C 12

```
   SSSSSSSSSSSS  MMM         MMM     GGGGGGGGGGGG  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
   SSSSSSSSSSSS  MMM         MMM     GGGGGGGGGGGG  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
   SSSSSSSSSSSS  MMM         MMM     GGGGGGGGGGGG  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
```

```
_$2

Val
---
001
001
001
001
001
001
001
001
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
```

```
**FILE**ID**SMGDISDHW                                M  4                                                                SMG
                                                                                                                          1-0

  SSSSSSSS  MM      MM    GGGGGGGG  DDDDDDDD      IIIIII      SSSSSSSS  DDDDDDDD    HH      HH  WW      WW
  SSSSSSSS  MM      MM    GGGGGGGG  DDDDDDDD      IIIIII      SSSSSSSS  DDDDDDDD    HH      HH  WW      WW
SS          MMMM  MMMM  GG          DD      DD      II      SS          DD      DD  HH      HH  WW      WW
SS          MMMM  MMMM  GG          DD      DD      II      SS          DD      DD  HH      HH  WW      WW
SS          MM  MM  MM  GG          DD      DD      II      SS          DD      DD  HH      HH  WW      WW
SS          MM  MM  MM  GG          DD      DD      II      SS          DD      DD  HH      HH  WW      WW
  SSSSSS    MM      MM  GG          DD      DD      II        SSSSSS    DD      DD  HHHHHHHHHH  WW      WW
  SSSSSS    MM      MM  GG          DD      DD      II        SSSSSS    DD      DD  HHHHHHHHHH  WW      WW
        SS  MM      MM  GG  GGGGGG  DD      DD      II              SS  DD      DD  HH      HH  WW  WW  WW
        SS  MM      MM  GG  GGGGGG  DD      DD      II              SS  DD      DD  HH      HH  WW  WW  WW
        SS  MM      MM  GG      GG  DD      DD      II              SS  DD      DD  HH      HH  WWWW  WWWW       ....
        SS  MM      MM  GG      GG  DD      DD      II              SS  DD      DD  HH      HH  WWWW  WWWW       ....
SSSSSSSS    MM      MM    GGGGGG    DDDDDDDD      IIIIII    SSSSSSSS    DDDDDDDD    HH      HH  WW      WW       ....
SSSSSSSS    MM      MM    GGGGGG    DDDDDDDD      IIIIII    SSSSSSSS    DDDDDDDD    HH      HH  WW      WW       ....


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 MODULE SMG$DISPLAY_DHDW ( %TITLE 'Display double high/double wide chars'
0                 IDENT = '1-004'         ! File: SMGDISDHW.B32 Edit: STAN1004
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 !
1 !       This module contains routines to write double high/double wide
1 !       characters to a virtual display.
1 !
1 ! ENVIRONMENT:  User mode - AST reentrant
1 !
1 ! AUTHOR: P. Levesque, CREATION DATE: 20-Jul-1983
1 !
1 ! MODIFIED BY:
1 !
1 ! 1-001 - Original.  PLL 20-Jul-1983
1 ! 1-002 - More tweaks to cursor position.  PLL 31-Aug-1983
1 ! 1-003 - Check the length of the string before moving it into the
1 !         DCB buffer.  PLL 7-Oct-1983
1 ! 1-004 - Don't say just one line has changed in DHW. STAN 7-Jul-1984.
1 !--
1
```

```
        0051  1 %SBTTL 'Declarations'
        0052  1 !
        0053  1 ! SWITCHES:
        0054  1 !
        0055  1
        0056  1 REQUIRE 'RTLIN:SMGPROLOG';
        0134  1
        0135  1 REQUIRE 'RTLIN:STRLNK';          ! JSB linkage for string routines
        0320  1 !
        0321  1 ! LINKAGES:
        0322  1 !
        0323  1 !       NONE
        0324  1 !
        0325  1 ! TABLE OF CONTENTS:
        0326  1 !
        0327  1
        0328  1 FORWARD ROUTINE
        0329  1         SMG$PUT_CHARS_WIDE,             ! Write dbl wide chars
        0330  1         SMG$PUT_CHARS_HIGHWIDE,         ! Write dbl high dbl wide chars
        0331  1         SMG$PUT_LINE_WIDE;              ! Write dbl wide w/advance
        0332  1
        0333  1 !
        0334  1 ! INCLUDE FILES:
        0335  1 !
        0336  1
        0337  1
        0338  1 !
        0339  1 ! MACROS:
        0340  1 !
        0341  1
        0342  1 !+
        0343  1 ! The following macro determines whether scrolling up, down, or neither
        0344  1 ! should occur.
        0345  1 !-
        0346  1
      M 0347  1 MACRO $SMG$SET_SCROLLING (SWITCH) =
      M 0348  1     BEGIN
      M 0349  1     SWITCH = 0;
      M 0350  1     IF .DCB [DCB_V_FULL] NEQ 0
      M 0351  1     THEN
      M 0352  1         BEGIN
      M 0353  1         IF .DCB [DCB_W_CURSOR_ROW] EQL .DCB [DCB_W_BOTTOM_OF_SCRREG]
      M 0354  1         THEN
      M 0355  1             SWITCH = 1  ! scroll up
      M 0356  1         ELSE
      M 0357  1             IF .DCB [DCB_W_CURSOR_ROW] EQL .DCB [DCB_W_TOP_OF_SCRREG]
      M 0358  1             THEN
      M 0359  1                 SWITCH = 2; ! scroll down
      M 0360  1         END;
        0361  1     END;%;
        0362  1
        0363  1 !
        0364  1 ! EQUATED SYMBOLS:
        0365  1 !
        0366  1 !       NONE
        0367  1 !
        0368  1 ! FIELDS:
```

```
1 !
1 !       NONE
1 !
1 ! PSECTS:
1 !
1
1 !
1 ! OWN STORAGE:
1 !
1 !       NONE
1 !
1 ! EXTERNAL REFERENCES:
1 !
1 EXTERNAL ROUTINE
1         LIB$ANALYZE_SDESC_R2 : LIB$ANALYZE_SDESC_JSB_LINK,
1         SMG$$SCROLL_AREA,                ! scroll virtual display area
1         SMG$$CHECK_FOR_OUTPUT_DCB,       ! check if time to repaint display
1         SMG$$PUT_TEXT_TO_BUFFER;         ! put text in DCB buffer
1 EXTERNAL LITERAL
1         SMG$_INVDIS_ID,                  ! Invalid display id
1         SMG$_INVARG,                     ! Invalid argument
1         SMG$_INVCOL,                     ! Invalid column number
1         SMG$_INVROW,                     ! Invalid row number
1         LIB$_INVSTRDES,                  ! Invalid string descriptor
1         SMG$_WRONUMARG;                  ! Wrong number of arguments
1
```

SM
1-

```
1 %SBTTL 'SMG$PUT_CHARS_WIDE - Write wide characters'
1 GLOBAL ROUTINE SMG$PUT_CHARS_WIDE (
1                                               DISPLAY_ID,
1                                               TEXT : REF BLOCK [,BYTE],
1                                               LINE_NO,
1                                               COL_NO,
1                                               RENDITION_SET,
1                                               RENDITION_COMPLEMENT,
1                                               CHAR_SET
1       ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine writes double wide characters to a virtual
1 !       display.  The line can not contain a mixture of single
1 !       wide and double wide characters; if the line previously
1 !       contained single wide, then the entire line will be re-
1 !       written, otherwise only the specified text is written.
1 !
1 !       The internal cursor position is left at the character
1 !       position following the text written.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = SMG$PUT_CHARS_WIDE (DISPLAY_ID.rl.r,
1 !                                               TEXT.rt.dx,
1 !                                               [,LINE_NO.rl.r, COL_NO.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 virtual display
1 !
1 !       TEXT.rt.dx              Address of descriptor of output string
1 !
1 !       LINE_NO.rl.r            O tional.  Address of line number at which
1 !                               to start output.  If omitted (=0), the
1 !                               current line number is used.
1 !
1 !       COL_NO.rl.r             Optional.  Address of column number at which
1 !                               to start output.  If omitted (=0), the
1 !                               current column number is used.
1 !
1 !       RENDITION_SET.rl.r      Optional.  Each 1 bit in this parameter
1 !                               causes the corresponding attribute to be
1 !                               set in the display.  (See below for list
1 !                               of settable attributes.)
1 !
1 !       RENDITION_COMPLEMENT.rl.r Optional.  Fach 1 bit attribute in this
1 !                                 parameter causes the corresponding attribute
1 !                                 to be complemented in the display.  (See
1 !                                 below for list of complementable 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 !
1 !        CHAR_SET.rl.r          Optional.  Character set to use.  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$_INVCOL      Invalid column number
1 !        SMG$_INVROW      Invalid row number
1 !        LIB$_INVSTRDES   Invalid string descriptor
1 !        SMG$_WRONUMARG   Wrong number of arguments
1 !
1 ! SIDE EFFECTS:
1 !
1 !        NONE
1 !
1 !--
1
2    BEGIN
```

```
;  250          0509  2         BUILTIN
;  251          0510  2             NULLPARAMETER;
;  252          0511  2
;  253          0512  2         LOCAL
;  254          0513  2             DCB : REF BLOCK [,BYTE],            ! address of virtual display
;  255          0514  2                                                 ! control block
;  256          0515  2             ROW,                                ! working row
;  257          0516  2             COL,                                ! working column
;  258          0517  2             REND_CODE,                          ! rendition code to use
;  259          0518  2             STR_LEN : INITIAL (0),              ! length of text string
;  260          0519  2             STR_ADDR,                           ! address of text string,
;  261          0520  2             STATUS;
;  262          0521  2
;  263          0522  2         LITERAL
;  264          0523  2             K_LINE_ARG = 3,
;  265          0524  2             K_COL_ARG = 4,
;  266          0525  2             K_SET_ARG = 5,
;  267          0526  2             K_COMP_ARG = 6,
;  268          0527  2             K_CHAR_ARG = 7;
;  269          0528  2
;  270          0529  2         $SMG$GET_DCB (.DISPLAY_ID, DCB);    ! get addr of virtual display
;  271          0530  2                                             ! control block
;  272          0531  2
;  273          0532  2         $SMG$VALIDATE_ARGCOUNT (2, 7);
;  274          0533  2
;  275          0534  2     !+
;  276          0535  2     ! Get the length and address of the text string.
;  277          0536  2     !-
;  278          0537  2
;  279          0538  3         IF NOT (STATUS = LIB$ANALYZE_SDESC_R2 (.TEXT;
;  280          0539  3                                                STR_LEN,
;  281          0540  3                                                STR_ADDR))
;  282          0541  2         THEN
;  283          0542  2             RETURN (.STATUS);
;  284          0543  2
;  285          0544  2     !+
;  286          0545  2     ! Check for optional arguments.  Set local variables to caller's
;  287          0546  2     ! values, when available, and defaults when arguments omitted.
;  288          0547  2     !-
;  289          0548  2
;  290          0549  2         IF NOT NULLPARAMETER (K_LINE_ARG) AND
;  291          0550  2            NOT NULLPARAMETER (K_COL_ARG)
;  292          0551  2         THEN
;  293          0552  3             BEGIN
;  294          0553  3             ROW = ..LINE_NO;
;  295          0554  3             COL = ..COL_NO;
;  296          0555  3             $SMG$VALIDATE_ROW_COL (.ROW, .COL);
;  297          0556  3                                         ! verify row & col within display
;  298          0557  3             END
;  299          0558  2         ELSE
;  300          0559  3             BEGIN
;  301          0560  3             ROW = .DCB [DCB_W_CURSOR_ROW];
;  302          0561  3             COL = .DCB [DCB_W_CURSOR_COL];
;  303          0562  2             END;
;  304          0563  2
;  305          0564  2         $SMG$SET_REND_CODE (K_SET_ARG, K_COMP_ARG);
;  306          0565  2                                         ! macro to use caller's args if present
```

```
2
2      IF NOT NULLPARAMETER (K_CHAR_ARG)
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$C_ALT_GRAPHICS]:
3                  ;
3
3          [INRANGE, OUTRANGE]:
3                  RETURN (SMG$_INVARG);
3
3          TES;
2          END;
2
2  !+
2  ! Double wide characters occupy two positions instead of one on the
2  ! screen.  However, for mapping purposes we store the text only half
2  ! way over in the text buffer.
2  !-
2
2      COL = (.COL + 1)/2;          ! col in half for dbl wide
2
2  !+
2  ! Set the double wide characteristic in the DCB.
2  !-
2
3      BEGIN
3      BIND
3          DCB_LCV = .DCB [DCB_A_LINE_CHAR];
3      MAP
3          DCB_LCV : VECTOR [,BYTE];
3
3      IF .DCB_LCV [.ROW] NEQ LINE_K_WIDE  ! previously single wide
3      THEN                                ! or double high
4          BEGIN
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          DCB_LCV [.ROW] = LINE_K_WIDE;   ! set this row to dbl wide
3          END;
3
3      DCB_LCV [0] = 1;                    ! mark that there are wide or
3                                          ! dbl high/wide chars in display
2      END;
2
2  !+
2  ! All local variables are set up.  Call routine to put text into
2  ! the display buffer.
2  !-
2
3      BEGIN
3      LOCAL
3          PRINT_LEN;
```

```
3      $SMG$FIND_PRINT_LENGTH (STR_LEN, .STR_ADDR, PRINT_LEN);
3                                      ! don't count non-printable chars
3      !+
3      ! SMG$$PUT_TEXT_TO_BUFFER doesn't realize that wide characters
3      ! occupy 2 spaces so it won't recognize overflow.  Make sure
3      ! we don't try to put more chars in buffer than will fit on
3      ! this line.
3      !-
4      IF .PRINT_LEN GTR ((.DCB [DCB_W_NO_COLS] - 1)/2)
3      THEN
3          PRINT_LEN = (.DCB [DCB_W_NO_COLS] - 1)/2;
3
3      DCB [DCB_W_CURSOR_ROW] = .ROW;
3      DCB [DCB_W_CURSOR_COL] = .COL;
3                                      ! set position for put_text
4      IF NOT (STATUS = SMG$$PUT_TEXT_TO_BUFFER (.DCB,
4                                              .REND_CODE,
4                                              .PRINT_LEN,
4                                              .STR_ADDR,
4                                              IF NOT NULLPARAMETER (K_CHAR_ARG)
4                                              THEN ..CHAR_SET
4                                              ELSE SMG$C_ASCII))
3      THEN
3          RETURN (.STATUS);
3
3  !+
3  ! Correct the cursor position.  We stored our text half way over in the
3  ! buffer, but the screen cursor position should be calculated based on
3  ! the actual starting column specified by the caller.  Also take into
3  ! account that characters occupy 2 positions.
3  !-
3
3      DCB [DCB_W_CURSOR_COL] = (2 * .COL) + (2 * .PRINT_LEN) - 1;
2      END;
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$C_PUT_CHARS,
2                                         .ROW));
2
1      END;                               ! End of routine SMG$PUT_CHARS_WIDE


                                              .TITLE  SMG$DISPLAY_DHDW Display double high/double wid
;                                                                     e chars
                                              .IDENT  \1-004\

                                              .EXTRN  LIB$ANALYZE_SDESC_R2
                                              .EXTRN  SMG$$SCROLL_AREA
                                              .EXTRN  SMG$$CHECK_FOR_OUTPUT_DCB
                                              .EXTRN  SMG$$PUT_TEXT_TO_BUFFER
                                              .EXTRN  SMG$_INVDIS_ID, SMG$_INVARG
                                              .EXTRN  SMG$_INVCOL, SMG$_INVROW
                                              .EXTRN  LIB$_INVSTRDES, SMG$_WRONUMARG
```

```
                                                          .EXTRN   SMG$_FATERRLIB, CHAR_TABLE

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

                                  0FFC 00000              .ENTRY   SMG$PUT_CHARS_WIDE, Save R2,R3,R4,R5,R6,R7,-; 0396
                                                                   R8,R9,R10,R11                               :
                     5E        0C  C2 00002               SUBL2    #12, SP                                     :
                               7E  D4 00005               CLRL     STR_LEN                                     : 0508
                     50    04  BC  D0 00007               MOVL     @DISPLAY_ID, R0                             : 0529
                04   BC    38  A0  D1 0000B               CMPL     56(R0), @DISPLAY_ID                         :
                               06  12 00010               BNEQ     1$                                          :
                     11    44  A0  91 00012               CMPB     68(R0), #17                                 :
                               08  13 00016               BEQL     2$                                          :
                     50 00000000G  8F  D0 00018 1$:       MOVL     #SMG$_INVDIS_ID, R0                         :
                               04  0001F                  RET                                                  :
                     58    04  BC  D0 00020 2$:           MOVL     @DISPLAY_ID, DCB                            :
          50                   6C  02  83 00024           SUBB3    #2, (AP), DIFF                              : 0532
                               05  50  91 00028           CMPB     DIFF, #5                                    :
                               08  1B 0002B               BLEQU    3$                                          :
                     50 00000000G  8F  D0 0002D           MOVL     #SMG$_WRONUMARG, R0                         :
                               04  00034                  RET                                                  :
                     50    08  AC  D0 00035 3$:           MOVL     TEXT, R0                                    : 0538
                        00000000G  00  16 00039           JSB      LIB$ANALYZE_SDESC_R2                        :
                08   AE            50  D0 0003F           MOVL     R0, STATUS                                  :
                     6E            51  D0 00043           MOVL     R1, (SP)                                    :
                0C   AE            52  D0 00046           MOVL     R2, 12(SP)                                  :
                     03    08  AE  E8 0004A               BLBS     STATUS, 4$                                  :
                             018A  31 0004E               BRW      24$                                         :
                     03        6C  91 00051 4$:           CMPB     (AP), #3                                    : 0549
                               3F  1F 00054               BLSSU    8$                                          :
                           0C  AC  D5 00056               TSTL     12(AP)                                      :
                               3A  13 00059               BEQL     8$                                          :
                     04        6C  91 0005B               CMPB     (AP), #4                                    : 0550
                               35  1F 0005E               BLSSU    8$                                          :
                           10  AC  D5 00060               TSTL     16(AP)                                      :
                               30  13 00063               BEQL     8$                                          :
                     57    0C  BC  D0 00065               MOVL     @LINE_NO, ROW                               : 0553
                     56    10  BC  D0 00069               MOVL     @COL_NO, COL                                : 0554
                               57  D5 0006D               TSTL     ROW                                         : 0555
                               08  15 0006F               BLEQ     5$                                          :
57     02   A8       10            00  ED 00071           CMPZV    #0, #16, 2(DCB), ROW                        :
                               08  18 00077               BGEQ     6$                                          :
                     50 00000000G  8F  D0 00079 5$:       MOVL     #SMG$_INVROW, R0                            :
                               04  00080                  RET                                                  :
                               56  D5 00081 6$:           TSTL     COL                                         :
                               08  15 00083               BLEQ     7$                                          :
56     06   A8       10            00  ED 00085           CMPZV    #0, #16, 6(DCB), COL                        :
                               10  18 0008B               BGEQ     9$                                          :
                     50 00000000G  8F  D0 0008D 7$:       MOVL     #SMG$_INVCOL, R0                            :
                               04  00094                  RET                                                  :
                     57    28  A8  3C 00095 8$:           MOVZWL   40(DCB), ROW                                : 0560
                     56    2A  A8  3C 00099               MOVZWL   42(DCB), COL                                : 0561
                04   AE    2E  A8  9A 0009D 9$:           MOVZBL   46(DCB), REND_CODE                          : 0564
                     05        6C  91 000A2               CMPB     (AP), #5                                    :
                               0A  1F 000A5               BLSSU    10$                                         :
                           14  AC  D5 000A7               TSTL     20(AP)                                      :
                               05  13 000AA               BEQL     10$                                         :
```

```
                        04    AE         14    BC  C8 000AC        BISL2   @RENDITION_SET, REND_CODE                        :
                              06               6C  91 000B1 10$:   CMPB    (AP), #6                                         :
                                               0A  1F 000B4        BLSSU   11$                                              :
                                         18    AC  D5 000B6        TSTL    24(AP)                                           :
                                               05  13 000B9        BEQL    11$                                              :
                        04    AE         18    BC  CC 000BB        XORL2   @RENDITION_COMPLEMENT, REND_CODE                 :
                              07               6C  91 000C0 11$:   CMPB    (AP), #7                                         : 0567
                                               1C  1F 000C3        BLSSU   13$                                              :
                                         1C    AC  D5 000C5        TSTL    28(AP)                                           :
                                               17  13 000C8        BEQL    13$                                              :
                 04           00         1C    BC  CF 000CA        CASEL   @CHAR_SET, #0, #4                                : 0570
    0012         0012         0012           0012     000CF 12$:   .WORD   13$-12$,-                                        :
                                             0012     000D7                13$-12$,-                                        :
                                                                           13$-12$,-                                        :
                                                                           13$-12$,-                                        :
                                                                           13$-12$                                          :
                              50 00000000G     8F  D0 000D9        MOVL    #SMG$_INVARG, R0                                 : 0578
                                                   04 000E0        RET                                                      :
                              50         01    A6  9E 000E1 13$:   MOVAB   1(R6), R0                                        : 0589
                 56           50               02  C7 000E5        DIVL3   #2, R0, COL                                      :
                              01         4C  B847  91 000E9        CMPB    @76(DCB)[ROW], #1                                : 0601
                                               39  13 000EE        BEQL    15$                                              :
                              50         FF    A7  9E 000F0        MOVAB   -1(R7), R0                                       : 0606
                              51         06    A8  3C 000F4        MOVZWL  6(DCB), R1                                       :
                              50               51  C4 000F8        MULL2   R1, R0                                           :
                              5B               50  D0 000FB        MOVL    R0, START_INDEX                                  :
                              50         10    A8  D0 000FE        MOVL    16(DCB), TEXT_BUF                                : 0607
                              59         14    A8  7D 00102        MOVQ    20(DCB), ATTR_BUF                                :
  06    A8       20           6E               00  2C 00106        MOVC5   #0, (SP), #32, 6(DCB), (START_INDEX)-            :
                                             6B40     0010C                [TEXT_BUF]                                       :
  06    A8    2E    A8        6E               00  2C 0010E        MOVC5   #0, (SP), 46(DCB), 6(DCB), (START_INDEX)-        :
                                             6B49     00115                [ATTR_BUF]                                       :
                                               5A  D5 00117        TSTL    CHAR_BUF                                         :
                                               09  13 00119        BEQL    14$                                              :
  06    A8    30    A8        6E               00  2C 0011B        MOVC5   #0, (SP), 48(DCB), 6(DCB), (START_INDEX)-        :
                                             6B4A     00122                [CHAR_BUF]                                       :
                        4C  B847               01  90 00124 14$:   MOVB    #1, @76(DCB)[ROW]                                : 0608
                        4C    B8               01  90 00129 15$:   MOVB    #1, @76(DCB)                                     : 0611
                              59               01  8E 0012D        MNEGB   #1, ALLONES                                      : 0623
                                               54  D4 00130        CLRL    PRINT_LEN                                        :
                              55               6C  D0 00132        MOVL    STR_LEN, BYTES_REMAINING                         :
                              5B         0C    AE  D0 00135        MOVL    STR_ADDR, IN_POINTER                             :
                                               6E  D4 00139        CLRL    STR_LEN                                          :
                                               55  D5 0013B 16$:   TSTL    BYTES_REMAINING                                  :
                                               59  13 0013D        BEQL    20$                                              :
      59 00000000G  00        6B               55  2A 0013F        SCANC   BYTES_REMAINING, (IN_POINTER), CHAR_TABLE, -     :
                                                                           ALLONES                                          :
                 52           55               50  C3 00148        SUBL3   NEW_BYTES_REMAINING, BYTES_REMAINING, R2         :
                              5B               52  C0 0014C        ADDL2   R2, IN_POINTER                                   :
                              54               52  C0 0014F        ADDL2   R2, PRINT_LEN                                    :
                              6E               52  C0 00152        ADDL2   R2, STR_LEN                                      :
                              55               50  D0 00155        MOVL    NEW_BYTES_REMAINING, BYTES_REMAINING             :
                                               3E  13 00158        BEQL    20$                                              :
                              50               61  9A 0015A        MOVZBL  (ADDR_DIFF), R0                                  :
                 09           01 00000000G0040 8F  8F 0015D        CASEB   CHAR_TABLE[R0], #1, #9                           :
    001C         002C         002C           002C     00166 17$:   .WORD   19$-17$,-                                        :
    0032         0032         0032           0032     0016E                19$-17$,-                                        :
```

```
                     002C        0032     00176                       19$-17$,-                      ;
                                                                      18$-17$,-                      ;
                                                                      20$-17$,-                      ;
                                                                      20$-17$,-                      ;
                                                                      20$-17$,-                      ;
                                                                      20$-17$,-                      ;
                                                                      20$-17$,-                      ;
                                                                      19$-17$                        ;
                     50 00000000G  8F  D0 0017A           MOVL        #SMG$_FATERRLIB, R0            ;
                                       04 00181           RET                                        ;
                     50        2A  A8  3C 00182 18$:      MOVZWL      42(DCB), R0                    ;
                                   50  D7 00186           DECL        R0                             ;
                     50            08  C6 00188           DIVL2       #8, R0                         ;
                     54      09 A440  7E 0018B            MOVAQ       9(PRINT_LEN)[R0], PRINT_LEN    ;
                                   6E  D6 00190           INCL        STR_LEN                        ;
                                   5B  D6 00192 19$:      INCL        IN_POINTER                     ;
                                   55  D7 00194           DECL        BYTES_REMAINING                ;
                                   A3  11 00196           BRB         16$                            ;
                     50        06  A8  3C 00198 20$:      MOVZWL      6(DCB), R0                     ; 0631
                                   50  D7 0019C           DECL        R0                             ;
                     50            02  C6 0019E           DIVL2       #2, R0                         ;
                     50            54  D1 001A1           CMPL        PRINT_LEN, R0                  ;
                                   03  15 001A4           BLEQ        21$                            ;
                     54            50  D0 001A6           MOVL        R0, PRINT_LEN                  ; 0633
                28   A8            57  B0 001A9 21$:      MOVW        ROW, 40(DCB)                   ; 0635
                2A   A8            56  B0 001AD           MOVW        COL, 42(DCB)                   ; 0636
                     07            6C  91 001B1           CMPB        (AP), #7                       ; 0642
                                   0A  1F 001B4           BLSSU       22$                            ;
                               1C  AC  D5 001B6           TSTL        28(AP)                         ;
                                   05  13 001B9           BEQL        22$                            ;
                               1C  BC  DD 001BB           PUSHL       @CHAR_SET                      ; 0643
                                   02  11 001BE           BRB         23$                            ;
                                   01  DD 001C0 22$:      PUSHL       #1                             ; 0642
                               10  AE  DD 001C2 23$:      PUSHL       STR_ADDR                       ; 0641
                                   54  DD 001C5           PUSHL       PRINT_LEN                      ; 0640
                               10  AE  DD 001C7           PUSHL       REND_CODE                      ; 0639
                                   58  DD 001CA           PUSHL       DCB                            ; 0638
           00000000G 00            05  FB 001CC           CALLS       #5, SMG$$PUT_TEXT_TO_BUFFER    ;
                08   AE            50  D0 001D3           MOVL        R0, STATUS                     ;
                     05        08  AE  E8 001D7           BLBS        STATUS, 25$                    ;
                     50        08  AE  D0 001DB 24$:      MOVL        STATUS, R0                     ; 0646
                                       04 001DF           RET                                        ;
                     54            02  C4 001E0 25$:      MULL2       #2, R4                         ; 0655
                     50      FF A446  3E 001E3            MOVAW       -1(R4)[COL], R0                ;
                2A   A8            50  B0 001E8           MOVW        R0, 42(DCB)                    ;
                                   57  DD 001EC           PUSHL       ROW                            ; 0664
                                   11  DD 001EE           PUSHL       #17                            ; 0662
                                   58  DD 001F0           PUSHL       DCB                            ;
           00000000G 00            03  FB 001F2           CALLS       #3, SMG$$CHECK_FOR_OUTPUT_DCB  ;
                                       04 001F9           RET                                        ; 0666

; Routine Size:  506 bytes,    Routine Base:  _SMG$CODE + 0000
```

```
1 %SBTTL 'SMG$PUT_CHARS_HIGHWIDE - Write double high double wide characters'
1 GLOBAL ROUTINE SMG$PUT_CHARS_HIGHWIDE (
1                                                     DISPLAY_ID,
1                                                     TEXT : REF BLOCK [,BYTE],
1                                                     LINE_NO,
1                                                     COL_NO,
1                                                     RENDITION_SET,
1                                                     RENDITION_COMPLEMENT,
1                                                     CHAR_SET
1       ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine writes double high/double wide characters to a
1 !       virtual display.  The line can not contain a mixture of single
1 !       high/wide and double high/wide characters; if the line previously
1 !       contained single high/wide, then the entire line will be re-
1 !       written, otherwise only the specified text is written.
1 !
1 !       The internal cursor position is left at the character
1 !       position following the text written.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = SMG$PUT_CHARS_HIGHWIDE (DISPLAY_ID.rl.r,
1 !                                                  TEXT.rt.dx,
1 !                                                  [,LINE_NO.rl.r, COL_NO.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 virtual display
1 !
1 !       TEXT.rt.dx              Address of descriptor of output string
1 !
1 !       LINE_NO.rl.r            Optional.  Address of line number at which
1 !                               to start output.  This line will contain the
1 !                               upper half of the double high text.  If omitted
1 !                               (=0), the current line number is used.
1 !
1 !       COL_NO.rl.r             Optional.  Address of column number at which
1 !                               to start output.  If omitted (=0), the
1 !                               current column number is used.
1 !
1 !       RENDITION_SET.rl.r      Optional.  Each 1 bit in this parameter
1 !                               causes the corresponding attribute to be
1 !                               set in the display.  (See below for list
1 !                               of settable attributes.)
1 !
1 !       RENDITION_COMPLEMENT.rl.r Optional.  Each 1 bit attribute in this
1 !                                 parameter causes the corresponding attribute
1 !                                 to be complemented in the display.  (See
1 !                                 below for list of complementable attributes.)
1 !
```

```
: 466     0724  1 !
: 467     0725  1 !       If the same bit is specified in both the RENDITION_SET parameter
: 468     0726  1 !       and in the RENDITION_COMPLEMENT parameter, the application is
: 469     0727  1 !       RENDITION_SET followed by RENDITION complement.  Using these two
: 470     0728  1 !       parameters together the caller can exercise arbitrary and
: 471     0729  1 !       independent control over each attribute on a single call.  On an
: 472     0730  1 !       attribute by attribute basis he can cause the following
: 473     0731  1 !       transformations:
: 474     0732  1 !
: 475     0733  1 !               SET     COMPLEMENT      Action
: 476     0734  1 !               ---     ----------      ------
: 477     0735  1 !               0       0               Attribute unchanged.
: 478     0736  1 !               1       0               Attribute set to "on"
: 479     0737  1 !               0       1               Attribute set to complement of
: 480     0738  1 !                                       current setting.
: 481     0739  1 !               1       1               Attribute set to "off".
: 482     0740  1 !
: 483     0741  1 !
: 484     0742  1 !       Attributes which can be manipulated in this manner are:
: 485     0743  1 !
: 486     0744  1 !       SMG$M_BLINK  displays characters blinking.
: 487     0745  1 !       SMG$M_BOLD  displays characters in higher-than-normal
: 488     0746  1 !                   intensity.
: 489     0747  1 !       SMG$M_REVERSE  displays characters in reverse video -- that is,
: 490     0748  1 !                   using the opposite default rendition of the
: 491     0749  1 !                   virtual display.
: 492     0750  1 !       SMG$M_UNDERLINE  displays characters underlined.
: 493     0751  1 !
: 494     0752  1 !
: 495     0753  1 !       CHAR_SET.rl.r           Optional.  Character set to use.  Choices are:
: 496     0754  1 !                                       SMG$C_UNITED_KINGDOM
: 497     0755  1 !                                       SMG$C_ASCII      (default)
: 498     0756  1 !                                       SMG$C_SPEC_GRAPHICS
: 499     0757  1 !                                       SMG$C_ALT_CHAR
: 500     0758  1 !                                       SMG$C_ALT_GRAPHICS
: 501     0759  1 !
: 502     0760  1 ! IMPLICIT INPUTS:
: 503     0761  1 !
: 504     0762  1 !       NONE
: 505     0763  1 !
: 506     0764  1 ! IMPLICIT OUTPUTS:
: 507     0765  1 !
: 508     0766  1 !       NONE
: 509     0767  1 !
: 510     0768  1 ! COMPLETION STATUS:
: 511     0769  1 !
: 512     0770  1 !       SS$_NORMAL      Normal successful completion
: 513     0771  1 !       SMG$_INVCOL     Invalid column number
: 514     0772  1 !       SMG$_INVROW     Invalid row number
: 515     0773  1 !       LIB$_INVSTRDES  Invalid string descriptor
: 516     0774  1 !       SMG$_WRONUMARG  Wrong number of arguments
: 517     0775  1 !
: 518     0776  1 ! SIDE EFFECTS:
: 519     0777  1 !
: 520     0778  1 !       NONE
: 521     0779  1 !
: 522     0780  1 !--
```

```
1
2       BEGIN
2       BUILTIN
2           NULLPARAMETER;
2
2       LOCAL
2           DCB : REF BLOCK [,BYTE],
2           ROW,                                ! working row
2           COL,                                ! working column
2           REND_CODE,                          ! rendition code to use
2           STR_LEN : INITIAL (0),              ! length of text string
2           STR_ADDR,                           ! address of text string,
2           LOWER_HALF,                         ! flag to output lower half
2                                               ! of dbl high
2           STATUS;
2
2       LITERAL
2           K_LINE_ARG = 3,
2           K_COL_ARG = 4,
2           K_SET_ARG = 5,
2           K_COMP_ARG = 6,
2           K_CHAR_ARG = 7;
2
2       $SMG$GET_DCB (.DISPLAY_ID, DCB);      ! get addr of virtual display
2                                               ! control block
2
2       $SMG$VALIDATE_ARGCOUNT (2, 7);
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 optional arguments.  Set local variables to caller's
2   ! values, when available, and defaults when arguments omitted.
2   !-
2
2       IF NOT NULLPARAMETER (K_LINE_ARG) AND
2          NOT NULLPARAMETER (K_COL_ARG)
2       THEN
3           BEGIN
3           ROW = ..LINE_NO;
3           COL = ..COL_NO;
3           $SMG$VALIDATE_ROW_COL (.ROW, .COL);
3                                       ! verify row & col within display
3           END
2       ELSE
3           BEGIN
3           ROW = .DCB [DCB_W_CURSOR_ROW];
3           COL = .DCB [DCB_W_CURSOR_COL];
2           END;
```

```
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 (K_CHAR_ARG)
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$C_ALT_GRAPHICS]:
3                  ;
3
3          [INRANGE, OUTRANGE]:
3                  RETURN (SMG$_INVARG);
3
3          TES;
2          END;
2
2 !+
2 ! Double wide characters occupy two positions instead of one on the
2 ! screen.  However,  or mapping purposes we store the text only half
2 ! way over in the text buffer.
2 !-
2
2      COL = (.COL + 1)/2;          ! col in half for dbl wide
2
2 !+
2 ! Set the double wide/double high characteristic in the DCB.
2 !-
2
3      BEGIN
3      BIND
3          DCB_LCV = .DCB [DCB_A_LINE_CHAR];
3      MAP
3          DCB_LCV : VECTOR [,BYTE];
3
3      IF .DCB_LCV [.ROW] NEQ LINE_K_UPPER_HIGH    ! previously single wide
3      THEN                                        ! or just wide
4          BEGIN
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          DCB_LCV [.ROW] = LINE_K_UPPER_HIGH;       ! set this row to dbl high
3          END;                                      ! (note that this implies
3                                                    !  dbl wide also)
3      IF .ROW + 1 LEQ .DCB [DCB_W_NO_ROWS]
3      THEN
4          BEGIN                                    ! we can fit other half
4          LOWER_HALF = 1;
4          IF .DCB_LCV [.ROW] NEQ LINE_K_LOWER_HIGH ! previously single wide
4          THEN                                      ! or just wide
5              BEGIN
5              LOCAL
5                  START_INDEX;
```

```
5                START_INDEX = $SMG$LINEAR (.ROW, 1);
5                $SMG$BLANK_FILL_DCB (.DCB [DCB_W_NO_COLS], .START_INDEX);
5                DCB_LCV [.ROW + 1] = LINE_K_LOWER_HIGH;
4                END;
3            END;                                       ! we can fit other half
3
3        DCB_LCV [0] = 1;                       ! mark that there are wide or
3                                               ! dbl high/wide chars in display
2        END;
2
2   !+
2   ! All local variables are set up.  Call routine to put text into
2   ! the display buffer.
2   !-
2
3        BEGIN
3        LOCAL
3            PRINT_LEN;
3        $SMG$FIND_PRINT_LENGTH (STR_LEN, .STR_ADDR, PRINT_LEN);
3                                   ! don't count non-printable chars
3        !+
3        ! SMG$$PUT_TEXT_TO_BUFFER doesn't realize that wide characters
3        ! occupy 2 spaces so it won't recognize overflow.  Make sure
3        ! we don't try to put more chars in buffer than will fit on
3        ! this line.
3        !-
4        IF .PRINT_LEN GTR ((.DCB [DCB_W_NO_COLS] - 1)/2)
3        THEN
3            PRINT_LEN = (.DCB [DCB_W_NO_COLS] - 1)/2;
3
3        DCB [DCB_W_CURSOR_ROW] = .ROW;
3        DCB [DCB_W_CURSOR_COL] = .COL;
3                                   ! set position for put_text
4        IF NOT (STATUS = SMG$$PUT_TEXT_TO_BUFFER (.DCB,
4                                                   .REND_CODE,
4                                                   .PRINT_LEN.
4                                                   .STR_ADDR,
4                                                   IF NOT NULLPARAMETER (K_CHAR_ARG)
4                                                   THEN ..CHAR_SET
4                                                   ELSE SMG$C_ASCII))
3        THEN
3            RETURN (.STATUS);
3
3        IF .LOWER_HALF
3        THEN
4            BEGIN                              ! write lower half of dbl high
4
4            DCB [DCB_W_CURSOR_ROW] = .ROW + 1;
4            DCB [DCB_W_CURSOR_COL] = .COL;
4                                               ! set position for put_text
5            IF NOT (STATUS = SMG$$PUT_TEXT_TO_BUFFER (.DCB,
5                                                       .REND_CODE,
5                                                       .PRINT_LEN,
5                                                       .STR_ADDR,
5                                                       IF NOT NULLPARAMETER (K_CHAR_ARG)
5                                                       THEN ..CHAR_SET
5                                                       ELSE SMG$C_ASCII))
```

```
;  694         0952  4           THEN
;  695         0953  4               RETURN (.STATUS);
;  696         0954  4
;  697         0955  4  !+
;  698         0956  4  ! Correct the cursor position.  We stored our text half way over in the
;  699         0957  4  ! buffer, but the screen cursor position should be calculated based on
;  700         0958  4  ! the actual starting column specified by the caller.  Also take into
;  701         0959  4  ! account that characters occupy 2 positions.
;  702         0960  4  !-
;  703         0961  4
;  704         0962  4           DCB [DCB_W_CURSOR_COL] = (2 * .COL) + (2 * .PRINT_LEN) - 1;
;  705         0963  4
;  706         0964  4           IF .LOWER_HALF                       ! we used 2 rows if we could
;  707         0965  4           THEN                                 ! write the lower half of dbl high
;  708         0966  4               DCB [DCB_W_CURSOR_ROW] = .ROW + 1
;  709         0967  4           ELSE
;  710         0968  4               DCB [DCB_W_CURSOR_ROW] = .ROW;
;  711         0969  3           END;
;  712         0970  3
;  713         0971  2       END;
;  714         0972  2
;  715         0973  2  !+
;  716         0974  2  ! See if this change should be reflected on the screen.
;  717         0975  2  !-
;  718         0976  2
;  719         0977  3       RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
;  720         0978  3                                          SMG$C_PUT_CHARS,
;  721         0979  2                                          0));
;  722         0980  2
;  723         0981  1       END;                                 ! End of routine SMG$PUT_CHARS_HIGHWIDE



                                  0FFC 00000          .ENTRY  SMG$PUT_CHARS_HIGHWIDE, Save R2,R3,R4,R5,-  ; 0668
                                                              R6,R7,R8,R9,R10,R11                          ;
                 5E          14   C2 00002            SUBL2   #20, SP                                      ;
                 7E               D4 00005            CLRL    STR_LEN                                      ; 0782
                 50      04  BC   D0 00007            MOVL    @DISPLAY_ID, R0                              ; 0804
          04     BC      38  A0   D1 0000B            CMPL    56(R0), @DISPLAY_ID                          ;
                             06   12 00010            BNEQ    1$                                           ;
                 11      44  A0   91 00012            CMPB    68(R0), #17                                  ;
                             08   13 00016            BEQL    2$                                           ;
                 50 00000000G     8F   D0 00018 1$:   MOVL    #SMG$_INVDIS_ID, R0                          ;
                                  04 0001F            RET                                                  ;
                 58      04  BC   D0 00020 2$:        MOVL    @DISPLAY_ID, DCB                             ;
      50         6C          02   83 00024            SUBB3   #2, (AP), DIFF                               ; 0807
                 05          50   91 00028            CMPB    DIFF, #5                                     ;
                             08   1B 0002B            BLEQU   3$                                           ;
                 50 00000000G     8F   D0 0002D       MOVL    #SMG$_WRONUMARG, R0                          ;
                                  04 00034            RET                                                  ;
                 50      08  AC   D0 00035 3$:        MOVL    TEXT, R0                                     ; 0813
                    00000000G     00   16 00039       JSB     LIB$ANALYZE_SDESC_R2                         ;
          08     AE          50   D0 0003F            MOVL    R0, STATUS                                   ;
                 6E          51   D0 00043            MOVL    R1, (SP)                                     ;
          14     AE          52   D0 00046            MOVL    R2, 20(SP)                                   ;
```

```
                          03     08    AE E8 0004A       BLBS    STATUS, 4$                      ;
                                     0214 31 0004E       BRW     28$                             ;
                          03           6C 91 00051 4$:   CMPB    (AP), #3                        ; 0824
                                       3F 1F 00054       BLSSU   8$                              ;
                                 0C    AC D5 00056       TSTL    12(AP)                          ;
                                       3A 13 00059       BEQL    8$                              ;
                          04           6C 91 0005B       CMPB    (AP), #4                        ; 0825
                                       35 1F 0005E       BLSSU   8$                              ;
                                 10    AC D5 00060       TSTL    16(AP)                          ;
                                       30 13 00063       BEQL    8$                              ;
                          57     0C    BC D0 00065       MOVL    @LINE_NO, ROW                   ; 0828
                          56     10    BC D0 00069       MOVL    @COL_NO, COL                    ; 0829
                                       57 D5 0006D       TSTL    ROW                             ; 0830
                                       08 15 0006F       BLEQ    5$                              ;
        57    02 A8       10           00 ED 00071       CMPZV   #0, #16, 2(DCB), ROW            ;
                                       08 18 00077       BGEQ    6$                              ;
                          50 00000000G 8F D0 00079 5$:   MOVL    #SMG$_INVROW, R0                ;
                                          04 00080       RET                                     ;
                                       56 D5 00081 6$:   TSTL    COL                             ;
                                       08 15 00083       BLEQ    7$                              ;
        56    06 A8       10           00 ED 00085       CMPZV   #0, #16, 6(DCB), COL            ;
                                       10 18 0008B       BGEQ    9$                              ;
                          50 00000000G 8F D0 0008D 7$:   MOVL    #SMG$_INVCOL, R0                ;
                                          04 00094       RET                                     ;
                          57     28    A8 3C 00095 8$:   MOVZWL  40(DCB), ROW                    ; 0835
                          56     2A    A8 3C 00099       MOVZWL  42(DCB), COL                    ; 0836
                       0C AE     2E    A8 9A 0009D 9$:   MOVZBL  46(DCB), REND_CODE              ; 0839
                          05           6C 91 000A2       CMPB    (AP), #5                        ;
                                       0A 1F 000A5       BLSSU   10$                             ;
                                 14    AC D5 000A7       TSTL    20(AP)                          ;
                                       05 13 000AA       BEQL    10$                             ;
                       0C AE     14    BC C8 000AC       BISL2   @RENDITION_SET, REND_CODE       ;
                          06           6C 91 000B1 10$:  CMPB    (AP), #6                        ;
                                       0A 1F 000B4       BLSSU   11$                             ;
                                 18    AC D5 000B6       TSTL    24(AP)                          ;
                                       05 13 000B9       BEQL    11$                             ;
                       0C AE     18    BC CC 000BB       XORL2   @RENDITION_COMPLEMENT, REND_CODE ;
                          07           6C 91 000C0 11$:  CMPB    (AP), #7                        ; 0842
                                       1C 1F 000C3       BLSSU   13$                             ;
                                 1C    AC D5 000C5       TSTL    28(AP)                          ;
                                       17 13 000C8       BEQL    13$                             ;
                 04       00     1C    BC CF 000CA       CASEL   @CHAR_SET, #0, #4               ; 0845
      0012     0012     0012         0012    000CF 12$:  .WORD   13$-12$,-                       ;
                                     0012    000D7               13$-12$,-                       ;
                                                                 13$-12$,-                       ;
                                                                 13$-12$,-                       ;
                                                                 13$-12$                         ;
                          50 00000000G 8F D0 000D9       MOVL    #SMG$_INVARG, R0                ; 0853
                                          04 000E0       RET                                     ;
                          50     01    A6 9E 000E1 13$:  MOVAB   1(R6), R0                       ; 0864
                 56       50           02 C7 000E5       DIVL3   #2, R0, COL                     ;
                       04 AE     4C  B847 9E 000E9       MOVAB   @76(DCB)[ROW], 4(SP)            ; 0876
                          02     04    BE 91 000EF       CMPB    @4(SP), #2                      ;
                                       35 13 000F3       BEQL    15$                             ;
                          5B     FF    A7 9E 000F5       MOVAB   -1(R7), R11                     ; 0881
                          50     06    A8 3C 000F9       MOVZWL  6(DCB), R0                      ;
                          5B           50 C4 000FD       MULL2   R0, R11                         ;
```

```
                                50      10   A8  D0 00100        MOVL    16(DCB), TEXT_BUF                              : 0882
                                59      14   A8  7D 00104        MOVQ    20(DCB), ATTR_BUF                              :
     06   A8        20          6E           00  2C 00108        MOVC5   #0, (SP), #32, 6(DCB), (START_INDEX)-          :
                                           6B40     0010E                [TEXT_BUF]                                     :
     06   A8   2E   A8          6E           00  2C 00110        MOVC5   #0, (SP), 46(DCB), 6(DCB), (START_INDEX)-      :
                                           6B49     00117                [ATTR_BUF]                                     :
                                             5A  D5 00119        TSTL    CHAR_BUF                                       :
                                             09  13 0011B        BEQL    14$                                            :
     06   A8   30   A8          6E           00  2C 0011D        MOVC5   #0, (SP), 48(DCB), 6(DCB), (START_INDEX)-      :
                                           6B4A     00124                [CHAR_BUF]                                     :
                           04   BE           02  90 00126 14$:   MOVB    #2, @4(SP)                                     : 0883
                                5B      01   A7  9E 0012A 15$:   MOVAB   1(R7), R11                                     : 0886
     5B        02   A8          10           00  ED 0012E        CMPZV   #0, #16, 2(DCB), R11                           :
                                             47  19 00134        BLSS    17$                                            :
                           10   AE           01  D0 00136        MOVL    #1, LOWER_HALF                                 : 0889
                                03      04   BE  91 0013A        CMPB    @4(SP), #3                                     : 0890
                                             3D  13 0013E        BEQL    17$                                            :
                                50      FF   A7  9E 00140        MOVAB   -1(R7), R0                                     : 0895
                                51      06   A8  3C 00144        MOVZWL  6(DCB), R1                                     :
                                50           51  C4 00148        MULL2   R1, R0                                         :
                           04   AE           50  D0 0014B        MOVL    R0, START_INDEX                                :
                                50      10   A8  D0 0014F        MOVL    16(DCB), TEXT_BUF                              : 0896
                                59      14   A8  7D 00153        MOVQ    20(DCB), ATTR_BUF                              :
     06   A8        20          6E           00  2C 00157        MOVC5   #0, (SP), #32, 6(DCB), @START_INDEX-           :
                                        04 BE40     0015D                [TEXT_BUF]                                     :
     06   A8   2E   A8          6E           00  2C 00160        MOVC5   #0, (SP), 46(DCB), 6(DCB), @START_INDEX-       :
                                        04 BE49     00167                [ATTR_BUF]                                     :
                                             5A  D5 0016A        TSTL    CHAR_BUF                                       :
                                             0A  13 0016C        BEQL    16$                                            :
     06   A8   30   A8          6E           00  2C 0016E        MOVC5   #0, (SP), 48(DCB), 6(DCB), @START_INDEX-       :
                                        04 BE4A     00175                [CHAR_BUF]                                     :
                          4C B84B            03  90 00178 16$:   MOVB    #3, @76(DCB)[R11]                              : 0897
                          4C    B8           01  90 0017D 17$:   MOVB    #1, @76(DCB)                                   : 0901
                                5A           01  8E 00181        MNEGB   #1, ALLONES                                    : 0913
                                             55  D4 00184        CLRL    PRINT_LEN                                      :
                                54           6E  D0 00186        MOVL    STR_LEN, BYTES_REMAINING                       :
                                59      14   AE  D0 00189        MOVL    STR_ADDR, IN_POINTER                           :
                                             6E  D4 0018D        CLRL    STR_LEN                                        :
                                             54  D5 0018F 18$:   TSTL    BYTES_REMAINING                                :
                                             59  13 00191        BEQL    22$                                            :
          5A 00000000G  00      69           54  2A 00193        SCANC   BYTES_REMAINING, (IN_POINTER), CHAR_TABLE, -;
                                                                         ALLONES                                        :
                    52          54           50  C3 0019C        SUBL3   NEW_BYTES_REMAINING, BYTES_REMAINING, R2       :
                                59           52  C0 001A0        ADDL2   R2, IN_POINTER                                 :
                                55           52  C0 001A3        ADDL2   R2, PRINT_LEN                                  :
                                6E           52  C0 001A6        ADDL2   R2, STR_LEN                                    :
                                54           50  D0 001A9        MOVL    NEW_BYTES_REMAINING, BYTES_REMAINING           :
                                             3E  13 001AC        BEQL    22$                                            :
                                50           61  9A 001AE        MOVZBL  (ADDR_DIFF), R0                                :
                    09          01 00000000G0040  8F 001B1        CASEB   CHAR_TABLE[R0], #1, #9                         :
       001C       002C        002C         002C     001BA 19$:   .WORD   21$-19$,-                                      :
       0032       0032        0032         0032     001C2                21$-19$,-                                      :
                              002C         0032     001CA                21$-19$,-                                      :
                                                                         20$-19$,-                                      :
                                                                         22$-19$,-                                      :
                                                                         22$-19$,-                                      :
                                                                         22$-19$,-                                      :
```

```
                                                                  22$-19$,-                                        ;
                                                                  22$-19$,-                                        ;
                                                                  21$-19$                                          ;
                  50 0000000G  8F   D0 001CE          MOVL     #SMG$_FATERRLIB, R0                                ;
                                    04 001D5          RET                                                          ;
                  50        2A  A8  3C 001D6 20$:     MOVZWL   42(DCB), R0                                        ;
                                50  D7 001DA          DECL     R0                                                 ;
                  50            08  C6 001DC          DIVL2    #8, R0                                             ;
                  55       09 A540  7E 001DF          MOVAQ    9(PRINT_LEN)[R0], PRINT_LEN                        ;
                                6E  D6 001E4          INCL     STR_LEN                                            ;
                                59  D6 001E6 21$:     INCL     IN_POINTER                                         ;
                                54  D7 001E8          DECL     BYTES_REMAINING                                    ;
                                A3  11 001EA          BRB      18$                                                ;
                  50        06  A8  3C 001EC 22$:     MOVZWL   6(DCB), R0                                         ; 0921
                                50  D7 001F0          DECL     R0                                                 ;
                  50            02  C6 001F2          DIVL2    #2, R0                                             ;
                  50            55  D1 001F5          CMPL     PRINT_LEN, R0                                      ;
                                03  15 001F8          BLEQ     23$                                                ;
                  55            50  D0 001FA          MOVL     R0, PRINT_LEN                                      ; 0923
             28   A8            57  B0 001FD 23$:     MOVW     ROW, 40(DCB)                                       ; 0925
             2A   A8            56  B0 00201          MOVW     COL, 42(DCB)                                       ; 0926
                  07            6C  91 00205          CMPB     (AP), #7                                           ; 0932
                                0A  1F 00208          BLSSU    24$                                                ;
                            1C  AC  D5 0020A          TSTL     28(AP)                                             ;
                                05  13 0020D          BEQL     24$                                                ;
                            1C  BC  DD 0020F          PUSHL    @CHAR_SET                                          ; 0933
                                02  11 00212          BRB      25$                                                ;
                                01  DD 00214 24$:     PUSHL    #1                                                 ; 0932
                            18  AE  DD 00216 25$:     PUSHL    STR_ADDR                                           ; 0931
                                55  DD 00219          PUSHL    PRINT_LEN                                          ; 0930
                            18  AE  DD 0021B          PUSHL    REND_CODE                                          ; 0929
                                58  DD 0021E          PUSHL    DCB                                                ; 0928
        0000000G  00            05  FB 00220          CALLS    #5, SMG$$PUT_TEXT_TO_BUFFER                        ;
              08  AE            50  D0 00227          MOVL     R0, STATUS                                         ;
                  36        08  AE  E9 0022B          BLBC     STATUS, 28$                                        ;
                  51        10  AE  E9 0022F          BLBC     LOWER_HALF, 31$                                    ; 0938
             28   A8            5B  B0 00233          MOVW     R11, 40(DCB)                                       ; 0942
             2A   A8            56  B0 00237          MOVW     COL, 42(DCB)                                       ; 0943
                  07            6C  91 0023B          CMPB     (AP), #7                                           ; 0949
                                0A  1F 0023E          BLSSU    26$                                                ;
                            1C  AC  D5 00240          TSTL     28(AP)                                             ;
                                05  13 00243          BEQL     26$                                                ;
                            1C  BC  DD 00245          PUSHL    @CHAR_SET                                          ; 0950
                                02  11 00248          BRB      27$                                                ;
                                01  DD 0024A 26$:     PUSHL    #1                                                 ; 0949
                            18  AE  DD 0024C 27$:     PUSHL    STR_ADDR                                           ; 0948
                                55  DD 0024F          PUSHL    PRINT_LEN                                          ; 0947
                            18  AE  DD 00251          PUSHL    REND_CODE                                          ; 0946
                                58  DD 00254          PUSHL    DCB                                                ; 0945
        0000000G  00            05  FB 00256          CALLS    #5, SMG$$PUT_TEXT_TO_BUFFER                        ;
              08  AE            50  D0 0025D          MOVL     R0, STATUS                                         ;
                  05        08  AE  E8 00261          BLBS     STATUS, 29$                                        ;
                  50        08  AE  D0 00265 28$:     MOVL     STATUS, R0                                         ; 0953
                                    04 00269          RET                                                          ;
                  55            02  C4 0026A 29$:     MULL2    #2, R5                                             ; 0962
                  50       FF A546  3E 0026D          MOVAW    -1(R5)[COL], R0                                    ;
             2A   A8            50  B0 00272          MOVW     R0, 42(DCB)                                        ;
```

```
                            06        10  AE  E9 00276          BLBC    LOWER_HALF, 30$                              ; 0964
                       28   A8            5B  B0 0027A          MOVW    R11, 40(DCB)                                 ; 0966
                                          04  11 0027E          BRB     31$                                          ;
                       28   A8            57  B0 00280 30$:     MOVW    ROW, 40(DCB)                                 ; 0968
                            7E            11  7D 00284 31$:     MOVQ    #17, -(SP)                                   ; 0977
                                          58  DD 00287          PUSHL   DCB                                          ;
                  00000000G 00            03  FB 00289          CALLS   #3, SMG$$CHECK_FOR_OUTPUT_DCB                ;
                                              04 00290          RET                                                  ; 0981

; Routine Size:  657 bytes,    Routine Base:  _SMG$CODE + 01FA
```

```
1 %SBTTL 'SMG$PUT_LINE_WIDE - Put Wide Text to Display in Line Mode'
1 GLOBAL ROUTINE SMG$PUT_LINE_WIDE (
1                                   DISPLAY_ID,
1                                   TEXT     : REF BLOCK [,BYTE],
1                                   LINE_ADV,
1                                   RENDITION_SET,
1                                   RENDITION_COMPLEMENT,
1                                   WRAP_FLAG,
1                                   CHAR_SET
1                               ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine is used to write lines with wide characters to the
1 !       virtual display optionally followed by cursor movement sequences.
1 !       SMG$PUT_LINE_WIDE writes from the current cursor position to the
1 !       end of the line. If the caller's text does not span to the end of
1 !       the line, blank fill is added.
1 !
1 !       Treatment of text which exceeds the rightmost bounds of the
1 !       display depends on WRAP_FLAG.  If WRAP_FLAG is set, lines are
1 !       scrolled LINE_ADV times to make room for the overflow characters
1 !       in the 'next' line.  If wrap is off, overflow characters are lost.
1 !
1 !       Following a call to SMG$PUT_LINE_WIDE, the internal display cursor
1 !       position is set to column 1 of the next line where output should
1 !       occur.  The next line where output should occur is determined by
1 !       LINE_ADV;  LINE_ADV defaults to 1 so that subsequent calls to
1 !       SMG$PUT_LINE_WIDE will not cause overprinting.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = SMG$PUT_LINE_WIDE (DISPLAY_ID.rl.r,
1 !                                             TEXT.rt.dx
1 !                                             [,LINE_ADV.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 virtual display
1 !
1 !       TEXT.rt.dx      Address of descriptor of output string.
1 !
1 !       LINE_ADV.rl.r   Optional.  Address of signed number of lines
1 !                       to advance after output.
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.
!
!       WRAP_FLAG.rl.r  = 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
!       SMG$_WRONUMARG  Wrong number (of combination of) arguments
!       LIB$_INVSTRDES  Invalid string descriptor
```

```
  1 !
  1 ! SIDE EFFECTS:
  1 !
  1 !     NONE
  1 !--
  1
  2    BEGIN
  2
  2    BUILTIN
  2        NULLPARAMETER;
  2
  2    LOCAL
  2        HALF_NO_COLS,
  2        DONE,
  2        DCB : REF BLOCK [,BYTE],        ! Address of virtual display
  2                                        ! control block.
  2        STR_LEN : INITIAL (0),          ! Length of text string
  2        STR_ADDR,                       ! Address of text string
  2        REND_CODE,                      ! Rendition code to use
  2        WRAPPED_CHARS :INITIAL (0),     ! Number of chars that don't fit on
  2                                        ! the current line
  2        SCROLL_FLAG : INITIAL (0),      ! Flag to scroll up, down, or neither
  2        STATUS;                         ! Status of subroutine calls
  2
  2    LITERAL
  2        K_ADV_ARG = 3,
  2        K_SET_ARG = 4,
  2        K_COMP_ARG = 5,
  2        K_WRAP_ARG = 6,
  2        K_CHAR_ARG = 7;
  2
M 2    MACRO $SCROLL_UP (COUNT) =
M 2            SMG$$SCROLL_AREA (.DCB,
M 2                              .DCB [DCB_W_TOP_OF_SCRREG],
M 2                              .DCB [DCB_W_COL_START],
M 2                              (.DCB [DCB_B_BOTTOM_OF_SCRREG] -
M 2                               .DCB [DCB_W_TOP_OF_SCRREG] + 1),
M 2                              .DCB [DCB_B_NO_COLS],
  2                              SMG$M_UP, COUNT) %;
  2
  2
  2    $SMG$GET_DCB (.DISPLAY_ID, DCB);    ! Get address of virtual display
  2                                        ! control block.
  2
  2    $SMG$VALIDATE_ARGCOUNT (2,7);
  2
  2    IF NOT NULLPARAMETER (K_ADV_ARG) AND
  2       ..LINE_ADV LSS 0
  2    THEN
  2        RETURN (SMG$_INVARG);           ! positive advancing only
  2
  2 !+
  2 ! Select rendition code to use, based on whether one was provided by
  2 ! caller.
  2 !-
  2
  2    $SMG$SET_REND_CODE (K_SET_ARG, K_COMP_ARG);
  2
  2 !+
  2 ! Get the length and address of the text string.
  2 !-
  3     IF NOT (STATUS = LIB$ANALYZE_SDESC_R2 ( .TEXT,
  3                                             STR_LEN,
  3                                             STR_ADDR))
  2     THEN
  2         RETURN (.STATUS);
  2
  2 !+
  2 ! Compute the number of columns in a line.  We can fit only half
  2 ! as many wide characters since they occupy 2 positions.
  2 !-
  2
  2     HALF_NO_COLS = (.DCB [DCB_W_NO_COLS] - 1)/2;
  2
  2 !+
  2 ! If the previous line written was the last line in the display, we
  2 ! did not scroll at the end of the operation.  (This would've always
  2 ! left the last line blank - effectively the display would have one
  2 ! less useable line.)  If we're at the bottom, scroll up one before writing.
  2 !
  2 !-
  2
  2     $SMG$SET_SCROLLING (SCROLL_FLAG);
  2
  2     IF .SCROLL_FLAG EQL 1
  2     THEN                        ! we're at the last line in the display
  3         BEGIN                   ! and display is full
  3         IF NOT NULLPARAMETER (K_ADV_ARG) AND
  3             ..LINE_ADV GTR 0
  3         THEN                    ! positive line advancing
  3             $SCROLL_UP (..LINE_ADV)
  3         ELSE
  3             IF NULLPARAMETER (K_ADV_ARG)
  3             THEN
  3                 $SCROLL_UP (1)  ! default advancing
  3
  2         END;                    ! we're at the last line in the display
  2
  2 !+
  2 ! Blank out the line before writing new text.
  2 !-
  2
  3     BEGIN
  3     LOCAL
  3         START_INDEX;
  3
  3     START_INDEX = $SMG$LINEAR (.DCB [DCB_W_CURSOR_ROW], .DCB [DCB_W_CURSOR_COL]);
P 3     $SMG$BLANK_FILL_DCB ((.DCB [DCB_W_NO_COLS] - .DCB [DCB_W_CURSOR_COL] + 1),
  3                         .START_INDEX);
  3
  2     END;
  2
  2 !+
  2 ! Reset the line characteristics in case the line was previously
```

```
2 ! double high or single.
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]] = LINE_K_WIDE;
3     LINE_CHAR [0] = 1;                          ! mark that there are dbl chars
3                                                 ! in display
3     END;
2
2 !+
2 ! Move the text string into our virtual display buffer.
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 (K_CHAR_ARG)
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 (K_ADV_ARG)
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_FULL] = 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                           $SCROLL_UP (..LINE_ADV); ! scroll if w/in scroll region
5                   IF .DCB [DCB_W_CURSOR_ROW] LEQ .DCB [DCB_W_BOTTOM_OF_SCRREG]
5                   THEN
5                       DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_BOTTOM_OF_SCRREG];
5                   DCB [DCB_V_FULL] = 1; ! remember that we used last line
5                   END                 ! scrolling up
3               END;                    ! upspace action
3
2           END;                        ! line_adv specified
2
2       !+
2       ! If wrapping was requested and some characters overflowed the line,
2       ! call ourself again with the remainder of the characters.
2       !-
2
2       IF .WRAPPED_CHARS NEQ 0 AND
3          ((NOT NULLPARAMETER (K_ADV_ARG) AND ..LINE_ADV GTR 0) OR
3          NULLPARAMETER (K_ADV_ARG))
2       THEN
3           BEGIN                               ! overflow chars
4           IF (NOT NULLPARAMETER (K_WRAP_ARG) AND
4              ..WRAP_FLAG NEQ 0)
3           THEN
4               BEGIN                           ! wrap set - recurse w/overflow
4               LOCAL
4                   STR_DESC : BLOCK [8, BYTE],
4                   C_SET;
4
4               STR_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
```

```
4              STR_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
4              STR_DESC [DSC$W_LENGTH] = .STR_LEN - .HALF_NO_COLS;
4              STR_DESC [DSC$A_POINTER] = .STR_ADDR + .HALF_NO_COLS;
4
4              C_SET = SMG$C_ASCII;
4              IF NOT NULLPARAMETER( K_CHAR_ARG )
4              THEN
4                  C_SET = ..CHAR_SET;
4
4              SMG$PUT_LINE_WIDE (.DISPLAY_ID,
4                                 STR_DESC,
4                                 .LINE_ADV,
4                                 .RENDITION_SET,
4                                 .RENDITION_COMPLEMENT,
4                                 .WRAP_FLAG,
4                                 C_SET);
4              DONE = 0;
4              RETURN 1;              ! to keep Bliss happy
4              END                    ! wrap set - recurse w/overflow
3          ELSE
4              BEGIN                  ! wrap not set - truncation
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                  IF NOT .DCB [DCB_V_FULL]
5                  THEN
5                      DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_CURSOR_ROW] - 1;
5                  DCB [DCB_W_CURSOR_COL] = .HALF_NO_COLS;
5
5                  SMG$$PUT_TEXT_TO_BUFFER (.DCB,
5                                           .REND_CODE + ATTR_M_USER_GRAPHIC,
5                                           1, UPLIT (BYTE (DIAMOND)),
5                                           SMG$C_ASCII);
5
5                  IF NOT .DCB [DCB_V_FULL]
5                  THEN
5                      DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_CURSOR_ROW] + 1;
5                                           ! restore for next call
5                  DCB [DCB_W_CURSOR_COL] = 1;
4                  END;
4
4              DONE = 1;
3              END;                         ! wrap not set - truncation
3          END
2      ELSE
2          DONE = 1;                        ! no wrap chars
2
2      !+
2      ! See if this change should be reflected on the screen.
2      ! Even if we call SMG$PUT_LINE_WIDE again, SMG$$CHECK_FOR_OUTPUT_DCB
2      ! should be called only once.
2      !-
2
```

```
; 1124          1381  2      IF .DONE
; 1125          1382  2      THEN
; 1126          1383  3          BEGIN
; 1127          1384  3          LOCAL
; 1128          1385  3              LINE_CHANGED;
; 1129          1386  3          LINE_CHANGED = .DCB [DCB_W_CURSOR_ROW] -
; 1130          1387  4                          (IF NOT NULLPARAMETER (K_ADV_ARG) THEN
; 1131          1388  3                           ABS (..LINE_ADV) ELSE 0);
; 1132          1389  4          RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
; 1133          1390  4                                             SMG$C_PUT_LINE,
; 1134          1391  3                                             .LINE_CHANGED));
; 1135          1392  3          END
; 1136          1393  2      ELSE
; 1137          1394  2          RETURN 1;
; 1138          1395  2
; 1139          1396  1      END;                                      ! End of routine SMG$PUT_LINE_WIDE


                                        0048B           .BLKB   1
                                     60 0048C P.AAA:    .BYTE   96                                                    ;


                                   0FFC 00000           .ENTRY  SMG$PUT_LINE_WIDE, Save R2,R3,R4,R5,R6,R7,-  ; 0983
                                                                R8,R9,R10,R11                                 ;
                    5E           24  C2 00002           SUBL2   #36, SP                                       ;
                                 7E  D4 00005           CLRL    STR_LEN                                       ; 1102
                           18    AE  D4 00007           CLRL    WRAPPED_CHARS                                 ;
                                 7E  D4 0000A           CLRL    SCROLL_FLAG                                   ;
                    50     04    BC  D0 0000C           MOVL    @DISPLAY_ID, R0                               ; 1137
               04   BC     38    A0  D1 00010           CMPL    56(R0), @DISPLAY_ID                           ;
                                 06  12 00015           BNEQ    1$                                            ;
                    11     44    A0  91 00017           CMPB    68(R0), #17                                   ;
                                 08  13 0001B           BEQL    2$                                            ;
                    50 00000000G 8F  D0 0001D 1$:       MOVL    #SMG$_INVDIS_ID, R0                           ;
                                     04 00024           RET                                                   ;
                    56     04    BC  D0 00025 2$:       MOVL    @DISPLAY_ID, DCB                              ;
          50        6C           02  83 00029           SUBB3   #2, (AP), DIFF                                ; 1140
                    05           50  91 0002D           CMPB    DIFF, #5                                      ;
                                 08  1B 00030           BLEQU   3$                                            ;
                    50 00000000G 8F  D0 00032           MOVL    #SMG$_WRONUMARG, R0                           ;
                                     04 00039           RET                                                   ;
                    03           6C  91 0003A 3$:       CMPB    (AP), #3                                      ; 1142
                                 12  1F 0003D           BLSSU   4$                                            ;
                           0C    AC  D5 0003F           TSTL    12(AP)                                        ;
                                 0D  13 00042           BEQL    4$                                            ;
                           0C    BC  D5 00044           TSTL    @LINE_ADV                                     ; 1143
                                 08  18 00047           BGEQ    4$                                            ;
                    50 00000000G 8F  D0 00049           MOVL    #SMG$_INVARG, R0                              ; 1145
                                     04 00050           RET                                                   ;
               10   AE     2E    A6  9A 00051 4$:       MOVZBL  46(DCB), REND_CODE                            ; 1152
                    04           6C  91 00056           CMPB    (AP), #4                                      ;
                                 0A  1F 00059           BLSSU   5$                                            ;
                           10    AC  D5 0005B           TSTL    16(AP)                                        ;
                                 05  13 0005E           BEQL    5$                                            ;
               10   AE     10    BC  C8 00060           BISL2   @RENDITION_SET, REND_CODE                     ;
```

```
                    05             6C  91 00065 5$:     CMPB    (AP), #5                              :
                                   0A  1F 00068         BLSSU   6$                                    :
                              14   AC  D5 0006A         TSTL    20(AP)                                :
                                   05  13 0006D         BEQL    6$                                    :
              10    AE        14   BC  CC 0006F         XORL2   @RENDITION_COMPLEMENT, REND_CODE      :
                    50        08   AC  D0 00074 6$:     MOVL    TEXT, R0                              : 1157
                       00000000G   00  16 00078         JSB     LIB$ANALYZE_SDESC_R2                  :
              0C    AE             50  D0 0007E         MOVL    R0, STATUS                            :
              04    AE             51  D0 00082         MOVL    R1, 4(SP)                             :
              18    AE             52  D0 00086         MOVL    R2, 24(SP)                            :
                    03        0C   AE  E8 0008A         BLBS    STATUS, 7$                            :
                                 00FB  31 0008E         BRW     17$                                   :
                    50        06   A6  3C 00091 7$:     MOVZWL  6(DCB), R0                            : 1168
                                   50  D7 00095         DECL    R0                                    :
     14     AE      50             02  C7 00097         DIVL3   #2, R0, HALF_NO_COLS                  :
                                   6E  D4 0009C         CLRL    SCROLL_FLAG                           : 1178
              08    AE        34   A6  9E 0009E         MOVAB   52(DCB), 8(SP)                        :
              16    08             BE  E9 000A3         BLBC    @8(SP), 9$                            :
              4A    A6        28   A6  B1 000A7         CMPW    40(DCB), 74(DCB)                      :
                                   05  12 000AC         BNEQ    8$                                    :
                    6E             01  D0 000AE         MOVL    #1, SCROLL_FLAG                       :
                                   0A  11 000B1         BRB     9$                                    :
              48    A6        28   A6  B1 000B3 8$:     CMPW    40(DCB), 72(DCB)                      :
                                   03  12 000B8         BNEQ    9$                                    :
                    6E             02  D0 000BA         MOVL    #2, SCROLL_FLAG                       :
                    01             6E  D1 000BD 9$:     CMPL    SCROLL_FLAG, #1                       : 1180
                                   45  12 000C0         BNEQ    13$                                   :
                    03             6C  91 000C2         CMPB    (AP), #3                              : 1183
                                   0F  1F 000C5         BLSSU   10$                                   :
                              0C   AC  D5 000C7         TSTL    12(AP)                                :
                                   0A  13 000CA         BEQL    10$                                   :
                              0C   BC  D5 000CC         TSTL    @LINE_ADV                             : 1184
                                   05  15 000CF         BLEQ    10$                                   :
                              0C   BC  DD 000D1         PUSHL   @LINE_ADV                             : 1186
                                   0C  11 000D4         BRB     12$                                   :
                    03             6C  91 000D6 10$:    CMPB    (AP), #3                              : 1188
                                   05  1F 000D9         BLSSU   11$                                   :
                              0C   AC  D5 000DB         TSTL    12(AP)                                :
                                   27  12 000DE         BNEQ    13$                                   :
                                   01  DD 000E0 11$:    PUSHL   #1                                    : 1190
                                   01  DD 000E2 12$:    PUSHL   #1                                    :
                    7E        06   A6  3C 000E4         MOVZWL  6(DCB), -(SP)                         :
                    50        4A   A6  3C 000E8         MOVZWL  74(DCB), R0                           :
                    51        48   A6  3C 000EC         MOVZWL  72(DCB), R1                           :
                    50             51  C2 000F0         SUBL2   R1, R0                                :
                              01   A0  9F 000F3         PUSHAB  1(R0)                                 :
                    7E        04   A6  3C 000F6         MOVZWL  4(DCB), -(SP)                         :
                    7E        48   A6  3C 000FA         MOVZWL  72(DCB), -(SP)                        :
                                   56  DD 000FE         PUSHL   DCB                                   :
          00000000G 00             07  FB 00100         CALLS   #7, SMG$$SCROLL_AREA                  :
                    57        28   A6  9E 00107 13$:    MOVAB   40(DCB), R7                           : 1202
                    50             67  3C 0010B         MOVZWL  (R7), R0                              :
                                   50  D7 0010E         DECL    R0                                    :
                    51        06   A6  3C 00110         MOVZWL  6(DCB), R1                            :
                    50             51  C4 00114         MULL2   R1, R0                                :
                    51        2A   A6  3C 00117         MOVZWL  42(DCB), R1                           :
                    5B        FF A140  9E 0011B         MOVAB   -1(R1)[R0], START_INDEX               :
                            50       10  A6  D0 00120         MOVL    16(DCB), TEXT_BUF                            ; 1204
                            59       14  A6  D0 00124         MOVL    20(DCB), ATTR_BUF                            ;
                            58       18  A6  D0 00128         MOVL    24(DCB), CHAR_BUF                            ;
                            5A       06  A6  3C 0012C         MOVZWL  6(DCB), R10                                  ;
                            5A           51  C2 00130         SUBL2   R1, R10                                      ;
                                         5A  D6 00133         INCL    R10                                          ;
   5A        20             6E           00  2C 00135         MOVC5   #0, (SP), #32, R10, (START_INDEX)[TEXT_BUF]  ;
                                       6B40     0013A                                                          ;
   5A    2E  A6             6E           00  2C 0013C         MOVC5   #0, (SP), 46(DCB), R10, (START_INDEX)-       ;
                                       6B49     00142                 [ATTR_BUF]                                   ;
                                         58  D5 00144         TSTL    CHAR_BUF                                     ;
                                         08  13 00146         BEQL    14$                                          ;
   5A    30  A6             6E           00  2C 00148         MOVC5   #0, (SP), 48(DCB), R10, (START_INDEX)-       ;
                                       6B48     0014E                 [CHAR_BUF]                                   ;
                            50           67  3C 00150 14$:    MOVZWL  (R7), R0                                     ; 1218
                            50       4C  A6  C0 00153         ADDL2   76(DCB), R0                                  ;
                            60           01  90 00157         MOVB    #1, (R0)                                     ;
                        4C  B6           01  90 0015A         MOVB    #1, @76(DCB)                                 ; 1219
                                     1C  AE  9F 0015E         PUSHAB  WRAPPED_CHARS                                ; 1227
                            07           6C  91 00161         CMPB    (AP), #7                                     ; 1231
                                         0A  1F 00164         BLSSU   15$                                          ;
                                     1C  AC  D5 00166         TSTL    28(AP)                                       ;
                                         05  13 00169         BEQL    15$                                          ;
                                     1C  BC  DD 0016B         PUSHL   @CHAR_SET                                    ; 1232
                                         02  11 0016E         BRB     16$                                          ;
                                         01  DD 00170 15$:    PUSHL   #1                                           ; 1231
                                     20  AE  DD 00172 16$:    PUSHL   STR_ADDR                                     ; 1230
                                     10  AE  DD 00175         PUSHL   STR_LEN                                      ; 1229
                                     20  AE  DD 00178         PUSHL   REND_CODE                                    ; 1228
                                         56  DD 0017B         PUSHL   DCB                                          ; 1227
                 00000000G  00           06  FB 0017D         CALLS   #6, SMG$$PUT_TEXT_TO_BUFFER                  ;
                        0C  AE           50  D0 00184         MOVL    R0, STATUS                                   ;
                            05       0C  AE  E8 00188         BLBS    STATUS, 18$                                  ;
                            50       0C  AE  D0 0018C 17$:    MOVL    STATUS, R0                                   ; 1236
                                             04 00190         RET                                                  ;
                        2A  A6           01  B0 00191 18$:    MOVW    #1, 42(DCB)                                  ; 1244
                            03           6C  91 00195         CMPB    (AP), #3                                     ; 1250
                                         05  1F 00198         BLSSU   19$                                          ;
                                     0C  AC  D5 0019A         TSTL    12(AP)                                       ;
                                         65  12 0019D         BNEQ    24$                                          ;
                            50           67  3C 0019F 19$:    MOVZWL  (R7), R0                                     ; 1253
                                         50  D6 001A2         INCL    R0                                           ;
   50    4A  A6             10           00  ED 001A4         CMPZV   #0, #16, 74(DCB), R0                         ;
                                         04  19 001AA         BLSS    20$                                          ;
                                         67  B6 001AC         INCW    (R7)                                         ; 1256
                                         6B  11 001AE         BRB     25$                                          ;
                                         6E  D4 001B0 20$:    CLRL    SCROLL_FLAG                                  ; 1259
                            14       08  BE  E9 001B2         BLBC    @8(SP), 22$                                  ;
                        4A  A6           67  B1 001B6         CMPW    (R7), 74(DCB)                                ;
                                         05  12 001BA         BNEQ    21$                                          ;
                            6E           01  D0 001BC         MOVL    #1, SCROLL_FLAG                              ;
                                         09  11 001BF         BRB     22$                                          ;
                        48  A6           67  B1 001C1 21$:    CMPW    (R7), 72(DCB)                                ;
                                         03  12 001C5         BNEQ    22$                                          ;
                            6E           02  D0 001C7         MOVL    #2, SCROLL_FLAG                              ;
                            01           6E  D1 001CA 22$:    CMPL    SCROLL_FLAG, #1                              ; 1260
                                         2D  12 001CD         BNEQ    23$                                          ;
```

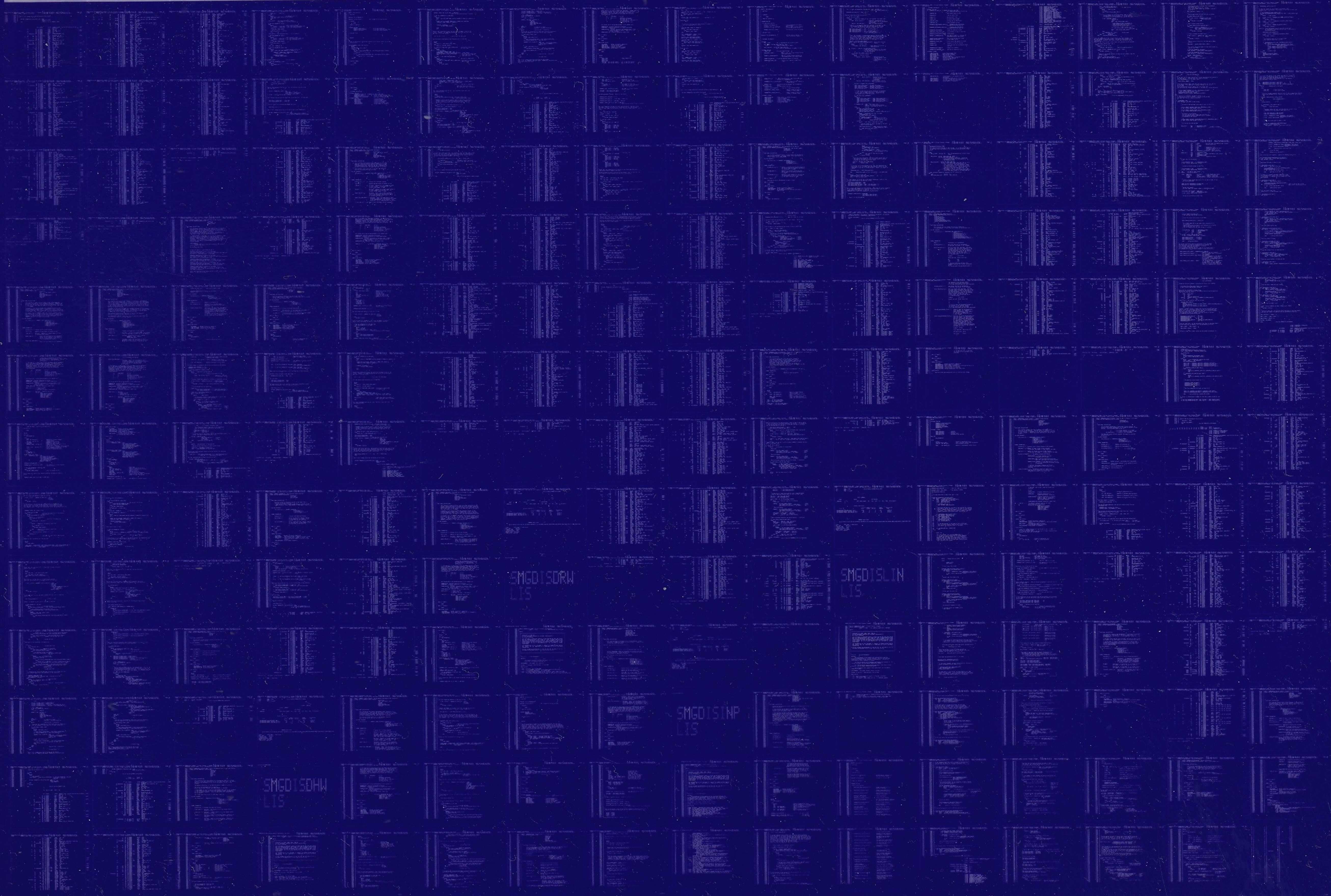
```
                       4A   A6            67  B1 001CF         CMPW     (R7), 74(DCB)                          ; 1262
                                          27  13 001D3         BEQL     23$                                    ;
                                          01  DD 001D5         PUSHL    #1                                     ; 1264
                                          01  DD 001D7         PUSHL    #1                                     ;
                            7E       06   A6  3C 001D9         MOVZWL   6(DCB), -(SP)                          ;
                            50       4A   A6  3C 001DD         MOVZWL   74(DCB), R0                            ;
                            51       48   A6  3C 001E1         MOVZWL   72(DCB), R1                            ;
                            50            51  C2 001E5         SUBL2    R1, R0                                 ;
                                     01   A0  9F 001E8         PUSHAB   1(R0)                                  ;
                            7E       04   A6  3C 001EB         MOVZWL   4(DCB), -(SP)                          ;
                            7E       48   A6  3C 001EF         MOVZWL   72(DCB), -(SP)                         ;
                                          56  DD 001F3         PUSHL    DCB                                    ;
                 00000000G  00            07  FB 001F5         CALLS    #7, SMG$$SCROLL_AREA                   ;
                       4A   A6            67  B1 001FC 23$:    CMPW     (R7), 74(DCB)                          ; 1265
                                          6D  1B 00200         BLEQU    30$                                    ;
                                          6F  11 00202         BRB      31$                                    ; 1268
                            50       0C   BC  D0 00204 24$:    MOVL     @LINE_ADV, R0                          ; 1279
                                          6D  15 00208         BLEQ     32$                                    ;
                            51            67  3C 0020A         MOVZWL   (R7), R1                               ; 1282
                            51            50  C0 0020D         ADDL2    R0, R1                                 ;
        51       4A   A6    10            00  ED 00210         CMPZV    #0, #16, 74(DCB), R1                   ; 1283
                                          05  19 00216         BLSS     26$                                    ;
                            67            50  A0 00218         ADDW2    R0, (R7)                               ; 1287
                                          5A  11 0021B 25$:    BRB      32$                                    ; 1286
                                          6E  D4 0021D 26$:    CLRL     SCROLL_FLAG                            ; 1290
                            14       08   BE  E9 0021F         BLBC     @8(SP), 28$                            ;
                       4A   A6            67  B1 00223         CMPW     (R7), 74(DCB)                          ;
                                          05  12 00227         BNEQ     27$                                    ;
                            6E            01  D0 00229         MOVL     #1, SCROLL_FLAG                        ;
                                          09  11 0022C         BRB      28$                                    ;
                       48   A6            67  B1 0022E 27$:    CMPW     (R7), 72(DCB)                          ;
                                          03  12 00232         BNEQ     28$                                    ;
                            6E            02  D0 00234         MOVL     #2, SCROLL_FLAG                        ;
                            01            6E  D1 00237 28$:    CMPL     SCROLL_FLAG, #1                        ; 1291
                                          2D  12 0023A         BNEQ     29$                                    ;
                       4A   A6            67  B1 0023C         CMPW     (R7), 74(DCB)                          ; 1293
                                          27  13 00240         BEQL     29$                                    ;
                                          50  DD 00242         PUSHL    R0                                     ; 1295
                                          01  DD 00244         PUSHL    #1                                     ;
                            7E       06   A6  3C 00246         MOVZWL   6(DCB), -(SP)                          ;
                            50       4A   A6  3C 0024A         MOVZWL   74(DCB), R0                            ;
                            51       48   A6  3C 0024E         MOVZWL   72(DCB), R1                            ;
                            50            51  C2 00252         SUBL2    R1, R0                                 ;
                                     01   A0  9F 00255         PUSHAB   1(R0)                                  ;
                            7E       04   A6  3C 00258         MOVZWL   4(DCB), -(SP)                          ;
                            7E       48   A6  3C 0025C         MOVZWL   72(DCB), -(SP)                         ;
                                          56  DD 00260         PUSHL    DCB                                    ;
                 00000000G  00            07  FB 00262         CALLS    #7, SMG$$SCROLL_AREA                   ;
                       4A   A6            67  B1 00269 29$:    CMPW     (R7), 74(DCB)                          ; 1296
                                          04  1A 0026D         BGTRU    31$                                    ;
                            67       4A   A6  B0 0026F 30$:    MOVW     74(DCB), (R7)                          ; 1298
                       08   BE            01  88 00273 31$:    BISB2    #1, @8(SP)                             ; 1299
                                     1C   AE  D5 00277 32$:    TSTL     WRAPPED_CHARS                          ; 1310
                                          03  12 0027A         BNEQ     34$                                    ;
                                        009F  31 0027C 33$:    BRW      41$                                    ;
                            03            6C  91 0027F 34$:    CMPB     (AP), #3                               ; 1311
                                          0A  1F 00282         BLSSU    35$                                    ;
```

```
                           0C    AC   D5 00284          TSTL    12(AP)                                  ;
                                 05   13 00287          BEQL    35$                                     ;
                           0C    BC   D5 00289          TSTL    @LINE_ADV                               ;
                                 0A   14 0028C          BGTR    36$                                     ;
                  03             6C   91 0028E 35$:     CMPB    (AP), #3                                ; 1312
                                 05   1F 00291          BLSSU   36$                                     ;
                           0C    AC   D5 00293          TSTL    12(AP)                                  ;
                                 E4   12 00296          BNEQ    33$                                     ;
                  06             6C   91 00298 36$:     CMPB    (AP), #6                                ; 1315
                                 4B   1F 0029B          BLSSU   38$                                     ;
                           18    AC   D5 0029D          TSTL    24(AP)                                  ;
                                 46   13 002A0          BEQL    38$                                     ;
                           18    BC   D5 002A2          TSTL    @WRAP_FLAG                              ; 1316
                                 41   13 002A5          BEQL    38$                                     ;
             26   AE     010E    8F   B0 002A7          MOVW    #270, STR_DESC+2                        ; 1324
 24   AE     04   AE       14    AE   A3 002AD          SUBW3   HALF_NO_COLS, STR_LEN, STR_DESC         ; 1325
 28   AE     18   AE       14    AE   C1 002B4          ADDL3   HALF_NO_COLS, STR_ADDR, STR_DESC+4      ; 1326
             20   AE             01   D0 002BB          MOVL    #1, C_SET                               ; 1328
                  07             6C   91 002BF          CMPB    (AP), #7                                ; 1329
                                 0A   1F 002C2          BLSSU   37$                                     ;
                           1C    AC   D5 002C4          TSTL    28(AP)                                  ;
                                 05   13 002C7          BEQL    37$                                     ;
             20   AE       1C    BC   D0 002C9          MOVL    @CHAR_SET, C_SET                        ; 1331
                           20    AE   9F 002CE 37$:     PUSHAB  C_SET                                   ; 1333
                  7E       14    AC   7D 002D1          MOVQ    RENDITION_COMPLEMENT, -(SP)             ; 1337
                  7E       0C    AC   7D 002D5          MOVQ    LINE_ADV, -(SP)                         ; 1335
                           38    AE   9F 002D9          PUSHAB  STR_DESC                                ; 1333
                           04    AC   DD 002DC          PUSHL   DISPLAY_ID                              ;
           FD1C   CF             07   FB 002DF          CALLS   #7, SMG$PUT_LINE_WIDE                   ;
                                 52   D4 002E4          CLRL    DONE                                    ; 1340
                                 68   11 002E6          BRB     44$                                     ; 1341
      31     2F   A6             01   E1 002E8 38$:     BBC     #1, 47(DCB), 41$                        ; 1349
                  02       08    BE   E8 002ED          BLBS    @8(SP), 39$                             ; 1352
                                 67   B7 002F1          DECW    (R7)                                    ; 1354
             2A   A6       14    AE   B0 002F3 39$:     MOVW    HALF_NO_COLS, 42(DCB)                   ; 1355
                                 01   DD 002F8          PUSHL   #1                                      ; 1357
                         FD01    CF   9F 002FA          PUSHAB  P.AAA                                   ; 1359
                                 01   DD 002FE          PUSHL   #1                                      ; 1357
      50     1C   AE 00000040    8F   C1 00300          ADDL3   #64, REND_CODE, R0                      ; 1358
                                 50   DD 00309          PUSHL   R0                                      ;
                                 56   DD 0030B          PUSHL   DCB                                     ; 1357
      00000000G   00             05   FB 0030D          CALLS   #5, SMG$$PUT_TEXT_TO_BUFFER             ;
                  02       08    BE   E8 00314          BLBS    @8(SP), 40$                             ; 1362
                                 67   B6 00318          INCW    (R7)                                    ; 1364
             2A   A6             01   B0 0031A 40$:     MOVW    #1, 42(DCB)                             ; 1366
                  52             01   D0 0031E 41$:     MOVL    #1, DONE                                ; 1373
                  2C             52   E9 00321          BLBC    DONE, 44$                               ; 1381
                  03             6C   91 00324          CMPB    (AP), #3                                ; 1387
                                 10   1F 00327          BLSSU   42$                                     ;
                           0C    AC   D5 00329          TSTL    12(AP)                                  ;
                                 0B   13 0032C          BEQL    42$                                     ;
                  50       0C    BC   D0 0032E          MOVL    @LINE_ADV, R0                           ; 1388
                                 07   18 00332          BGEQ    43$                                     ;
                  50             50   CE 00334          MNEGL   R0, R0                                  ;
                                 02   11 00337          BRB     43$                                     ;
                                 50   D4 00339 42$:     CLRL    R0                                      ; 1387
                  51             67   3C 0033B 43$:     MOVZWL  (R7), R1                                ;
```

```
              50              51          50  C3 0033E          SUBL3   R0, R1, LINE_CHANGED                              ;
                                          50  DD 00342          PUSHL   LINE_CHANGED                                      ; 1391
                                          12  DD 00344          PUSHL   #18                                               ; 1389
                                          56  DD 00346          PUSHL   DCB                                               ;
                 00000000G    00          03  FB 00348          CALLS   #3, SMG$$CHECK_FOR_OUTPUT_DCB                     ;
                                              04 0034F          RET                                                       ; 1394
                              50          01  D0 00350 44$:     MOVL    #1, R0                                            ;
                                              04 00353          RET                                                       ; 1396

; Routine Size:  852 bytes,    Routine Base:  _SMG$CODE + 048D


; 1140          1397  1 !<BLF/PAGE>
; 1142          1398  1 END                                      ! End of module SMG$DISPLAY_DHDW
; 1143          1399  1
; 1144          1400  0 ELUDOM



;                                  PSECT SUMMARY
;
;       Name                     Bytes                       Attributes
;
;  _SMG$CODE                         2017  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        16         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        30         6           38        00:00.3



;                                   COMMAND QUALIFIERS

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

; Size:          2015 code + 2 data bytes
; Run Time:         00:45.6
; Elapsed Time:     02:08.8
; Lines/CPU Min:     1841
; Lexemes/CPU-Min: 17479
; Memory Used:  339 pages
; Compilation Complete
```

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

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY


SMGDISDRW
LIS

SMGDISLIN
LIS

SMGDISINP
LIS

SMGDISDHW
LIS