDEX                     ;
                               50      10   A7   D0 00103        MOVL    16(DCB), TEXT_BUF                   ; 2405
                               58      14   A7   7D 00107        MOVQ    20(DCB), ATTR_BUF                   ;
   06   A7             20      6E           00   2C 0010B        MOVC5   #0, (SP), #32, 6(DCB), (START_INDEX)- ;
                                          6A40      00111                [TEXT_BUF]                          ;
   06   A7       2E    A7      6E           00   2C 00113        MOVC5   #0, (SP), 46(DCB), 6(DCB), (START_INDEX)- ;
                                          6A48      0011A                [ATTR_BUF]                          ;
                                            59   D5 0011C        TSTL    CHAR_BUF                            ;
                                            09   13 0011E        BEQL    17$                                 ;
   06   A7       30    A7      6E           00   2C 00120        MOVC5   #0, (SP), 48(DCB), 6(DCB), (START_INDEX)- ;
                                          6A49      00127                [CHAR_BUF]                          ;
                                     4C B746     94 00129 17$:   CLRB    @76(DCB)[ROW]                       ; 2406
                               05           6C   91 0012D 18$:   CMPB    (AP), #5                            ; 2414
                                            3E   1F 00130        BLSSU   19$                                 ;
                                       14   AC   D5 00132        TSTL    20(AP)                              ;
                                            39   13 00135        BEQL    19$                                 ;
                                       14   BC   D5 00137        TSTL    @ERASE_LINE_FLAG                    ; 2415
                                            34   13 0013A        BEQL    19$                                 ;
                               50      FF   A6   9E 0013C        MOVAB   -1(R6), R0                          ; 2420
```

```
                        51     06    A7   3C 00140          MOVZWL  6(DCB), R1                                  ;
                        50           51   C4 00144          MULL2   R1, R0                                      ;
                        5A           50   D0 00147          MOVL    R0, START_INDEX                             ;
                        50     10    A7   D0 0014A          MOVL    16(DCB), TEXT_BUF                           ; 2421
                        58     14    A7   7D 0014E          MOVQ    20(DCB), ATTR_BUF                           ;
    06   A7        20                6E   00 2C 00152       MOVC5   #0, (SP), #32, 6(DCB), (START_INDEX)-       ;
                                   6A40      00158                  [TEXT_BUF]                                  ;
    06   A7    2E  A7                6E   00 2C 0015A       MOVC5   #0, (SP), 46(DCB), 6(DCB), (START_INDEX)-   ;
                                   6A48      00161                  [ATTR_BUF]                                  ;
                                     59   D5 00163          TSTL    CHAR_BUF                                    ;
                                     09   13 00165          BEQL    19$                                         ;
    06   A7    30  A7                6E   00 2C 00167       MOVC5   #0, (SP), 48(DCB), 6(DCB), (START_INDEX)-   ;
                                   6A49      0016E                  [CHAR_BUF]                                  ;
                   28  A7            56   B0 00170 19$:     MOVW    ROW, 40(DCB)                                ; 2429
                   2A  A7            5B   B0 00174          MOVW    COL, 42(DCB)                                ; 2430
                       08            6C   91 00178          CMPB    (AP), #8                                    ; 2437
                                     0A   1F 0017B          BLSSU   20$                                         ;
                             20      AC   D5 0017D          TSTL    32(AP)                                      ;
                                     05   13 00180          BEQL    20$                                         ;
                             20      BC   DD 00182          PUSHL   @CHAR_SET                                   ; 2438
                                     02   11 00185          BRB     21$                                         ;
                                     01   DD 00187 20$:     PUSHL   #1                                          ; 2437
                             10      AE   DD 00189 21$:     PUSHL   STR_ADDR                                    ; 2436
                             08      AE   DD 0018C          PUSHL   STR_LEN                                     ; 2435
                             14      AE   DD 0018F          PUSHL   REND_CODE                                   ; 2434
                                     57   DD 00192          PUSHL   DCB                                         ; 2433
               00000000G 00          05   FB 00194          CALLS   #5, SMG$$PUT_TEXT_TO_BUFFER                 ;
                      04 AE          50   D0 0019B          MOVL    R0, STATUS                                  ;
                         05    04    AE   E8 0019F          BLBS    STATUS, 23$                                 ;
                         50    04    AE   D0 001A3 22$:     MOVL    STATUS, R0                                  ; 2441
                                          04 001A7          RET                                                 ;
                         5B          6E   D0 001A8 23$:     MOVL    STR_LEN, SAVED_STRING_LEN                   ; 2456
                         5A          01   8E 001AB          MNEGB   #1, ALLONES                                 ; 2458
                                     55   D4 001AE          CLRL    PRINT_LEN                                   ;
                         54          6E   D0 001B0          MOVL    STR_LEN, BYTES_REMAINING                    ;
                         58    0C    AE   D0 001B3          MOVL    STR_ADDR, IN_POINTER                        ;
                                     6E   D4 001B7          CLRL    STR_LEN                                     ;
                                     54   D5 001B9 24$:     TSTL    BYTES_REMAINING                             ;
                                     59   13 001BB          BEQL    28$                                         ;
        5A 00000000G 00        68    54   2A 001BD          SCANC   BYTES_REMAINING, (IN_POINTER), CHAR_TABLE, -;
                                                                    ALLONES                                     ;
                   52          54    50   C3 001C6          SUBL3   NEW_BYTES_REMAINING, BYTES_REMAINING, R2    ;
                               58    52   C0 001CA          ADDL2   R2, IN_POINTER                              ;
                               55    52   C0 001CD          ADDL2   R2, PRINT_LEN                               ;
                               6E    52   C0 001D0          ADDL2   R2, STR_LEN                                 ;
                               54    50   D0 001D3          MOVL    NEW_BYTES_REMAINING, BYTES_REMAINING        ;
                                     3E   13 001D6          BEQL    28$                                         ;
                               50    61   9A 001D8          MOVZBL  (ADDR_DIFF), R0                             ;
                   09          01 00000000G0040 8F 001DB    CASEB   CHAR_TABLE[R0], #1, #9                      ;
      001C         002C      002C      002C    001E4 25$:   .WORD   27$-25$,-                                   ;
      0032         0032      0032      0032    001EC                27$-25$,-                                   ;
                             002C      0032    001F4                27$-25$,-                                   ;
                                                                    26$-25$,-                                   ;
                                                                    28$-25$,-                                   ;
                                                                    28$-25$,-                                   ;
                                                                    28$-25$,-                                   ;
                                                                    28$-25$,-                                   ;
```

```
                                                                28$-25$,-                                 ;
                                                                27$-25$                                   ;
                     50 00000000G  8F  D0 001F8         MOVL    #SMG$_FATERRLIB, R0                       ;
                                       04 001FF         RET                                               ;
                     50        2A  A7  3C 00200 26$:    MOVZWL  42(DCB), R0                               ;
                                   50  D7 00204         DECL    R0                                        ;
                     50            08  C6 00206         DIVL2   #8, R0                                    ;
                     55       09 A540  7E 00209         MOVAQ   9(PRINT_LEN)[R0], PRINT_LEN               ;
                                   6E  D6 0020E         INCL    STR_LEN                                   ;
                                   58  D6 00210 27$:    INCL    IN_POINTER                                ;
                                   54  D7 00212         DECL    BYTES_REMAINING                           ;
                                   A3  11 00214         BRB     24$                                       ;
                     5B            55  D1 00216 28$:    CMPL    PRINT_LEN, SAVED_STRING_LEN               ; 2460
                                   02  13 00219         BEQL    29$                                       ;
                                   56  D4 0021B         CLRL    ROW                                       ; 2462
              34     A7            01  8A 0021D 29$:    BICB2   #1, 52(DCB)                               ; 2469
                                   56  DD 00221         PUSHL   ROW                                       ; 2477
                                   11  DD 00223         PUSHL   #17                                       ; 2475
                                   57  DD 00225         PUSHL   DCB                                       ;
         00000000G   00            03  FB 00227         CALLS   #3, SMG$$CHECK_FOR_OUTPUT_DCB             ;
                                       04 0022E         RET                                               ; 2478

; Routine Size:  559 bytes,    Routine Base:  _SMG$CODE + 0A91

; 2231          2479  1 !<BLF/PAGE>
```

```
%SBTTL 'SMG$PUT_LINE - Put Text to Display in Line Mode'
GLOBAL ROUTINE SMG$PUT_LINE (
                                DISPLAY_ID,
                                TEXT      : REF BLOCK [,BYTE],
                                LINE_ADV,
                                RENDITION_SET,
                                RENDITION_COMPLEMENT,
                                WRAP_FLAG,
                                CHAR_SET
                            ) =

!++
! FUNCTIONAL DESCRIPTION:
!
!       This routine is used to write lines to the virtual display
!       optionally followed by cursor movement sequences.  SMG$PUT_LINE
!       writes from the current cursor position to the end of the line.
!       If the caller's text does not span to the end of the line, blank
!       fill is added.
!
!       Treatment of text which exceeds the rightmost bounds of the
!       display depends on WRAP_FLAG.  If WRAP_FLAG is set, lines are
!       scrolled LINE_ADV times to make room for the overflow characters
!       in the 'next' line.  If wrap is off, overflow characters are lost.
!
!       Following a call to SMG$PUT_LINE, the internal display cursor
!       position is set to column 1 of the next line where output should
!       occur.  The next line where output should occur is determined by
!       LINE_ADV;  LINE_ADV defaults to 1 so that subsequent calls to
!       SMG$PUT_LINE will not cause overprinting.
!
! CALLING SEQUENCE:
!
!       ret_status.wlc.v = SMG$PUT_LINE (DISPLAY_ID.rl.r,
!                                        TEXT.rt.dx
!                                        [,LINE_ADV.rl.r]
!                                        [,RENDITION_SET.rl.r]
!                                        [,RENDITION_COMPLEMENT.rl.r]
!                                        [,WRAP_FLAG.rl.r]
!                                        [,CHAR_SET.rl.r])
!
! FORMAL PARAMETERS:
!
!       DISPLAY_ID.rl.r Display id of virtual display
!
!       TEXT.rt.dx      Address of descriptor of output string.
!
!       LINE_ADV.rl.r   Optional.  Address of signed number of lines
!                       to advance after output.
!
!       RENDITION_SET.rl.r      Each 1 bit attribute in this parameter
!                               causes the corresponding attribute to
!                               be set in the display.  (See below for
!                               list of settable attributes.)
!
!       RENDITION_COMPLEMENT.rl.r
!                               Each 1 bit attribute in this parameter
```

```
1 !                                  causes the corresponding attribute to
1 !                                  be complemented in the display.  (See
1 !                                  below for list of complementable
1 !                                  attributes.)
1 !
1 !          If the same bit is specified in both the RENDITION_SET parameter
1 !          and in the RENDITION_COMPLEMENT parameter, the application is
1 !          RENDITION_SET followed by RENDITION complement.  Using these two
1 !          parameters together the caller can exercise arbitrary and
1 !          independent control over each attribute on a single call.  On an
1 !          attribute by attribute basis he can cause the following
1 !          transformations:
1 !
1 !                  SET     COMPLEMENT       Action
1 !                  ---     ----------       ------
1 !                  0       0                Attribute unchanged.
1 !                  1       0                Attribute set to "on"
1 !                  0       1                Attribute set to complement of
1 !                                           current setting.
1 !                  1       1                Attribute set to "off".
1 !
1 !          Attributes which can be manipulated in this manner are:
1 !
1 !          SMG$M_BLINK  displays characters blinking.
1 !          SMG$M_BOLD  displays characters in higher-than-normal
1 !                      intensity.
1 !          SMG$M_REVERSE  displays characters in reverse video -- that is,
1 !                        using the opposite default rendition of the
1 !                        virtual display.
1 !          SMG$M_UNDERLINE  displays characters underlined.
1 !
1 !          WRAP_FLAG.rl.r   = 0 means no wrap
1 !                           = 1 means wrap
1 !                           If omitted, no wrap is the default.
1 !
1 !          CHAR_SET.rl.r    Optional.  Character set to use.
1 !                           Choices are:
1 !                                   SMG$C_UNITED_KINGDOM
1 !                                   SMG$C_ASCII (default)
1 !                                   SMG$C_SPEC_GRAPHICS
1 !                                   SMG$C_ALT_CHAR
1 !                                   SMG$C_ALT_GRAPHICS
1 ! IMPLICIT INPUTS:
1 !
1 !          NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !          NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !          SS$_NORMAL       Normal successful completion
1 !          SMG$_WRONUMARG   Wrong number (of combination of) arguments
1 !          SMG$_INVARG      Negative LINE_ADV specified
```

```
!       LIB$_INVSTRDES  Invalid string descriptor
!
! SIDE EFFECTS:
!
!       NONE
!--

    BEGIN

    BUILTIN
        NULLPARAMETER;

    LOCAL
        DONE,
        DCB : REF $DCB_DECL,            ! Address of virtual display
                                        ! control block.
        STR_LEN : INITIAL (0),          ! Length of text string
        STR_ADDR,                       ! Address of text string
        REND_CODE,                      ! Rendition code to use
        WRAPPED_CHARS :INITIAL (0),     ! Number of chars that don't fit on
                                        ! the current line
                                        ! Set by call to SMG$$PUT_TEXT_TO_BUFFER
        SCROLL_FLAG : INITIAL (0),      ! Flag to scroll up, down, or neither
                                        ! Set by $SMG$SET_SCROLLING
        STATUS;                         ! Status of subroutine calls

    LITERAL
        K_SET_ARG = 4,
        K_COMP_ARG = 5;

    MACRO $SCROLL_UP (COUNT) =
            SMG$$SCROLL_AREA (.DCB,
                              .DCB [DCB_W_TOP_OF_SCRREG],
                              .DCB [DCB_W_COL_START],
                              (.DCB [DCB_W_BOTTOM_OF_SCRREG] -
                               .DCB [DCB_W_TOP_OF_SCRREG] + 1),
                              .DCB [DCB_W_NO_COLS],
                              SMG$M_UP, COUNT) %;

    $SMG$VALIDATE_ARGCOUNT (2,7);

    $SMG$GET_DCB (.DISPLAY_ID, DCB);    ! Get address of virtual display
                                        ! control block.

    IF NOT NULLPARAMETER (LINE_ADV) AND
       ..LINE_ADV LSS 0
    THEN
        RETURN (SMG$_INVARG);           ! positive advancing only

!+
! Select rendition code to use, based on whether one was provided by
! caller.
!-

    $SMG$SET_REND_CODE (K_SET_ARG, K_COMP_ARG);

!+
```

```
! Get the length and address of the text string.
!-
    IF NOT (STATUS = LIB$ANALYZE_SDESC_R2 ( .TEXT;
                                             STR_LEN,
                                             STR_ADDR))
    THEN
        RETURN (.STATUS);

!+
! If the previous line written was the last line in the display, we
! did not scroll at the end of the operation.  (This would've always
! left the last line blank - effectively the display would have one
! less useable line.)  If we're at the bottom, scroll up one before writing.
!
!-

    $SMG$SET_SCROLLING (SCROLL_FLAG);

    IF .SCROLL_FLAG EQL 1
    THEN                            ! we're at the last line in the display
        BEGIN                       ! and display is full
        IF NOT NULLPARAMETER (LINE_ADV)
        THEN
            BEGIN       ! Arg present
            IF ..LINE_ADV GTR 0
            THEN                            ! positive line advancing

                $SCROLL_UP (..LINE_ADV)

            ELSE
                BEGIN   ! Negative line advance specified

                IF ..LINE_ADV LSS 0
                THEN
                    RETURN (SMG$_INVARG);

                END     ! Negative line advance specified
            END         ! Arg present

        ELSE    ! Arg not present, default to 1 line

            $SCROLL_UP (1)

        END;                        ! we're at the last line in the display

!+
! Blank out the line before writing new text.
!-

    BEGIN
    LOCAL
        START_INDEX;

    START_INDEX = $SMG$LINEAR (.DCB [DCB_W_CURSOR_ROW], .DCB [DCB_W_CURSOR_COL]);
    $SMG$BLANK_FILL_DCB ((.DCB [DCB_W_NO_COLS] - .DCB [DCB_W_CURSOR_COL] + 1),
                        .START_INDEX);
```

```
2       END;
2
2    !+
2    ! Reset the line characteristics in case the line was previously
2    ! double high or wide.
2    !-
2
3       BEGIN
3       BIND
3           LINE_CHAR = .DCB [DCB_A_LINE_CHAR];
3       MAP
3           LINE_CHAR : VECTOR [,BYTE];
3       LINE_CHAR [.DCB [DCB_W_CURSOR_ROW]] = 0;
2       END;
2
2    !+
2    ! Move the text string into our virtual display buffer.
2    ! Note the cursor position is already correct for the call.
2    !-
2
3       IF NOT ( STATUS = SMG$$PUT_TEXT_TO_BUFFER ( .DCB,
3                                                   .REND_CODE,
3                                                   .STR_LEN,
3                                                   .STR_ADDR,
3                                                   IF NOT NULLPARAMETER (CHAR_SET)
3                                                       THEN ..CHAR_SET
3                                                       ELSE SMG$C_ASCII,
3                                                  WRAPPED_CHARS))
2       THEN
2           RETURN (.STATUS);
2
2
2    !+
2    ! If all went well so far, we need to enter the <CR>,<LF> to form the
2    ! end of line.
2    !-
2
2       DCB [DCB_W_CURSOR_COL] = 1;          ! Effect of <CR>
2
2    !+
2    ! Default to advancing one line if LINE_ADV is omitted.
2    !-
2
2       IF NULLPARAMETER (LINE_ADV)
2       THEN
3           BEGIN                        ! line adv omitted
3           IF .DCB [DCB_W_CURSOR_ROW] + 1 LEQ .DCB [DCB_W_BOTTOM_OF_SCRREG]
3           THEN
3               ! Just advance cursor row to next line
3               DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_CURSOR_ROW] + 1
3           ELSE
4               BEGIN
4               $SMG$SET_SCROLLING (SCROLL_FLAG);
4               IF .SCROLL_FLAG EQL 1
4               THEN
4                   IF .DCB [DCB_W_CURSOR_ROW] NEQ .DCB [DCB_W_BOTTOM_OF_SCRREG]
4                   THEN
```

```
4                      $SCROLL_UP (1);      ! scroll if within scrolling region
4                  IF .DCB [DCB_W_CURSOR_ROW] LEQ .DCB [DCB_W_BOTTOM_OF_SCRREG]
4                  THEN
4                      DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_BOTTOM_OF_SCRREG];
4                  DCB [DCB_V_FOL] = 1;  ! remember that we used last line
3                  END;
3
3          END                         ! line_adv omitted
3
3  !+
3  ! Take care of the requested line advancing.
3  !-
3
2      ELSE
3          BEGIN                       !  line_adv specified
3          IF ..LINE_ADV GTR 0
3          THEN
4              BEGIN                   ! upspacing requested
4              IF .DCB [DCB_W_CURSOR_ROW] + ..LINE_ADV LEQ
4                  .DCB [DCB_W_BOTTOM_OF_SCRREG]
4              THEN
4                  ! Just advance cursor row number
4                  DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_CURSOR_ROW]
4                                          + ..LINE_ADV
4              ELSE
5                  BEGIN               ! scrolling up
5                  $SMG$SET_SCROLLING (SCROLL_FLAG);
5                  IF .SCROLL_FLAG EQL 1
5                  THEN
5                      IF .DCB [DCB_W_CURSOR_ROW] NEQ .DCB [DCB_W_BOTTOM_OF_SCRREG]
5                      THEN
5                          $SCROLL_UP (..LINE_ADV); ! scroll if w/in scroll region
5                  IF .DCB [DCB_W_CURSOR_ROW] LEQ .DCB [DCB_W_BOTTOM_OF_SCRREG]
5                  THEN
5                      DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_BOTTOM_OF_SCRREG];
5                  DCB [DCB_V_FOL] = 1;  ! remember that we used last line
5                  END                 ! scrolling up
3              END;                    ! upspace action
3
2          END;                        ! line_adv specified
2
2      !+
2      ! If wrapping was requested and some characters overflowed the line,
2      ! call ourself again with the remainder of the characters.
2      !-
2
2      IF .WRAPPED_CHARS NEQ 0 AND
3          ((NOT NULLPARAMETER (LINE_ADV) AND ..LINE_ADV GTR 0) OR
3          NULLPARAMETER (LINE_ADV))
2      THEN
3          BEGIN                           ! overflow chars
4          IF (NOT NULLPARAMETER (WRAP_FLAG) AND
4              ..WRAP_FLAG NEQ 0)
3          THEN
4              BEGIN                       ! wrap set - recurse w/overflow
4              LOCAL
4                  STR_DESC : BLOCK [8, BYTE],
```

```
                C_SET;

            STR_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
            STR_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
            STR_DESC [DSC$W_LENGTH] = .STR_LEN - .DCB [DCB_W_NO_COLS];
            STR_DESC [DSC$A_POINTER] = .STR_ADDR + .DCB [DCB_W_NO_COLS];

            C_SET = SMG$C_ASCII;
            IF NOT NULLPARAMETER( CHAR_SET )
            THEN
                C_SET = ..CHAR_SET;

            SMG$PUT_LINE (.DISPLAY_ID,
                          STR_DESC,
                          .LINE_ADV,
                          .RENDITION_SET,
                          .RENDITION_COMPLEMENT,
                          .WRAP_FLAG,
                          C_SET);
            DONE = 0;
            RETURN 1;               ! to keep Bliss happy
            END                     ! wrap set - recurse w/overflow
        ELSE
            BEGIN                   ! wrap not set - truncation
            !+
            ! Wrap was not requested but there were overflow characters.
            ! Put out diamond in last position to show truncation.
            !-
            IF .DCB [DCB_V_TRUNC_ICON]
            THEN
                BEGIN
                IF NOT .DCB [DCB_V_FULL]
                THEN
                    DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_CURSOR_ROW] - 1;
                DCB [DCB_W_CURSOR_COL] = .DCB [DCB_W_NO_COLS];

                SMG$$PUT_TEXT_TO_BUFFER (.DCB,
                                        .REND_CODE + ATTR_M_USER_GRAPHIC,
                                        1, UPLIT (BYTE (DIAMOND)),
                                        SMG$C_ASCII);

                IF NOT .DCB [DCB_V_FULL]
                THEN
                    DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_CURSOR_ROW] + 1;
                                        ! restore for next call
                DCB [DCB_W_CURSOR_COL] = 1;
                END;

            DONE = 1;
            END;                            ! wrap not set - truncation
        END
    ELSE
        DONE = 1;                           ! no wrap chars

    !+
    ! See if this change should be reflected on the screen.
    ! Even if we call SMG$PUT_LINE again, SMG$$CHECK_FOR_OUTPUT_DCB
```

```
; 2632      2879 2      ! should be called only once.
; 2633      2880 2      !-
; 2634      2881 2
; 2635      2882 2      IF .DONE
; 2636      2883 2      THEN
; 2637      2884 3          BEGIN
; 2638      2885 3          LOCAL
; 2639      2886 3              LINE_CHANGED;
; 2640      2887 4          LINE_CHANGED = .DCB [DCB_W_CURSOR_ROW] -(IF NOT NULLPARAMETER (LINE_ADV)
; 2641      2888 4                                                  THEN ABS (..LINE_ADV)
; 2642      2889 3                                                  ELSE 0);
; 2643      2890 4          RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
; 2644      2891 4                                            SMG$C_PUT_LINE,
; 2645      2892 3                                            .LINE_CHANGED));
; 2646      2893 3          END
; 2647      2894 2      ELSE
; 2648      2895 2          RETURN 1;
; 2649      2896 2
; 2650      2897 1      END;                                    ! End of routine SMG$PUT_LINE


                                60  000C0 P.AAB:  .BYTE   96                                              ;


                              0FFC  00000         .ENTRY  SMG$PUT_LINE, Save R2,R3,R4,R5,R6,R7,R8,R9,-;  2481
                                                          R10,R11                                     ;
            5E             24  C2  00002          SUBL2   #36, SP                                     ;
                           7E  D4  00005          CLRL    STR_LEN                                     ;  2601
                    18     AE  D4  00007          CLRL    WRAPPED_CHARS                               ;
                           7E  D4  0000A          CLRL    SCROLL_FLAG                                 ;
    50      6C             02  83  0000C          SUBB3   #2, (AP), DIFF                              ;  2633
            05             50  91  00010          CMPB    DIFF, #5                                    ;
                           08  1B  00013          BLEQU   1$                                          ;
            50 00000000G   8F  D0  00015          MOVL    #SMG$_WRONUMARG, R0                         ;
                           04  0001C              RET                                                 ;
            50      04     BC  D0  0001D 1$:      MOVL    @DISPLAY_ID, R0                             ;  2635
        04  BC      38     A0  D1  00021          CMPL    56(R0), @DISPLAY_ID                         ;
                           06  12  00026          BNEQ    2$                                          ;
            11      44     A0  91  00028          CMPB    68(R0), #17                                 ;
                           08  13  0002C          BEQL    3$                                          ;
            50 00000000G   8F  D0  0002E 2$:      MOVL    #SMG$_INVDIS_ID, R0                         ;
                           04  00035              RET                                                 ;
            56      04     BC  D0  00036 3$:      MOVL    @DISPLAY_ID, DCB                            ;
            03             6C  91  0003A          CMPB    (AP), #3                                    ;  2638
                           0A  1F  0003D          BLSSU   4$                                          ;
                    0C     AC  D5  0003F          TSTL    12(AP)                                      ;
                           05  13  00042          BEQL    4$                                          ;
                    0C     BC  D5  00044          TSTL    @LINE_ADV                                   ;  2639
                           7C  19  00047          BLSS    11$                                         ;
        14  AE      2E     A6  9A  00049 4$:      MOVZBL  46(DCB), REND_CODE                          ;  2648
            04             6C  91  0004E          CMPB    (AP), #4                                    ;
                           0A  1F  00051          BLSSU   5$                                          ;
                    10     AC  D5  00053          TSTL    16(AP)                                      ;
                           05  13  00056          BEQL    5$                                          ;
        14  AE      10     BC  C8  00058          BISL2   @RENDITION_SET, REND_CODE                   ;
```

```
                    05          6C  91 0005D 5$:    CMPB    (AP), #5                          ;
                                0A  1F 00060        BLSSU   6$                                ;
                          14    AC  D5 00062        TSTL    20(AP)                            ;
                                05  13 00065        BEQL    6$                                ;
         14   AE          14    BC  CC 00067        XORL2   @RENDITION_COMPLEMENT, REND_CODE  ;
              50          08    AC  D0 0006C 6$:    MOVL    TEXT, R0                          ; 2653
                   00000000G    00  16 00070        JSB     LIB$ANALYZE_SDESC_R2              ;
         10   AE                50  D0 00076        MOVL    R0, STATUS                        ;
         04   AE                51  D0 0007A        MOVL    R1, 4(SP)                         ;
         18   AE                52  D0 0007E        MOVL    R2, 24(SP)                        ;
              03          10    AE  E8 00082        BLBS    STATUS, 7$                        ;
                              00F1  31 00086        BRW     18$                               ;
                                6E  D4 00089 7$:    CLRL    SCROLL_FLAG                       ; 2667
         0C   AE          34    A6  9E 0008B        MOVAB   52(DCB), 12(SP)                   ;
              16          0C    BE  E9 00090        BLBC    @12(SP), 9$                       ;
         4A   A6          28    A6  B1 00094        CMPW    40(DCB), 74(DCB)                  ;
                                05  12 00099        BNEQ    8$                                ;
              6E                01  D0 0009B        MOVL    #1, SCROLL_FLAG                   ;
                                0A  11 0009E        BRB     9$                                ;
         48   A6          28    A6  B1 000A0 8$:    CMPW    40(DCB), 72(DCB)                  ;
                                03  12 000A5        BNEQ    9$                                ;
              6E                02  D0 000A7        MOVL    #2, SCROLL_FLAG                   ;
              01                6E  D1 000AA 9$:    CMPL    SCROLL_FLAG, #1                   ; 2669
                                45  12 000AD        BNEQ    14$                               ;
              03                6C  91 000AF        CMPB    (AP), #3                          ; 2672
                                19  1F 000B2        BLSSU   12$                               ;
                          0C    AC  D5 000B4        TSTL    12(AP)                            ;
                                14  13 000B7        BEQL    12$                               ;
              50          0C    BC  D0 000B9        MOVL    @LINE_ADV, R0                     ; 2675
                                04  15 000BD        BLEQ    10$                               ;
                                50  DD 000BF        PUSHL   R0                                ; 2678
                                0C  11 000C1        BRB     13$                               ;
                                2F  18 000C3 10$:   BGEQ    14$                               ; 2683
              50 00000000G      00  9E 000C5 11$:   MOVAB   SMG$_INVARG, R0                   ; 2685
                                    04 000CC        RET                                       ;
                                01  DD 000CD 12$:   PUSHL   #1                                ; 2692
                                01  DD 000CF 13$:   PUSHL   #1                                ;
              7E          06    A6  3C 000D1        MOVZWL  6(DCB), -(SP)                     ;
              50          4A    A6  3C 000D5        MOVZWL  74(DCB), R0                       ;
              51          48    A6  3C 000D9        MOVZWL  72(DCB), R1                       ;
              50                51  C2 000DD        SUBL2   R1, R0                            ;
                          01    A0  9F 000E0        PUSHAB  1(R0)                             ;
              7E          04    A6  3C 000E3        MOVZWL  4(DCB), -(SP)                     ;
              7E          48    A6  3C 000E7        MOVZWL  72(DCB), -(SP)                    ;
                                56  DD 000EB        PUSHL   DCB                               ;
    00000000G 00                07  FB 000ED        CALLS   #7, SMG$$SCROLL_AREA              ;
              57          28    A6  9E 000F4 14$:   MOVAB   40(DCB), R7                       ; 2704
              50                67  3C 000F8        MOVZWL  (R7), R0                          ;
                                50  D7 000FB        DECL    R0                                ;
              5B          06    A6  9E 000FD        MOVAB   6(DCB), R11                       ;
              51                6B  3C 00101        MOVZWL  (R11), R1                         ;
              50                51  C4 00104        MULL2   R1, R0                            ;
              51          2A    A6  3C 00107        MOVZWL  42(DCB), R1                       ;
         08   AE          FF  A140  9E 0010B        MOVAB   -1(R1)[R0], START_INDEX           ;
              50          10    A6  D0 00111        MOVL    16(DCB), TEXT_BUF                 ; 2706
              59          14    A6  D0 00115        MOVL    20(DCB), ATTR_BUF                 ;
              58          18    A6  D0 00119        MOVL    24(DCB), CHAR_BUF                 ;
```

```
                           5A          6B  3C 0011D          MOVZWL  (R11), R10                              :
                           5A          51  C2 00120          SUBL2   R1, R10                                 :
                                       5A  D6 00123          INCL    R10                                     :
      5A         20        6E          00  2C 00125          MOVC5   #0, (SP), #32, R10, @START_INDEX[TEXT_BUF] :
                               08 BE40        0012A                                                          :
      5A    2E   A6        6E          00  2C 0012D          MOVC5   #0, (SP), 46(DCB), R10, @START_INDEX-   :
                               08 BE49        00133                  [ATTR_BUF]                              :
                                       58  D5 00136          TSTL    CHAR_BUF                                :
                                       09  13 00138          BEQL    15$                                     :
      5A    30   A6        6E          00  2C 0013A          MOVC5   #0, (SP), 48(DCB), R10, @START_INDEX-   :
                               08 BE48        00140                  [CHAR_BUF]                              :
                           50          67  3C 00143 15$:     MOVZWL  (R7), R0                                : 2720
                           50      4C  A6  C0 00146          ADDL2   76(DCB), R0                             :
                                       60  94 0014A          CLRB    (R0)                                    :
                                   1C  AE  9F 0014C          PUSHAB  WRAPPED_CHARS                           : 2728
                           07          6C  91 0014F          CMPB    (AP), #7                                : 2732
                                       0A  1F 00152          BLSSU   16$                                     :
                                   1C  AC  D5 00154          TSTL    28(AP)                                  :
                                       05  13 00157          BEQL    16$                                     :
                                   1C  BC  DD 00159          PUSHL   @CHAR_SET                               : 2733
                                       02  11 0015C          BRB     17$                                     :
                                       01  DD 0015E 16$:     PUSHL   #1                                      : 2732
                                   20  AE  DD 00160 17$:     PUSHL   STR_ADDR                                : 2731
                                   10  AE  DD 00163          PUSHL   STR_LEN                                 : 2730
                                   24  AE  DD 00166          PUSHL   REND_CODE                               : 2729
                                       56  DD 00169          PUSHL   DCB                                     : 2728
                  00000000G  00        06  FB 0016B          CALLS   #6, SMG$$PUT_TEXT_TO_BUFFER             :
                         10  AE        50  D0 00172          MOVL    R0, STATUS                              :
                             05    10  AE  E8 00176          BLBS    STATUS, 19$                             :
                             50    10  AE  D0 0017A 18$:     MOVL    STATUS, R0                              : 2737
                                       04     0017E          RET                                             :
                     2A      A6        01  B0 0017F 19$:     MOVW    #1, 42(DCB)                             : 2745
                             03        6C  91 00183          CMPB    (AP), #3                                : 2751
                                       05  1F 00186          BLSSU   20$                                     :
                                   0C  AC  D5 00188          TSTL    12(AP)                                  :
                                       64  12 0018B          BNEQ    25$                                     :
                             50        67  3C 0018D 20$:     MOVZWL  (R7), R0                                : 2754
                                       50  D6 00190          INCL    R0                                      :
      50    4A   A6          10        00  ED 00192          CMPZV   #0, #16, 74(DCB), R0                    :
                                       04  19 00198          BLSS    21$                                     :
                                       67  B6 0019A          INCW    (R7)                                    : 2757
                                       6A  11 0019C          BRB     26$                                     :
                                       6E  D4 0019E 21$:     CLRL    SCROLL_FLAG                             : 2760
                             14    0C  BE  E9 001A0          BLBC    @12(SP), 23$                            :
                     4A      A6        67  B1 001A4          CMPW    (R7), 74(DCB)                           :
                                       05  12 001A8          BNEQ    22$                                     :
                             6E        01  D0 001AA          MOVL    #1, SCROLL_FLAG                         :
                                       09  11 001AD          BRB     23$                                     :
                     48      A6        67  B1 001AF 22$:     CMPW    (R7), 72(DCB)                           :
                                       03  12 001B3          BNEQ    23$                                     :
                             6E        02  D0 001B5          MOVL    #2, SCROLL_FLAG                         :
                             01        6E  D1 001B8 23$:     CMPL    SCROLL_FLAG, #1                         : 2761
                                       2C  12 001BB          BNEQ    24$                                     :
                     4A      A6        67  B1 001BD          CMPW    (R7), 74(DCB)                           : 2763
                                       26  13 001C1          BEQL    24$                                     :
                                       01  DD 001C3          PUSHL   #1                                      : 2765
                                       01  DD 001C5          PUSHL   #1                                      :
```

```
                              7E          6B  3C 001C7        MOVZWL  (R11), -(SP)              :
                              50    4A    A6  3C 001CA        MOVZWL  74(DCB), R0               :
                              51    48    A6  3C 001CE        MOVZWL  72(DCB), R1               :
                              50          51  C2 001D2        SUBL2   R1, R0                    :
                                    01    A0  9F 001D5        PUSHAB  1(R0)                     :
                              7E    04    A6  3C 001D8        MOVZWL  4(DCB), -(SP)             :
                              7E    48    A6  3C 001DC        MOVZWL  72(DCB), -(SP)            :
                                          56  DD 001E0        PUSHL   DCB                       :
                      00000000G   00      07  FB 001E2        CALLS   #7, SMG$$SCROLL_AREA      :
                           4A    A6       67  B1 001E9 24$:   CMPW    (R7), 74(DCB)             : 2766
                                          6C  1B 001ED        BLEQU   31$                       :
                                          6E  11 001EF        BRB     32$                       : 2769
                              50    0C    BC  D0 001F1 25$:   MOVL    @LINE_ADV, R0             : 2780
                                          6C  15 001F5        BLEQ    33$                       :
                              51          67  3C 001F7        MOVZWL  (R7), R1                  : 2783
                              51          50  C0 001FA        ADDL2   R0, R1                    :
         51       4A    A6    10          00  ED 001FD        CMPZV   #0, #16, 74(DCB), R1      : 2784
                                          05  19 00203        BLSS    27$                       :
                              67          50  A0 00205        ADDW2   R0, (R7)                  : 2788
                                          59  11 00208 26$:   BRB     33$                       : 2787
                                          6E  D4 0020A 27$:   CLRL    SCROLL_FLAG               : 2791
                              14    0C    BE  E9 0020C        BLBC    @12(SP), 29$              :
                           4A    A6       67  B1 00210        CMPW    (R7), 74(DCB)             :
                                          05  12 00214        BNEQ    28$                       :
                              6E          01  D0 00216        MOVL    #1, SCROLL_FLAG           :
                                          09  11 00219        BRB     29$                       :
                           48    A6       67  B1 0021B 28$:   CMPW    (R7), 72(DCB)             :
                                          03  12 0021F        BNEQ    29$                       :
                              6E          02  D0 00221        MOVL    #2, SCROLL_FLAG           :
                              01          6E  D1 00224 29$:   CMPL    SCROLL_FLAG, #1           : 2792
                                          2C  12 00227        BNEQ    30$                       :
                           4A    A6       67  B1 00229        CMPW    (R7), 74(DCB)             : 2794
                                          26  13 0022D        BEQL    30$                       :
                                          50  DD 0022F        PUSHL   R0                        : 2796
                                          01  DD 00231        PUSHL   #1                        :
                              7E          6B  3C 00233        MOVZWL  (R11), -(SP)              :
                              50    4A    A6  3C 00236        MOVZWL  74(DCB), R0               :
                              51    48    A6  3C 0023A        MOVZWL  72(DCB), R1               :
                              50          51  C2 0023E        SUBL2   R1, R0                    :
                                    01    A0  9F 00241        PUSHAB  1(R0)                     :
                              7E    04    A6  3C 00244        MOVZWL  4(DCB), -(SP)             :
                              7E    48    A6  3C 00248        MOVZWL  72(DCB), -(SP)            :
                                          56  DD 0024C        PUSHL   DCB                       :
                      00000000G   00      07  FB 0024E        CALLS   #7, SMG$$SCROLL_AREA      :
                           4A    A6       67  B1 00255 30$:   CMPW    (R7), 74(DCB)             : 2797
                                          04  1A 00259        BGTRU   32$                       :
                              67    4A    A6  B0 0025B 31$:   MOVW    74(DCB), (R7)             : 2799
                           0C    BE       01  88 0025F 32$:   BISB2   #1, @12(SP)               : 2800
                                    1C    AE  D5 00263 33$:   TSTL    WRAPPED_CHARS             : 2811
                                          03  12 00266        BNEQ    35$                       :
                                      009F    31 00268 34$:   BRW     42$                       :
                              03          6C  91 0026B 35$:   CMPB    (AP), #3                  : 2812
                                          0A  1F 0026E        BLSSU   36$                       :
                                    0C    AC  D5 00270        TSTL    12(AP)                    :
                                          05  13 00273        BEQL    36$                       :
                                    0C    BC  D5 00275        TSTL    @LINE_ADV                 :
                                          0A  14 00278        BGTR    37$                       :
```

```
                    03          6C  91 0027A 36$:   CMPB    (AP), #3                              ; 2813
                    05          1F  0027D           BLSSU   37$                                   ;
                          0C    AC  D5 0027F        TSTL    12(AP)                                ;
                    E4          12  00282           BNEQ    34$                                   ;
                    06          6C  91 00284 37$:   CMPB    (AP), #6                              ; 2816
                    4C          1F  00287           BLSSU   39$                                   ;
                          18    AC  D5 00289        TSTL    24(AP)                                ;
                    47          13  0028C           BEQL    39$                                   ;
                          18    BC  D5 0028E        TSTL    @WRAP_FLAG                            ; 2817
                    42          13  00291           BEQL    39$                                   ;
              26    AE    010E  8F  B0 00293        MOVW    #270, STR_DESC+2                      ; 2825
   24   AE    04    AE          6B  A3 00299        SUBW3   (R11), STR_LEN, STR_DESC              ; 2826
                    50          6B  3C 0029F        MOVZWL  (R11), R0                             ; 2827
              28    AE       18 BE40 9E 002A2       MOVAB   @STR_ADDR[R0], STR_DESC+4             ;
              20    AE          01  D0 002A8        MOVL    #1, C_SET                             ; 2829
                    07          6C  91 002AC        CMPB    (AP), #7                              ; 2830
                    0A          1F  002AF           BLSSU   38$                                   ;
                          1C    AC  D5 002B1        TSTL    28(AP)                                ;
                    05          13  002B4           BEQL    38$                                   ;
              20    AE    1C    BC  D0 002B6        MOVL    @CHAR_SET, C_SET                      ; 2832
                          20    AE  9F 002BB 38$:   PUSHAB  C_SET                                 ; 2834
                    7E    14    AC  7D 002BE        MOVQ    RENDITION_COMPLEMENT, -(SP)           ; 2838
                    7E    0C    AC  7D 002C2        MOVQ    LINE_ADV, -(SP)                       ; 2836
                          38    AE  9F 002C6        PUSHAB  STR_DESC                              ; 2834
                          04    AC  DD 002C9        PUSHL   DISPLAY_ID                            ;
              FD2F  CF          07  FB 002CC        CALLS   #7, SMG$PUT_LINE                      ;
                                52  D4 002D1        CLRL    DONE                                  ; 2841
                                67  11 002D3        BRB     45$                                   ; 2842
        30    2F    A6          01  E1 002D5 39$:   BBC     #1, 47(DCB), 42$                      ; 2850
                    02    0C    BE  E8 002DA        BLBS    @12(SP), 40$                          ; 2853
                                67  B7 002DE        DECW    (R7)                                  ; 2855
              2A    A6          6B  B0 002E0 40$:   MOVW    (R11), 42(DCB)                        ; 2856
                                01  DD 002E4        PUSHL   #1                                    ; 2858
                          FD15  CF  9F 002E6        PUSHAB  P.AAB                                 ; 2860
                                01  DD 002EA        PUSHL   #1                                    ; 2858
        50    20    AE 00000040 8F  C1 002EC        ADDL3   #64, REND_CODE, R0                    ; 2859
                                50  DD 002F5        PUSHL   R0                                    ;
                                56  DD 002F7        PUSHL   DCB                                   ; 2858
           00000000G  00        05  FB 002F9        CALLS   #5, SMG$$PUT_TEXT_TO_BUFFER           ;
                    02    0C    BE  E8 00300        BLBS    @12(SP), 41$                          ; 2863
                                67  B6 00304        INCW    (R7)                                  ; 2865
              2A    A6          01  B0 00306 41$:   MOVW    #1, 42(DCB)                           ; 2867
                    52          01  D0 0030A 42$:   MOVL    #1, DONE                              ; 2874
                    2C          52  E9 0030D        BLBC    DONE, 45$                             ; 2882
                    03          6C  91 00310        CMPB    (AP), #3                              ; 2887
                    10          1F  00313           BLSSU   43$                                   ;
                          0C    AC  D5 00315        TSTL    12(AP)                                ;
                    0B          13  00318           BEQL    43$                                   ;
                    50    0C    BC  D0 0031A        MOVL    @LINE_ADV, R0                         ; 2888
                    07          18  0031E           BGEQ    44$                                   ;
                    50          50  CE 00320        MNEGL   R0, R0                                ;
                    02          11  00323           BRB     44$                                   ;
                                50  D4 00325 43$:   CLRL    R0                                    ; 2887
                    51          67  3C 00327 44$:   MOVZWL  (R7), R1                              ;
        50          51          50  C3 0032A        SUBL3   R0, R1, LINE_CHANGED                  ;
                                50  DD 0032E        PUSHL   LINE_CHANGED                          ; 2892
                                12  DD 00330        PUSHL   #18                                   ; 2890
```

```
                                56 DD 00332          PUSHL   DCB                                    ;
              00000000G  00     03 FB 00334          CALLS   #3, SMG$$CHECK_FOR_OUTPUT_DCB          ;
                                   04 0033B          RET                                            ; 2895
                         50     01 D0 0033C 45$:     MOVL    #1, R0                                 ;
                                   04 0033F          RET                                            ; 2897

; Routine Size:  832 bytes,    Routine Base:  _SMG$CODE + 0CC1


; 2651          2898  1 !<BLF/PAGE>
```

```
%SBTTL 'SMG$PUT_WITH_SCROLL - Scroll virtual display and output'
GLOBAL ROUTINE SMG$PUT_WITH_SCROLL (
                        DISPLAY_ID,
                        STRING,
                        DIRECTION,
                        RENDITION_SET,
                        RENDITION_COMPLEMENT,
                        WRAP_FLAG,
                        CHAR_SET
                                ) =
!++
! FUNCTIONAL DESCRIPTION:
!
!       This routine scrolls a virtual display, if necessary,  and deposits
!       the string in the opened area.  If no string is specified, the opened
!       line is left blank.  This routine imitates a physical scrolling region
!       in a virtual display;  scrolling occurs when all the lines in the
!       virtual display have been filled.  A new line is opened at the bottom
!       of the virtual display for upscrolling, and at the top of the virtual
!       display for downscrolling.
!
!       Treatment of text which exceeds the rightmost boundary of the display
!       depends on the setting of WRAP_FLAG.  If WRAP_FLAG is on, another line
!       scroll occurs and overflow characters are written in the new space.
!       If WRAP_FLAG is off, overflow characters are lost.
!
!       The internal display cursor position is left at the character
!       position following the last character written.
!
!       See SMG$INSERT_LINE for inserting lines at locations other than the
!       first or last line of a virtual display.
!
! CALLING SEQUENCE:
!
!       ret_status.wlc.v = SMG$PUT_WITH_SCROLL (
!                               DISPLAY_ID.rl.r
!                               [,STRING.rt.dx]
!                               [,DIRECTION.rlu.r]
!                               [,RENDITION_SET.rl.r]
!                               [,RENDITION_COMPLEMENT.rl.r]
!                               [,WRAP_FLAG.rl.r]
!                               [,CHAR_SET.rl.r])
!
! FORMAL PARAMETERS:
!
!       DISPLAY_ID.rl.r         Display id of desired virtual display.
!
!       STRING.rt.dx            Optional.  String to be deposited in the opened
!                               line.  If omitted, a blank line is the default.
!
!       DIRECTION.rlu.r         Optional.  Direction to scroll.  Values:
!                                       SMG$M_UP   (up)   Default
!                                       SMG$M_DOWN (down)
!
!       RENDITION_SET.rl.r      Optional.  Each 1 bit attribute in this
!                               parameter causes the corresponding attribute
!                               to be set in the display.  (See below for
```

```
!                                   list of settable attributes.)
!
!       RENDITION_COMPLEMENT.rl.r
!                                   Optional.  Each 1 bit attribute in this
!                                   parameter causes the corresponding attribute
!                                   to be complemented in the display.  (See
!                                   below for list of complementable
!                                   attributes.)
!
!       If the same bit is specified in both the RENDITION_SET parameter
!       and in the RENDITION_COMPLEMENT parameter, the application is
!       RENDITION_SET followed by RENDITION complement.  Using these two
!       parameters together the caller can exercise arbitrary and
!       independent control over each attribute on a single call.  On an
!       attribute by attribute basis he can cause the following
!       transformations:
!
!               SET     COMPLEMENT      Action
!               ---     ----------      ------
!               0       0               Attribute unchanged.
!               1       0               Attribute set to "on"
!               0       1               Attribute set to complement of
!                                       current setting.
!               1       1               Attribute set to "off".
!
!
!       Attributes which can be manipulated in this manner are:
!
!       SMG$M_BLINK  displays characters blinking.
!       SMG$M_BOLD  displays characters in higher-than-normal
!                   intensity.
!       SMG$M_REVERSE  displays characters in reverse video -- that is,
!                   using the opposite default rendition of the
!                   virtual display.
!       SMG$M_UNDERLINE  displays characters underlined.
!
!
!       WRAP_FLAG.rl.r  Optional.
!                       = 0 means no wrap
!                       = 1 means wrap
!                       If omitted, no wrap is the default.
!
!       CHAR_SET.rl.r   Optional.  Character set to use.
!                       Choices are:
!                               SMG$C_UNITED_KINGDOM
!                               SMG$C_ASCII (default)
!                               SMG$C_SPEC_GRAPHICS
!                               SMG$C_ALT_CHAR
!                               SMG$C_ALT_GRAPHICS
!
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
!       NONE
```

```
!
! COMPLETION STATUS:
!
!       SS$_NORMAL       Normal successful completion
!       LIB$_INVSTRDES   Invalid string descriptor
!       SMG$_INVDIS_ID   Invalid display id.
!       SMG$_WRONUMARG   Wrong number of arguments.
!                     or ACTION is given but RENDITION is not.
!       SMG$_INVARG      Unrecognized action code
!                     or Unrecognized rendition code
!
! SIDE EFFECTS:
!
!       NONE
!--
    BEGIN

    MACRO $SCROLL_UP (COUNT) =
            SMG$$SCROLL_AREA (.DCB,
                              .DCB [DCB_W_TOP_OF_SCRREG],
                              .DCB [DCB_W_COL_START],
                              (.DCB [DCB_W_BOTTOM_OF_SCRREG] -
                               .DCB [DCB_W_TOP_OF_SCRREG] + 1),
                              .DCB [DCB_W_NO_COLS],
                              SMG$M_UP, COUNT) %;

    MACRO $SCROLL_DOWN (COUNT) =
            SMG$$SCROLL_AREA (.DCB,
                              .DCB [DCB_W_TOP_OF_SCRREG],
                              .DCB [DCB_W_COL_START],
                              (.DCB [DCB_W_BOTTOM_OF_SCRREG] -
                               .DCB [DCB_W_TOP_OF_SCRREG] + 1),
                              .DCB [DCB_W_NO_COLS],
                              SMG$M_DOWN, COUNT) %;

    BUILTIN
        NULLPARAMETER;

    LITERAL
        K_STRING_ARG = 2,
        K_SET_ARG = 4,
        K_COMP_ARG = 5;

    LOCAL
        DONE,                           ! Done flag
        STATUS,                         ! Status of subroutine calls
        DCB : REF $DCB_DECL,            ! Addr of display control block
        STR_LEN : INITIAL (0),          ! Length of string
        STR_ADDR,                       ! Addr of string
        WORK_DIR,                       ! Working direction
        WRAP_CHARS : INITIAL (0),       ! No. of overflow characters
        SCROLL_FLAG,                    ! Flag to scroll up, down, or neither
        REND_CODE;                      ! Rendition code from params

    $SMG$VALIDATE_ARGCOUNT (1, 7);

    $SMG$GET_DCB ( .DISPLAY_ID, DCB);   ! Get address of display control
```

I 2

```
2                                                       ! block
2
2   !+
2   ! Validity check the arguments.
2   !-
2
2       IF NOT NULLPARAMETER( DIRECTION )
2       THEN
3           BEGIN                               ! use caller's direction
4           IF (..DIRECTION NEQ SMG$M_UP AND
4               ..DIRECTION NEQ SMG$M_DOWN)
3           THEN
4               RETURN (SMG$_INVARG)
3           ELSE
3               WORK_DIR = ..DIRECTION;
3           END
2       ELSE
2           WORK_DIR = SMG$M_UP;                ! use default direction
2
2       IF NOT NULLPARAMETER( CHAR_SET )
2       THEN
3           BEGIN
3           CASE ..CHAR_SET FROM SMG$C_UNITED_KINGDOM TO SMG$C_ALT_GRAPHICS OF
3           SET
3
3           [SMG$C_UNITED_KINGDOM, SMG$C_ASCII, SMG$C_SPEC_GRAPHICS,
3            SMG$C_ALT_CHAR, SMG$C_ALT_GRAPHICS]:
3                   ;
3           [INRANGE, OUTRANGE]:
3                   RETURN (SMG$_INVARG);
3           TES;
2           END;
2
2   !+
2   ! Check user's text, if any.
2   !-
2
2       IF NOT NULLPARAMETER (STRING)
2       THEN
3           BEGIN                               ! use caller's text
4           IF NOT (STATUS = LIB$ANALYZE_SDESC_R2 (.STRING; STR_LEN, STR_ADDR))
3           THEN
3               RETURN (.STATUS);
2           END;                                ! end use caller's text
2
2   !+
2   ! If the display is full and we're at the top or bottom line (depending on
2   ! scrolling direction), then scroll to make a space.
2   !-
2
2       $SMG$SET_SCROLLING (SCROLL_FLAG);
2
2       IF .SCROLL_FLAG EQL 1 AND
2          .WORK_DIR EQL SMG$M_UP
2       THEN
2           $SCROLL_UP (1)          ! on bottom line
2       ELSE
```

```
	IF .SCROLL_FLAG EQL 2 AND
	   .WORK_DIR EQL SMG$M_DOWN
	THEN
	    $SCROLL_DOWN (1);	! on top line

    !+
    ! Reset the line characteristics in case the line was previously
    ! double high or wide.
    !-

    BEGIN
    BIND
	LINE_CHAR = .DCB [DCB_A_LINE_CHAR];
    MAP
	LINE_CHAR : VECTOR [,BYTE];
    LINE_CHAR [.DCB [DCB_W_CURSOR_ROW]] = 0;
    END;

    !+
    ! Write text into the space.  SMG$$PUT_TEXT_TO_BUFFER checks for
    ! funny characters (tabs, etc) and also updates the cursor position.
    !-

    IF .STR_LEN NEQ 0
    THEN
	BEGIN
	$SMG$SET_REND_CODE (K_SET_ARG, K_COMP_ARG);

	DCB [DCB_W_CURSOR_COL] = .DCB [DCB_W_COL_START];
					! set cursor for PUT_TEXT
	SMG$$PUT_TEXT_TO_BUFFER (.DCB, .REND_CODE,
				 .STR_LEN,
				 .STR_ADDR,
				 (IF NULLPARAMETER( CHAR_SET )
				      THEN SMG$C_ASCII
				      ELSE ..CHAR_SET),
				 WRAP_CHARS);
	END;

    !+
    ! Update the cursor position.  Set bits in the DCB which may affect
    ! future scrolling.
    !-

    DCB [DCB_W_CURSOR_COL] = 1;

    CASE .WORK_DIR FROM SMG$M_UP TO SMG$M_DOWN OF
    SET
	[SMG$M_UP]:
	    BEGIN
	    IF .DCB [DCB_W_CURSOR_ROW] + 1 LEQ .DCB [DCB_W_BOTTOM_OF_SCRREG]
	    THEN
		DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_CURSOR_ROW] + 1
	    ELSE
		DCB [DCB_V_FULL] = 1;
	    END;
```

```
2          [SMG$M_DOWN]:
3              BEGIN
3              IF .DCB [DCB_W_CURSOR_ROW] - 1 GEQ .DCB [DCB_W_TOP_OF_SCRREG]
3              THEN
3                  DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_CURSOR_ROW] - 1
3              ELSE
3                  DCB [DCB_V_FULL] = 1;
2              END;
2      TES;
2
2  !+
2  ! Check for overflow characters which should cause us to scroll 2 lines
2  ! in order to output everything (if wrap requested).
2  !-
2
2      IF .WRAP_CHARS NEQ 0
2      THEN
3          BEGIN
3          IF NOT NULLPARAMETER( WRAP_FLAG ) AND
3              ..WRAP_FLAG NEQ 0                ! wrap requested
3          THEN
4              BEGIN                            ! recurse until whole string output
4              LOCAL
4                  TEMP_DESC : BLOCK [8, BYTE],
4                  C_SET;
4
4              TEMP_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
4              TEMP_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
4              TEMP_DESC [DSC$W_LENGTH] = .STR_LEN - .DCB [DCB_W_NO_COLS];
4              TEMP_DESC [DSC$A_POINTER] = .STR_ADDR +
4                                          .DCB [DCB_W_NO_COLS];
4
4              C_SET = SMG$C_ASCII;
4              IF NOT NULLPARAMETER( CHAR_SET )
4              THEN
4                  C_SET = ..CHAR_SET;
4
4              !+
4              ! Notice that we always upscroll for wrapped text.  This is
4              ! because downscrolling would cause the last part of the line
4              ! to be displayed before the first part.  This approach may not
4              ! be perfect but it at least lines appear in a readable manner.
4              !-
4              IF .WORK_DIR EQL SMG$M_DOWN
4              THEN
5                  BEGIN
5                  LOCAL
5                      OVERFLOW_LINE;
5                  OVERFLOW_LINE = .DCB [DCB_W_CURSOR_ROW] + 1;
5                  SMG$INSERT_LINE (.DISPLAY_ID,
5                                   OVERFLOW_LINE,
5                                   TEMP_DESC, %REF (SMG$M_DOWN),
6                                   (IF NOT NULLPARAMETER (RENDITION_SET)
6                                        THEN .RENDITION_SET
5                                        ELSE 0),
6                                   (IF NOT NULLPARAMETER (RENDITION_COMPLEMENT)
6                                        THEN .RENDITION_COMPLEMENT
```

```
                                    ELSE 0),
                                .WRAP_FLAG, C_SET);
              DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_ROW_START];
              END
          ELSE
              SMG$PUT_WITH_SCROLL (.DISPLAY_ID,
                                   TEMP_DESC, %REF (SMG$M_UP),
                                   (IF NOT NULLPARAMETER (RENDITION_SET)
                                       THEN .RENDITION_SET
                                       ELSE 0),
                                   (IF NOT NULLPARAMETER (RENDITION_COMPLEMENT)
                                       THEN .RENDITION_COMPLEMENT
                                       ELSE 0),
                                   .WRAP_FLAG,
                                   C_SET);
          DONE = 0;
          RETURN 1;                      ! to keep Bliss happy
          END
      ELSE
          BEGIN
          !+
          ! Wrap was not requested but there were overflow characters.
          ! Put out diamond in last position to show truncation.
          !-
          IF .DCB [DCB_V_TRUNC_ICON]
          THEN
              BEGIN
              IF NOT .DCB [DCB_V_FULL]
              THEN
                  DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_CURSOR_ROW] - 1;
              DCB [DCB_W_CURSOR_COL] = .DCB [DCB_W_NO_COLS];

              SMG$$PUT_TEXT_TO_BUFFER (.DCB,
                                       .REND_CODE + ATTR_M_USER_GRAPHIC,
                                       1, UPLIT (BYTE (DIAMOND)),
                                       SMG$C_ASCII);

              IF NOT .DCB [DCB_V_FULL]
              THEN
                  DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_CURSOR_ROW] + 1;
                                         ! restore for next call
              END;

          DONE = 1;
          END;
      END
  ELSE
      DONE = 1;                          ! no wrap chars

!+
! See if this change should be reflected on the screen.
! Even if SMG$PUT_WITH_SCROLL is called multiple times, call
! SMG$$CHECK_FOR_OUTPUT_DCB only once.
!-

  IF .DONE
  THEN
```

```
      3         RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
      3                                           SMG$C_PUT_WITH_SCROLL))
      2     ELSE
      2         RETURN 1;
      2
      1     END;                   ! End of routine SMG$PUT_WITH_SCROLL


                                          01001            .BLKB    3
                                     60   01004 P.AAC:     .BYTE    96                                                  ;

                                   03FC   00000            .ENTRY   SMG$PUT_WITH_SCROLL, Save R2,R3,R4,R5,R6,-         ; 2900
                                                                    R7,R8,R9                                            ;
                59 00000000G   00   9E   00002            MOVAB    SMG$$PUT_TEXT_TO_BUFFER, R9                         ;
                5E             18   C2   00009            SUBL2    #24, SP                                             ;
                               58   D4   0000C            CLRL     STR_LEN                                             ; 3028
                        04     AE   D4   0000E            CLRL     WRAP_CHARS                                          ;
      50        6C             01   83   00011            SUBB3    #1, (AP), DIFF                                      ; 3067
                06             50   91   00015            CMPB     DIFF, #6                                            ;
                               08   1B   00018            BLEQU    1$                                                  ;
                50 00000000G   8F   D0   0001A            MOVL     #SMG$_WRONUMARG, R0                                 ;
                                    04   00021            RET                                                          ;
                50      04     BC   D0   00022 1$:        MOVL     @DISPLAY_ID, R0                                     ; 3069
           04   BC      38     A0   D1   00026            CMPL     56(R0), @DISPLAY_ID                                 ;
                               06   12   0002B            BNEQ     2$                                                  ;
                11      44     A0   91   0002D            CMPB     68(R0), #17                                         ;
                               08   13   00031            BEQL     3$                                                  ;
                50 00000000G   8F   D0   00033 2$:        MOVL     #SMG$_INVDIS_ID, R0                                 ;
                                    04   0003A            RET                                                          ;
                53      04     BC   D0   0003B 3$:        MOVL     @DISPLAY_ID, DCB                                    ;
                03             6C   91   0003F            CMPB     (AP), #3                                            ; 3076
                               17   1F   00042            BLSSU    5$                                                  ;
                        0C     AC   D5   00044            TSTL     12(AP)                                              ;
                               12   13   00047            BEQL     5$                                                  ;
                01      0C     BC   D1   00049            CMPL     @DIRECTION, #1                                      ; 3079
                               06   13   0004D            BEQL     4$                                                  ;
                02      0C     BC   D1   0004F            CMPL     @DIRECTION, #2                                      ; 3080
                               22   12   00053            BNEQ     8$                                                  ;
                57      0C     BC   D0   00055 4$:        MOVL     @DIRECTION, WORK_DIR                                ; 3084
                               03   11   00059            BRB      6$                                                  ; 3076
                57             01   D0   0005B 5$:        MOVL     #1, WORK_DIR                                        ; 3087
                07             6C   91   0005E 6$:        CMPB     (AP), #7                                            ; 3089
                               1C   1F   00061            BLSSU    9$                                                  ;
                        1C     AC   D5   00063            TSTL     28(AP)                                              ;
                               17   13   00066            BEQL     9$                                                  ;
         04     00      1C     BC   CF   00068            CASEL    @CHAR_SET, #0, #4                                   ; 3092
 0012  0012   0012            0012       0006D 7$:        .WORD    9$-7$,-                                             ;
                              0012       00075                     9$-7$,-                                             ;
                                                                   9$-7$,-                                             ;
                                                                   9$-7$,-                                             ;
                                                                   9$-7$                                               ;
                50 00000000G   00   9E   00077 8$:        MOVAB    SMG$_INVARG, R0                                     ; 3099
                                    04   0007E            RET                                                          ;
                02             6C   91   0007F 9$:        CMPB     (AP), #2                                            ; 3107
```

```
                        16  1F 00082       BLSSU   10$                                   ;
               08       AC  D5 00084       TSTL    8(AP)                                 ;
                        11  13 00087       BEQL    10$                                   ;
          50   08       AC  D0 00089       MOVL    STRING, R0                            ; 3110
        00000000G       00  16 0008D       JSB     LIB$ANALYZE_SDESC_R2                  ;
          58            51  D0 00093       MOVL    R1, R8                                ;
          01            50  E8 00096       BLBS    STATUS, 10$                           ;
                            04 00099       RET                                           ;
                        50  D4 0009A 10$:  CLRL    SCROLL_FLAG                           ; 3120
          56   34       A3  9E 0009C       MOVAB   52(DCB), R6                           ;
          16            66  E9 000A0       BLBC    (R6), 12$                             ;
     4A   A3   28       A3  B1 000A3       CMPW    40(DCB), 74(DCB)                      ;
                        05  12 000A8       BNEQ    11$                                   ;
          50            01  D0 000AA       MOVL    #1, SCROLL_FLAG                       ;
                        0A  11 000AD       BRB     12$                                   ;
     48   A3   28       A3  B1 000AF 11$:  CMPW    40(DCB), 72(DCB)                      ;
                        03  12 000B4       BNEQ    12$                                   ;
          50            02  D0 000B6       MOVL    #2, SCROLL_FLAG                       ;
          01            50  D1 000B9 12$:  CMPL    SCROLL_FLAG, #1                       ; 3122
                        0B  12 000BC       BNEQ    13$                                   ;
          01            57  D1 000BE       CMPL    WORK_DIR, #1                          ; 3123
                        06  12 000C1       BNEQ    13$                                   ;
                        01  DD 000C3       PUSHL   #1                                    ; 3125
                        01  DD 000C5       PUSHL   #1                                    ;
                        0E  11 000C7       BRB     14$                                   ;
          02            50  D1 000C9 13$:  CMPL    SCROLL_FLAG, #2                       ; 3127
                        2C  12 000CC       BNEQ    15$                                   ;
          02            57  D1 000CE       CMPL    WORK_DIR, #2                          ; 3128
                        27  12 000D1       BNEQ    15$                                   ;
                        01  DD 000D3       PUSHL   #1                                    ; 3130
                        02  DD 000D5       PUSHL   #2                                    ;
          7E   06       A3  3C 000D7 14$:  MOVZWL  6(DCB), -(SP)                         ;
          50   4A       A3  3C 000DB       MOVZWL  74(DCB), R0                           ;
          51   48       A3  3C 000DF       MOVZWL  72(DCB), R1                           ;
          50            51  C2 000E3       SUBL2   R1, R0                                ;
               01       A0  9F 000E6       PUSHAB  1(R0)                                 ;
          7E   04       A3  3C 000E9       MOVZWL  4(DCB), -(SP)                         ;
          7E   48       A3  3C 000ED       MOVZWL  72(DCB), -(SP)                        ;
                        53  DD 000F1       PUSHL   DCB                                   ;
 00000000G      00      07  FB 000F3       CALLS   #7, SMG$$SCROLL_AREA                  ;
          54   28       A3  9E 000FA 15$:  MOVAB   40(DCB), R4                           ; 3142
          50            64  3C 000FE       MOVZWL  (R4), R0                              ;
          50   4C       A3  C0 00101       ADDL2   76(DCB), R0                           ;
                        60  94 00105       CLRB    (R0)                                  ;
                        58  D5 00107       TSTL    STR_LEN                               ; 3150
                        42  13 00109       BEQL    21$                                   ;
          55   2E       A3  9A 0010B       MOVZBL  46(DCB), REND_CODE                    ; 3153
          04            6C  91 0010F       CMPB    (AP), #4                              ;
                        09  1F 00112       BLSSU   16$                                   ;
               10       AC  D5 00114       TSTL    16(AP)                                ;
                        04  13 00117       BEQL    16$                                   ;
          55   10       BC  C8 00119       BISL2   @RENDITION_SET, REND_CODE             ;
          05            6C  91 0011D 16$:  CMPB    (AP), #5                              ;
                        09  1F 00120       BLSSU   17$                                   ;
               14       AC  D5 00122       TSTL    20(AP)                                ;
                        04  13 00125       BEQL    17$                                   ;
          55   14       BC  CC 00127       XORL2   @RENDITION_COMPLEMENT, REND_CODE      ;
```

```
                    2A   A3      04   A3  B0 0012B 17$:    MOVW     4(DCB), 42(DCB)                    : 3155
                                 04   AE  9F 00130         PUSHAB   WRAP_CHARS                         : 3157
                         07           6C  91 00133         CMPB     (AP), #7                           : 3160
                                      05  1F 00136         BLSSU    18$                                :
                                 1C   AC  D5 00138         TSTL     28(AP)                             :
                                      04  12 0013B         BNEQ     19$                                :
                                      01  DD 0013D 18$:    PUSHL    #1                                 :
                                      03  11 0013F         BRB      20$                                :
                                 1C   BC  DD 00141 19$:    PUSHL    @CHAR_SET                          : 3162
                                      52  DD 00144 20$:    PUSHL    STR_ADDR                           : 3159
                               0128   8F  BB 00146         PUSHR    #^M<R3,R5,R8>                      : 3157
                         69           06  FB 0014A         CALLS    #6, SMG$$PUT_TEXT_TO_BUFFER        :
                    2A   A3           01  B0 0014D 21$:    MOVW     #1, 42(DCB)                        : 3171
               01        01           57  CF 00151         CASEL    WORK_DIR, #1, #1                   : 3173
                       0015     0004         00155 22$:    .WORD    23$-22$,-                          :
                                                                    24$-22$                            :
                         50           64  3C 00159 23$:    MOVZWL   (R4), R0                           : 3177
                                      50  D6 0015C         INCL     R0                                 :
     50        4A   A3   10           00  ED 0015E         CMPZV    #0, #16, 74(DCB), R0               :
                                      15  19 00164         BLSS     25$                                :
                                      64  B6 00166         INCW     (R4)                               : 3179
                                      14  11 00168         BRB      26$                                :
                         50           64  3C 0016A 24$:    MOVZWL   (R4), R0                           : 3186
                                      50  D7 0016D         DECL     R0                                 :
     50        48   A3   10           00  ED 0016F         CMPZV    #0, #16, 72(DCB), R0               :
                                      04  14 00175         BGTR     25$                                :
                                      64  B7 00177         DECW     (R4)                               : 3188
                                      03  11 00179         BRB      26$                                :
                         66           01  88 0017B 25$:    BISB2    #1, (R6)                           : 3190
                                 04   AE  D5 0017E 26$:    TSTL     WRAP_CHARS                         : 3199
                                      03  12 00181         BNEQ     27$                                :
                               00EA   31 00183             BRW      43$                                :
                         06           6C  91 00186 27$:    CMPB     (AP), #6                           : 3202
                                      03  1E 00189         BGEQU    29$                                :
                               00BE   31 0018B 28$:        BRW      41$                                :
                                 18   AC  D5 0018E 29$:    TSTL     24(AP)                             :
                                      F8  13 00191         BEQL     28$                                :
                                 18   BC  D5 00193         TSTL     @WRAP_FLAG                         : 3203
                                      F3  13 00196         BEQL     28$                                :
                    12   AE    010E   8F  B0 00198         MOVW     #270, TEMP_DESC+2                  : 3211
               10   AE   58      06   A3  A3 0019E         SUBW3    6(DCB), STR_LEN, TEMP_DESC         : 3212
                         50      06   A3  3C 001A4         MOVZWL   6(DCB), R0                         : 3214
               14   AE   52           50  C1 001A8         ADDL3    R0, STR_ADDR, TEMP_DESC+4          :
                    0C   AE           01  D0 001AD         MOVL     #1, C_SET                          : 3216
                         07           6C  91 001B1         CMPB     (AP), #7                           : 3217
                                      0A  1F 001B4         BLSSU    30$                                :
                                 1C   AC  D5 001B6         TSTL     28(AP)                             :
                                      05  13 001B9         BEQL     30$                                :
                    0C   AE      1C   BC  D0 001BB         MOVL     @CHAR_SET, C_SET                   : 3219
                         02           57  D1 001C0 30$:    CMPL     WORK_DIR, #2                       : 3227
                                      49  12 001C3         BNEQ     35$                                :
                    08   AE           64  3C 001C5         MOVZWL   (R4), OVERFLOW_LINE                : 3232
                                 08   AE  D6 001C9         INCL     OVERFLOW_LINE                      :
                                 0C   AE  9F 001CC         PUSHAB   C_SET                              : 3233
                                 18   AC  DD 001CF         PUSHL    WRAP_FLAG                          : 3242
                         05           6C  91 001D2         CMPB     (AP), #5                           : 3239
                                      0A  1F 001D5         BLSSU    31$                                :
```

```
                        14  AC  D5 001D7          TSTL     20(AP)                              ;
                            05  13 001DA          BEQL     31$                                 ;
                        14  AC  DD 001DC          PUSHL    RENDITION_COMPLEMENT                ; 3240
                            02  11 001DF          BRB      32$                                 ;
                            7E  D4 001E1 31$:     CLRL     -(SP)                               ; 3239
                04          6C  91 001E3 32$:     CMPB     (AP), #4                            ; 3236
                            0A  1F 001E6          BLSSU    33$                                 ;
                        10  AC  D5 001E8          TSTL     16(AP)                              ;
                            05  13 001EB          BEQL     33$                                 ;
                        10  AC  DD 001ED          PUSHL    RENDITION_SET                       ; 3237
                            02  11 001F0          BRB      34$                                 ;
                            7E  D4 001F2 33$:     CLRL     -(SP)                               ; 3236
           10   AE          02  D0 001F4 34$:     MOVL     #2, 16(SP)                          ; 3235
                        10  AE  9F 001F8          PUSHAB   16(SP)                              ;
                        24  AE  9F 001FB          PUSHAB   TEMP_DESC                           ; 3233
                        20  AE  9F 001FE          PUSHAB   OVERFLOW_LINE                       ;
                        04  AC  DD 00201          PUSHL    DISPLAY_ID                          ;
         F637   CF          08  FB 00204          CALLS    #8, SMG$INSERT_LINE                 ;
                64          63  B0 00209          MOVW     (DCB), (R4)                         ; 3243
                            3A  11 0020C          BRB      40$                                 ; 3227
                        0C  AE  9F 0020E 35$:     PUSHAB   C_SET                               ; 3246
                        18  AC  DD 00211          PUSHL    WRAP_FLAG                           ; 3254
                05          6C  91 00214          CMPB     (AP), #5                            ; 3251
                            0A  1F 00217          BLSSU    36$                                 ;
                        14  AC  D5 00219          TSTL     20(AP)                              ;
                            05  13 0021C          BEQL     36$                                 ;
                        14  AC  DD 0021E          PUSHL    RENDITION_COMPLEMENT                ; 3252
                            02  11 00221          BRB      37$                                 ;
                            7E  D4 00223 36$:     CLRL     -(SP)                               ; 3251
                04          6C  91 00225 37$:     CMPB     (AP), #4                            ; 3248
                            0A  1F 00228          BLSSU    38$                                 ;
                        10  AC  D5 0022A          TSTL     16(AP)                              ;
                            05  13 0022D          BEQL     38$                                 ;
                        10  AC  DD 0022F          PUSHL    RENDITION_SET                       ; 3249
                            02  11 00232          BRB      39$                                 ;
                            7E  D4 00234 38$:     CLRL     -(SP)                               ; 3248
           10   AE          01  D0 00236 39$:     MOVL     #1, 16(SP)                          ; 3247
                        10  AE  9F 0023A          PUSHAB   16(SP)                              ;
                        24  AE  9F 0023D          PUSHAB   TEMP_DESC                           ; 3246
                        04  AC  DD 00240          PUSHL    DISPLAY_ID                          ;
         FDB8   CF          07  FB 00243          CALLS    #7, SMG$PUT_WITH_SCROLL             ;
                            52  D4 00248 40$:     CLRL     DONE                                ; 3256
                            36  11 0024A          BRB      44$                                 ; 3257
     1F      2F   A3        01  E1 0024C 41$:     BBC      #1, 47(DCB), 43$                    ; 3265
                  02        66  E8 00251          BLBS     (R6), 42$                           ; 3268
                            64  B7 00254          DECW     (R4)                                ; 3270
           2A     A3    06  A3  B0 00256 42$:     MOVW     6(DCB), 42(DCB)                     ; 3271
                            01  DD 0025B          PUSHL    #1                                  ; 3273
                   FD9E     CF  9F 0025D          PUSHAB   P.AAC                               ; 3275
                            01  DD 00261          PUSHL    #1                                  ; 3273
                        40  A5  9F 00263          PUSHAB   64(REND_CODE)                       ; 3274
                            53  DD 00266          PUSHL    DCB                                 ; 3273
                  69        05  FB 00268          CALLS    #5, SMG$$PUT_TEXT_TO_BUFFER         ;
                  02        66  E8 0026B          BLBS     (R6), 43$                           ; 3278
                            64  B6 0026E          INCW     (R4)                                ; 3280
                  52        01  D0 00270 43$:     MOVL     #1, DONE                            ; 3288
                  0C        52  E9 00273          BLBC     DONE, 44$                           ; 3296
```

```
                                        15 DD 00276          PUSHL   #21                                          ; 3298
                                        53 DD 00278          PUSHL   DCB                                          ;
                     00000000G  00      02 FB 0027A          CALLS   #2, SMG$$CHECK_FOR_OUTPUT_DCB                ;
                                           04 00281          RET                                                  ; 3301
                                50      01 D0 00282 44$:     MOVL    #1, R0                                       ;
                                           04 00285          RET                                                  ; 3303

; Routine Size:  646 bytes,    Routine Base:  _SMG$CODE + 1005


; 3058          3304  1 !<BLF/PAGE>
```

```
%SBTTL 'SMG$READ_FROM_DISPLAY - Read text from virtual display'
GLOBAL ROUTINE SMG$READ_FROM_DISPLAY (
                                        DISPLAY_ID,
                                        RET_STRING,
                                        SEARCH_STRING
                                       ) =
!++
! FUNCTIONAL DESCRIPTION:
!
!       This procedure returns back to the caller a text string which
!       represents the current contents of a portion of a virtual
!       display line.  The current cursor position row determines which
!       line is to act as the source of the text.
!       If the SEARCH_STRING parameter is omitted, the contents of
!       the current line, from the current cursor column to the
!       right-most column, is returned in RET_STRING.
!
!       If SEARCH_STRING is provided, it is a list of character codes
!       that are to act as terminators in "back searching" -- the
!       process of determining the first character position to be
!       returned.  Each character in this string serves as a potential
!       terminator.  If no terminator is encountered during "back
!       searching", the search is terminated by column 1 of the
!       virtual display.
!
!       Some examples:
!               Assume the line of text which contains the current
!       cursor contains the text:
!
!       "$ COPY/LOG FOO1 FOO2      "
!                 ^
!
!       Assume the cursor is positioned at the "G".
!
!       SMG$READ_FROM_DISPLAY (Display_id, OUTSTRING )
!       will return "G FOO1 FOO2      " since no terminator was supplied,
!       and hence no "back-searching" is performed.
!
!       SMG$READ_FROM_DISPLAY (Display_id, OUTSTRING, '/' )
!       will return "/LOG FOO1 FOO2      " since the specified terminator
!       was found prior to encountering column 1.
!
!       SMG$READ_FROM_DISPLAY (Display_id, OUTSTRING, '' )
!       will return "$ COPY/LOG FOO1 FOO2      " since the specified
!       terminator was not found prior to encountering column 1.
!
!       SMG$READ_FROM_DISPLAY (Display_id, OUTSTRING, '#' )
!       will return "$ COPY/LOG FOO1 FOO2      " since the specified
!       terminator was not seen before column 1 was encountered.
!
!       SMG$READ_FROM_DISPLAY (Display_id, OUTSTRING, "Y/" )
!       will return "/LOG FOO1 FOO2      " since, of the specified
!       terminators, the "/" was encountered before column 1 or any of
!       the other terminators.
!
!       NOTE: The "back-searching" begins at the current cursor column.
!       Hence,  if the text string is:
```

```
!        "$COPY/LOG F001 F002      " and cursor is positioned
!                       ^
!
!        SMG$READ_FROM_DISPLAY (Display_id, OUTSTRING, ' ' ) will
!        return " F002       ".
!
! CALLING SEQUENCE:
!
!        ret_status.wlc.v = SMG$READ_FROM_DISPLAY (
!                                        DISPLAY_ID.rl.r,
!                                        RET_STRING.wt.dx
!                                        [,SEARCH_STRING.rt.dx])
!
! FORMAL PARAMETERS:
!
!        DISPLAY_ID.rl.r         Address of a display id.
!
!        RET_STRING.wt.dx        Address of a string descriptor,
!                                describing the output string to be
!                                produced.
!
!        SEARCH_STRING.rt.dx     [Optional].  If specified, the address
!                                of a string descriptor, describing a
!                                list of terminators which is to stop the
!                                backward search in determining the
!                                starting position of the string to be
!                                returned.
!                                If omitted, the returned string starts
!                                at the current cursor column position --
!                                no back searching is performed.
!
! IMPLICIT INPUTS:
!
!        NONE
!
! IMPLICIT OUTPUTS:
!
!        NONE
!
! COMPLETION STATUS:
!
!        SS$_NORMAL      Normal successful completion
!        LIB$_INVSTRDES  Invalid string descriptor
!        LIB$_INSVIRMEM  Insufficient virtual memory
!
! SIDE EFFECTS:
!
!        NONE
!--

    BEGIN
    BUILTIN
        NULLPARAMETER;

    LOCAL
        STATUS,                 ! Status of subroutine calls
```

```
2           START_COL,              ! Computed column number where source-
2                                   ! to-be-returned starts.
2
2           START_LOC : REF VECTOR [1, BYTE], ! Address of text byte
2                                   ! currently being inspected
2
2           DCB : REF $DCB_DECL;    ! Address of Display Control Block
2
2   !+
2   ! Validate number of arguments and extract corresponding DCB address
2   !-
2       $SMG$VALIDATE_ARGCOUNT (1, 3);
2       $SMG$GET_DCB (.DISPLAY_ID, DCB)
2
2       IF NOT NULLPARAMETER (SEARCH_STRING)
2       THEN
3           BEGIN   ! Terminator string supplied
3           LOCAL
3               T_LEN,              ! Length of terminator string
3
3               T_ADDR : REF VECTOR [,BYTE];        ! Address of 1st byte
3                                                   ! of terminator string
3
3           !+
3           ! Extract length and address of 1st byte of terminator string.
3           !-
4           IF NOT (STATUS = LIB$ANALYZE_SDESC_R2 ( .SEARCH_STRING ;
4                                                   T_LEN, T_ADDR))
3           THEN
3               RETURN (.STATUS);   ! Can't access terminator string
3
3           !+
3           ! Initialize pointer for searching.
3           !-
3           START_LOC =
3               (.DCB [DCB_W_CURSOR_ROW] - 1) * .DCB [DCB_W_NO_COLS] +
3                .DCB [DCB_W_CURSOR_COL] - 1 + .DCB [DCB_A_TEXT_BUF];
3
3           START_COL = .DCB [DCB_W_CURSOR_COL];
3           WHILE .START_COL NEQ 1
3           DO
4               BEGIN       ! Back search
4               !+
4               ! If text byte currently under inspection is one of the
4               ! the terminator bytes, exit from loop.
4               !-
4               INCR I FROM 0 TO .T_LEN -1
4               DO
5                   BEGIN   ! Try to match terminator loop
5                   IF .START_LOC [0] EQL .T_ADDR [.I]
5                   THEN
6                       BEGIN       ! Terminator found exit path
7                       RETURN (LIB$SCOPY_R_DX6 (
7                           .DCB [DCB_W_NO_COLS] - .START_COL + 1, ! # bytes
7                           .START_LOC,                           ! from
6                           .RET_STRING));                        ! To desc
5                       END;        ! Terminator found exit path
```

```
; 3231          3476  4                 END;    ! Try to match terminator loop
; 3232          3477  4               !+
; 3233          3478  4               ! This character doesn't match any of the terminator.
; 3234          3479  4               ! Back up where we are searching by one column.
; 3235          3480  4               !-
; 3236          3481  4               START_COL = .START_COL -1;
; 3237          3482  4               START_LOC = .START_LOC -1;
; 3238          3483  3               END;    ! Back search
; 3239          3484  3           !+
; 3240          3485  3           ! If we fall out of loop above, none of the terminators were
; 3241          3486  3           ! found.  We need to return the whole line.
; 3242          3487  3           !-
; 3243          3488  4           RETURN (LIB$SCOPY_R_DX6 (
; 3244          3489  4                       .DCB [DCB_W_NO_COLS],     ! # bytes
; 3245          3490  4                       .START_LOC,               ! from
; 3246          3491  3                       .RET_STRING));            ! To desc
; 3247          3492  3           END     ! Terminator string supplied
; 3248          3493  3
; 3249          3494  2       ELSE
; 3250          3495  2
; 3251          3496  3           BEGIN   ! No terminator string supplied
; 3252          3497  3           !+
; 3253          3498  3           ! Return string starting at current cursor column.
; 3254          3499  3           !-
; 3255          3500  3           START_COL = .DCB [DCB_W_CURSOR_COL];
; 3256          3501  2           END;    ! No terminator string supplied
; 3257          3502  2
; 3258          3503  2   !+
; 3259          3504  2   ! Recalculate starting byte offset within text buffer for selected
; 3260          3505  2   ! starting postion.
; 3261          3506  2   !-
; 3262          3507  2       START_LOC  = (.DCB [DCB_W_CURSOR_ROW] - 1) * .DCB [DCB_W_NO_COLS]
; 3263          3508  2                   + .START_COL - 1 + .DCB [DCB_A_TEXT_BUF];
; 3264          3509  2   !+
; 3265          3510  2   ! Deliver selected text string to callers string.
; 3266          3511  2   !-
; 3267          3512  3       RETURN (LIB$SCOPY_R_DX6 (
; 3268          3513  3               .DCB [DCB_W_NO_COLS] - .START_COL + 1,   ! # bytes
; 3269          3514  3               .START_LOC,                              ! start address
; 3270          3515  2               .RET_STRING));                           ! Dest. descr.
; 3271          3516  1       END;            ! End of routine SMG$READ_FROM_DISPLAY


                                 0FFC 00000         .ENTRY  SMG$READ_FROM_DISPLAY, Save R2,R3,R4,R5,R6,-  ; 3306
                                                            R7,R8,R9,R10,R11                              ;
                    5E        04  C2 00002          SUBL2   #4, SP                                        ;
          50        6C        01  83 00005          SUBB3   #1, (AP), DIFF                                ; 3430
                    02        50  91 00009          CMPB    DIFF, #2                                      ;
                              08  1B 0000C          BLEQU   1$                                            ;
                    50 00000000G  8F  D0 0000E      MOVL    #SMG$_WRONUMARG, R0                           ;
                              04  00015             RET                                                   ;
                    50     04  BC  D0 00016 1$:     MOVL    @DISPLAY_ID, R0                               ; 3431
          04  BC     38  A0  D1 0001A               CMPL    56(R0), @DISPLAY_ID                           ;
                              06  12 0001F          BNEQ    2$                                            ;
```

```
                 11        44  A0  91 00021          CMPB     68(R0), #17                 ;
                                08  13 00025          BEQL     3$                          ;
                 50 00000000G   8F  D0 00027 2$:      MOVL     #SMG$_INVDIS_ID, R0         ;
                                    04 0002E          RET                                  ;
                 57        04  BC  D0 0002F 3$:      MOVL     @DISPLAY_ID, DCB            ;
                 53        2A  A7  9E 00033          MOVAB    42(DCB), R3                 ; 3455
                 03            6C  91 00037          CMPB     (AP), #3                    ; 3433
                               6E  1F 0003A          BLSSU    9$                          ;
                           0C  AC  D5 0003C          TSTL     12(AP)                      ;
                               69  13 0003F          BEQL     9$                          ;
                 50        0C  AC  D0 00041          MOVL     SEARCH_STRING, R0           ; 3445
                    00000000G  00  16 00045          JSB      LIB$ANALYZE_SDESC_R2        ;
                 54            51  D0 0004B          MOVL     R1, R4                      ;
                 59            52  D0 0004E          MOVL     R2, R9                      ;
                 01            50  E8 00051          BLBS     STATUS, 4$                  ;
                                   04 00054          RET                                  ;
                 50        28  A7  3C 00055 4$:      MOVZWL   40(DCB), R0                 ; 3454
                               50  D7 00059          DECL     R0                          ;
                 51        06  A7  3C 0005B          MOVZWL   6(DCB), R1                  ;
                 50            51  C4 0005F          MULL2    R1, R0                      ;
                 52            63  3C 00062          MOVZWL   (R3), R2                    ; 3455
         51      50            52  C1 00065          ADDL3    R2, R0, R1                  ;
                 51        10  A7  C0 00069          ADDL2    16(DCB), R1                 ;
                 5A        FF  A1  9E 0006D          MOVAB    -1(R1), START_LOC           ;
                 5B            52  D0 00071          MOVL     R2, START_COL               ; 3457
                 6E        FF  A4  9E 00074          MOVAB    -1(R4), (SP)                ; 3465
                 01            5B  D1 00078 5$:      CMPL     START_COL, #1               ; 3458
                               20  13 0007B          BEQL     8$                          ;
                 58            01  CE 0007D          MNEGL    #1, I                       ; 3468
                               11  11 00080          BRB      7$                          ;
               6849            6A  91 00082 6$:      CMPB     (START_LOC), (I)[T_ADDR]    ;
                               0B  12 00086          BNEQ     7$                          ;
                 50        06  A7  3C 00088          MOVZWL   6(DCB), R0                  ; 3472
                 50            5B  C2 0008C          SUBL2    START_COL, R0               ;
                               50  D6 0008F          INCL     R0                          ;
                               3E  11 00091          BRB      10$                         ;
         EB      58            6E  F3 00093 7$:      AOBLEQ   (SP), I, 6$                 ; 3465
                               5B  D7 00097          DECL     START_COL                   ; 3481
                               5A  D7 00099          DECL     START_LOC                   ; 3482
                               DB  11 0009B          BRB      5$                          ; 3458
                 52        08  AC  D0 0009D 8$:      MOVL     RET_STRING, R2              ; 3489
                 51            5A  D0 000A1          MOVL     START_LOC, R1               ;
                 50        06  A7  3C 000A4          MOVZWL   6(DCB), R0                  ;
                               2E  11 000A8          BRB      11$                         ;
                 5B            63  3C 000AA 9$:      MOVZWL   (R3), START_COL             ; 3500
                 50        28  A7  3C 000AD          MOVZWL   40(DCB), R0                 ; 3507
                               50  D7 000B1          DECL     R0                          ;
                 51        06  A7  3C 000B3          MOVZWL   6(DCB), R1                  ;
                 50            51  C4 000B7          MULL2    R1, R0                      ;
         51      50            5B  C1 000BA          ADDL3    START_COL, R0, R1           ; 3508
                 51        10  A7  C0 000BE          ADDL2    16(DCB), R1                 ;
                 5A        FF  A1  9E 000C2          MOVAB    -1(R1), START_LOC           ;
                 57        06  A7  3C 000C6          MOVZWL   6(DCB), R7                  ; 3513
                 57            5B  C2 000CA          SUBL2    START_COL, R7               ;
                 50        01  A7  9E 000CD          MOVAB    1(R7), R0                   ;
                 52        08  AC  D0 000D1 10$:     MOVL     RET_STRING, R2              ;
                 51            5A  D0 000D5          MOVL     START_LOC, R1               ;
```

```
                    00000000G  00  16 000D8 11$:    JSB     LIB$SCOPY_R_DX6                 ;
                               04 000DE             RET                                     ; 3516

; Routine Size:  223 bytes,    Routine Base:  _SMG$CODE + 128B

; 3272          3517  1 !<BLF/PAGE>
```

```
%SBTTL 'SMG$RETURN_CURSOR_POS - Return current cursor position'
GLOBAL ROUTINE SMG$RETURN_CURSOR_POS (
                                        DISPLAY_ID,
                                        ROW_NUM,
                                        COL_NUM
                                       ) =
!++
! FUNCTIONAL DESCRIPTION:
!
!       Returns the current cursor row and column position in the
!       specified virtual display.
!
! CALLING SEQUENCE:
!
!       ret_status.wlc.v = SMG$RETURN_CURSOR_POS (
!                                               DISPLAY_ID.rl.r,
!                                               ROW_NUM.wl.r,
!                                               COL_NUM.wl.r)
!
! FORMAL PARAMETERS:
!
!       DISPLAY_ID.rl.r Display id of desired display.
!
!       ROW_NUM.wl.r    Returned row position.
!
!       COL_NUM.w.r     Returned column position.
!
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! COMPLETION STATUS:
!
!       SS$_NORMAL      Normal successful completion
!       SMG$_WRONUMARG  Wrong number arguments.
!       SMG$_INVDIS_ID  Invalid display Id.
!
! SIDE EFFECTS:
!
!       NONE
!--
    BEGIN
    LOCAL
        DCB : REF $DCB_DECL;            ! Addr. of display control block

    $SMG$VALIDATE_ARGCOUNT (3,3);       ! Check for right no. of args

    $SMG$GET_DCB ( .DISPLAY_ID, DCB);   ! Get address of display control
                                        ! block

    .ROW_NUM = .DCB [DCB_W_CURSOR_ROW];
    .COL_NUM = .DCB [DCB_W_CURSOR_COL];
```

```
; 3331          3575 2     RETURN (SS$_NORMAL);
; 3332          3576 1     END;                              ! End of routine SMG$RETURN_CURSOR_POS


                                  0000 00000         .ENTRY  SMG$RETURN_CURSOR_POS, Save nothing          ; 3519
               03              6C   91 00002         CMPB    (AP), #3                                     ; 3567
                               08   13 00005         BEQL    1$                                           ;
               50 00000000G    8F   D0 00007         MOVL    #SMG$_WRONUMARG, R0                          ;
                                    04 0000E         RET                                                  ;
               50          04  BC   D0 0000F 1$:     MOVL    @DISPLAY_ID, R0                              ; 3569
         04    BC          38  A0   D1 00013         CMPL    56(R0), @DISPLAY_ID                          ;
                               06   12 00018         BNEQ    2$                                           ;
               11          44  A0   91 0001A         CMPB    68(R0), #17                                  ;
                               08   13 0001E         BEQL    3$                                           ;
               50 00000000G    8F   D0 00020 2$:     MOVL    #SMG$_INVDIS_ID, R0                          ;
                                    04 00027         RET                                                  ;
               50          04  BC   D0 00028 3$:     MOVL    @DISPLAY_ID, DCB                             ;
         08    BC          28  A0   3C 0002C         MOVZWL  40(DCB), @ROW_NUM                            ; 3572
         0C    BC          2A  A0   3C 00031         MOVZWL  42(DCB), @COL_NUM                            ; 3573
               50              01   D0 00036         MOVL    #1, R0                                       ; 3575
                                    04 00039         RET                                                  ; 3576

; Routine Size:  58 bytes,    Routine Base:  _SMG$CODE + 136A
```

```
%SBTTL 'SMG$SCROLL_DISPLAY_AREA - Scroll rectangular area in a virtual display'
GLOBAL ROUTINE SMG$SCROLL_DISPLAY_AREA (
                                DISPLAY_ID,
                                START_ROW,
                                START_COL,
                                HEIGHT,
                                WIDTH,
                                DIRECTION,
                                COUNT
                               ) =
!++
! FUNCTIONAL DESCRIPTION:
!
!       This routine scrolls the contents of a rectangular area in a virtual
!       display in the specified direction for the specified count.  By default
!       the scrolling rectangle is the 'scrolling region' in the virtual
!       display.
!
!       The default direction is 'up' and the default number of units is 1.
!
!       The scrolling rectangle must lie entirely
!       within the virtual display, else SMG$_INVARG is returned.
!
!       If the number of units to be scrolled exceeds the size of the rectangle
!       in that direction the text within the rectangle simply "disappears"
!       at the rectangle boundary.  Locations emptied by this operation are
!       set to spaces.
!
!       The cursor position is left at START_ROW, START_COL.
!
! CALLING SEQUENCE:
!
!       ret_status.wlc.v = SMG$SCROLL_DISPLAY_AREA (
!                                       DISPLAY_ID.rl.r
!                                       [,START_ROW.rl.r, START_COL_NUM.rl.r]
!                                       [,HEIGHT.rl.r]
!                                       [,WIDTH.rl.r]
!                                       [,DIRECTION.rl.r]
!                                       [,COUNT.rl.r]
!
! FORMAL PARAMETERS:
!
!       DISPLAY_ID.rl.r Display id of desired display.
!
!       START_ROW.rl.r  Optional.  Row in which scrolling region starts.
!                       Defaults to row 1.
!
!       START_COL.rl.r  Optional.  Column in which scrolling region starts.
!                       Defaults to column 1.
!
!       HEIGHT.rl.r     Optional.  Height of scrolling region.  Defaults to
!                       the height of the scrolling region in the virtual
!                       display.
!
!       WIDTH.rl.r      Optional.  Width of scrolling region.  Defaults to
!                       the width of the scrolling region in the virtual
!                       display.
```

```
!
!       DIRECTION.rl.r  Optional.  Direction in which to scroll.  Defaults
!                       to SMG$M_UP.  Choices are:       SMG$M_UP
!                                                        SMG$M_DOWN
!                                                        SMG$M_RIGHT
!                                                        SMG$M_LEFT
!
!       COUNT.rl.r      Optional.  Number of positions to scroll.  Defaults
!                       to 1 line.
!
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! COMPLETION STATUS:
!
!       SS$_NORMAL      Normal successful completion
!       SMG$_WRONUMARG  Wrong number arguments.
!       SMG$_INVDIS_ID  Invalid display Id.
!
! SIDE EFFECTS:
!
!       NONE
!--
    BEGIN
    LOCAL
        DCB : REF $DCB_DECL,            ! Addr. of display control block
        LOCAL_ROW,                      ! working start row
        LOCAL_COL,                      ! working start column
        LOCAL_HEIGHT,                   ! working height
        LOCAL_WIDTH,                    ! working width
        LOCAL_DIR,                      ! working direction
        LOCAL_COUNT,                    ! working count
        STATUS;                         ! status of called routine

    BUILTIN
        NULLPARAMETER;

    $SMG$VALIDATE_ARGCOUNT (1,7);

    $SMG$GET_DCB ( .DISPLAY_ID, DCB);   ! Get address of display control
                                        ! block

!+
! Validate arguments and fill in local variables.
!-

    IF NOT NULLPARAMETER (START_ROW) AND
       NOT NULLPARAMETER (START_COL)
    THEN
        BEGIN                   ! use caller's row, col
        $SMG$VALIDATE_ROW_COL (..START_ROW, ..START_COL);
        LOCAL_ROW = ..START_ROW;
```

```
3               LOCAL_COL = ..START_COL;
3               END
2           ELSE
3               BEGIN                   ! use default row, col
3               LOCAL_ROW = .DCB [DCB_W_TOP_OF_SCRREG];
3               LOCAL_COL = 1;
2               END;
2
2           LOCAL_HEIGHT = .DCB [DCB_W_BOTTOM_OF_SCRREG] -
2                          .LOCAL_ROW + 1; ! init to default
2
2           IF NOT NULLPARAMETER (HEIGHT)
2           THEN
3               BEGIN                   ! check caller's heigth
3               IF ..HEIGHT + .LOCAL_ROW - 1 LSS .DCB [DCB_W_ROW_START] OR
3                  ..HEIGHT + .LOCAL_ROW - 1 GTR .DCB [DCB_W_NO_ROWS]
3               THEN
4                   RETURN (SMG$_INVARG)
3               ELSE
3                   LOCAL_HEIGHT = ..HEIGHT;
2               END;
2
2           LOCAL_WIDTH = .DCB [DCB_W_NO_COLS]; ! init to default
2
2           IF NOT NULLPARAMETER (WIDTH)
2           THEN
3               BEGIN                   ! check caller's width
3               IF ..WIDTH + .LOCAL_COL -1 LSS .DCB [DCB_W_COL_START] OR
3                  ..WIDTH + .LOCAL_COL -1 GTR .DCB [DCB_W_NO_COLS]
3               THEN
4                   RETURN (SMG$_INVARG)
3               ELSE
3                   LOCAL_WIDTH = ..WIDTH;
2               END;
2
2           LOCAL_DIR = SMG$M_UP;
2
2           IF NOT NULLPARAMETER (DIRECTION)
2           THEN
3               BEGIN                   ! check caller's direction
3               CASE ..DIRECTION FROM SMG$M_UP TO SMG$M_LEFT OF
3               SET
3                   [SMG$M_UP, SMG$M_DOWN, SMG$M_RIGHT, SMG$M_LEFT]:
3                       LOCAL_DIR = ..DIRECTION;
3
3                   [INRANGE,OUTRANGE]:
3                       RETURN (SMG$_INVARG);
3               TES;
2               END;
2
2           LOCAL_COUNT = 1;                ! init to default
2
2           IF NOT NULLPARAMETER (COUNT)
2           THEN
3               BEGIN                   ! check caller's count
3               IF ..COUNT GTR 0
3               THEN
```

```
; 3505          3748  3            LOCAL_COUNT = ..COUNT
; 3506          3749  3         ELSE
; 3507          3750  3            RETURN (SMG$_INVARG);
; 3508          3751  2         END;
; 3509          3752  2
; 3510          3753  2  !+
; 3511          3754  2  ! Now all the arguments are set up.  Call an internal routine to do
; 3512          3755  2  ! the real work.
; 3513          3756  2  !-
; 3514          3757  2
; 3515          3758  2      STATUS = SMG$$SCROLL_AREA (.DCB,
; 3516          3759  2                                 .LOCAL_ROW, .LOCAL_COL,
; 3517          3760  2                                 .LOCAL_HEIGHT, .LOCAL_WIDTH,
; 3518          3761  2                                 .LOCAL_DIR, .LOCAL_COUNT);
; 3519          3762  2      IF NOT (.STATUS) THEN RETURN (.STATUS);
; 3520          3763  2
; 3521          3764  2  !+
; 3522          3765  2  ! The internal routine did not reset the cursor position.  Do that here.
; 3523          3766  2  !-
; 3524          3767  2
; 3525          3768  2      DCB [DCB_W_CURSOR_ROW] = .LOCAL_ROW;
; 3526          3769  2      DCB [DCB_W_CURSOR_COL] = .LOCAL_COL;
; 3527          3770  2
; 3528          3771  2  !+
; 3529          3772  2  ! Turn off scrolling since we have moved the cursor.
; 3530          3773  2  !-
; 3531          3774  2
; 3532          3775  2      DCB [DCB_V_FULL] = 0;
; 3533          3776  2
; 3534          3777  2  !+
; 3535          3778  2  ! See if this change should be reflected on the screen immediately.
; 3536          3779  2  !-
; 3537          3780  2
; 3538          3781  3      RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
; 3539          3782  2                                  SMG$C_SCROLL_DISPLAY_AREA));
; 3540          3783  1      END;                          ! End of routine SMG$SCROLL_DISPLAY_AREA


                                 01FC 00000          .ENTRY  SMG$SCROLL_DISPLAY_AREA, Save R2,R3,R4,R5,- ; 3578
                                                             R6,R7,R8
                   58 00000000G   00   9E 00002      MOVAB   SMG$_INVARG, R8                            ;
          50       6C             01   83 00009      SUBB3   #1, (AP), DIFF                             ; 3676
          06                      50   91 0000D      CMPB    DIFF, #6                                   ;
          08                           1B 00010      BLEQU   1$                                         ;
                   50 00000000G   8F   D0 00012      MOVL    #SMG$_WRONUMARG, R0                        ;
                                       04 00019      RET                                                ;
                   50        04   BC   D0 0001A 1$:  MOVL    @DISPLAY_ID, R0                            ; 3678
              04   BC        38   A0   D1 0001E      CMPL    56(R0), @DISPLAY_ID                        ;
                                  06   12 00023      BNEQ    2$                                         ;
                   11        44   A0   91 00025      CMPB    68(R0), #17                                ;
                                  08   13 00029      BEQL    3$                                         ;
                   50 00000000G   8F   D0 0002B 2$:  MOVL    #SMG$_INVDIS_ID, R0                        ;
                                       04 00032      RET                                                ;
                   52        04   BC   D0 00033 3$:  MOVL    @DISPLAY_ID, DCB                           ;
```

```
                        02         6C  91 00037        CMPB    (AP), #2                       ; 3685
                                   43  1F 0003A        BLSSU   8$                             ;
                              08   AC  D5 0003C        TSTL    8(AP)                          ;
                                   3E  13 0003F        BEQL    8$                             ;
                        03         6C  91 00041        CMPB    (AP), #3                       ; 3686
                                   39  1F 00044        BLSSU   8$                             ;
                              0C   AC  D5 00046        TSTL    12(AP)                         ;
                                   34  13 00049        BEQL    8$                             ;
                        51    08   BC  D0 0004B        MOVL    @START_ROW, R1                 ; 3689
                                   08  15 0004F        BLEQ    4$                             ;
  51       02      A2   10         00  ED 00051        CMPZV   #0, #16, 2(DCB), R1            ;
                                   08  18 00057        BGEQ    5$                             ;
                        50 00000000G  8F  D0 00059 4$:  MOVL    #SMG$_INVROW, R0               ;
                                   04 00060            RET                                    ;
                        50    0C   BC  D0 00061 5$:    MOVL    @START_COL, R0                 ;
                                   08  15 00065        BLEQ    6$                             ;
  50       06      A2   10         00  ED 00067        CMPZV   #0, #16, 6(DCB), R0            ;
                                   08  18 0006D        BGEQ    7$                             ;
                        50 00000000G  8F  D0 0006F 6$:  MOVL    #SMG$_INVCOL, R0               ;
                                   04 00076            RET                                    ;
                        56         51  D0 00077 7$:    MOVL    R1, LOCAL_ROW                  ; 3690
                        57         50  D0 0007A        MOVL    R0, LOCAL_COL                  ; 3691
                                   07  11 0007D        BRB     9$                             ; 3685
                        56    48   A2  3C 0007F 8$:    MOVZWL  72(DCB), LOCAL_ROW             ; 3695
                        57         01  D0 00083        MOVL    #1, LOCAL_COL                  ; 3696
                        50    4A   A2  3C 00086 9$:    MOVZWL  74(DCB), R0                    ; 3700
                        50         56  C2 0008A        SUBL2   LOCAL_ROW, R0                  ;
                                   50  D6 0008D        INCL    LOCAL_HEIGHT                   ;
                        04         6C  91 0008F        CMPB    (AP), #4                       ; 3702
                                   20  1F 00092        BLSSU   10$                            ;
                              10   AC  D5 00094        TSTL    16(AP)                         ;
                                   1B  13 00097        BEQL    10$                            ;
                        51    10   BC  D0 00099        MOVL    @HEIGHT, R1                    ; 3705
                        53    FF A641  9E 0009D        MOVAB   -1(LOCAL_ROW)[R1], R3          ;
  53               62   10         00  ED 000A2        CMPZV   #0, #16, (DCB), R3             ;
                                   32  14 000A7        BGTR    11$                            ;
  53       02      A2   10         00  ED 000A9        CMPZV   #0, #16, 2(DCB), R3            ; 3706
                                   2A  19 000AF        BLSS    11$                            ;
                        50         51  D0 000B1        MOVL    R1, LOCAL_HEIGHT               ; 3710
                        55    06   A2  3C 000B4 10$:   MOVZWL  6(DCB), LOCAL_WIDTH            ; 3713
                        05         6C  91 000B8        CMPB    (AP), #5                       ; 3715
                                   28  1F 000BB        BLSSU   13$                            ;
                              14   AC  D5 000BD        TSTL    20(AP)                         ;
                                   23  13 000C0        BEQL    13$                            ;
                        51    14   BC  D0 000C2        MOVL    @WIDTH, R1                     ; 3718
                        53    FF A741  9E 000C6        MOVAB   -1(LOCAL_COL)[R1], R3          ;
  53       04      A2   10         00  ED 000CB        CMPZV   #0, #16, 4(DCB), R3            ;
                                   08  14 000D1        BGTR    11$                            ;
  53       06      A2   10         00  ED 000D3        CMPZV   #0, #16, 6(DCB), R3            ; 3719
                                   07  18 000D9        BGEQ    12$                            ;
                        53         68  9E 000DB 11$:   MOVAB   SMG$_INVARG, R3                ; 3721
                        50         53  D0 000DE        MOVL    R3, R0                         ;
                                   04 000E1            RET                                    ;
                        55         51  D0 000E2 12$:   MOVL    R1, LOCAL_WIDTH                ; 3723
                        54         01  D0 000E5 13$:   MOVL    #1, LOCAL_DIR                  ; 3726
                        06         6C  91 000E8        CMPB    (AP), #6                       ; 3728
                                   20  1F 000EB        BLSSU   16$                            ;
```

```
                                         18   AC  D5 000ED         TSTL    24(AP)                               ;
                                              1B  13 000F0         BEQL    16$                                  ;
                     07            01    18   BC  CF 000F2         CASEL   @DIRECTION, #1, #7                   ; 3731
      0012         002E          0012       0012     000F7 14$:    .WORD   15$-14$,-                            ;
      0012         002E          002E       002E     000FF                 15$-14$,-                            ;
                                                                           17$-14$,-                            ;
                                                                           15$-14$,-                            ;
                                                                           17$-14$,-                            ;
                                                                           17$-14$,-                            ;
                                                                           17$-14$,-                            ;
                                                                           15$-14$                              ;
                                              1C  11 00107         BRB     17$                                  ; 3737
                                   54    18   BC  D0 00109 15$:    MOVL    @DIRECTION, LOCAL_DIR                ; 3734
                                   53         01  D0 0010D 16$:    MOVL    #1, LOCAL_COUNT                      ; 3741
                                   07         6C  91 00110         CMPB    (AP), #7                             ; 3743
                                              17  1F 00113         BLSSU   18$                                  ;
                                         1C   AC  D5 00115         TSTL    28(AP)                               ;
                                              12  13 00118         BEQL    18$                                  ;
                                         1C   BC  D5 0011A         TSTL    @COUNT                               ; 3746
                                              06  15 0011D         BLEQ    17$                                  ;
                                   53    1C   BC  D0 0011F         MOVL    @COUNT, LOCAL_COUNT                  ; 3748
                                              07  11 00123         BRB     18$                                  ;
                                   51         68  9E 00125 17$:    MOVAB   SMG$_INVARG, R1                      ; 3750
                                   50         51  D0 00128         MOVL    R1, R0                               ;
                                              04     0012B         RET                                          ;
                                              53  DD 0012C 18$:    PUSHL   LOCAL_COUNT                          ; 3761
                                              54  DD 0012E         PUSHL   LOCAL_DIR                            ;
                                              21  BB 00130         PUSHR   #^M<R0,R5>                           ; 3760
                                       00C4   8F  BB 00132         PUSHR   #^M<R2,R6,R7>                        ; 3758
                      00000000G    00         07  FB 00136         CALLS   #7, SMG$$SCROLL_AREA                 ;
                                   17         50  E9 0013D         BLBC    STATUS, 19$                          ; 3762
                           28      A2         56  B0 00140         MOVW    LOCAL_ROW, 40(DCB)                   ; 3768
                           2A      A2         57  B0 00144         MOVW    LOCAL_COL, 42(DCB)                   ; 3769
                           34      A2         01  8A 00148         BICB2   #1, 52(DCB)                          ; 3775
                                              1A  DD 0014C         PUSHL   #26                                  ; 3781
                                              52  DD 0014E         PUSHL   DCB                                  ;
                      00000000G    00         02  FB 00150         CALLS   #2, SMG$$CHECK_FOR_OUTPUT_DCB        ;
                                              04     00157 19$:    RET                                          ; 3783

; Routine Size:  344 bytes,    Routine Base:  _SMG$CODE + 13A4


; 3541         3784  1 !<BLF/PAGE>
```

```
1 %SBTTL 'SMG$SET_CURSOR_ABS - Set absolute cursor position'
1 GLOBAL ROUTINE SMG$SET_CURSOR_ABS (
1                                       DISPLAY_ID,
1                                       ROW_NUM,
1                                       COL_NUM
1                                       ) =
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       Position cursor to absolute row and column specified.
1 !       Return SMG$_INVROW or SMG$_INVCOL if either position is outside
1 !       of the virtual display area.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = SMG$SET_CURSOR_ABS (
1 !                                       DISPLAY_ID.rl.r
1 !                                       [,ROW_NUM.rl.r]
1 !                                       [,COL_NUM.rl.r])
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       DISPLAY_ID.rl.r Display id of desired display.
1 !
1 !       ROW_NUM.rl.r    Row position.  If not supplied, row number is
1 !                       not altered from current position.
1 !
1 !       COL_NUM.rl.r    Column position.  If not supplied, column number
1 !                       is not altered from current position.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !       SS$_NORMAL      Normal successful completion
1 !       SMG$_INVROW     Invalid row number
1 !       SMG$_INVCOL     Invalid column number
1 !       SMG$_WRONUMARG  Wrong number arguments.
1 !       SMG$_INVDIS_ID  Invalid display Id.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !--
2   BEGIN
2   BUILTIN
2       NULLPARAMETER;
2
2   LOCAL
2       STATUS,                         ! Status of subroutine calls
2       ROW,                            ! Working row number
```

```
2          COL,                                ! Working column number
2          DCB : REF $DCB_DECL;                ! Addr. of display control block
2
2      $SMG$VALIDATE_ARGCOUNT (1,3);
2
2      $SMG$GET_DCB ( .DISPLAY_ID, DCB);       ! Get address of display control
2                                              ! block
2
2
3      ROW = (IF NOT NULLPARAMETER (ROW_NUM) THEN ..ROW_NUM
2                                             ELSE .DCB [DCB_W_CURSOR_ROW]);
2
3      COL = (IF NOT NULLPARAMETER (COL_NUM) THEN ..COL_NUM
2                                             ELSE .DCB [DCB_W_CURSOR_COL]);
2
2   !+
2   ! Make sure resulting position is still within the virtual display.
2   !-
2   !+
2   ! Validity check row and column specified
2   !-
2      $SMG$VALIDATE_ROW_COL ( .ROW, .COL);
2
2   !+
2   ! Ok, set them.
2   !-
2      DCB [DCB_W_CURSOR_ROW] = .ROW;
2      DCB [DCB_W_CURSOR_COL] = .COL;
2
2   !+
2   ! We have altered the cursor position so obviously sequential output
2   ! has been interupted.  Turn off the scrolling bit.
2   !-
2
2      DCB [DCB_V_FULL] = 0;
2
3      RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
2                                        SMG$C_SET_CURSOR_ABS));
1      END;                          ! End of routine SMG$SET_CURSOR_ABS


                                   0004 00000       .ENTRY  SMG$SET_CURSOR_ABS, Save R2          ; 3786
          50        6C          01  83 00002        SUBB3   #1, (AP), DIFF                       ; 3845
                    02          50  91 00006        CMPB    DIFF, #2                             ;
                                08  1B 00009        BLEQU   1$                                   ;
                    50 00000000G 8F D0 0000B        MOVL    #SMG$_WRONUMARG, R0                  ;
                                04  00012           RET                                          ;
                    50      04  BC  D0 00013 1$:    MOVL    @DISPLAY_ID, R0                      ; 3847
               04   BC      38  A0  D1 00017        CMPL    56(R0), @DISPLAY_ID                  ;
                                06  12 0001C        BNEQ    2$                                   ;
                    11      44  A0  91 0001E        CMPB    68(R0), #17                          ;
                                08  13 00022        BEQL    3$                                   ;
                    50 00000000G 8F D0 00024 2$:    MOVL    #SMG$_INVDIS_ID, R0                  ;
                                04  0002B           RET                                          ;
```

```
                         50        04  BC  D0 0002C 3$:     MOVL    @DISPLAY_ID, DCB                ;
                         02            6C  91 00030         CMPB    (AP), #2                        ; 3851
                                       0B  1F 00033         BLSSU   4$                              ;
                                   08  AC  D5 00035         TSTL    8(AP)                           ;
                                       06  13 00038         BEQL    4$                              ;
                         52        08  BC  D0 0003A         MOVL    @ROW_NUM, ROW                   ;
                                       04  11 0003E         BRB     5$                              ;
                         52        28  A0  3C 00040 4$:     MOVZWL  40(DCB), ROW                    ; 3852
                         03            6C  91 00044 5$:     CMPB    (AP), #3                        ; 3854
                                       0B  1F 00047         BLSSU   6$                              ;
                                   0C  AC  D5 00049         TSTL    12(AP)                          ;
                                       06  13 0004C         BEQL    6$                              ;
                         51        0C  BC  D0 0004E         MOVL    @COL_NUM, COL                   ;
                                       04  11 00052         BRB     7$                              ;
                         51        2A  A0  3C 00054 6$:     MOVZWL  42(DCB), COL                    ; 3855
                                       52  D5 00058 7$:     TSTL    ROW                             ; 3863
                                       08  15 0005A         BLEQ    8$                              ;
      52      02   A0              10  00  ED 0005C         CMPZV   #0, #16, 2(DCB), ROW            ;
                                       08  18 00062         BGEQ    9$                              ;
                         50 00000000G  8F  D0 00064 8$:     MOVL    #SMG$_INVROW, R0                ;
                                           04 0006B         RET                                     ;
                                       51  D5 0006C 9$:     TSTL    COL                             ;
                                       08  15 0006E         BLEQ    10$                             ;
      51      06   A0              10  00  ED 00070         CMPZV   #0, #16, 6(DCB), COL            ;
                                       08  18 00076         BGEQ    11$                             ;
                         50 00000000G  8F  D0 00078 10$:    MOVL    #SMG$_INVCOL, R0                ;
                                           04 0007F         RET                                     ;
                      28   A0          52  B0 00080 11$:    MOVW    ROW, 40(DCB)                    ; 3868
                      2A   A0          51  B0 00084         MOVW    COL, 42(DCB)                    ; 3869
                      34   A0          01  8A 00088         BICB2   #1, 52(DCB)                     ; 3876
                                       16  DD 0008C         PUSHL   #22                             ; 3878
                                       50  DD 0008E         PUSHL   DCB                             ;
                 00000000G 00          02  FB 00090         CALLS   #2, SMG$$CHECK_FOR_OUTPUT_DCB   ;
                                           04 00097         RET                                     ; 3880

; Routine Size:  152 bytes,    Routine Base:  _SMG$CODE + 14FC


; 3639          3881  1 !<BLF/PAGE>
```

```
%SBTTL 'SMG$SET_CURSOR_REL - Set relative cursor position'
GLOBAL ROUTINE SMG$SET_CURSOR_REL (
                                    DISPLAY_ID,
                                    DELTA_ROW,
                                    DELTA_COLUMN
                                  ) =
!++
! FUNCTIONAL DESCRIPTION:
!
!       Position cursor to relative row and column specified --
!       relative to current position.
!       If adding DELTA_ROW or DELTA_COLUMN takes us outside of the
!       virtual display, return SMG$_INVROW or SMG$_INVCOL respectively.
!
!
! CALLING SEQUENCE:
!
!       ret_status.wlc.v = SMG$SET_CURSOR_REL (
!                                   DISPLAY_ID.rl.r
!                                   [,DELTA_ROW.rl.r]
!                                   [,DELTA_COLUMN.rl.r])
!
! FORMAL PARAMETERS:
!
!       DISPLAY_ID.rl.r         Display id of desired display.
!
!       DELTA_ROW.rl.r          Delta row position (-delta, +delta)
!                               If not supplied, row number is not
!                               altered from current position.
!
!       DELTA_COLUMN.rl.r       Delta column position (-delta, +delta)
!                               If not supplied, column number is not
!                               altered from current position.
!
!
! IMPLICIT INPUTS:
!
!       NONE
!
! IMPLICIT OUTPUTS:
!
!       NONE
!
! COMPLETION STATUS:
!
!       SS$_NORMAL      Normal successful completion
!       SMG$_INVROW     When delta_column is added to current column
!       SMG$_INVCOL     and/or delta_row is added to current row,
!                       resulting row or column is outside virtual
!                       display.
!       SMG$_WRONUMARG  Wrong number arguments.
!
! SIDE EFFECTS:
!
!       NONE
!--
    BEGIN
```

```
        BUILTIN
            NULLPARAMETER;

        LOCAL
            STATUS,                 ! Status of subroutine calls
            D_ROW,                  ! Working delta row
            D_COL,                  ! Working delta column
            DCB : REF $DCB_DECL;    ! Addr. of display control block

        $SMG$VALIDATE_ARGCOUNT (1,3);

        $SMG$GET_DCB ( .DISPLAY_ID, DCB);      ! Get address of display control
                                               ! block


        D_ROW = (IF NOT NULLPARAMETER (DELTA_ROW) THEN ..DELTA_ROW
                                                 ELSE 0);

        D_COL = (IF NOT NULLPARAMETER (DELTA_COLUMN) THEN ..DELTA_COLUMN
                                                    ELSE 0);

    !+
    ! Make sure resulting position is still within the virtual display.
    !-
        IF .DCB [DCB_W_CURSOR_ROW] + .D_ROW LSS 1  OR
           .DCB [DCB_W_CURSOR_ROW] + .D_ROW GTR .DCB [DCB_W_NO_ROWS]
        THEN
            RETURN (SMG$_INVROW);

        IF .DCB [DCB_W_CURSOR_COL] + .D_COL LSS 1  OR
           .DCB [DCB_W_CURSOR_COL] + .D_COL GTR .DCB [DCB_W_NO_COLS]
        THEN
            RETURN (SMG$_INVCOL);

    !+
    ! Ok, set them.
    !-
        DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_CURSOR_ROW] + .D_ROW;
        DCB [DCB_W_CURSOR_COL] = .DCB [DCB_W_CURSOR_COL] + .D_COL;

    !+
    ! Turn off scrolling.
    !-

        DCB [DCB_V_FULL] = 0;

        RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
                                    SMG$C_SET_CURSOR_REL));
        END;                        ! End of routine SMG$SET_CURSOR_REL



                            000C 00000        .ENTRY  SMG$SET_CURSOR_REL, Save R2,R3           ; 3883
    50          6C       01  83 00002         SUBB3   #1, (AP), DIFF                           ; 3948
                02       50  91 00006         CMPB    DIFF, #2                                 ;
```

```
                                08  1B 00009          BLEQU   1$                                          ;
                   50 00000000G 8F  D0 0000B          MOVL    #SMG$_WRONUMARG, R0                         ;
                                04     00012          RET                                                 ;
                   50        04 BC  D0 00013 1$:      MOVL    @DISPLAY_ID, R0                             ; 3950
            04  BC           38 A0  D1 00017          CMPL    56(R0), @DISPLAY_ID                         ;
                                06  12 0001C          BNEQ    2$                                          ;
                   11        44 A0  91 0001E          CMPB    68(R0), #17                                 ;
                                08  13 00022          BEQL    3$                                          ;
                   50 00000000G 8F  D0 00024 2$:      MOVL    #SMG$_INVDIS_ID, R0                         ;
                                04     0002B          RET                                                 ;
                   50        04 BC  D0 0002C 3$:      MOVL    @DISPLAY_ID, DCB                            ;
                   02           6C  91 00030          CMPB    (AP), #2                                    ; 3954
                                0B  1F 00033          BLSSU   4$                                          ;
                             08 AC  D5 00035          TSTL    8(AP)                                       ;
                                06  13 00038          BEQL    4$                                          ;
                   53        08 BC  D0 0003A          MOVL    @DELTA_ROW, D_ROW                           ;
                                02  11 0003E          BRB     5$                                          ;
                                53  D4 00040 4$:      CLRL    D_ROW                                       ;
                   03           6C  91 00042 5$:      CMPB    (AP), #3                                    ; 3957
                                0B  1F 00045          BLSSU   6$                                          ;
                             0C AC  D5 00047          TSTL    12(AP)                                      ;
                                06  13 0004A          BEQL    6$                                          ;
                   52        0C BC  D0 0004C          MOVL    @DELTA_COLUMN, D_COL                        ;
                                02  11 00050          BRB     7$                                          ;
                                52  D4 00052 6$:      CLRL    D_COL                                       ;
                   51        28 A0  3C 00054 7$:      MOVZWL  40(DCB), R1                                 ; 3963
                   51           53  C0 00058          ADDL2   D_ROW, R1                                   ;
                                08  15 0005B          BLEQ    8$                                          ;
  51      02   A0  10           00  ED 0005D          CMPZV   #0, #16, 2(DCB), R1                         ; 3964
                                09  18 00063          BGEQ    9$                                          ;
                   51 00000000G 00  9E 00065 8$:      MOVAB   SMG$_INVROW, R1                             ; 3966
                                18  11 0006C          BRB     11$                                         ;
                   51        2A A0  3C 0006E 9$:      MOVZWL  42(DCB), R1                                 ; 3968
                   51           52  C0 00072          ADDL2   D_COL, R1                                   ;
                                08  15 00075          BLEQ    10$                                         ;
  51      06   A0  10           00  ED 00077          CMPZV   #0, #16, 6(DCB), R1                         ; 3969
                                0B  18 0007D          BGEQ    12$                                         ;
                   51 00000000G 00  9E 0007F 10$:     MOVAB   SMG$_INVCOL, R1                             ; 3971
                   50           51  D0 00086 11$:     MOVL    R1, R0                                      ;
                                04     00089          RET                                                 ;
            28  A0              53  A0 0008A 12$:     ADDW2   D_ROW, 40(DCB)                              ; 3976
            2A  A0              52  A0 0008E          ADDW2   D_COL, 42(DCB)                              ; 3977
            34  A0              01  8A 00092          BICB2   #1, 52(DCB)                                 ; 3983
                                17  DD 00096          PUSHL   #23                                         ; 3985
                                50  DD 00098          PUSHL   DCB                                         ;
         00000000G  00          02  FB 0009A          CALLS   #2, SMG$$CHECK_FOR_OUTPUT_DCB               ;
                                04     000A1          RET                                                 ; 3987

; Routine Size:  162 bytes,    Routine Base:  _SMG$CODE + 1594


; 3747          3988  1 !<BLF/PAGE>
```

```
; 3749          3989  1 END                     ! End of module SMG$DISPLAY_CHANGE
; 3750          3990  1
; 3751          3991  0 ELUDOM


                                                                .EXTRN  LIB$STOP
;                               PSECT SUMMARY
;
;       Name                        Bytes                       Attributes
;
;   _SMG$CODE                          5686  NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)


;                           Library Statistics
;
;                                           -------- Symbols --------        Pages        Processing
;       File                                Total    Loaded   Percent        Mapped       Time
;
;   _$255$DUA28:[SYSLIB]STARLET.L32;1       9776        41         0           581         00:01.0
;   _$255$DUA28:[SMGRTL.OBJ]RTLLIB.L32;1      36         0         0             8         00:00.1
;   _$255$DUA28:[SMGRTL.OBJ]SMGLIB.L32;1     469        31         6            38         00:00.4


;                               COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:SMGDISCHA/OBJ=OBJ$:SMGDISCHA MSRC$:SMGDISCHA/UPDATE=(ENH$:SMGDISCHA
;       )

; Size:          5679 code + 7 data bytes
; Run Time:         01:51.9
; Elapsed Time:     05:20.2
; Lines/CPU Min:     2140
; Lexemes/CPU-Min: 18059
; Memory Used:  354 pages
; Compilation Complete
```

0355 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SMGDATSTR
REQ

SMGLNK
REQ

SMGKQB
SDL

SMGBLDTRM
LIS

SMGDISCHA
LIS

SMGKTH
SDL

SMGMACROS
REQ

SMGPROLOG
REQ

SMGTERM
REQ

SMGTRMSTR
R32

SMGTRMPTR
SDL

SMGSCRMAC
REQ

SMGALLESC
LIS

SMGTPACTL
REQ

SMGSCRTCB
REQ

SMGLIB
REQ

SMGTPALIB
REQ

SMGDEFKEY
LIS

SMGTABDEF
REQ

SMGTRMMAC
REQ

SMGBOOTAB
LIS

0356 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SMGDISDRW
LIS

SMGDISLIN
LIS

SMGDISINP
LIS

SMGDISDHW
LIS