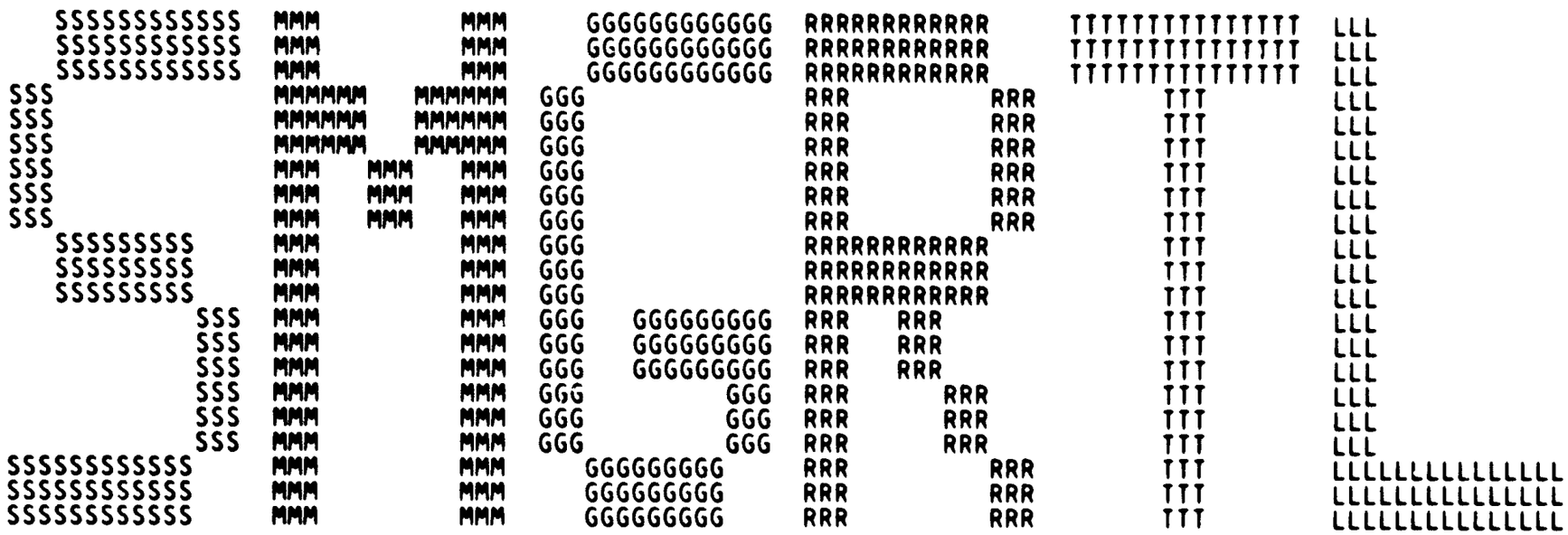

_$2

Val

001
001
001
001
001
001
001
001
001
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF

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

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

SMG
1-0

```
;    1          0001  0 %TITLE 'SMG$DISPLAY_CHANGE - Virtual Display -- Changes to Displays'
;    2          0002  0 MODULE SMG$DISPLAY_CHANGE (
;    3          0003  0                 IDENT = '1-048' ! File: SMGDISCHA.B32 Edit: PLL1048
;    4          0004  0                 ) =
;    5          0005  1 BEGIN
;    6          0006  1 !
;    7          0007  1 !*****************************************************************************
;    8          0008  1 !*                                                                           *
;    9          0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;   10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;   11          0011  1 !*  ALL RIGHTS RESERVED.                                                      *
;   12          0012  1 !*                                                                           *
;   13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;   14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;   15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;   16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;   17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;   18          0018  1 !*  TRANSFERRED.                                                              *
;   19          0019  1 !*                                                                           *
;   20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;   21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;   22          0022  1 !*  CORPORATION.                                                              *
;   23          0023  1 !*                                                                           *
;   24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;   25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;   26          0026  1 !*                                                                           *
;   27          0027  1 !*                                                                           *
;   28          0028  1 !*****************************************************************************
;   29          0029  1 !
;   30          0030  1
;   31          0031  1 !++
;   32          0032  1 ! FACILITY:     Screen Management
;   33          0033  1 !
;   34          0034  1 ! ABSTRACT:
;   35          0035  1 !       The procedures in this module update and maintain the contents
;   36          0036  1 ! of the in-memory representations of the virtual displays.  The data
;   37          0037  1 ! areas themselves are allocated/deallocated, pasted/upasted, etc.                ; R
;   38          0038  1 ! by the procedures in module SMG$DISPLAY_LINKS.  Output from these
;   39          0039  1 ! virtual displays is handled by procedures in SMG$DISPLAY_OUTPUT.
;   40          0040  1 ! The module SMG$DISPLAY_INPUT contains the routines that support                ;
;   41          0041  1 ! input operations on displays.
;   42          0042  1 !
;   43          0043  1 !
;   44          0044  1 !
;   45          0045  1 ! ENVIRONMENT:  User mode, Shared library routines.
;   46          0046  1 !
;   47          0047  1 ! AUTHOR: R. Reichert, CREATION DATE: 26-Jan-1983
;   48          0048  1 !
;   49          0049  1 ! MODIFIED BY:
;   50          0050  1 !
;   51          0051  1 ! 1-048 - Change error messages by SMG$SET_CURSOR_REL. STAN 3-Jun-1984.
;   52          0052  1 ! 1-047 - Turn off the scrolling bit whenever sequential output ends, ie.
;   53          0053  1 !         whenever the cursor may be randomly positioned.  PLL 3-May-1984
;   54          0054  1 ! 1-001 - Original.  Skeleton for future code.  RKR 26-Jan-1983
;   55          0055  1 !--
```

```
;    57          0056  1 %SBTTL 'Declarations'
;    58          0057  1 !
;    59          0058  1 ! SWITCHES:
;    60          0059  1 !
;    61          0060  1 !
;    62          0061  1 !
;    63          0062  1 ! LINKAGES:
;    64          0063  1 !
;    65          0064  1 !       NONE
;    66          0065  1 !
;    67          0066  1 ! TABLE OF CONTENTS:
;    68          0067  1 !
;    69          0068  1
;    70          0069  1 FORWARD ROUTINE
;    71          0070  1
;    72          0071  1 ! Public entry points:
;    73          0072  1
;    74          0073  1     SMG$CHANGE_RENDITION,           ! Change video attributes of
;    75          0074  1                                     ! something already in a virtual
;    76          0075  1                                     ! display
;    77          0076  1
;    78          0077  1     SMG$CURSOR_COLUMN,              ! Function to return current
;    79          0078  1                                     ! cursor col. within virt. disp.
;    80          0079  1
;    81          0080  1     SMG$CURSOR_ROW,                 ! Function to return current
;    82          0081  1                                     ! cursor row within virt. disp.
;    83          0082  1
;    84          0083  1     SMG$DELETE_CHARS,               ! Delete characters from within
;    85          0084  1                                     ! a line in a virtual display.
;    86          0085  1
;    87          0086  1     SMG$DELETE_LINE,                ! Delete line in a display &
;    88          0087  1                                     ! scroll remaining lines
;    89          0088  1
;    90          0089  1     SMG$ERASE_CHARS,                ! Erase characters in a display
;    91          0090  1                                     ! w/out moving remaining text
;    92          0091  1
;    93          0092  1     SMG$ERASE_DISPLAY,              ! Erase virtual display
;    94          0093  1
;    95          0094  1     SMG$ERASE_LINE,                 ! Erase from position to end
;    96          0095  1                                     ! of line
;    97          0096  1
;    98          0097  1     SMG$HOME_CURSOR,                ! Move cursor to one of the 4
;    99          0098  1                                     ! corners of the virtual display
;   100          0099  1
;   101          0100  1     SMG$INSERT_CHARS,               ! Insert characters in a line
;   102          0101  1                                     ! of a virtual display.
;   103          0102  1
;   104          0103  1     SMG$INSERT_LINE,                ! Insert line in a virtual
;   105          0104  1                                     ! display
;   106          0105  1
;   107          0106  1     SMG$PUT_CHARS,                  ! Put text to display without
;   108          0107  1                                     ! additonal cursor motion.
;   109          0108  1
;   110          0109  1     SMG$PUT_LINE,                   ! Put text line to display with
;   111          0110  1                                     ! optional additonal cursor
;   112          0111  1                                     ! motion.
;   113          0112  1
```

```
:  114          0113  1     SMG$READ_FROM_DISPLAY,          ! Read text back from virtual
:  115          0114  1                                     ! display
:  116          0115  1
:  117          0116  1     SMG$PUT_WITH_SCROLL,            ! Scroll a virtual display and
:  118          0117  1                                     ! place callers text into opened
:  119          0118  1                                     ! area.
:  120          0119  1
:  121          0120  1     SMG$RETURN_CURSOR_POS,          ! Returns current cursor
:  122          0121  1                                     ! position (row/column).
:  123          0122  1
:  124          0123  1     SMG$SCROLL_DISPLAY_AREA,        ! Scroll a rectangular area
:  125          0124  1
:  126          0125  1     SMG$SET_CURSOR_ABS,             ! Move cursor to absolute
:  127          0126  1                                     ! position specified.
:  128          0127  1
:  129          0128  1     SMG$SET_CURSOR_REL;             ! Move cursor relative to its
:  130          0129  1                                     ! current position.
:  131          0130  1
:  132          0131  1
:  133          0132  1 ! INCLUDE FILES
:  134          0133  1 !
:  135          0134  1
:  136          0135  1 REQUIRE 'RTLIN:SMGPROLOG';          ! defines psects, macros, tcb,
:  137          0213  1                                     !  wcb, & terminal symbols
:  138          0214  1 REQUIRE 'RTLIN:STRLNK';             ! JSB linkage
:  139          0399  1
:  140          0400  1 !
:  141          0401  1 ! EXTERNAL REFERENCES
:  142          0402  1 !
:  143          0403  1 EXTERNAL ROUTINE
:  144          0404  1     LIB$ANALYZE_SDESC_R2 : LIB$ANALYZE_SDESC_JSB_LINK,
:  145          0405  1
:  146          0406  1     LIB$SCOPY_R_DX6 : STRING_JSB,       ! String copier
:  147          0407  1
:  148          0408  1     SMG$$CHECK_FOR_OUTPUT_DCB,  ! Force output if now is the time
:  149          0409  1
:  150          0410  1     SMG$$PUT_TEXT_TO_BUFFER,    ! Move callers text to virtual
:  151          0411  1                                 ! display text buffer checking
:  152          0412  1                                 ! for characters that affect
:  153          0413  1                                 ! where characters land in
:  154          0414  1                                 ! buffer.
:  155          0415  1
:  156          0416  1     SMG$$SCROLL_AREA;           ! Scroll a rectangular area
:  157          0417  1
:  158          0418  1 EXTERNAL
:  159          0419  1     SMG$_FATERRLIB,     ! Fatal error in SMG$ -- internal consistancy
:  160          0420  1                         ! check failed.
:  161          0421  1     SMG$_INVARG,        ! Invalid argument
:  162          0422  1     SMG$_INVCOL,        ! Invalid column number
:  163          0423  1     SMG$_INVDIS_ID,     ! Invalid virtual display id
:  164          0424  1     SMG$_INVPAS_ID,     ! Invalid pasteboard id
:  165          0425  1     SMG$_INVROW,        ! Invalid row number
:  166          0426  1     SMG$_NO_CHADIS,     ! No change to virtual display
:  167          0427  1     SMG$_WRONUMARG;     ! Wrong number of arguments
:  168          0428  1
:  169          0429  1 !+
:  170          0430  1 ! The following macro determines whether scrolling up, down, or neither
```

```
;  171          0431  1 ! should occur.
;  172          0432  1 !-
;  173        M 0433  1 MACRO $SMG$SET_SCROLLING (SWITCH) =
;  174        M 0434  1     BEGIN
;  175        M 0435  1     SWITCH = 0;          ! init scroll to off
;  176        M 0436  1     IF .DCB [DCB_V_FULL] NEQ 0
;  177        M 0437  1     THEN                 ! display is full, check other conditions
;  178        M 0438  1         BEGIN
;  179        M 0439  1         IF .DCB [DCB_W_CURSOR_ROW] EQL .DCB [DCB_W_BOTTOM_OF_SCRREG]
;  180        M 0440  1         THEN
;  181        M 0441  1             SWITCH = 1  ! scroll up
;  182        M 0442  1         ELSE
;  183        M 0443  1             IF .DCB [DCB_W_CURSOR_ROW] EQL .DCB [DCB_W_TOP_OF_SCRREG]
;  184        M 0444  1             THEN
;  185        M 0445  1                 SWITCH = 2;  ! scroll down
;  186        M 0446  1         END;
;  187        M 0447  1     END;%;
;  188          0448  1
;  189          0449  1 !<BLF/PAGE>
```

```
;  191          0450  1 %SBTTL 'SMG$CHANGE_RENDITION - Change Rendition'
;  192          0451  1 GLOBAL ROUTINE SMG$CHANGE_RENDITION (
;  193          0452  1                               DISPLAY_ID,
;  194          0453  1                               START_ROW,
;  195          0454  1                               START_COL,
;  196          0455  1                               NO_ROW,
;  197          0456  1                               NO_COL,
;  198          0457  1                               RENDITION_SET,
;  199          0458  1                               RENDITION_COMPLEMENT
;  200          0459  1                                     ) =
;  201          0460  1 !++
;  202          0461  1 ! FUNCTIONAL DESCRIPTION:
;  203          0462  1 !
;  204          0463  1 !       This routine changes the video rendition of a rectangular block of
;  205          0464  1 !       text already in the specified virtual display.
;  206          0465  1 !       Use of this routine makes it easy, for example, to redisplay a
;  207          0466  1 !       particular column in reverse video.
;  208          0467  1 !
;  209          0468  1 ! CALLING SEQUENCE:
;  210          0469  1 !
;  211          0470  1 !       ret_status.wlc.v = SMG$CHANGE_RENDITION (
;  212          0471  1 !                                  DISPLAY_ID.rl.r,
;  213          0472  1 !                                  START_ROW.rl.r,
;  214          0473  1 !                                  START_COL.rl.r,
;  215          0474  1 !                                  NO_ROW.rl.r,
;  216          0475  1 !                                  NO_COL.rl.r,
;  217          0476  1 !                                  [,RENDITION_SET.rl.r]
;  218          0477  1                                   [,RENDITION_COMPLEMENT.rl.r])
;  219          0478  1 !
;  220          0479  1 ! FORMAL PARAMETERS:
;  221          0480  1 !
;  222          0481  1 !       DISPLAY_ID.rl.r         Display id of desired virtual display.
;  223          0482  1 !
;  224          0483  1 !       START_ROW.rl.r          Starting row position to receive new
;  225          0484  1 !                               rendition
;  226          0485  1 !
;  227          0486  1 !       START_COL.rl.r          Starting column position to receive
;  228          0487  1 !                               new rendition.
;  229          0488  1 !
;  230          0489  1 !       NO_ROW.rl.r             No. of rows to receive new rendition
;  231          0490  1 !
;  232          0491  1 !       NO_COL.rl.r             No. of columns to receive new
;  233          0492  1 !                               rendition
;  234          0493  1 !
;  235          0494  1 !       RENDITION_SET.rl.r      Each 1 bit attribute in this parameter
;  236          0495  1 !                               causes the corresponding attribute to
;  237          0496  1 !                               be set in the display.  (See below for
;  238          0497  1 !                               list of settable attributes.)
;  239          0498  1 !
;  240          0499  1 !       RENDITION_COMPLEMENT.rl.r
;  241          0500  1 !                               Each 1 bit attribute in this parameter
;  242          0501  1 !                               causes the corresponding attribute to
;  243          0502  1 !                               be complemented in the display.  (See
;  244          0503  1 !                               below for list of complementable
;  245          0504  1 !                               attributes.)
;  246          0505  1 !
;  247          0506  1 !       If the same bit is specified in both the RENDITION_SET parameter
```

```
;  248          0507  1 !       and in the RENDITION_COMPLEMENT parameter, the application is
;  249          0508  1 !       RENDITION_SET followed by RENDITION complement.  Using these two
;  250          0509  1 !       parameters together the caller can exercise arbitrary and
;  251          0510  1 !       independent control over each attribute on a single call.  On an
;  252          0511  1 !       attribute by attribute basis he can cause the following
;  253          0512  1 !       transformations:
;  254          0513  1 !
;  255          0514  1 !               SET     COMPLEMENT      Action
;  256          0515  1 !               ---     ----------      ------
;  257          0516  1 !               0       0               Attribute unchanged.
;  258          0517  1 !               1       0               Attribute set to "on"
;  259          0518  1 !               0       1               Attribute set to complement of
;  260          0519  1 !                                       current setting.
;  261          0520  1 !               1       1               Attribute set to "off".
;  262          0521  1 !
;  263          0522  1 !
;  264          0523  1 !       Attributes which can be manipulated in this manner are:
;  265          0524  1 !
;  266          0525  1 !       SMG$M_BLINK  displays characters blinking.
;  267          0526  1 !       SMG$M_BOLD  displays characters in higher-than-normal
;  268          0527  1 !                   intensity.
;  269          0528  1 !       SMG$M_REVERSE  displays characters in reverse video -- that is,
;  270          0529  1 !                   using the opposite default rendition of the
;  271          0530  1 !                   virtual display.
;  272          0531  1 !       SMG$M_UNDERLINE  displays characters underlined.
;  273          0532  1 !
;  274          0533  1 ! IMPLICIT INPUTS:
;  275          0534  1 !
;  276          0535  1 !       NONE
;  277          0536  1 !
;  278          0537  1 ! IMPLICIT OUTPUTS:
;  279          0538  1 !
;  280          0539  1 !       NONE
;  281          0540  1 !
;  282          0541  1 ! COMPLETION STATUS:
;  283          0542  1 !
;  284          0543  1 !       SS$_NORMAL        Normal successful completion
;  285          0544  1 !       SMG$_INVROW       Invalid start row -- outside virtual display.
;  286          0545  1 !       SMG$_INVCOL       Invalid start column -- outside virtual display.
;  287          0546  1 !       SMG$_INVDIS_ID    Invalid display id.
;  288          0547  1 !       SMG$_INVARG       Invalid no. of rows -- outside virtual display
;  289          0548  1 !                      or Invalid no. of columns -- outside virtual display.
;  290          0549  1 !                      or Unrecognized action code.
;  291          0550  1 !                      or Unrecognized rendition code.
;  292          0551  1 !       SMG$_WRONUMARG  Wrong number of arguments.
;  293          0552  1 !
;  294          0553  1 ! SIDE EFFECTS:
;  295          0554  1 !
;  296          0555  1 !       NONE
;  297          0556  1 !--
;  298          0557  2     BEGIN
;  299          0558  2
;  300          0559  2     LOCAL
;  301          0560  2         START_INDEX,            ! Byte offset into attribute buffer
;  302          0561  2
;  303          0562  2         DCB : REF $DCB_DECL,    ! Addr of display control block
;  304          0563  2
```

```
;   305           0564  2          REND_CODE,                  ! Video rendition -- set up by
;   306           0565  2                                      ! $SMG$SET_REND_CODE
;   307           0566  2
;   308           0567  2          ATTR_BUF : REF VECTOR [,BYTE];  ! Addr of attribute buffer for
;   309           0568  2                                         ! this virtual display.
;   310           0569  2
;   311           0570  2      LITERAL
;   312           0571  2          K_SET_ARG = 6,
;   313           0572  2          K_COMP_ARG = 7;
;   314           0573  2
;   315           0574  2      $SMG$VALIDATE_ARGCOUNT (5,7);
;   316           0575  2
;   317           0576  2      $SMG$GET_DCB ( .DISPLAY_ID, DCB);   ! Get addr of display control
;   318           0577  2                                         ! block
;   319           0578  2
;   320           0579  2  !+
;   321           0580  2  ! Validate arguments, make sure row & column are within the display.
;   322           0581  2  !-
;   323           0582  2
;   324           0583  2      $SMG$VALIDATE_ROW_COL (..START_ROW, ..START_COL);
;   325           0584  2
;   326           0585  2      IF ..NO_COL LEQ 0 OR
;   327           0586  2         ..NO_ROW LEQ 0
;   328           0587  2      THEN
;   329           0588  2          RETURN (SMG$_NO_CHADIS);
;   330           0589  2
;   331         P 0590  2      $SMG$VALIDATE_ROW_COL (..START_ROW + ..NO_ROW - 1,
;   332           0591  2                             ..START_COL + ..NO_COL - 1);
;   333           0592  2
;   334           0593  2      ATTR_BUF = .DCB [DCB_A_ATTR_BUF];
;   335           0594  2                                      ! addr of attribute buffer
;   336           0595  2
;   337           0596  2      $SMG$SET_REND_CODE (K_SET_ARG, K_COMP_ARG);
;   338           0597  2                                      ! set rend_code w/default or caller's args
;   339           0598  2
;   340           0599  2      !+
;   341           0600  2      ! Set the rendition bytes in the attribute buffer on a line by line
;   342           0601  2      ! basis, in case the change does not include the entire line.
;   343           0602  2      ! Be careful to preserve the other bits in the rendition byte that have
;   344           0603  2      ! nothing to do with rendition.
;   345           0604  2      !-
;   346           0605  2
;   347           0606  2      START_INDEX = $SMG$LINEAR (..START_ROW, ..START_COL);
;   348           0607  2      INCR LINE_COUNT FROM 1 TO ..NO_ROW
;   349           0608  2      DO
;   350           0609  3          BEGIN   ! For each row
;   351           0610  4          INCR COL_COUNT FROM 0 TO (..NO_COL - 1)
;   352           0611  3          DO
;   353           0612  4              BEGIN       ! For each column
;   354           0613  4
;   355           0614  4              ATTR_BUF [.START_INDEX + .COL_COUNT ] =
;   356           0615  5                (.ATTR_BUF [.START_INDEX + .COL_COUNT]
;   357           0616  5                           AND
;   358           0617  5                ( NOT (SMG$M_BLINK+SMG$M_BOLD+SMG$M_REVERSE+SMG$M_UNDERLINE)) )
;   359           0618  4                           OR
;   360           0619  4                   .REND_CODE ;
;   361           0620  4
```

```
;  362          0621  3              END;        ! For each column
;  363          0622  3
;  364          0623  3          START_INDEX = .START_INDEX + .DCB [DCB_W_NO_COLS]; ! To next row start
;  365          0624  2          END;    ! For each row
;  366          0625  2
;  367          0626  2      !+
;  368          0627  2      ! See if this change should be reflected on the screen right away.
;  369          0628  2      !-
;  370          0629  2
;  371          0630  3      RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
;  372          0631  2                                         SMG$C_CHANGE_RENDITION));
;  373          0632  1      END;                           ! End of routine SMG$CHANGE_RENDITION


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

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

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

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

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

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

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

```
;  376          0634  1 %SBTTL 'SMG$CURSOR_COLUMN - Return current cursor column position'
;  377          0635  1 GLOBAL ROUTINE SMG$CURSOR_COLUMN ( DISPLAY_ID ) =
;  378          0636  1 !++
;  379          0637  1 ! FUNCTIONAL DESCRIPTION:
;  380          0638  1 !
;  381          0639  1 !       This functions returns the current cursor column position in the
;  382          0640  1 !       specified virtual display.
;  383          0641  1 !
;  384          0642  1 !       If the specified virtual display id is invalid, SMG$_INVDIS_ID
;  385          0643  1 !       is signaled.
;  386          0644  1 !
;  387          0645  1 ! CALLING SEQUENCE:
;  388          0646  1 !
;  389          0647  1 !       CURSOR_COLUMN.wl.r = SMG$CURSOR_COLUMN ( DISPLAY_ID.rl.r )
;  390          0648  1 !
;  391          0649  1 ! FORMAL PARAMETERS:
;  392          0650  1 !
;  393          0651  1 !       DISPLAY_ID.rl.r Display id of desired display.
;  394          0652  1 !
;  395          0653  1 ! IMPLICIT INPUTS:
;  396          0654  1 !
;  397          0655  1 !       NONE
;  398          0656  1 !
;  399          0657  1 ! IMPLICIT OUTPUTS:
;  400          0658  1 !
;  401          0659  1 !       NONE
;  402          0660  1 !
;  403          0661  1 ! ROUTINE VALUE:
;  404          0662  1 !
;  405          0663  1 !       CURSOR_COLUMN.wl.v      The current cursor column number in the
;  406          0664  1 !                               display.
;  407          0665  1 !
;  408          0666  1 ! SIDE EFFECTS:
;  409          0667  1 !
;  410          0668  1 !       SMG$_INVDIS_ID  will be signaled if DISPLAY_ID in invalid.
;  411          0669  1 !       SMG$_WRONUMARG  will be signaled if DISPLAY_ID is omitted.
;  412          0670  1 !--
;  413          0671  2     BEGIN
;  414          0672  2     BUILTIN
;  415          0673  2         NULLPARAMETER;
;  416          0674  2
;  417          0675  2     LOCAL
;  418          0676  2         DCB : REF $DCB_DECL;    ! Addr. of display control block
;  419          0677  2
;  420          0678  2     IF NULLPARAMETER(DISPLAY_ID)
;  421          0679  2     THEN
;  422          0680  2         SIGNAL_STOP ( SMG$_WRONUMARG);  ! Display id has to be there
;  423          0681  2
;  424          0682  2 !+
;  425          0683  2 ! The next few lines of code validate the supplied device id.  It
;  426          0684  2 ! does precisely what $SMG$GET_DCB  does except it signals where that
;  427          0685  2 ! macro would return a status.
;  428          0686  2 !-
;  429          0687  2     DCB = ..DISPLAY_ID;                 ! assume it is correct
;  430          0688  2
;  431          0689  2     IF .DCB [DCB_L_DID] NEQ ..DISPLAY_ID
;  432          0690  2     THEN
```

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


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

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

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

```
;  446          0703  1 %SBTTL 'SMG$CURSOR_ROW - Return current cursor row position'
;  447          0704  1 GLOBAL ROUTINE SMG$CURSOR_ROW ( DISPLAY_ID ) =
;  448          0705  1 !++
;  449          0706  1 ! FUNCTIONAL DESCRIPTION:
;  450          0707  1 !
;  451          0708  1 !       This functions returns the current cursor row position in the
;  452          0709  1 !       specified virtual display.
;  453          0710  1 !
;  454          0711  1 !       If the specified virtual display id is invalid, SMG$_INVDIS_ID
;  455          0712  1 !       is signaled.
;  456          0713  1 !
;  457          0714  1 ! CALLING SEQUENCE:
;  458          0715  1 !
;  459          0716  1 !       CURSOR_ROW.wl.r = SMG$CURSOR_ROW ( DISPLAY_ID.rl.r )
;  460          0717  1 !
;  461          0718  1 ! FORMAL PARAMETERS:
;  462          0719  1 !
;  463          0720  1 !       DISPLAY_ID.rl.r Display id of desired display.
;  464          0721  1 !
;  465          0722  1 ! IMPLICIT INPUTS:
;  466          0723  1 !
;  467          0724  1 !       NONE
;  468          0725  1 !
;  469          0726  1 ! IMPLICIT OUTPUTS:
;  470          0727  1 !
;  471          0728  1 !       NONE
;  472          0729  1 !
;  473          0730  1 ! ROUTINE VALUE:
;  474          0731  1 !
;  475          0732  1 !       CURSOR_ROW.wl.v The current cursor row number in the display.
;  476          0733  1 !
;  477          0734  1 ! SIDE EFFECTS:
;  478          0735  1 !
;  479          0736  1 !       SMG$_INVDIS_ID  will be signaled if DISPLAY_ID in invalid.
;  480          0737  1 !       SMG$_WRONUMARG  will be signaled if DISPLAY_ID is omitted.
;  481          0738  1 !--
;  482          0739  2     BEGIN
;  483          0740  2     BUILTIN
;  484          0741  2         NULLPARAMETER;
;  485          0742  2
;  486          0743  2     LOCAL
;  487          0744  2         DCB : REF $DCB_DECL;    ! Addr. of display control block
;  488          0745  2
;  489          0746  2     IF NULLPARAMETER(DISPLAY_ID)
;  490          0747  2     THEN
;  491          0748  2         SIGNAL_STOP ( SMG$_WRONUMARG);  ! Display id has to be there
;  492          0749  2
;  493          0750  2 !+
;  494          0751  2 ! The next few lines of code validate the supplied device id.  It
;  495          0752  2 ! does precisely what $SMG$GET_DCB  does except it signals where that
;  496          0753  2 ! macro would return a status.
;  497          0754  2 !-
;  498          0755  2     DCB = ..DISPLAY_ID;                 ! assume it is correct
;  499          0756  2
;  500          0757  2     IF .DCB [DCB_L_DID] NEQ ..DISPLAY_ID
;  501          0758  2     THEN
;  502          0759  2         SIGNAL_STOP (SMG$_INVDIS_ID);   ! Not pointing to one of our
```

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


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

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

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

```
;  515          0771  1 %SBTTL 'SMG$DELETE_CHARS - Delete Characters'
;  516          0772  1 GLOBAL ROUTINE SMG$DELETE_CHARS (
;  517          0773  1                                  DISPLAY_ID,
;  518          0774  1                                  NUM_CHAR,
;  519          0775  1                                  ROW,
;  520          0776  1                                  COL
;  521          0777  1                                 ) =
;  522          0778  1 !++
;  523          0779  1 ! FUNCTIONAL   CRIPTION:
;  524          0780  1 !
;  525          0781  1 !       This  ..ut ne deletes the number of characters specified starting
;  526          0782  1 !       at the row/column specified.
;  527          0783  1 !
;  528          0784  1 !       Remaining characters on the line to the right of the deleted
;  529          0785  1 !       characters are shifted to the left to occupy the vacated
;  530          0786  1 !       space(s).
;  531          0787  1 !
;  532          0788  1 !       The internal display cursor position is left at the specified
;  533          0789  1 !       start position after the operation.
;  534          0790  1 !
;  535          0791  1 ! CALLING SEQUENCE:
;  536          0792  1 !
;  537          0793  1 !
;  538          0794  1 !       ret_status.wlc.v = SMG$DELETE_CHARS (
;  539          0795  1 !                                  DISPLAY_ID.rl.r,
;  540          0796  1 !                                  NUM_CHAR.rl.r,
;  541          0797  1 !                                  ROW.rl.r,
;  542          0798  1 !                                  COL.rl.r)
;  543          0799  1 !
;  544          0800  1 ! FORMAL PARAMETERS:
;  545          0801  1 !
;  546          0802  1 !       DISPLAY_ID.rl.r         Display id of desired virtual display.
;  547          0803  1 !
;  548          0804  1 !       NUM_CHAR.rl.r           Number of characters to be deleted
;  549          0805  1 !
;  550          0806  1 !       ROW.rl.r                Row position at which to start deletion.
;  551          0807  1 !
;  552          0808  1 !       COL.rl.r                Column position at which to start
;  553          0809  1 !                               deletion.
;  554          0810  1 !
;  555          0811  1 ! IMPLICIT INPUTS:
;  556          0812  1 !
;  557          0813  1 !       NONE
;  558          0814  1 !
;  559          0815  1 ! IMPLICIT OUTPUTS:
;  560          0816  1 !
;  561          0817  1 !       NONE
;  562          0818  1 !
;  563          0819  1 ! COMPLETION STATUS:
;  564          0820  1 !
;  565          0821  1 !       SS$_NORMAL      Normal successful completion
;  566          0822  1 !       SMG$_INVROW     Invalid row specification-- outside virtual
;  567          0823  1 !                                                   display.
;  568          0824  1 !       SMG$_INVCOL     Invalid column specification -- outside virtual
;  569          0825  1 !                                                   display.
;  570          0826  1 !       SMG$_INVDIS_ID  Invalid display id.
;  571          0827  1 !       SMG$_WRONUMARG  Wrong number of arguments.
```

```
;  572          0828  1 !       SMG$_INVARG     Number of characters specified starting at
;  573          0829  1 !                       specified row and column extends outside of
;  574          0830  1 !                       virtual display.
;  575          0831  1 !
;  576          0832  1 ! SIDE EFFECTS:
;  577          0833  1 !
;  578          0834  1 !       NONE
;  579          0835  1 !--
;  580          0836  2     BEGIN
;  581          0837  2     LOCAL
;  582          0838  2         NUM_TO_MOVE,            ! No. of char positions to move left
;  583          0839  2
;  584          0840  2         DCB : REF $DCB_DECL;    ! Addr of display control block
;  585          0841  2
;  586          0842  2
;  587          0843  2     $SMG$VALIDATE_ARGCOUNT (4,4);
;  588          0844  2
;  589          0845  2     $SMG$GET_DCB ( .DISPLAY_ID, DCB);   ! Get addr of display control
;  590          0846  2
;  591          0847  2 !+
;  592          0848  2 ! Validity check row and column specified
;  593          0849  2 !-
;  594          0850  2
;  595          0851  2     $SMG$VALIDATE_ROW_COL ( ..ROW, ..COL);
;  596          0852  2
;  597          0853  2 !+
;  598          0854  2 ! Call SMG$$SCROLL_AREA to do the work.  It will shift the text left and
;  599          0855  2 ! blank fill the vacated space.
;  600          0856  2 !-
;  601          0857  2
;  602          0858  2     NUM_TO_MOVE = ..NUM_CHAR;           ! Assume number is ok
;  603          0859  2     IF ..COL + ..NUM_CHAR -1 GTR .DCB[DCB_W_NO_COLS]
;  604          0860  2     THEN                                ! Truncate to what's left on line
;  605          0861  2         NUM_TO_MOVE = .DCB [DCB_W_NO_COLS] - ..COL + 1;
;  606          0862  2
;  607          0863  2     SMG$$SCROLL_AREA (.DCB, ..ROW, ..COL,
;  608          0864  2                       1, .DCB [DCB_W_NO_COLS],
;  609          0865  2                       SMG$M_LEFT, .NUM_TO_MOVE);
;  610          0866  2
;  611          0867  2 !+
;  612          0868  2 ! Make sure cursor points to first requested delete position.
;  613          0869  2 !-
;  614          0870  2
;  615          0871  2     DCB [DCB_W_CURSOR_ROW] = ..ROW;
;  616          0872  2     DCB [DCB_W_CURSOR_COL] = ..COL;
;  617          0873  2     DCB [DCB_V_FULL] = 0;               ! turn off scrolling
;  618          0874  2
;  619          0875  2 !+
;  620          0876  2 ! See if this change should be reflected on the screen.
;  621          0877  2 !-
;  622          0878  2
;  623          0879  3     RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
;  624          0880  3                                        SMG$C_DELETE_CHARS,
;  625          0881  2                                        ..ROW));
;  626          0882  1     END;                        ! End of routine SMG$DELETE_CHARS
```

```
                                                 001C 00000          .ENTRY  SMG$DELETE_CHARS, Save R2,R3,R4                 ; 0772
                                          04     6C  91 00002          CMPB    (AP), #4                                        ; 0843
                                                 08  13 00005          BEQL    1$
                                          50 00000000G   8F  DO 00007          MOVL    #SMG$_WRONUMARG, R0
                                                         04  00000E          RET
                                          50        04   BC  DO 0000F 1$:     MOVL    @DISPLAY_ID, R0                         ; 0845
                                    04    BC        38   A0  D1 00013          CMPL    56(R0), @DISPLAY_ID
                                                         06  12 00018          BNEQ    2$
                                          11        44   A0  91 0001A          CMPB    68(R0), #17
                                                         08  13 0001E          BEQL    3$
                                          50 00000000G   8F  DO 00020 2$:     MOVL    #SMG$_INVDIS_ID, R0
                                                         04  00027          RET
                                          52        04   BC  DO 00028 3$:     MOVL    @DISPLAY_ID, DCB
                                          54        0C   BC  DO 0002C          MOVL    @ROW, R4                                ; 0851
                                                         08  15 00030          BLEQ    4$
                54        02    A2                   10      00  ED 00032          CMPZV   #0, #16, 2(DCB), R4
                                                         08  18 00038          BGEQ    5$
                                          50 00000000G   8F  DO 0003A 4$:     MOVL    #SMG$_INVROW, R0
                                                         04  00041          RET
                                          53        10   BC  DO 00042 5$:     MOVL    @COL, R3
                                                         08  15 00046          BLEQ    6$
                53        06    A2                   10      00  ED 00048          CMPZV   #0, #16, 6(DCB), R3
                                                         08  18 0004E          BGEQ    7$
                                          50 00000000G   8F  DO 00050 6$:     MOVL    #SMG$_INVCOL, R0
                                                         04  00057          RET
                                          51        08   BC  DO 00058 7$:     MOVL    @NUM_CHAR, R1                           ; 0858
                                          50             51  DO 0005C          MOVL    R1, NUM_TO_MOVE
                                          51        FF A143  9E 0005F          MOVAB   -1(R1)[R3], R1                          ; 0859
                51        06    A2                   10      00  ED 00064          CMPZV   #0, #16, 6(DCB), R1
                                                         0B  18 0006A          BGEQ    8$
                                          51        06   A2  3C 0006C          MOVZWL  6(DCB), R1                              ; 0861
                                          51             53  C2 00070          SUBL2   R3, R1
                                          50        01   A1  9E 00073          MOVAB   1(R1), NUM_TO_MOVE
                                                         50  DD 00077 8$:     PUSHL   NUM_TO_MOVE                             ; 0865
                                                         08  DD 00079          PUSHL   #8                                      ; 0863
                                          7E        06   A2  3C 0007B          MOVZWL  6(DCB), -(SP)                           ; 0864
                                                         01  DD 0007F          PUSHL   #1                                      ; 0863
                                                         53  DD 00081          PUSHL   R3
                                                         14  BB 00083          PUSHR   #^M<R2,R4>
                                    00000000G     00       07  FB 00085          CALLS   #7, SMG$$SCROLL_AREA
                                          28      A2       54  B0 00089          MOVW    R4, 40(DCB)                             ; 0871
                                          2A      A2       53  B0 00090          MOVW    R3, 42(DCB)                             ; 0872
                                          34      A2       01  8A 00094          BICB2   #1, 52(DCB)                             ; 0873
                                                         54  DD 00098          PUSHL   R4                                      ; 0881
                                                         0B  DD 0009A          PUSHL   #11                                     ; 0879
                                                         52  DD 0009C          PUSHL   DCB
                                    00000000G     00       03  FB 0009E          CALLS   #3, SMG$$CHECK_FOR_OUTPUT_DCB
                                                         04  000A5          RET                                             ; 0882

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

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

```
;  629          0884  1 %SBTTL 'SMG$DELETE_LINE - Delete Line in Display'
;  630          0885  1 GLOBAL ROUTINE SMG$DELETE_LINE (
;  631          0886  1                                 DISPLAY_ID,
;  632          0887  1                                 START_LINE_NO,
;  633          0888  1                                 NUM_LINES
;  634          0889  1                               ) =
;  635          0890  1 !++
;  636          0891  1 ! FUNCTIONAL DESCRIPTION:
;  637          0892  1 !
;  638          0893  1 !       This routine deletes one or more lines in a virtual display
;  639          0894  1 !       and scrolls up remaining lines into the created space.
;  640          0895  1 !
;  641          0896  1 !       The internal display cursor position is left at the first column
;  642          0897  1 !       of the start line.
;  643          0898  1 !
;  644          0899  1 ! CALLING SEQUENCE:
;  645          0900  1 !
;  646          0901  1 !       ret_status.wlc.v = SMG$DELETE_LINE (DISPLAY_ID.rl.r,
;  647          0902  1 !                                           START_LINE_NO.rl.r
;  648          0903  1 !                                           [,NUM_LINES.rl.r])
;  649          0904  1 !
;  650          0905  1 ! FORMAL PARAMETERS:
;  651          0906  1 !
;  652          0907  1 !       DISPLAY_ID      Display id of desired virtual display.
;  653          0908  1 !
;  654          0909  1 !       START_LINE_NO.rl.r
;  655          0910  1 !                       Address of line number where deleting/scrolling
;  656          0911  1 !                       starts.
;  657          0912  1 !
;  658          0913  1 !       NUM_LINES.rl.r  Optional.  Number of lines to delete/scroll.
;  659          0914  1 !                       If not specified, defaults to 1.
;  660          0915  1 !
;  661          0916  1 ! IMPLICIT INPUTS:
;  662          0917  1 !
;  663          0918  1 !       NONE
;  664          0919  1 !
;  665          0920  1 ! IMPLICIT OUTPUTS:
;  666          0921  1 !
;  667          0922  1 !       NONE
;  668          0923  1 !
;  669          0924  1 ! COMPLETION STATUS:
;  670          0925  1 !
;  671          0926  1 !       SS$_NORMAL      Normal successful completion
;  672          0927  1 !
;  673          0928  1 ! SIDE EFFECTS:
;  674          0929  1 !
;  675          0930  1 !       NONE
;  676          0931  1 !--
;  677          0932  2     BEGIN
;  678          0933  2     BUILTIN
;  679          0934  2         NULLPARAMETER;
;  680          0935  2
;  681          0936  2     LOCAL
;  682          0937  2         DCB : REF $DCB_DECL,    ! addr of display control block
;  683          0938  2         WORK_NUM_LINES;         ! working number of lines
;  684          0939  2
;  685          0940  2     $SMG$VALIDATE_ARGCOUNT (2,3);
```

```
;  686     0941  2
;  687     0942  2      $SMG$GET_DCB (.DISPLAY_ID, DCB);
;  688     0943  2                                    ! get address of display control block
;  689     0944  2
;  690     0945  2  !+
;  691     0946  2  ! Validate arguments and set up working temporaries.
;  692     0947  2  !-
;  693     0948  2
;  694     0949  2      IF ..START_LINE_NO LSS .DCB [DCB_W_ROW_START] OR
;  695     0950  2         ..START_LINE_NO GTR .DCB [DCB_W_NO_ROWS]
;  696     0951  2      THEN
;  697     0952  2          RETURN (SMG$_INVROW);
;  698     0953  2
;  699     0954  2      IF NOT NULLPARAMETER (NUM_LINES)
;  700     0955  2      THEN
;  701     0956  3          BEGIN
;  702     0957  3          WORK_NUM_LINES = ..NUM_LINES;
;  703     0958  3          IF ..START_LINE_NO + .WORK_NUM_LINES -1 GTR .DCB [DCB_W_NO_ROWS]
;  704     0959  3          THEN
;  705     0960  3              RETURN (SMG$_INVARG);
;  706     0961  3          END
;  707     0962  2      ELSE
;  708     0963  2          WORK_NUM_LINES = 1;
;  709     0964  2
;  710     0965  2  !+
;  711     0966  2  ! Reset the line characteristics in case the line was previously
;  712     0967  2  ! double high or wide.
;  713     0968  2  !-
;  714     0969  2
;  715     0970  3      BEGIN
;  716     0971  3      BIND
;  717     0972  3          LINE_CHAR = .DCB [DCB_A_LINE_CHAR];
;  718     0973  3      MAP
;  719     0974  3          LINE_CHAR : VECTOR [,BYTE];
;  720     0975  3
;  721     0976  3      INCR ROW_NUM FROM 1 TO .WORK_NUM_LINES DO
;  722     0977  3          LINE_CHAR [.ROW_NUM] = 0;
;  723     0978  2      END;
;  724     0979  2
;  725     0980  2  !+
;  726     0981  2  ! Call SMG$$SCROLL_AREA to do the work.  It will move lines up or down,
;  727     0982  2  ! overwriting the lines that are to be deleted.
;  728     0983  2  !-
;  729     0984  2
;  730     0985  2      SMG$$SCROLL_AREA (.DCB,
;  731     0986  2                        ..START_LINE_NO,
;  732     0987  2                        .DCB [DCB_W_COL_START],
;  733     0988  2                        (.DCB [DCB_W_NO_ROWS] - ..START_LINE_NO + 1),
;  734     0989  2                        .DCB [DCB_W_NO_COLS],
;  735     0990  2                        SMG$M_UP,
;  736     0991  2                        .WORK_NUM_LINES);
;  737     0992  2
;  738     0993  2  !+
;  739     0994  2  ! Set cursor position to start of first line that was 'deleted'.
;  740     0995  2  !-
;  741     0996  2
;  742     0997  2      DCB [DCB_W_CURSOR_ROW] = ..START_LINE_NO;
```

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


                                        000C 00000          .ENTRY  SMG$DELETE_LINE, Save R2,R3                     ; 0885
                50          6C          02  83 00002        SUBB3   #2, (AP), DIFF                                  ; 0940
                            01          50  91 00006        CMPB    DIFF, #1
                                        08  1B 00009        BLEQU   1$
                            50 00000000G 8F  D0 0000B       MOVL    #SMG$_WRONUMARG, R0
                                        04  00012           RET
                            50      04  BC  D0 00013 1$:    MOVL    @DISPLAY_ID, R0                                 ; 0942
                        C4  BC      38  A0  D1 00017        CMPL    56(R0), @DISPLAY_ID
                                        06  12 0001C        BNEQ    2$
                            11      44  A0  91 0001E        CMPB    68(R0), #17
                                        08  13 00022        BEQL    3$
                            50 00000000G 8F  D0 00024 2$:   MOVL    #SMG$_INVDIS_ID, R0
                                        04  0002B          RET
                            52      04  BC  D0 0002C 3$:    MOVL    @DISPLAY_ID, DCB
                            53      08  BC  D0 00030        MOVL    @START_LINE_NO, R3                              ; 0949
                53              62      10  00  ED 00034    CMPZV   #0, #16, (DCB), R3
                                        08  14 00039        BGTR    4$
                53      02      A2      10  00  ED 0003B    CMPZV   #0, #16, 2(DCB), R3                             ; 0950
                                        08  18 00041        BGEQ    5$
                            50 00000000G 00  9E 00043 4$:   MOVAB   SMG$_INVROW, R0                                 ; 0952
                                        04  0004A          RET
                            03          6C  91 0004B 5$:    CMPB    (AP), #3                                        ; 0954
                                        1E  1F 0004E        BLSSU   6$
                                    0C  AC  D5 00050        TSTL    12(AP)
                                        19  13 00053        BEQL    6$
                            51      0C  BC  D0 00055        MOVL    @NUM_LINES, WORK_NUM_LINES                      ; 0957
                            50      FF A143 9E 00059        MOVAB   -1(WORK_NUM_LINES)[R3], R0                       ; 0958
                50      02      A2      10  00  ED 0005E    CMPZV   #0, #16, 2(DCB), R0
                                        0B  18 00064        BGEQ    7$
                            50 00000000G 00  9E 00066       MOVAB   SMG$_INVARG, R0                                 ; 0960
                                        04  0006D          RET
                            51          01  D0 0006E 6$:    MOVL    #1, WORK_NUM_LINES                              ; 0963
                                        50  D4 00071 7$:    CLRL    ROW_NUM                                         ; 0976
                                        04  11 00073        BRB     9$
                                    4C B240 94 00075 8$:    CLRB    @76(DCB)[ROW_NUM]                               ; 0977
                                F8      51  F3 00079 9$:    AOBLEQ  WORK_NUM_LINES, ROW_NUM, 8$
                50                      51  DD 0007D        PUSHL   WORK_NUM_LINES                                  ; 0991
                                        01  DD 0007F        PUSHL   #1                                              ; 0985
                7E          06      A2  3C  00081           MOVZWL  6(DCB), -(SP)                                   ; 0989
                50          02      A2  3C  00085           MOVZWL  2(DCB), R0                                      ; 0988
```

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

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

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

```
;  757          1011  1 %SBTTL 'SMG$ERASE_CHARS - Erase Characters in Display'
;  758          1012  1 GLOBAL ROUTINE SMG$ERASE_CHARS (
;  759          1013  1                                 DISPLAY_ID,
;  760          1014  1                                 NUM_CHARS,
;  761          1015  1                                 LINE_NO,
;  762          1016  1                                 COL_NO
;  763          1017  1                               ) =
;  764          1018  1 !++
;  765          1019  1 ! FUNCTIONAL DESCRIPTION:
;  766          1020  1 !
;  767          1021  1 !       This routine erases characters in a display by replacing them
;  768          1022  1 !       with blanks.  Remaining text is not moved over.
;  769          1023  1 !
;  770          1024  1 !       Erasing is limited to the specified line.  If NUM_CHARS would
;  771          1025  1 !       cause characters past the end of line to be erased, it is ignored
;  772          1026  1 !       and only to the end of the line is blanked.
;  773          1027  1 !
;  774          1028  1 !       The internal display cursor position is set to the first free
;  775          1029  1 !       position following the erase.
;  776          1030  1 !
;  777          1031  1 ! CALLING SEQUENCE:
;  778          1032  1 !
;  779          1033  1 !       ret_status.wlc.v = SMG$ERASE_CHARS (DISPLAY_ID.rl.r,
;  780          1034  1 !                                           NUM_CHARS.rl.r
;  781          1035  1 !                                           LINE_NO.rl.r,
;  782          1036  1 !                                           COL_NO.rl.r)
;  783          1037  1 !
;  784          1038  1 ! FORMAL PARAMETERS:
;  785          1039  1 !
;  786          1040  1 !       DISPLAY_ID.rl.r Display id of desired virtual display.
;  787          1041  1 !
;  788          1042  1 !       NUM_CHARS.rl.r  Number of characters to erase.
;  789          1043  1 !
;  790          1044  1 !       LINE_NO.rl.r    Address of line number where erasing starts.
;  791          1045  1 !
;  792          1046  1 !       COL_NO.rl.r     Address of column number where erasing starts.
;  793          1047  1 !
;  794          1048  1 ! IMPLICIT INPUTS:
;  795          1049  1 !
;  796          1050  1 !       NONE
;  797          1051  1 !
;  798          1052  1 ! IMPLICIT OUTPUTS:
;  799          1053  1 !
;  800          1054  1 !       NONE
;  801          1055  1 !
;  802          1056  1 ! COMPLETION STATUS:
;  803          1057  1 !
;  804          1058  1 !       SS$_NORMAL      Normal successful completion
;  805          1059  1 !
;  806          1060  1 ! SIDE EFFECTS:
;  807          1061  1 !
;  808          1062  1 !       NONE
;  809          1063  1 !--
;  810          1064  1
;  811          1065  2     BEGIN
;  812          1066  2
;  813          1067  2     LOCAL
```

```
;  814          1068  2          DCB : REF $DCB_DECL,     ! Addr of virtual display
;  815          1069  2                                   ! control block.
;  816          1070  2          DEST_COL,                ! destination column number
;  817          1071  2          START_INDEX,             ! starting linear index into buffers
;  818          1072  2          WORK_NUM_CHARS;          ! working number of characters
;  819          1073  2
;  820          1074  2      $SMG$VALIDATE_ARGCOUNT (4, 4);
;  821          1075  2
;  822          1076  2      $SMG$GET_DCB ( .DISPLAY_ID, DCB);    ! Get address of virtual
;  823          1077  2                                           ! display control block.
;  824          1078  2  !+
;  825          1079  2  ! Validity check the parameters.  Don't try to erase more characters
;  826          1080  2  ! than are on the line.
;  827          1081  2  !-
;  828          1082  2
;  829          1083  2      $SMG$VALIDATE_ROW_COL ( ..LINE_NO, ..COL_NO);
;  830          1084  2
;  831          1085  2      WORK_NUM_CHARS = ..NUM_CHARS;
;  832          1086  2      DEST_COL = ..COL_NO + ..NUM_CHARS - 1;
;  833          1087  2      IF .DEST_COL GTR .DCB [DCB_W_NO_COLS]
;  834          1088  2      THEN
;  835          1089  2          WORK_NUM_CHARS = .DCB [DCB_W_NO_COLS] - ..COL_NO + 1;
;  836          1090  2
;  837          1091  2  !+
;  838          1092  2  ! Put blanks in the text buffer, the default attribute byte into the
;  839          1093  2  ! attribute buffer, and the default character set byte into the character
;  840          1094  2  ! buffer.
;  841          1095  2  !-
;  842          1096  2
;  843          1097  2      START_INDEX = $SMG$LINEAR (..LINE_NO, ..COL_NO);
;  844          1098  2
;  845          1099  2      $SMG$BLANK_FILL_DCB (.WORK_NUM_CHARS, .START_INDEX);
;  846          1100  2
;  847          1101  2  !+
;  848          1102  2  ! Virtual display buffers have been erased.  Init cursor to first
;  849          1103  2  ! free position.
;  850          1104  2  !-
;  851          1105  2
;  852          1106  2      DCB [DCB_W_CURSOR_ROW] = ..LINE_NO;
;  853          1107  2      DCB [DCB_W_CURSOR_COL] = ..COL_NO;
;  854          1108  2      DCB [DCB_V_FULL] = 0;                ! turn off scrolling
;  855          1109  2
;  856          1110  2  !+
;  857          1111  2  ! Call the output routine to see if this change in the virtual display
;  858          1112  2  ! should be reflected on the screen immediately.
;  859          1113  2  !-
;  860          1114  2
;  861          1115  3      RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
;  862          1116  3                                         SMG$C_ERASE_CHARS,
;  863          1117  2                                         ..LINE_NO));
;  864          1118  1      END;                               ! End of routine SMG$ERASE_CHARS


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

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

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

; 865   1119 1 !<BLF/PAGE>

```
;  867          1120  1 %SBTTL 'SMG$ERASE_DISPLAY - Erase Display'
;  868          1121  1 GLOBAL ROUTINE SMG$ERASE_DISPLAY (
;  869          1122  1                                   DISPLAY_ID,
;  870          1123  1                                   START_LINE_NO,
;  871          1124  1                                   START_COL_NO,
;  872          1125  1                                   END_LINE_NO,
;  873          1126  1                                   END_COL_NO
;  874          1127  1                                    ) =
;  875          1128  1 !++
;  876          1129  1 ! FUNCTIONAL DESCRIPTION:
;  877          1130  1 !
;  878          1131  1 !       This routine causes the specified display to be erased from the
;  879          1132  1 !       specified start position to the specified end position.  If omitted,
;  880          1133  1 !       the start position defaults to (1,1).  If the end position is omitted,
;  881          1134  1 !       it defaults to the last character position in the display.  Thus the
;  882          1135  1 !       entire display is erased if only the display_id is passed.
;  883          1136  1 !
;  884          1137  1 !       The internal display cursor position is set to the first free
;  885          1138  1 !       position (1,1) following the erase.
;  886          1139  1 !
;  887          1140  1 ! CALLING SEQUENCE:
;  888          1141  1 !
;  889          1142  1 !       ret_status.wlc.v = SMG$ERASE_DISPLAY (DISPLAY_ID.rl.r
;  890          1143  1 !                                             [,START_LINE_NO.rl.r,
;  891          1144  1 !                                             START_COL_NO.rl.r]
;  892          1145  1 !                                             [,END_LINE_NO.rl.r,
;  893          1146  1 !                                             END_COL_NO.rl.r])
;  894          1147  1 !
;  895          1148  1 ! FORMAL PARAMETERS:
;  896          1149  1 !
;  897          1150  1 !       DISPLAY_ID.rl.r         Display id of the desired virtual display.
;  898          1151  1 !
;  899          1152  1 !       START_LINE_NO.rl.r      Optional.  Address of line number at which to
;  900          1153  1 !                               start erasing.  If omitted, START_COL_NO will be
;  901          1154  1 !                               ignored as well and whole display will be
;  902          1155  1 !                               erased.
;  903          1156  1 !
;  904          1157  1 !       START_COL_NO.rl.r       Optional.  Address of Column number at which
;  905          1158  1 !                               to start erasing.  If omitted, START_LINE_NO
;  906          1159  1 !                               will be ignored as well and entire display will
;  907          1160  1 !                               be erased.
;  908          1161  1 !
;  909          1162  1 !       END_LINE_NO.rl.r        Optional.  Address of line number at which to
;  910          1163  1 !                               end erasing.  If omitted, END_COL_NO is also
;  911          1164  1 !                               ignored and to the end of the display is erased.
;  912          1165  1 !
;  913          1166  1 !       END_COL_NO.rl.r         Optional.  Address of column number at which to
;  914          1167  1 !                               end erasing.  If omitted, END_LINE_NO is also
;  915          1168  1 !                               ignored and to the end of the display is erased.
;  916          1169  1 !
;  917          1170  1 ! IMPLICIT INPUTS:
;  918          1171  1 !
;  919          1172  1 !       NONE
;  920          1173  1 !
;  921          1174  1 ! IMPLICIT OUTPUTS:
;  922          1175  1 !
;  923          1176  1 !       NONE
```

```
;  924          1177  1 !
;  925          1178  1 ! COMPLETION STATUS:
;  926          1179  1 !
;  927          1180  1 !       SS$_NORMAL      Normal successful completion
;  928          1181  1 !
;  929          1182  1 ! SIDE EFFECTS:
;  930          1183  1 !
;  931          1184  1 !       NONE
;  932          1185  1 !--
;  933          1186  1
;  934          1187  2     BEGIN
;  935          1188  2
;  936          1189  2     LOCAL
;  937          1190  2         DCB : REF $DCB_DECL,    ! Addr of virtual display
;  938          1191  2                                 ! control block.
;  939          1192  2         LINE,                   ! temp row
;  940          1193  2         COL,                    ! temp column
;  941          1194  2         END_ROW,                ! temp end row
;  942          1195  2         END_COLUMN,             ! temp end col
;  943          1196  2         START_INDEX,            ! starting linear index into buffers
;  944          1197  2         END_INDEX,              ! ending linear index into buffers
;  945          1198  2         NUM_CHARS;              ! number of characters to put into buffers
;  946          1199  2
;  947          1200  2     BUILTIN
;  948          1201  2         ACTUALCOUNT,
;  949          1202  2         NULLPARAMETER;
;  950          1203  2
;  951          1204  2     LITERAL
;  952          1205  2         K_START_LINE_ARG = 2,   ! argument number of start_line_no
;  953          1206  2         K_END_LINE_ARG   = 4,   ! argument number of end_line_no
;  954          1207  2         K_END_COL_ARG    = 5;   ! argument number of end_col_no
;  955          1208  2
;  956          1209  2 !+
;  957          1210  2 ! Validity check the number of parameters.
;  958          1211  2 !-
;  959          1212  2
;  960          1213  2     $SMG$VALIDATE_ARGCOUNT (1, 5);
;  961          1214  2
;  962          1215  2     $SMG$GET_DCB ( .DISPLAY_ID, DCB);   ! Get address of virtual
;  963          1216  2                                         ! display control block.
;  964          1217  2 !+
;  965          1218  2 ! Calculate the linear start index, based on whether a cursor position
;  966          1219  2 ! was specified.
;  967          1220  2 !-
;  968          1221  2
;  969          1222  2     IF ACTUALCOUNT () LSS K_START_LINE_ARG  OR          ! Quick check
;  970          1223  2        NULLPARAMETER (START_LINE_NO)        OR          ! More elaborate checks
;  971          1224  2        NULLPARAMETER (START_COL_NO)
;  972          1225  2     THEN                            ! default start position
;  973          1226  3         BEGIN
;  974          1227  3         START_INDEX = 0;
;  975          1228  3         LINE = 1;
;  976          1229  3         COL = 1;
;  977          1230  3         END
;  978          1231  2     ELSE                            ! use specified position
;  979          1232  3         BEGIN
;  980          1233  3         $SMG$VALIDATE_ROW_COL ( ..START_LINE_NO, ..START_COL_NO);
```

```
;  981          1234  3          START_INDEX = $SMG$LINEAR (..START_LINE_NO, ..START_COL_NO);
;  982          1235  3          LINE = ..START_LINE_NO;
;  983          1236  3          COL = ..START_COL_NO;
;  984          1237  2          END;
;  985          1238  2
;  986          1239  2      IF ACTUALCOUNT() GEQ K_END_LINE_ARG    AND  ! Last 2 exist and are not null
;  987          1240  2      NOT NULLPARAMETER (END_LINE_NO)        AND
;  988          1241  2      NOT NULLPARAMETER (END_COL_NO)
;  989          1242  2      THEN
;  990          1243  3          BEGIN                          ! use specified end position
;  991          1244  3          $SMG$VALIDATE_ROW_COL (..END_LINE_NO, ..END_COL_NO);
;  992          1245  3          END_ROW = ..END_LINE_NO;
;  993          1246  3          END_COLUMN = ..END_COL_NO;
;  994          1247  3          END
;  995          1248  2      ELSE                               ! use default end position
;  996          1249  3          BEGIN
;  997          1250  3          END_ROW = .DCB [DCB_W_NO_ROWS];
;  998          1251  3          END_COLUMN = .DCB [DCB_W_NO_COLS];
;  999          1252  2          END;
; 1000          1253  2
; 1001          1254  2      END_INDEX = $SMG$LINEAR (.END_ROW, .END_COLUMN);
; 1002          1255  2
; 1003          1256  2
; 1004          1257  2 !+
; 1005          1258  2 ! Reset the line characteristics in case the line was previously
; 1006          1259  2 ! double high or wide.
; 1007          1260  2 !-
; 1008          1261  2
; 1009          1262  3      BEGIN
; 1010          1263  3      BIND
; 1011          1264  3          LINE_CHAR = .DCB [DCB_A_LINE_CHAR];
; 1012          1265  3      MAP
; 1013          1266  3          LINE_CHAR : VECTOR [,BYTE];
; 1014          1267  3      INCR ROW_NUM FROM .LINE TO (.END_ROW - .LINE) DO
; 1015          1268  3          LINE_CHAR [.ROW_NUM] = 0;
; 1016          1269  2      END;
; 1017          1270  2
; 1018          1271  2 !+
; 1019          1272  2 ! Put blanks in the text buffer, the default attribute byte into the
; 1020          1273  2 ! attribute buffer, and the default character set byte into the character
; 1021          1274  2 ! buffer.
; 1022          1275  2 !-
; 1023          1276  2
; 1024          1277  2      NUM_CHARS = .END_INDEX - .START_INDEX + 1;
; 1025          1278  2
; 1026          1279  2      $SMG$BLANK_FILL_DCB (.NUM_CHARS, .START_INDEX);
; 1027          1280  2
; 1028          1281  2 !+
; 1029          1282  2 ! Virtual display buffers have been erased.  Init the cursor to the first
; 1030          1283  2 ! free position.
; 1031          1284  2 !-
; 1032          1285  2
; 1033          1286  2      DCB [DCB_W_CURSOR_ROW] = .LINE;
; 1034          1287  2      DCB [DCB_W_CURSOR_COL] = .COL;
; 1035          1288  2
; 1036          1289  2      DCB [DCB_V_FULL] = 0;       ! flag that some lines are free
; 1037          1290  2
```

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


                                        0FFC 00000          .ENTRY  SMG$ERASE_DISPLAY, Save R2,R3,R4,R5,R6,R7,- ; 1121
                                                                    R8,R9,R10,R11
                                5E          04  C2 00002            SUBL2   #4, SP
                          50    6C          01  83 00005            SUBB3   #1, (AP), DIFF                      ; 1213
                                04          50  91 00009            CMPB    DIFF, #4
                                            08  1B 0000C            BLEQU   1$
                                50 00000000G    8F  D0 0000E        MOVL    #SMG$_WRONUMARG, R0
                                                04 00015            RET
                                50      04      BC  D0 00016 1$:    MOVL    @DISPLAY_ID, R0                     ; 1215
                          04    BC      38      A0  D1 0001A        CMPL    56(R0), @DISPLAY_ID
                                                06  12 0001F        BNEQ    2$
                                11      44      A0  91 00021        CMPB    68(R0), #17
                                                08  13 00025        BEQL    3$
                                50 00000000G    8F  D0 00027 2$:    MOVL    #SMG$_INVDIS_ID, R0
                                                04 0002E            RET
                                58      04      BC  D0 0002F 3$:    MOVL    @DISPLAY_ID, DCB
                                02              6C  91 00033        CMPB    (AP), #2                            ; 1222
                                                0F  1F 00036        BLSSU   4$
                                        08      AC  D5 00038        TSTL    8(AP)                               ; 1223
                                                0A  13 0003B        BEQL    4$
                                03              6C  91 0003D        CMPB    (AP)  #3                            ; 1224
                                                05  1F 00040        BLSSU   4$
                                        0C      AC  D5 00042        TSTL    12(AP)
                                                0A  12 00045        BNEQ    5$
                                                5B  D4 00047 4$:    CLRL    START_INDEX                         ; 1227
                                57              01  D0 00049        MOVL    #1, LINE                            ; 1228
                                6E              01  D0 0004C        MOVL    #1, COL                             ; 1229
                                                42  11 0004F        BRB     10$                                 ; 1232
                                52      08      BC  D0 00051 5$:    MOVL    @START_LINE_NO, R2                  ; 1233
                                                08  15 00055        BLEQ    6$
52      02      A8              10              00  ED 00057        CMPZV   #0, #16, 2(DCB), R2
                                                08  18 0005D        BGEQ    7$
                                50 00000000G    8F  D0 0005F 6$:    MOVL    #SMG$_INVROW, R0
                                                04 00066            RET
                                51      0C      BC  D0 00067 7$:    MOVL    @START_COL_NO, R1
                                                08  15 0006B        BLEQ    8$
51      06      A8              10              00  ED 0006D        CMPZV   #0, #16, 6(DCB), R1
                                                08  18 00073        BGEQ    9$
                                50 00000000G    8F  D0 00075 8$:    MOVL    #SMG$_INVCOL, R0
                                                04 0007C            RET
                                50      FF      A2  9E 0007D 9$:    MOVAB   -1(R2), R0                          ; 1234
                                53      06      A8  3C 00081        MOVZWL  6(DCB), R3
                                50              53  C4 00085        MULL2   R3, R0
                                5B      FF A140     9E 00088        MOVAB   -1(R1)[R0], START_INDEX
```

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

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

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

```
;  1048          1300  1 %SBTTL 'SMG$ERASE_LINE - Erase Line from Display'
;  1049          1301  1 GLOBAL ROUTINE SMG$ERASE_LINE (
;  1050          1302  1                                 DISPLAY_ID,
;  1051          1303  1                                 LINE_NO,
;  1052          1304  1                                 COL_NO
;  1053          1305  1                               ) =
;  1054          1306  1 !++
;  1055          1307  1 ! FUNCTIONAL DESCRIPTION:
;  1056          1308  1 !
;  1057          1309  1 !       This routine causes the display to be erased from the specified
;  1058          1310  1 !       position to the end of the line.  If the position is not
;  1059          1311  1 !       specified, the current cursor position on the screen is assumed.
;  1060          1312  1 !
;  1061          1313  1 !       The internal display cursor position is set to the first free
;  1062          1314  1 !       position following the erase.
;  1063          1315  1 !
;  1064          1316  1 ! CALLING SEQUENCE:
;  1065          1317  1 !
;  1066          1318  1 !       ret_status.wlc.v = SMG$ERASE_LINE (DISPLAY_ID.rl.r,
;  1067          1319  1 !                                          [LINE_NO.rl.r,
;  1068          1320  1 !                                          COL_NO.rl.r])
;  1069          1321  1 !
;  1070          1322  1 ! FORMAL PARAMETERS:
;  1071          1323  1 !
;  1072          1324  1 !       DISPLAY_ID.rl.r Display id of desired virtual display.
;  1073          1325  1 !
;  1074          1326  1 !       LINE_NO.rl.r    Optional.  Address of line number where erasing
;  1075          1327  1 !                       starts.  If this argument is omitted, COL_NO is
;  1076          1328  1 !                       ignored as well, and current position will be
;  1077          1329  1 !                       used.
;  1078          1330  1 !
;  1079          1331  1 !       COL_NO.rl.r     Optional.  Address of column number where
;  1080          1332  1 !                       erasing starts.  If this argument is omitted,
;  1081          1333  1 !                       LINE_NO is ignored as well, and current position
;  1082          1334  1 !                       will be used.
;  1083          1335  1 !
;  1084          1336  1 ! IMPLICIT INPUTS:
;  1085          1337  1 !
;  1086          1338  1 !       NONE
;  1087          1339  1 !
;  1088          1340  1 ! IMPLICIT OUTPUTS:
;  1089          1341  1 !
;  1090          1342  1 !       NONE
;  1091          1343  1 !
;  1092          1344  1 ! COMPLETION STATUS:
;  1093          1345  1 !
;  1094          1346  1 !       SS$_NORMAL      Normal successful completion
;  1095          1347  1 !
;  1096          1348  1 ! SIDE EFFECTS:
;  1097          1349  1 !
;  1098          1350  1 !       NONE
;  1099          1351  1 !--
;  1100          1352  1
;  1101          1353  2     BEGIN
;  1102          1354  2
;  1103          1355  2     LOCAL
;  1104          1356  2         DCB : REF $DCB_DECL,    ! Addr of virtual display
```

```
;  1105          1357  2                                    ! control block.
;  1106          1358  2          LINE,                     ! temp line
;  1107          1359  2          COL,                      ! temp column
;  1108          1360  2          START_INDEX,              ! starting linear index into buffers
;  1109          1361  2          END_INDEX,                ! ending linear index into buffers
;  1110          1362  2          NUM_CHARS;                ! number of characters to put into buffers
;  1111          1363  2
;  1112          1364  2      BUILTIN
;  1113          1365  2          ACTUALCOUNT,
;  1114          1366  2          NULLPARAMETER;
;  1115          1367  2
;  1116          1368  2  !+
;  1117          1369  2  ! Validity check the number of parameters.
;  1118          1370  2  !-
;  1119          1371  2
;  1120          1372  2      $SMG$VALIDATE_ARGCOUNT (1, 3);
;  1121          1373  2
;  1122          1374  2      $SMG$GET_DCB ( .DISPLAY_ID, DCB);    ! Get address of virtual
;  1123          1375  2                                           ! display control block.
;  1124          1376  2
;  1125          1377  2
;  1126          1378  2  !+
;  1127          1379  2  ! Calculate the linear start index, based on whether a cursor position
;  1128          1380  2  ! was specified.
;  1129          1381  2  !-
;  1130          1382  2
;  1131          1383  2      IF NULLPARAMETER (LINE_NO) OR
;  1132          1384  2         NULLPARAMETER (COL_NO)
;  1133          1385  2      THEN                             ! default start position
;  1134          1386  3          BEGIN
;  1135          1387  3          LINE = .DCB [DCB_W_CURSOR_ROW];
;  1136          1388  3          COL = .DCB [DCB_W_CURSOR_COL];
;  1137          1389  3          END
;  1138          1390  2      ELSE                             ! use specified position
;  1139          1391  3          BEGIN
;  1140          1392  3          $SMG$VALIDATE_ROW_COL ( ..LINE_NO, ..COL_NO);
;  1141          1393  3          LINE = ..LINE_NO;
;  1142          1394  3          COL = ..COL_NO;
;  1143          1395  2          END;
;  1144          1396  2  !+
;  1145          1397  2  ! Put blanks in the text buffer, the default attribute byte into the
;  1146          1398  2  ! attribute buffer, and the default character set byte into the character
;  1147          1399  2  ! buffer.
;  1148          1400  2  !-
;  1149          1401  2
;  1150          1402  2      START_INDEX = $SMG$LINEAR (.LINE, .COL);
;  1151          1403  2      END_INDEX = $SMG$LINEAR (.LINE, .DCB [DCB_W_NO_COLS]);
;  1152          1404  2      NUM_CHARS = .END_INDEX - .START_INDEX + 1;
;  1153          1405  2
;  1154          1406  2      $SMG$BLANK_FILL_DCB (.NUM_CHARS, .START_INDEX);
;  1155          1407  2
;  1156          1408  2  !+
;  1157          1409  2  ! Reset the line characteristics in case the line was previously
;  1158          1410  2  ! double high or wide.
;  1159          1411  2  !-
;  1160          1412  2
;  1161          1413  3      BEGIN
```

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


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

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

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

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

```
;1191          1442  1 %SBTTL 'SMG$HOME_CURSOR - Home cursor to a corner position'
;1192          1443  1 GLOBAL ROUTINE SMG$HOME_CURSOR (
;1193          1444  1                                 DISPLAY_ID,
;1194          1445  1                                 POSITION
;1195          1446  1                                ) =
;1196          1447  1 !++
;1197          1448  1 ! FUNCTIONAL DESCRIPTION:
;1198          1449  1 !
;1199          1450  1 !       This routine moves the cursor to the corner row and column
;1200          1451  1 !       of the specified virtual display according to position specified
;1201          1452  1 !       by the POSITION parameter.
;1202          1453  1 !       The caller does not need to know the dimensions of the
;1203          1454  1 !       virtual display.
;1204          1455  1 !
;1205          1456  1 ! CALLING SEQUENCE:
;1206          1457  1 !
;1207          1458  1 !       ret_status.wlc.v = SMG$HOME_CURSOR (
;1208          1459  1 !                                           DISPLAY_ID.rl.r
;1209          1460  1 !                                           [,POSITION.rl.r])
;1210          1461  1 !
;1211          1462  1 ! FORMAL PARAMETERS:
;1212          1463  1 !
;1213          1464  1 !       DISPLAY_ID.rl.r The display id of the desired virtual display.
;1214          1465  1 !
;1215          1466  1 !       POSITION.rl.r   Optional.  Position to move cursor to according
;1216          1467  1 !                       to following code:
;1217          1468  1 !
;1218          1469  1 !        0 = SMG$C_UPPER_LEFT   Row 1 Column 1 (upper left-hand corner).
;1219          1470  1 !                               This is the default if argument is
;1220          1471  1 !                               omitted.
;1221          1472  1 !
;1222          1473  1 !        1 = SMG$C_LOWER_LEFT   Row n Column 1 (Lower left-hand corner).
;1223          1474  1 !                               This is a useful position for inputting
;1224          1475  1 !                               into an upward scrolling virtual display
;1225          1476  1 !
;1226          1477  1 !        2 = SMG$C_UPPER_RIGHT  Row 1 Column m (Upper right-hand corner)
;1227          1478  1 !
;1228          1479  1 !        3 = SMG$C_LOWER_RIGHT  Row n Column m (Lower right-hand corner)
;1229          1480  1 !
;1230          1481  1 !                       +------------------+
;1231          1482  1 !                       |0                2|
;1232          1483  1 !                       |                  |
;1233          1484  1 !                       |                  |
;1234          1485  1 !                       |1                3|
;1235          1486  1 !                       +------------------+
;1236          1487  1 ! IMPLICIT INPUTS:
;1237          1488  1 !
;1238          1489  1 !       NONE
;1239          1490  1 !
;1240          1491  1 ! IMPLICIT OUTPUTS:
;1241          1492  1 !
;1242          1493  1 !       NONE
;1243          1494  1 !
;1244          1495  1 ! COMPLETION STATUS:
;1245          1496  1 !
;1246          1497  1 !       SS$_NORMAL      Normal successful completion
;1247          1498  1 !       SMG$_INVDIS_ID  Invalid virtual display id.
```

```
;  1248          1499  1 !       SMG$WRONUMARG   Wrong number of arguments.
;  1249          1500  1 !
;  1250          1501  1 ! SIDE EFFECTS:
;  1251          1502  1 !
;  1252          1503  1 !       NONE
;  1253          1504  1 !--
;  1254          1505  2     BEGIN
;  1255          1506  2     BUILTIN
;  1256          1507  2         NULLPARAMETER;
;  1257          1508  2
;  1258          1509  2     LOCAL
;  1259          1510  2         ROW,                                ! Working row number
;  1260          1511  2         COL,                                ! Working col number
;  1261          1512  2         DCB : REF $DCB_DECL;                ! Addr of display control block
;  1262          1513  2
;  1263          1514  2     $SMG$VALIDATE_ARGCOUNT (1, 2);
;  1264          1515  2
;  1265          1516  2     $SMG$GET_DCB ( .DISPLAY_ID, DCB);   ! Get addr of display control
;  1266          1517  2                                         ! block
;  1267          1518  2
;  1268          1519  2     ROW = 1;     ! Assume defaults
;  1269          1520  2     COL = 1;
;  1270          1521  2
;  1271          1522  2     IF NOT NULLPARAMETER (POSITION)
;  1272          1523  2     THEN
;  1273          1524  3         BEGIN   ! User-specified
;  1274          1525  3         IF ..POSITION LSS SMG$C_UPPER_LEFT OR
;  1275          1526  3            ..POSITION GTR SMG$C_LOWER_RIGHT
;  1276          1527  3         THEN
;  1277          1528  3             RETURN (SMG$_INVARG);
;  1278          1529  3
;  1279          1530  3         CASE ..POSITION FROM SMG$C_UPPER_LEFT TO SMG$C_LOWER_RIGHT OF
;  1280          1531  3         SET
;  1281          1532  3             [SMG$C_UPPER_LEFT]:
;  1282          1533  3                 ;
;  1283          1534  3
;  1284          1535  3             [SMG$C_LOWER_LEFT]:
;  1285          1536  3                 ROW = .DCB [DCB_W_NO_ROWS];
;  1286          1537  3
;  1287          1538  3             [SMG$C_UPPER_RIGHT]:
;  1288          1539  3                 COL = .DCB [DCB_W_NO_COLS];
;  1289          1540  3
;  1290          1541  3             [SMG$C_LOWER_RIGHT]:
;  1291          1542  4                 BEGIN
;  1292          1543  4                 ROW = .DCB [DCB_W_NO_ROWS];
;  1293          1544  4                 COL = .DCB [DCB_W_NO_COLS];
;  1294          1545  3                 END;
;  1295          1546  3
;  1296          1547  3             [INRANGE, OUTRANGE]:
;  1297          1548  3                 RETURN SMG$_INVARG;
;  1298          1549  3
;  1299          1550  3         TES;
;  1300          1551  3
;  1301          1552  2         END;     ! User-specified
;  1302          1553  2
;  1303          1554  2     DCB [DCB_W_CURSOR_ROW] = .ROW;
;  1304          1555  2     DCB [DCB_W_CURSOR_COL] = .COL;
```

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


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

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

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

```
;  1312          1562  1 %SBTTL 'SMG$INSERT_CHARS - Insert Characters'
;  1313          1563  1 GLOBAL ROUTINE SMG$INSERT_CHARS (
;  1314          1564  1                                  DISPLAY_ID,
;  1315          1565  1                                  CHAR_STRING,
;  1316          1566  1                                  ROW,
;  1317          1567  1                                  COL,
;  1318          1568  1                                  RENDITION_SET,
;  1319          1569  1                                  RENDITION_COMPLEMENT,
;  1320          1570  1                                  CHAR_SET
;  1321          1571  1                                 ) =
;  1322          1572  1 !++
;  1323          1573  1 ! FUNCTIONAL DESCRIPTION:
;  1324          1574  1 !
;  1325          1575  1 !       This routine inserts the specified character string at the
;  1326          1576  1 !       row and column position specified.
;  1327          1577  1 !
;  1328          1578  1 !       Characters supplied from CHAR_STRING are inserted into the
;  1329          1579  1 !       specified display at the specified ROW and COL, shifting
;  1330          1580  1 !       subsequent characters on the line to the right.
;  1331          1581  1 !
;  1332          1582  1 !       SMG$INSERT_CHARS is a character-oriented operation - any
;  1333          1583  1 !       characters which do not fit on the current line are thrown
;  1334          1584  1 !       away.  Wrapping is not allowed.
;  1335          1585  1 !
;  1336          1586  1 !       The internal display cursor position is left at the character
;  1337          1587  1 !       position following the last character written.
;  1338          1588  1 !
;  1339          1589  1 ! CALLING SEQUENCE:
;  1340          1590  1 !
;  1341          1591  1 !       ret_status.wlc.v = SMG$INSERT_CHARS (
;  1342          1592  1 !                                  DISPLAY_ID.rl.r,
;  1343          1593  1 !                                  CHAR_STRING.rt.dx,
;  1344          1594  1 !                                  ROW.rl.r,
;  1345          1595  1 !                                  COL.rl.r,
;  1346          1596  1 !                                  [,RENDITION_SET.rl.r]
;  1347          1597  1 !                                  [,RENDITION_COMPLEMENT.rl.r]
;  1348          1598  1 !                                  [,CHAR_SET.rl.r])
;  1349          1599  1 !
;  1350          1600  1 ! FORMAL PARAMETERS:
;  1351          1601  1 !
;  1352          1602  1 !       DISPLAY_ID.rl.r         Display id of desired virtual display.
;  1353          1603  1 !
;  1354          1604  1 !       CHAR_STRING.rt.dx       Character string to be inserted
;  1355          1605  1 !
;  1356          1606  1 !       ROW.rl.r                Row position at which to start
;  1357          1607  1 !                               insertion.
;  1358          1608  1 !
;  1359          1609  1 !       COL.rl.r                Column position at which to start
;  1360          1610  1 !                               insertion.
;  1361          1611  1 !
;  1362          1612  1 !       RENDITION_SET.rl.r      Each 1 bit attribute in this parameter
;  1363          1613  1 !                               causes the corresponding attribute to
;  1364          1614  1 !                               be set in the display.  (See below for
;  1365          1615  1 !                               list of settable attributes.)
;  1366          1616  1 !
;  1367          1617  1 !       RENDITION_COMPLEMENT.rl.r
;  1368          1618  1 !                               Each 1 bit attribute in this parameter
```

```
;                                causes the corresponding attribute to
;                                be complemented in the display.  (See
;                                below for list of complementable
;                                attributes.)

;       If the same bit is specified in both the RENDITION_SET parameter
;       and in the RENDITION_COMPLEMENT parameter, the application is
;       RENDITION_SET followed by RENDITION complement.  Using these two
;       parameters together the caller can exercise arbitrary and
;       independent control over each attribute on a single call.  On an
;       attribute by attribute basis he can cause the following
;       transformations:

;               SET     COMPLEMENT      Action
;               ---     ----------      ------
;               0       0               Attribute unchanged.
;               1       0               Attribute set to "on"
;               0       1               Attribute set to complement of
;                                       current setting.
;               1       1               Attribute set to "off".


;       Attributes which can be manipulated in this manner are:

;       SMG$M_BLINK  displays characters blinking.
;       SMG$M_BOLD  displays characters in higher-than-normal
;                   intensity.
;       SMG$M_REVERSE  displays characters in reverse video -- that is,
;                  using the opposite default rendition of the
;                  virtual display.
;       SMG$M_UNDERLINE  displays characters underlined.

;       CHAR_SET.rl.r           Optional.  Character set to use.
;                               Choices are:
;                                       SMG$C_UNITED_KINGDOM
;                                       SMG$C_ASCII (default)
;                                       SMG$C_SPEC_GRAPHICS
;                                       SMG$C_ALT_CHAR
;                                       SMG$C_ALT_GRAPHICS

; IMPLICIT INPUTS:

;       NONE

; IMPLICIT OUTPUTS:

;       NONE

; COMPLETION STATUS:

;       SS$_NORMAL      Normal successful completion
;       LIB$_INVSTRDES  Invalid string descriptor
;       SMG$_INVROW     Invalid row specification-- outside virtual
;                       display.
;       SMG$_INVCOL     Invalid column specification -- outside virtual
;                       display.
;       SMG$_INVDIS_ID  Invalid virtual display id.
```

```
;  1426          1676  1 !       SMG$_WRONUMARG  Wrong number or arguments.
;  1427          1677  1 !       SMG$_INVARG     Unrecognized rendition code.
;  1428          1678  1 !
;  1429          1679  1 ! SIDE EFFECTS:
;  1430          1680  1 !
;  1431          1681  1 !       NONE
;  1432          1682  1 !--
;  1433          1683  2     BEGIN
;  1434          1684  2
;  1435          1685  2     BUILTIN
;  1436          1686  2         NULLPARAMETER;
;  1437          1687  2
;  1438          1688  2     LITERAL
;  1439          1689  2         K_SET_ARG = 5,
;  1440          1690  2         K_COMP_ARG = 6;
;  1441          1691  2
;  1442          1692  2     LOCAL
;  1443          1693  2         DCB : REF $DCB_DECL,            ! Addr of display control block
;  1444          1694  2         STR_ADDR,                       ! Addr of text
;  1445          1695  2         STR_LEN : INITIAL(0),           ! Length of text
;  1446          1696  2         PRINT_LEN,                      ! Length of text w/tabs, backspaces, etc
;  1447          1697  2         REND_CODE,                      ! Rendition code
;  1448          1698  2         STATUS;                         ! Retd by called routine
;  1449          1699  2
;  1450          1700  2     $SMG$VALIDATE_ARGCOUNT (4, 7);
;  1451          1701  2
;  1452          1702  2     $SMG$GET_DCB ( .DISPLAY_ID, DCB);   ! Get addr of display control
;  1453          1703  2
;  1454          1704  2 !+
;  1455          1705  2 ! Validity check row and column specified.
;  1456          1706  2 !-
;  1457          1707  2
;  1458          1708  2     $SMG$VALIDATE_ROW_COL ( ..ROW, ..COL);
;  1459          1709  2
;  1460          1710  2     IF NOT NULLPARAMETER( CHAR_SET )
;  1461          1711  2     THEN
;  1462          1712  3         BEGIN
;  1463          1713  3         CASE ..CHAR_SET FROM SMG$C_UNITED_KINGDOM TO SMG$C_ALT_GRAPHICS OF
;  1464          1714  3         SET
;  1465          1715  3
;  1466          1716  3         [SMG$C_UNITED_KINGDOM, SMG$C_ASCII, SMG$C_SPEC_GRAPHICS,
;  1467          1717  3          SMG$C_ALT_CHAR, SMG$C_ALT_GRAPHICS]:
;  1468          1718  3                 ;
;  1469          1719  3         [INRANGE, OUTRANGE]:
;  1470          1720  3                 RETURN (SMG$_INVARG);
;  1471          1721  3         TES;
;  1472          1722  2         END;
;  1473          1723  2
;  1474          1724  3     IF NOT (STATUS = LIB$ANALYZE_SDESC_R2 (.CHAR_STRING;
;  1475          1725  3                                            STR_LEN,
;  1476          1726  3                                            STR_ADDR))
;  1477          1727  2     THEN
;  1478          1728  2         RETURN (.STATUS);
;  1479          1729  2
;  1480          1730  2 !+
;  1481          1731  2 ! Reset the line characteristics in case the line was previously
;  1482          1732  2 ! double high or wide.
```

```
;  1483          1733  2 !-
;  1484          1734  2
;  1485          1735  3     BEGIN
;  1486          1736  3     BIND
;  1487          1737  3         LINE_CHAR = .DCB [DCB_A_LINE_CHAR];
;  1488          1738  3     MAP
;  1489          1739  3         LINE_CHAR : VECTOR [,BYTE];
;  1490          1740  3     IF .LINE_CHAR [..ROW] NEQ 0         ! prev. wide or high chars
;  1491          1741  3     THEN
;  1492          1742  4         BEGIN                           ! blank line
;  1493          1743  4         LOCAL
;  1494          1744  4             START_INDEX;
;  1495          1745  4         START_INDEX = $SMG$LINEAR (..ROW, 1);
;  1496          1746  4         $SMG$BLANK_FILL_DCB (.DCB [DCB_W_NO_COLS], .START_INDEX);
;  1497          1747  4         LINE_CHAR [..ROW] = 0;
;  1498          1748  3         END;
;  1499          1749  2     END;
;  1500          1750  2
;  1501          1751  2 !+
;  1502          1752  2 ! Shift existing characters over to make room for new text.  If the
;  1503          1753  2 ! space is already blank, just write into it.  Don't write over a
;  1504          1754  2 ! blank if there are characters to the right of it - an embedded blank
;  1505          1755  2 ! should be treated as a regular character and shifted.
;  1506          1756  2 !
;  1507          1757  2 ! Macro $SMG$FIND_PRINT_LENGTH scans the string for funny characters
;  1508          1758  2 ! and determines how many printing characters there are.  Extra space
;  1509          1759  2 ! is allowed for tabs.  If a LF or CR is found, it terminates the string
;  1510          1760  2 ! since the effect of this routine is limited to a single line.  STR_LEN
;  1511          1761  2 ! is modified to reflect the number of characters that comprise PRINT_LEN.
;  1512          1762  2 !-
;  1513          1763  2
;  1514          1764  3     BEGIN
;  1515          1765  3     LOCAL
;  1516          1766  3         INDEX,
;  1517          1767  3         PTR,
;  1518          1768  3         LINE_LEN,
;  1519          1769  3         TEXT_BUF : REF VECTOR [,BYTE];
;  1520          1770  3
;  1521          1771  3     $SMG$FIND_PRINT_LENGTH (STR_LEN, .STR_ADDR, PRINT_LEN);
;  1522          1772  3
;  1523          1773  3     INDEX = $SMG$LINEAR (..ROW, ..COL);
;  1524          1774  3     TEXT_BUF = .DCB [DCB_A_TEXT_BUF];
;  1525          1775  3
;  1526          1776  3     LINE_LEN = .DCB [DCB_W_NO_COLS] - ..COL + 1;
;  1527          1777  3
;  1528          1778  3     PTR = CH$FIND_NOT_CH (.LINE_LEN, TEXT_BUF [.INDEX], %C' ');
;  1529          1779  3                                         ! get ptr to 1st non-blank
;  1530          1780  3     IF .PTR NEQ 0
;  1531          1781  3     THEN
;  1532          1782  3         SMG$$SCROLL_AREA (.DCB, ..ROW, ..COL,
;  1533          1783  3                           1, .DCB [DCB_W_NO_COLS],
;  1534          1784  3                           SMG$M_RIGHT, .PRINT_LEN);
;  1535          1785  3
;  1536          1786  2     END;
;  1537          1787  2
;  1538          1788  2 !+
;  1539          1789  2 ! Write text into created space.  SMG$$PUT_TEXT_TO_BUFFER checks for
```

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


                                                          .EXTRN  CHAR_TABLE

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

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

```
                                50              61  9A 000FE          MOVZBL  (ADDR_DIFF), R0                                    ;
                             09 01 00000000G0040  8F 00101          CASEB   CHAR_TABLE[R0], #1, #9                             ;
          001C        002C    002C        002C     0010A 14$:       .WORD   16$-T4$,-                                          ;
          0032        0032    0032        0032     00112                    16$-14$,-                                         ;
                              002C        0032     0011A                    16$-14$,-                                         ;
                                                                            15$-14$,-                                         ;
                                                                            17$-14$,-                                         ;
                                                                            17$-14$,-                                         ;
                                                                            17$-14$,-                                         ;
                                                                            17$-14$,-                                         ;
                                                                            17$-14$,-                                         ;
                                                                            16$-14$                                           ;
                                50 00000000G  8F  D0 0011E          MOVL    #SMG$_FATERRLIB, R0                               ;
                                              04  00125             RET                                                       ;
                                50       2A   A6  3C 00126 15$:     MOVZWL  42(DCB), R0                                       ;
                                              50  D7 0012A          DECL    R0                                                ;
                                50            08  C6 0012C          DIVL2   #8, R0                                            ;
                                55       09 A540  7E 0012F          MOVAQ   9(PRINT_LEN)[R0], PRINT_LEN                       ;
                                              6E  D6 00134          INCL    STR_LEN                                           ;
                                              57  D6 00136 16$:     INCL    IN_POINTER                                        ;
                                              54  D7 00138          DECL    BYTES_REMAINING                                   ;
                                              A3  11 0013A          BRB     13$                                               ;
                                50       FF   A9  9E 0013C 17$:     MOVAB   -1(R9), R0                                        ; 1773
                                51       06   A6  3C 00140          MOVZWL  6(DCB), R1                                        ;
                                50            51  C4 00144          MULL2   R1, R0                                            ;
                                52       FF AA40  9E 00147          MOVAB   -1(R10)[R0], INDEX                                ;
                                51       10   A6  D0 0014C          MOVL    16(DCB), TEXT_BUF                                 ; 1774
                                50       06   A6  3C 00150          MOVZWL  6(DCB), R0                                        ; 1776
                                50            5A  C2 00154          SUBL2   R10, R0                                           ;
                                              50  D6 00157          INCL    LINE_LEN                                          ;
                          6241  50            20  3B 00159          SKPC    #32, LINE_LEN, (INDEX)[TEXT_BUF]                  ; 1778
                                              02  12 0015E          BNEQ    18$                                               ;
                                              51  D4 00160          CLRL    R1                                                ;
                                              51  D5 00162 18$:     TSTL    PTR                                               ; 1780
                                              15  13 00164          BEQL    19$                                               ;
                                              55  DD 00166          PUSHL   PRINT_LEN                                         ; 1784
                                              04  DD 00168          PUSHL   #4                                                ; 1782
                             7E       06   A6  3C 0016A          MOVZWL  6(DCB), -(SP)                                     ; 1783
                                              01  DD 0016E          PUSHL   #1                                                ; 1782
                                    0640      8F  BB 00170          PUSHR   #^M<R6,R9,R10>                                    ;
          00000000G             00            07  FB 00174          CALLS   #7, SMG$$SCROLL_AREA                              ;
                                50       2E   A6  9A 0017B 19$:     MOVZBL  46(DCB), REND_CODE                                ; 1793
                                05            6C  91 0017F          CMPB    (AP), #5                                          ;
                                              09  1F 00182          BLSSU   20$                                               ;
                                          14  AC  D5 00184          TSTL    20(AP)                                            ;
                                              04  13 00187          BEQL    20$                                               ;
                                50        14  BC  C8 00189          BISL2   @RENDITION_SET, REND_CODE                         ;
                                06            6C  91 0018D 20$:     CMPB    (AP), #6                                          ;
                                              09  1F 00190          BLSSU   21$                                               ;
                                          18  AC  D5 00192          TSTL    24(AP)                                            ;
                                              04  13 00195          BEQL    21$                                               ;
                                50        18  BC  CC 00197          XORL2   @RENDITION_COMPLEMENT, REND_CODE                  ;
                       28       A6            59  B0 0019B 21$:     MOVW    R9, 40(DCB)                                       ; 1796
                       2A       A6            5A  B0 0019F          MOVW    R10, 42(DCB)                                      ; 1797
                                07            6C  91 001A3          CMPB    (AP), #7                                          ; 1802
                                              0A  1F 001A6          BLSSU   22$                                               ;
                                          1C  AC  D5 001A8          TSTL    28(AP)                                            ;
```

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

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

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

```
;1573          1822  1 %SBTTL 'SMG$INSERT_LINE - Insert a line into a virtual display'
;1574          1823  1 GLOBAL ROUTINE SMG$INSERT_LINE   ( DISPLAY_ID,
;1575          1824  1                                    LINE_NO,
;1576          1825  1                                    STRING,
;1577          1826  1                                    DIRECTION,
;1578          1827  1                                    RENDITION_SET,
;1579          1828  1                                    RENDITION_COMPLEMENT,
;1580          1829  1                                    WRAP_FLAG,
;1581          1830  1                                    CHAR_SET ) =
;1582          1831  1
;1583          1832  1 !++
;1584          1833  1 ! FUNCTIONAL DESCRIPTION:
;1585          1834  1 !
;1586          1835  1 !       This routine allows a line to be inserted into a virtual display
;1587          1836  1 !       at a location other than the first or last line.  Existing lines
;1588          1837  1 !       are scrolled in the specified direction to create an open space.
;1589          1838  1 !       If a string is passed, it will be written in the space created;
;1590          1839  1 !       otherwise, the new line will remain blank.  The entire line is
;1591          1840  1 !       rewritten; if the string does not span the width of the display,
;1592          1841  1 !       it is blank padded.
;1593          1842  1 !
;1594          1843  1 !       If WRAP_FLAG is set, characters which would exceed the rightmost
;1595          1844  1 !       boundary of the display cause another line scroll and the overflow
;1596          1845  1 !       characters are written in the created space.  If WRAP_FLAG is off,
;1597          1846  1 !       any overflow characters are lost.
;1598          1847  1 !
;1599          1848  1 !       The internal display cursor position is left at the character
;1600          1849  1 !       position following the text written.
;1601          1850  1 !
;1602          1851  1 !       See SMG$PUT_WITH_SCROLL to add lines at the first or last
;1603          1852  1 !       line of a virtual display with scrolling.
;1604          1853  1 !
;1605          1854  1 ! CALLING SEQUENCE:
;1606          1855  1 !
;1607          1856  1 !       ret_status.wlc.v = SMG$INSERT_LINE ( DISPLAY_ID.rl.r,
;1608          1857  1 !                                            LINE_NO.rlu.r,
;1609          1858  1 !                                            [,STRING.rt.dx]
;1610          1859  1 !                                            [,DIRECTION.rl.r]
;1611          1860  1 !                                            [,RENDITION_SET.rl.r]
;1612          1861  1 !                                            [,RENDITION_COMPLEMENT.rl.r]
;1613          1862  1 !                                            [,WRAP_FLAG.rl.r]
;1614          1863  1 !                                            [,CHAR_SET.rl.r])
;1615          1864  1 !
;1616          1865  1 ! FORMAL PARAMETERS:
;1617          1866  1 !
;1618          1867  1 !       DISPLAY_ID.rl.r         Display id of desired virtual display.
;1619          1868  1 !
;1620          1869  1 !       LINE_NO.rlu.r           Line number at which to start scroll.
;1621          1870  1 !
;1622          1871  1 !       STRING.rt.dx            Optional.  Text to insert into line
;1623          1872  1 !                               created.  If not specified, all blanks
;1624          1873  1 !                               are used.
;1625          1874  1 !
;1626          1875  1 !       DIRECTION.rlu.r         Optional.  Direction to scroll text.
;1627          1876  1 !                                       SMG$M_UP (Default)
;1628          1877  1 !                                       SMG$M_DOWN
;1629          1878  1 !
```

```
;1630   1879  1 !       RENDITION_SET.rl.r      Each 1 bit attribute in this parameter
;1631   1880  1 !                               causes the corresponding attribute to
;1632   1881  1 !                               be set in the display.  (See below for
;1633   1882  1 !                               list of settable attributes.)
;1634   1883  1 !
;1635   1884  1 !       RENDITION_COMPLEMENT.rl.r
;1636   1885  1 !                               Each 1 bit attribute in this parameter
;1637   1886  1 !                               causes the corresponding attribute to
;1638   1887  1 !                               be complemented in the display.  (See
;1639   1888  1 !                               below for list of complementable
;1640   1889  1 !                               attributes.)
;1641   1890  1 !
;1642   1891  1 !       If the same bit is specified in both the RENDITION_SET parameter
;1643   1892  1 !       and in the RENDITION_COMPLEMENT parameter, the application is
;1644   1893  1 !       RENDITION_SET followed by RENDITION complement.  Using these two
;1645   1894  1 !       parameters together the caller can exercise arbitrary and
;1646   1895  1 !       independent control over each attribute on a single call.  On an
;1647   1896  1 !       attribute by attribute basis he can cause the following
;1648   1897  1 !       transformations:
;1649   1898  1 !
;1650   1899  1 !               SET     COMPLEMENT      Action
;1651   1900  1 !               ---     ----------      ------
;1652   1901  1 !               0       0               Attribute unchanged.
;1653   1902  1 !               1       0               Attribute set to "on"
;1654   1903  1 !               0       1               Attribute set to complement of
;1655   1904  1 !                                       current setting.
;1656   1905  1 !               1       1               Attribute set to "off".
;1657   1906  1 !
;1658   1907  1 !       WRAP_FLAG.rl.r          Optional.  Action to take if text does
;1659   1908  1 !                               not fit on the line.
;1660   1909  1 !                               0 = nowrap (Default)
;1661   1910  1 !                               1 = wrap
;1662   1911  1 !
;1663   1912  1 !       CHAR_SET.rl.r           Optional.  Character set to use.
;1664   1913  1 !                               Choices are:
;1665   1914  1 !                                       SMG$C_UNITED_KINGDOM
;1666   1915  1 !                                       SMG$C_ASCII (default)
;1667   1916  1 !                                       SMG$C_SPEC_GRAPHICS
;1668   1917  1 !                                       SMG$C_ALT_CHAR
;1669   1918  1 !                                       SMG$C_ALT_GRAPHICS
;1670   1919  1 !
;1671   1920  1 ! IMPLICIT INPUTS:
;1672   1921  1 !
;1673   1922  1 !       NONE
;1674   1923  1 !
;1675   1924  1 ! IMPLICIT OUTPUTS:
;1676   1925  1 !
;1677   1926  1 !       NONE
;1678   1927  1 !
;1679   1928  1 ! COMPLETION STATUS:
;1680   1929  1 !
;1681   1930  1 !       SS$_NORMAL      Normal successful completion
;1682   1931  1 !       SMG$_INVDIS_ID  Invalid display id.
;1683   1932  1 !       SMG$_WRONUMARG  Wrong number of arguments.
;1684   1933  1 !
;1685   1934  1 ! SIDE EFFECTS:
;1686   1935  1 !
```

```
;  1687          1936  1 !       NONE
;  1688          1937  1 !--
;  1689          1938  2     BEGIN
;  1690          1939  2
;  1691          1940  2     BUILTIN
;  1692          1941  2         NULLPARAMETER;
;  1693          1942  2
;  1694          1943  2     LITERAL
;  1695          1944  2         K_SET_ARG = 5,
;  1696          1945  2         K_COMP_ARG = 6;
;  1697          1946  2
;  1698          1947  2     LOCAL
;  1699          1948  2         DONE,
;  1700          1949  2         STATUS,                          ! Status of subroutine calls
;  1701          1950  2         DCB : REF $DCB_DECL,             ! Addr of display control block
;  1702          1951  2         STR_LEN : INITIAL (0),           ! Length of text string
;  1703          1952  2         STR_ADDR,                        ! Addr of text string
;  1704          1953  2         DIR,                             ! Working direction
;  1705          1954  2         C_SET,                           ! Working character set
;  1706          1955  2         WRAP_CHARS : INITIAL (0),        ! No. of overflow characters
;  1707          1956  2                                          ! Reset by call to
;  1708          1957  2                                          ! SMG$$PUT_TEXT_TO_BUFFER
;  1709          1958  2         REND_CODE;                       ! Rendition code from params
;  1710          1959  2                                          ! Set by $SMG$SET_REND_CODE
;  1711          1960  2     $SMG$VALIDATE_ARGCOUNT (2, 8);
;  1712          1961  2
;  1713          1962  2     $SMG$GET_DCB ( .DISPLAY_ID, DCB);    ! Get address of display control
;  1714          1963  2                                          ! block
;  1715          1964  2
;  1716          1965  2 !+
;  1717          1966  2 ! Validity check the arguments.
;  1718          1967  2 !-
;  1719          1968  2
;  1720          1969  2     IF ..LINE_NO LSS .DCB [DCB_W_TOP_OF_SCRREG] OR
;  1721          1970  2        ..LINE_NO GTR .DCB [DCB_W_BOTTOM_OF_SCRREG]
;  1722          1971  2     THEN
;  1723          1972  2         RETURN (SMG$_INVROW);
;  1724          1973  2
;  1725          1974  2     DIR = SMG$M_UP;                      ! init to default
;  1726          1975  2
;  1727          1976  2     IF NOT NULLPARAMETER( DIRECTION )
;  1728          1977  2     THEN
;  1729          1978  3         BEGIN
;  1730          1979  4         IF (..DIRECTION NEQ SMG$M_UP AND
;  1731          1980  4             ..DIRECTION NEQ SMG$M_DOWN)
;  1732          1981  3         THEN
;  1733          1982  4             RETURN (SMG$_INVARG)
;  1734          1983  3         ELSE
;  1735          1984  3             DIR = ..DIRECTION;
;  1736          1985  2         END;
;  1737          1986  2
;  1738          1987  2     C_SET = SMG$C_ASCII;                 ! init to default
;  1739          1988  2
;  1740          1989  2     IF NOT NULLPARAMETER( CHAR_SET )
;  1741          1990  2     THEN
;  1742          1991  3         BEGIN
;  1743          1992  3         CASE ..CHAR_SET FROM SMG$C_UNITED_KINGDOM TO SMG$C_ALT_GRAPHICS OF
```

```
; 1744          1993  3          SET
; 1745          1994  3
; 1746          1995  3          [SMG$C_UNITED_KINGDOM, SMG$C_ASCII, SMG$C_SPEC_GRAPHICS,
; 1747          1996  3           SMG$C_ALT_CHAR, SMG$C_ALT_GRAPHICS]:
; 1748          1997  3                   C_SET = ..CHAR_SET;
; 1749          1998  3          [INRANGE, OUTRANGE]:
; 1750          1999  3                  RETURN (SMG$_INVARG);
; 1751          2000  3          TES;
; 1752          2001  2          END;
; 1753          2002  2
; 1754          2003  2      !+
; 1755          2004  2      ! Set up string to output, either the caller's text or a blank line.
; 1756          2005  2      !-
; 1757          2006  2
; 1758          2007  2          IF NOT NULLPARAMETER (STRING)
; 1759          2008  2          THEN
; 1760          2009  3              BEGIN                               ! use caller's text
; 1761          2010  4              IF NOT (STATUS = LIB$ANALYZE_SDESC_R2 (.STRING; STR_LEN, STR_ADDR))
; 1762          2011  3              THEN
; 1763          2012  3                  RETURN (.STATUS);
; 1764          2013  2              END;                                ! user caller's text
; 1765          2014  2
; 1766          2015  2      !+
; 1767          2016  2      ! Move existing lines up or down to create an empty space.  This space
; 1768          2017  2      ! will remain empty if the user did not specify a text string.
; 1769          2018  2      !-
; 1770          2019  2
; 1771          2020  2          IF .DIR EQL SMG$M_UP
; 1772          2021  2          THEN
; 1773          2022  2              SMG$$SCROLL_AREA (.DCB, .DCB [DCB_W_TOP_OF_SCRREG],
; 1774          2023  2                                .DCB [DCB_W_COL_START],
; 1775          2024  2                                (..LINE_NO = .DCB [DCB_W_TOP_OF_SCRREG] + 1),
; 1776          2025  2                                .DCB [DCB_W_NO_COLS],
; 1777          2026  2                                SMG$M_UP, 1)
; 1778          2027  2          ELSE
; 1779          2028  2              SMG$$SCROLL_AREA (.DCB, ..LINE_NO, .DCB [DCB_W_COL_START],
; 1780          2029  2                                (.DCB [DCB_W_BOTTOM_OF_SCRREG] - ..LINE_NO + 1),
; 1781          2030  2                                .DCB [DCB_W_NO_COLS],
; 1782          2031  2                                SMG$M_DOWN, 1);
; 1783          2032  2
; 1784          2033  2      !+
; 1785          2034  2      ! Reset the line characteristics in case the line was previously
; 1786          2035  2      ! double high or wide.
; 1787          2036  2      !-
; 1788          2037  2
; 1789          2038  3          BEGIN
; 1790          2039  3          BIND
; 1791          2040  3              LINE_CHAR = .DCB [DCB_A_LINE_CHAR];
; 1792          2041  3          MAP
; 1793          2042  3              LINE_CHAR : VECTOR [,BYTE];
; 1794          2043  3          LINE_CHAR [..LINE_NO] = 0;
; 1795          2044  2          END;
; 1796          2045  2
; 1797          2046  2      !+
; 1798          2047  2      ! Write text into the space just created.  SMG$$PUT_TEXT_TO_BUFFER
; 1799          2048  2      ! checks for funny characters (tabs, etc) and also updates the cursor
; 1800          2049  2      ! position.
```

```
;  1801          2050  2 !-
;  1802          2051  2
;  1803          2052  2      IF .STR_LEN NEQ 0
;  1804          2053  2      THEN
;  1805          2054  3          BEGIN                               ! parse & move caller's text
;  1806          2055  3          $SMG$SET_REND_CODE (K_SET_ARG, K_COMP_ARG);
;  1807          2056  3
;  1808          2057  3          DCB [DCB_W_CURSOR_ROW] = ..LINE_NO;
;  1809          2058  3          DCB [DCB_W_CURSOR_COL] = 1;
;  1810          2059  3                                              ! set current position for PUT_TEXT
;  1811          2060  3          DCB [DCB_V_FULL] = 0;
;  1812          2061  3
;  1813          2062  3          SMG$$PUT_TEXT_TO_BUFFER (.DCB,
;  1814          2063  3                                   .REND_CODE,
;  1815          2064  3                                   .STR_LEN,
;  1816          2065  3                                   .STR_ADDR,
;  1817          2066  3                                   C_SET,
;  1818          2067  3                                   WRAP_CHARS);
;  1819          2068  3          END
;  1820          2069  3
;  1821          2070  2      ELSE
;  1822          2071  3          BEGIN
;  1823          2072  3          !+
;  1824          2073  3          ! Used the short cut to put out a blank line (we let SMG$$SCROLL_AREA
;  1825          2074  3          ! do it), so we must update the cursor position ourselves (normally
;  1826          2075  3          ! SMG$$PUT_TEXT_TO_BUFFER would do it).
;  1827          2076  3          !-
;  1828          2077  3          DCB [DCB_W_CURSOR_ROW] = ..LINE_NO + 1;
;  1829          2078  3          IF .DCB [DCB_W_CURSOR_ROW] LEQ .DCB [DCB_W_BOTTOM_OF_SCRREG]
;  1830          2079  3          THEN
;  1831          2080  4              BEGIN
;  1832          2081  4              DCB [DCB_V_FULL] = 0;
;  1833          2082  4              DCB [DCB_W_CURSOR_COL] = .DCB [DCB_W_COL_START]
;  1834          2083  4              END
;  1835          2084  3          ELSE
;  1836          2085  4              BEGIN                           ! just wrote last line
;  1837          2086  4              DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_BOTTOM_OF_SCRREG];
;  1838          2087  4              DCB [DCB_W_CURSOR_COL] = .DCB [DCB_W_NO_COLS];
;  1839          2088  4              DCB [DCB_V_FULL] = 1;           ! remember used all lines
;  1840          2089  4              DCB [DCB_V_COL_80] = 1;         ! remember at last col position
;  1841          2090  3              END;
;  1842          2091  2          END;
;  1843          2092  2
;  1844          2093  2 !+
;  1845          2094  2 ! Check for overflow characters which should cause us to scroll 2 lines
;  1846          2095  2 ! in order to output everything (if wrap requested).  Need to pass the
;  1847          2096  2 ! number of characters (not the printing length) in the descriptor.
;  1848          2097  2 !-
;  1849          2098  2
;  1850          2099  2      IF .WRAP_CHARS NEQ 0
;  1851          2100  2      THEN
;  1852          2101  3          BEGIN
;  1853          2102  3          IF NOT NULLPARAMETER( WRAP_FLAG ) AND
;  1854          2103  3             ..WRAP_FLAG NEQ 0                ! wrap requested
;  1855          2104  3          THEN
;  1856          2105  4              BEGIN                           ! recurse until whole string output
;  1857          2106  4              LOCAL
```

```
;  1858          2107  4                  TEMP_DESC : BLOCK [8, BYTE];
;  1859          2108  4
;  1860          2109  4              TEMP_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
;  1861          2110  4              TEMP_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  1862          2111  4              TEMP_DESC [DSC$W_LENGTH] = .STR_LEN - .DCB [DCB_W_NO_COLS];
;  1863          2112  4              TEMP_DESC [DSC$A_POINTER] = .STR_ADDR +
;  1864          2113  4                                          .DCB [DCB_W_NO_COLS];
;  1865          2114  4
;  1866          2115  4              !+
;  1867          2116  4              ! Kludge for downscrolling so that pieces of the line appear
;  1868          2117  4              ! in the correct order.
;  1869          2118  4              !-
;  1870          2119  4              IF .DIR EQL SMG$M_DOWN
;  1871          2120  4              THEN
;  1872          2121  5                  BEGIN
;  1873          2122  5                  LOCAL
;  1874          2123  5                      OVERFLOW_LINE;
;  1875          2124  5                  OVERFLOW_LINE = ..LINE_NO + 1;
;  1876          2125  5                  SMG$INSERT_LINE (.DISPLAY_ID, OVERFLOW_LINE,
;  1877          2126  5                                   TEMP_DESC, %REF (SMG$M_DOWN),
;  1878          2127  6                                   (IF NOT NULLPARAMETER (RENDITION_SET)
;  1879          2128  6                                        THEN .RENDITION_SET
;  1880          2129  5                                        ELSE 0),
;  1881          2130  6                                   (IF NOT NULLPARAMETER (RENDITION_COMPLEMENT)
;  1882          2131  6                                        THEN .RENDITION_COMPLEMENT
;  1883          2132  5                                        ELSE 0),
;  1884          2133  5                                   .WRAP_FLAG, C_SET);
;  1885          2134  5                  END
;  1886          2135  4              ELSE
;  1887          2136  4                  SMG$INSERT_LINE (.DISPLAY_ID, .LINE_NO,
;  1888          2137  4                                   TEMP_DESC, %REF (SMG$M_UP),
;  1889          2138  5                                   (IF NOT NULLPARAMETER (RENDITION_SET)
;  1890          2139  5                                        THEN .RENDITION_SET
;  1891          2140  4                                        ELSE 0),
;  1892          2141  5                                   (IF NOT NULLPARAMETER (RENDITION_COMPLEMENT)
;  1893          2142  5                                        THEN .RENDITION_COMPLEMENT
;  1894          2143  4                                        ELSE 0),
;  1895          2144  4                                   .WRAP_FLAG,
;  1896          2145  4                                   C_SET);
;  1897          2146  4                  DONE = 0;
;  1898          2147  4                  RETURN 1;                   ! to keep Bliss happy
;  1899          2148  4                  END
;  1900          2149  3          ELSE
;  1901          2150  4              BEGIN
;  1902          2151  4              !+
;  1903          2152  4              ! Wrap was not requested but there were overflow characters.
;  1904          2153  4              ! Put out diamond in last position to show truncation.
;  1905          2154  4              !-
;  1906          2155  4              IF .DCB [DCB_V_TRUNC_ICON]
;  1907          2156  4              THEN
;  1908          2157  5                  BEGIN
;  1909          2158  5                  DCB [DCB_W_CURSOR_COL] = .DCB [DCB_W_NO_COLS];
;  1910          2159  5                  SMG$$PUT_TEXT_TO_BUFFER (.DCB,
;  1911          2160  5                                           .REND_CODE + ATTR_M_USER_GRAPHIC,
;  1912          2161  5                                           1, UPLIT (BYTE (DIAMOND)),
;  1913          2162  5                                           SMG$C_ASCII);
;  1914          2163  4                  END;
```

```
;  1915          2164  4              DONE = 1;
;  1916          2165  3              END;
;  1917          2166  3          END
;  1918          2167  2      ELSE
;  1919          2168  2          DONE = 1;                       ! no wrap chars
;  1920          2169  2
;  1921          2170  2  !+
;  1922          2171  2  ! See if this change should be reflected on the screen.  Even if
;  1923          2172  2  ! we call SMG$INSERT_LINE again, SMG$$CHECK_FOR_OUTPUT_DCB should
;  1924          2173  2  ! be executed only once.
;  1925          2174  2  !-
;  1926          2175  2
;  1927          2176  2      IF .DONE
;  1928          2177  2      THEN
;  1929          2178  3          RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
;  1930          2179  3                                             SMG$C_INSERT_LINE,
;  1931          2180  3                                             ..LINE_NO))
;  1932          2181  2      ELSE
;  1933          2182  2          RETURN 1;
;  1934          2183  2
;  1935          2184  1      END;                                ! End of SMG$INSERT_LINE


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

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

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

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

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

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

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

```
;  1938          2186  1 %SBTTL 'SMG$PUT_CHARS - Put Text to Display'
;  1939          2187  1 GLOBAL ROUTINE SMG$PUT_CHARS (
;  1940          2188  1                               DISPLAY_ID,
;  1941          2189  1                               TEXT    : REF BLOCK [,BYTE],
;  1942          2190  1                               LINE_NO,
;  1943          2191  1                               COL_NO,
;  1944          2192  1                               ERASE_LINE_FLAG,
;  1945          2193  1                               RENDITION_SET,
;  1946          2194  1                               RENDITION_COMPLEMENT,
;  1947          2195  1                               CHAR_SET
;  1948          2196  1                             ) =
;  1949          2197  1 !++
;  1950          2198  1 ! FUNCTIONAL DESCRIPTION:
;  1951          2199  1 !
;  1952          2200  1 !       This routine is used to write messages to the virtual display.
;  1953          2201  1 !       No additional cursor motion is added.  Existing text is NOT
;  1954          2202  1 !       shifted over - SMG$PUT_CHARS will overwrite anything already
;  1955          2203  1 !       in the specified positions.  (See SMG$INSERT_CHARS to write new
;  1956          2204  1 !       text while preserving existing text.)
;  1957          2205  1 !
;  1958          2206  1 !       By default, SMG$PUT_CHARS will write out only the text passed to
;  1959          2207  1 !       it - other character positions in the line are not modified.
;  1960          2208  1 !       Optionally, the caller can request that the line be erased before
;  1961          2209  1 !       the new characters are output by setting ERASE_LINE_FLAG.  This
;  1962          2210  1 !       option is provided for callers who do not want to preserve the
;  1963          2211  1 !       previous contents of the line.
;  1964          2212  1 !
;  1965          2213  1 !       The internal display cursor position is left at the character position
;  1966          2214  1 !       following the text written.
;  1967          2215  1 !
;  1968          2216  1 ! CALLING SEQUENCE:
;  1969          2217  1 !
;  1970          2218  1 !       ret_status.wlc.v = SMG$PUT_CHARS (DISPLAY_ID.rl.r,
;  1971          2219  1 !                                         TEXT.rt.dx
;  1972          2220  1 !                                         [,LINE_NO.rl.r, COL_NO.rl.r]
;  1973          2221  1 !                                         [,ERASE_LINE_FLAG.rl[,r]
;  1974          2222  1 !                                         [,RENDITION_SET.rl.r]
;  1975          2223  1 !                                         [,RENDITION_COMPLEMENT.rl.r]
;  1976          2224  1 !                                         [,CHAR_SET.rl.r])
;  1977          2225  1 !
;  1978          2226  1 ! FORMAL PARAMETERS:
;  1979          2227  1 !
;  1980          2228  1 !       DISPLAY_ID.rl.r Display id of virtual display
;  1981          2229  1 !
;  1982          2230  1 !       TEXT.rt.dx      Address of descriptor of output string.
;  1983          2231  1 !
;  1984          2232  1 !       LINE_NO.rl.r    Optional.  Address of line number at which to
;  1985          2233  1 !                       start output.  If omitted (=0), the current
;  1986          2234  1 !                       line number is used.
;  1987          2235  1 !
;  1988          2236  1 !       COL_NO.rl.r     Optional.  Address of column number at which
;  1989          2237  1 !                       to start output.  If omitted (=0), the current
;  1990          2238  1 !                       line number is used.
;  1991          2239  1 !
;  1992          2240  1 !       ERASE_LINE_FLAG.rl.r
;  1993          2241  1 !                       Optional.  Address of flag which specifies
;  1994          2242  1 !                       whether to erase the line before outputting
```

```
;1995           2243  1 !                           the specified text.
;1996           2244  1 !                           1==> Erase
;1997           2245  1 !                           0==> Don't erase
;1998           2246  1 !
;1999           2247  1 !       RENDITION_SET.rl.r      Each 1 bit attribute in this parameter
;2000           2248  1 !                               causes the corresponding attribute to
;2001           2249  1 !                               be set in the display.  (See below for
;2002           2250  1 !                               list of settable attributes.)
;2003           2251  1 !
;2004           2252  1 !       RENDITION_COMPLEMENT.rl.r
;2005           2253  1 !                               Each 1 bit attribute in this parameter
;2006           2254  1 !                               causes the corresponding attribute to
;2007           2255  1 !                               be complemented in the display.  (See
;2008           2256  1 !                               below for list of complementable
;2009           2257  1 !                               attributes.)
;2010           2258  1 !
;2011           2259  1 !       If the same bit is specified in both the RENDITION_SET parameter
;2012           2260  1 !       and in the RENDITION_COMPLEMENT parameter, the application is
;2013           2261  1 !       RENDITION_SET followed by RENDITION complement.  Using these two
;2014           2262  1 !       parameters together the caller can exercise arbitrary and
;2015           2263  1 !       independent control over each attribute on a single call.  On an
;2016           2264  1 !       attribute by attribute basis he can cause the following
;2017           2265  1 !       transformations:
;2018           2266  1 !
;2019           2267  1 !               SET     COMPLEMENT      Action
;2020           2268  1 !               ---     ----------      ------
;2021           2269  1 !               0       0               Attribute unchanged.
;2022           2270  1 !               1       0               Attribute set to "on"
;2023           2271  1 !               0       1               Attribute set to complement of
;2024           2272  1 !                                       current setting.
;2025           2273  1 !               1       1               Attribute set to "off".
;2026           2274  1 !
;2027           2275  1 !
;2028           2276  1 !       Attributes which can be manipulated in this manner are:
;2029           2277  1 !
;2030           2278  1 !       SMG$M_BLINK  displays characters blinking.
;2031           2279  1 !       SMG$M_BOLD  displays characters in higher-than-normal
;2032           2280  1 !                     intensity.
;2033           2281  1 !       SMG$M_REVERSE  displays characters in reverse video -- that is,
;2034           2282  1 !                     using the opposite default rendition of the
;2035           2283  1 !                     virtual display.
;2036           2284  1 !       SMG$M_UNDERLINE  displays characters underlined.
;2037           2285  1 !
;2038           2286  1 !       CHAR_SET.rl.r   Optional.  Character set to use.
;2039           2287  1 !                       Choices are:
;2040           2288  1 !                               SMG$C_UNITED_KINGDOM
;2041           2289  1 !                               SMG$C_ASCII (default)
;2042           2290  1 !                               SMG$C_SPEC_GRAPHICS
;2043           2291  1 !                               SMG$C_ALT_CHAR
;2044           2292  1 !                               SMG$C_ALT_GRAPHICS
;2045           2293  1 !
;2046           2294  1 ! IMPLICIT INPUTS:
;2047           2295  1 !
;2048           2296  1 !       NONE
;2049           2297  1 !
;2050           2298  1 ! IMPLICIT OUTPUTS:
;2051           2299  1 !
```

```
;  2052          2300  1 !       NONE
;  2053          2301  1 !
;  2054          2302  1 ! COMPLETION STATUS:
;  2055          2303  1 !
;  2056          2304  1 !       SS$_NORMAL      Normal successful completion
;  2057          2305  1 !       SMG$_INVCOL     Invalid column specification
;  2058          2306  1 !       SMG$_INVROW     Invalid row specification
;  2059          2307  1 !       LIB$_INVSTRDES  Invalid string descriptor
;  2060          2308  1 !       SMG$_WRONUMARG  Wrong number (or combination of) arguments
;  2061          2309  1 !
;  2062          2310  1 ! SIDE EFFECTS:
;  2063          2311  1 !
;  2064          2312  1 !       NONE
;  2065          2313  1 !--
;  2066          2314  1
;  2067          2315  2     BEGIN
;  2068          2316  2
;  2069          2317  2     BUILTIN
;  2070          2318  2         NULLPARAMETER;
;  2071          2319  2
;  2072          2320  2     LOCAL
;  2073          2321  2         ROW,                                ! Row where text goes
;  2074          2322  2         COL,                                ! Column where text goes
;  2075          2323  2         REND_CODE,                          ! RENDITION_CODE code to use.
;  2076          2324  2         STR_LEN : INITIAL (0),              ! Length of text string
;  2077          2325  2         STR_ADDR,                           ! Addr. of text string
;  2078          2326  2         DCB : REF $DCB_DECL,                ! Address of virtual display
;  2079          2327  2                                             ! control block
;  2080          2328  2         STATUS;                             ! Status of subroutine calls
;  2081          2329  2
;  2082          2330  2     LITERAL
;  2083          2331  2         K_SET_ARG = 6,
;  2084          2332  2         K_COMP_ARG = 7;
;  2085          2333  2
;  2086          2334  2     $SMG$VALIDATE_ARGCOUNT (2, 8);
;  2087          2335  2
;  2088          2336  2     $SMG$GET_DCB ( .DISPLAY_ID, DCB);   ! Get address of virtual display
;  2089          2337  2                                         ! control block.
;  2090          2338  2
;  2091          2339  2 !+
;  2092          2340  2 ! Get the length and address of the text string.
;  2093          2341  2 !-
;  2094          2342  2
;  2095          2343  3     IF NOT (STATUS = LIB$ANALYZE_SDESC_R2 ( .TEXT,
;  2096          2344  3                                             STR_LEN,
;  2097          2345  3                                             STR_ADDR))
;  2098          2346  2     THEN
;  2099          2347  2         RETURN (.STATUS);
;  2100          2348  2
;  2101          2349  2 !+
;  2102          2350  2 ! Check for all optional arguments.  Set local variables to caller's
;  2103          2351  2 ! values, when available, and defaults when arguments omitted.
;  2104          2352  2 !-
;  2105          2353  2
;  2106          2354  2     IF NOT NULLPARAMETER (LINE_NO) AND
;  2107          2355  2        ..LINE_NO NEQ 0
;  2108          2356  2     THEN
```

```
; 2109          2357  2          ROW = ..LINE_NO
; 2110          2358  2      ELSE
; 2111          2359  2          ROW = .DCB [DCB_W_CURSOR_ROW];
; 2112          2360  2
; 2113          2361  2      IF NOT NULLPARAMETER (COL_NO) AND
; 2114          2362  2         ..COL_NO NEQ 0
; 2115          2363  2      THEN
; 2116          2364  2          COL = ..COL_NO
; 2117          2365  2      ELSE
; 2118          2366  2          COL = .DCB [DCB_W_CURSOR_COL];
; 2119          2367  2
; 2120          2368  2      $SMG$VALIDATE_ROW_COL (.ROW, .COL);
; 2121          2369  2                                  ! verify row & col within display
; 2122          2370  2
; 2123          2371  2      $SMG$SET_REND_CODE (K_SET_ARG, K_COMP_ARG);
; 2124          2372  2                                  ! macro to use caller's args if present
; 2125          2373  2
; 2126          2374  2      IF NOT NULLPARAMETER( CHAR_SET )
; 2127          2375  2      THEN
; 2128          2376  3          BEGIN
; 2129          2377  3          CASE ..CHAR_SET FROM SMG$C_UNITED_KINGDOM TO SMG$C_ALT_GRAPHICS OF
; 2130          2378  3          SET
; 2131          2379  3
; 2132          2380  3          [SMG$C_UNITED_KINGDOM, SMG$C_ASCII, SMG$C_SPEC_GRAPHICS,
; 2133          2381  3           SMG$C_ALT_CHAR, SMG$C_ALT_GRAPHICS]:
; 2134          2382  3                  ;
; 2135          2383  3          [INRANGE, OUTRANGE]:
; 2136          2384  3                  RETURN (SMG$_INVARG);
; 2137          2385  3          TES;
; 2138          2386  2          END;
; 2139          2387  2
; 2140          2388  2
; 2141          2389  2  !+
; 2142          2390  2  ! Reset the line characteristics in case the line was previously
; 2143          2391  2  ! double high or wide.
; 2144          2392  2  !-
; 2145          2393  2
; 2146          2394  3      BEGIN
; 2147          2395  3      BIND
; 2148          2396  3          LINE_CHAR = .DCB [DCB_A_LINE_CHAR];
; 2149          2397  3      MAP
; 2150          2398  3          LINE_CHAR : VECTOR [,BYTE];
; 2151          2399  3      IF .LINE_CHAR [.ROW] NEQ 0          ! prev wide or high chars
; 2152          2400  3      THEN
; 2153          2401  4          BEGIN                           ! blank line
; 2154          2402  4          LOCAL
; 2155          2403  4              START_INDEX;
; 2156          2404  4          START_INDEX = $SMG$LINEAR (.ROW, 1);
; 2157          2405  4          $SMG$BLANK_FILL_DCB (.DCB [DCB_W_NO_COLS], .START_INDEX);
; 2158          2406  4          LINE_CHAR [.ROW] = 0;
; 2159          2407  3          END;
; 2160          2408  2      END;
; 2161          2409  2
; 2162          2410  2  !+
; 2163          2411  2  ! If erase_line_flag is set, blank out the line before outputting the text.
; 2164          2412  2  !-
; 2165          2413  2
```

```
; 2166          2414  2     IF NOT NULLPARAMETER (ERASE_LINE_FLAG) AND
; 2167          2415  2        ..ERASE_LINE_FLAG NEQ 0
; 2168          2416  2     THEN
; 2169          2417  3         BEGIN
; 2170          2418  3         LOCAL
; 2171          2419  3             START_INDEX;
; 2172          2420  3         START_INDEX = $SMG$LINEAR (.ROW, 1);
; 2173          2421  3         $SMG$BLANK_FILL_DCB (.DCB [DCB_W_NO_COLS], .START_INDEX);
; 2174          2422  2         END;
; 2175          2423  2
; 2176          2424  2 !+
; 2177          2425  2 ! All local variables are set up.  Call routine to put text into the
; 2178          2426  2 ! display buffer.
; 2179          2427  2 !-
; 2180          2428  2
; 2181          2429  2     DCB [DCB_W_CURSOR_ROW] = .ROW;
; 2182          2430  2     DCB [DCB_W_CURSOR_COL] = .COL;
; 2183          2431  2                                 ! set current position for PUT_TEXT
; 2184          2432  2
; 2185          2433  3     IF NOT (STATUS = SMG$$PUT_TEXT_TO_BUFFER ( .DCB,
; 2186          2434  3                                                .REND_CODE,
; 2187          2435  3                                                .STR_LEN,
; 2188          2436  3                                                .STR_ADDR,
; 2189          2437  3                                                IF NOT NULLPARAMETER (CHAR_SET)
; 2190          2438  3                                                THEN ..CHAR_SET
; 2191          2439  3                                                ELSE SMG$C_ASCII ))
; 2192          2440  2     THEN
; 2193          2441  2         RETURN (.STATUS);
; 2194          2442  2
; 2195          2443  2
; 2196          2444  2 !+
; 2197          2445  2 ! Check for non-printable characters, such as LF, that may cause this
; 2198          2446  2 ! call to affect several lines.  Normally we try to tell the output
; 2199          2447  2 ! routines exactly what line changed (so it can optimize), but if we
; 2200          2448  2 ! encountered funny characters which may have changed more than 1 line,
; 2201          2449  2 ! just pass a 0 (which will turn off optimization).
; 2202          2450  2 !-
; 2203          2451  3     BEGIN
; 2204          2452  3     LOCAL
; 2205          2453  3         SAVED_STRING_LEN,
; 2206          2454  3         PRINT_LEN;
; 2207          2455  3
; 2208          2456  3         SAVED_STRING_LEN = .STR_LEN;
; 2209          2457  3                                 ! side effect of macro is to change STR_LEN
; 2210          2458  3         $SMG$FIND_PRINT_LENGTH (STR_LEN, .STR_ADDR, PRINT_LEN);
; 2211          2459  3                                 ! macro to do SCANC
; 2212          2460  3         IF .PRINT_LEN NEQ .SAVED_STRING_LEN
; 2213          2461  3         THEN
; 2214          2462  3             ROW = 0;
; 2215          2463  2     END;
; 2216          2464  2
; 2217          2465  2 !+
; 2218          2466  2 ! Turn off scrolling.  This change just moved the cursor to some
; 2219          2467  2 ! random position.
; 2220          2468  2 !-
; 2221          2469  2     DCB [DCB_V_FULL] = 0;
; 2222          2470  2
```

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


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

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

```
                                51       06    A7   5C 00140         MOVZWL  6(DCB), R1                                        ;
                                50             51   C4 00144         MULL2   R1, R0                                            ;
                                5A             50   D0 00147         MOVL    R0, START_INDEX                                   ;
                                50       10    A7   D0 0014A         MOVL    16(DCB), TEXT_BUF                                 ; 2421
                                58       14    A7   7D 0014E         MOVQ    20(DCB), ATTR_BUF                                 ;
     06   A7         20         6E             00   2C 00152         MOVC5   #0, (SP), #32, 6(DCB), (START_INDEX)-             ;
                                             6A40      00158                 [TEXT_BUF]                                        ;
     06   A7     2E  A7         6E             00   2C 0015A         MOVC5   #0, (SP), 46(DCB), 6(DCB), (START_INDEX)-         ;
                                             6A48      00161                 [ATTR_BUF]                                        ;
                                               59   D5 00163         TSTL    CHAR_BUF                                          ;
                                               09   13 00165         BEQL    19$                                               ;
     06   A7     30  A7         6E             00   2C 00167         MOVC5   #0, (SP), 48(DCB), 6(DCB), (START_INDEX)-         ;
                                             6A49      0016E                 [CHAR_BUF]                                        ;
                           28   A7             56   B0 00170 19$:    MOVW    ROW, 40(DCB)                                      ; 2429
                           2A   A7             5B   B0 00174         MOVW    COL, 42(DCB)                                      ; 2430
                                08             6C   91 00178         CMPB    (AP), #8                                          ; 2437
                                               0A   1F 0017B         BLSSU   20$                                               ;
                                         20    AC   D5 0017D         TSTL    32(AP)                                            ;
                                               05   13 00180         BEQL    20$                                               ;
                                         20    BC   DD 00182         PUSHL   @CHAR_SET                                         ; 2438
                                               02   11 00185         BRB     21$                                               ;
                                               01   DD 00187 20$:    PUSHL   #1                                                ; 2437
                                         10    AE   DD 00189 21$:    PUSHL   STR_ADDR                                          ; 2436
                                         08    AE   DD 0018C         PUSHL   STR_LEN                                           ; 2435
                                         14    AE   DD 0018F         PUSHL   REND_CODE                                         ; 2434
                                               57   DD 00192         PUSHL   DCB                                               ; 2433
                     0C000000G  00             05   FB 00194         CALLS   #5, SMG$$PUT_TEXT_TO_BUFFER                       ;
                           04   AE             50   D0 0019B         MOVL    R0, STATUS                                        ;
                                05       04    AE   E8 0019F         BLBS    STATUS, 23$                                       ;
                                50       04    AE   D0 001A3 22$:    MOVL    STATUS, R0                                        ; 2441
                                                    04 001A7         RET                                                       ;
                                5B             6E   D0 001A8 23$:    MOVL    STR_LEN, SAVED_STRING_LEN                         ; 2456
                                5A             01   8E 001AB         MNEGB   #1, ALLONES                                       ; 2458
                                               55   D4 001AE         CLRL    PRINT_LEN                                         ;
                                54             6E   D0 001B0         MOVL    STR_LEN, BYTES_REMAINING                          ;
                                58       0C    AE   D0 001B3         MOVL    STR_ADDR, IN_POINTER                              ;
                                               6E   D4 001B7         CLRL    STR_LEN                                           ;
                                               54   D5 001B9 24$:    TSTL    BYTES_REMAINING                                   ;
                                               59   13 001BB         BEQL    28$                                               ;
          5A 00000000G  00      68             54   2A 001BD         SCANC   BYTES_REMAINING, (IN_POINTER), CHAR_TABLE, -     ;
                                                                             ALLONES                                           ;
                     52         54             50   C3 001C6         SUBL3   NEW_BYTES_REMAINING, BYTES_REMAINING, R2          ;
                                58             52   C0 001CA         ADDL2   R2, IN_POINTER                                    ;
                                55             52   C0 001CD         ADDL2   R2, PRINT_LEN                                     ;
                                6E             52   C0 001D0         ADDL2   R2, STR_LEN                                       ;
                                54             50   D0 001D3         MOVL    NEW_BYTES_REMAINING, BYTES_REMAINING              ;
                                               3E   13 001D6         BEQL    28$                                               ;
                                50             61   9A 001D8         MOVZBL  (ADDR_DIFF), R0                                   ;
                     09         01 00000000G0040    8F 001DB         CASEB   CHAR_TABLE[R0], #1, #9                            ;
        001C         002C     002C           002C      001E4 25$:    .WORD   27$-25$,-                                         ;
        0032         0032     0032           0032      001EC                 27$-25$,-                                         ;
                              002C           0032      001F4                 27$-25$,-                                         ;
                                                                             26$-25$,-                                         ;
                                                                             28$-25$,-                                         ;
                                                                             28$-25$,-                                         ;
                                                                             28$-25$,-                                         ;
                                                                             28$-25$,-                                         ;
```

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

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

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

```
;  2233          2480  1 %SBTTL 'SMG$PUT_LINE - Put Text to Display in Line Mode'
;  2234          2481  1 GLOBAL ROUTINE SMG$PUT_LINE (
;  2235          2482  1                              DISPLAY_ID,
;  2236          2483  1                              TEXT     : REF BLOCK [,BYTE],
;  2237          2484  1                              LINE_ADV,
;  2238          2485  1                              RENDITION_SET,
;  2239          2486  1                              RENDITION_COMPLEMENT,
;  2240          2487  1                              WRAP_FLAG,
;  2241          2488  1                              CHAR_SET
;  2242          2489  1                            ) =
;  2243          2490  1
;  2244          2491  1 !++
;  2245          2492  1 ! FUNCTIONAL DESCRIPTION:
;  2246          2493  1 !
;  2247          2494  1 !       This routine is used to write lines to the virtual display
;  2248          2495  1 !       optionally followed by cursor movement sequences.  SMG$PUT_LINE
;  2249          2496  1 !       writes from the current cursor position to the end of the line.
;  2250          2497  1 !       If the caller's text does not span to the end of the line, blank
;  2251          2498  1 !       fill is added.
;  2252          2499  1 !
;  2253          2500  1 !       Treatment of text which exceeds the rightmost bounds of the
;  2254          2501  1 !       display depends on WRAP_FLAG.  If WRAP_FLAG is set, lines are
;  2255          2502  1 !       scrolled LINE_ADV times to make room for the overflow characters
;  2256          2503  1 !       in the 'next' line.  If wrap is off, overflow characters are lost.
;  2257          2504  1 !
;  2258          2505  1 !       Following a call to SMG$PUT_LINE, the internal display cursor
;  2259          2506  1 !       position is set to column 1 of the next line where output should
;  2260          2507  1 !       occur.  The next line where output should occur is determined by
;  2261          2508  1 !       LINE_ADV;  LINE_ADV defaults to 1 so that subsequent calls to
;  2262          2509  1 !       SMG$PUT_LINE will not cause overprinting.
;  2263          2510  1 !
;  2264          2511  1 ! CALLING SEQUENCE:
;  2265          2512  1 !
;  2266          2513  1 !       ret_status.wlc.v = SMG$PUT_LINE (DISPLAY_ID.rl.r,
;  2267          2514  1 !                                        TEXT.rt.dx
;  2268          2515  1 !                                        [,LINE_ADV.rl.r]
;  2269          2516  1 !                                        [,RENDITION_SET.rl.r]
;  2270          2517  1 !                                        [,RENDITION_COMPLEMENT.rl.r]
;  2271          2518  1 !                                        [,WRAP_FLAG.rl.r]
;  2272          2519  1 !                                        [,CHAR_SET.rl.r])
;  2273          2520  1 !
;  2274          2521  1 ! FORMAL PARAMETERS:
;  2275          2522  1 !
;  2276          2523  1 !       DISPLAY_ID.rl.r Display id of virtual display
;  2277          2524  1 !
;  2278          2525  1 !       TEXT.rt.dx      Address of descriptor of output string.
;  2279          2526  1 !
;  2280          2527  1 !       LINE_ADV.rl.r   Optional.  Address of signed number of lines
;  2281          2528  1 !                       to advance after output.
;  2282          2529  1 !
;  2283          2530  1 !       RENDITION_SET.rl.r      Each 1 bit attribute in this parameter
;  2284          2531  1 !                               causes the corresponding attribute to
;  2285          2532  1 !                               be set in the display.  (See below for
;  2286          2533  1 !                               list of settable attributes.)
;  2287          2534  1 !
;  2288          2535  1 !       RENDITION_COMPLEMENT.rl.r
;  2289          2536  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
        SMG$_INVARG     Negative LINE_ADV specified
```

```
;  2347          2594  1 !       LIB$_INVSTRDES  Invalid string descriptor
;  2348          2595  1 !
;  2349          2596  1 ! SIDE EFFECTS:
;  2350          2597  1 !
;  2351          2598  1 !       NONE
;  2352          2599  1 !--
;  2353          2600  1
;  2354          2601  2     BEGIN
;  2355          2602  2
;  2356          2603  2     BUILTIN
;  2357          2604  2         NULLPARAMETER;
;  2358          2605  2
;  2359          2606  2     LOCAL
;  2360          2607  2         DONE,
;  2361          2608  2         DCB : REF $DCB_DECL,            ! Address of virtual display
;  2362          2609  2                                         ! control block.
;  2363          2610  2         STR_LEN : INITIAL (0),          ! Length of text string
;  2364          2611  2         STR_ADDR,                       ! Address of text string
;  2365          2612  2         REND_CODE,                      ! Rendition code to use
;  2366          2613  2         WRAPPED_CHARS :INITIAL (0),     ! Number of chars that don't fit on
;  2367          2614  2                                         ! the current line
;  2368          2615  2                                         ! Set by call to SMG$$PUT_TEXT_TO_BUFFER
;  2369          2616  2         SCROLL_FLAG : INITIAL (0),      ! Flag to scroll up, down, or neither
;  2370          2617  2                                         ! Set by $SMG$SET_SCROLLING
;  2371          2618  2         STATUS;                         ! Status of subroutine calls
;  2372          2619  2
;  2373          2620  2     LITERAL
;  2374          2621  2         K_SET_ARG = 4,
;  2375          2622  2         K_COMP_ARG = 5;
;  2376          2623  2
;  2377        M 2624  2     MACRO $SCROLL_UP (COUNT) =
;  2378        M 2625  2             SMG$$SCROLL_AREA (.DCB,
;  2379        M 2626  2                               .DCB [DCB_W_TOP_OF_SCRREG],
;  2380        M 2627  2                               .DCB [DCB_W_COL_START],
;  2381        M 2628  2                               (.DCB [DCB_W_BOTTOM_OF_SCRREG] -
;  2382        M 2629  2                                .DCB [DCB_W_TOP_OF_SCRREG] + 1),
;  2383        M 2630  2                               .DCB [DCB_W_NO_COLS],
;  2384          2631  2                               SMG$M_UP, COUNT) %;
;  2385          2632  2
;  2386          2633  2     $SMG$VALIDATE_ARGCOUNT (2,7);
;  2387          2634  2
;  2388          2635  2     $SMG$GET_DCB (.DISPLAY_ID, DCB);    ! Get address of virtual display
;  2389          2636  2                                         ! control block.
;  2390          2637  2
;  2391          2638  2     IF NOT NULLPARAMETER (LINE_ADV) AND
;  2392          2639  2        ..LINE_ADV LSS 0
;  2393          2640  2     THEN
;  2394          2641  2         RETURN (SMG$_INVARG);           ! positive advancirg only
;  2395          2642  2
;  2396          2643  2 !+
;  2397          2644  2 ! Select rendition code to use, based on whether one was provided by
;  2398          2645  2 ! caller.
;  2399          2646  2 !-
;  2400          2647  2
;  2401          2648  2     $SMG$SET_REND_CODE (K_SET_ARG, K_COMP_ARG);
;  2402          2649  2
;  2403          2650  2 !+
```

```
;  2404          2651  2 ! Get the length and address of the text string.
;  2405          2652  2 !-
;  2406          2653  3     IF NOT (STATUS = LIB$ANALYZE_SDESC_R2 ( .TEXT,
;  2407          2654  3                                             STR_LEN,
;  2408          2655  3                                             STR_ADDR))
;  2409          2656  2     THEN
;  2410          2657  2         RETURN (.STATUS);
;  2411          2658  2
;  2412          2659  2 !+
;  2413          2660  2 ! If the previous line written was the last line in the display, we
;  2414          2661  2 ! did not scroll at the end of the operation.  (This would've always
;  2415          2662  2 ! left the last line blank - effectively the display would have one
;  2416          2663  2 ! less useable line.)  If we're at the bottom, scroll up one before writing.
;  2417          2664  2 !
;  2418          2665  2 !-
;  2419          2666  2
;  2420          2667  2     $SMG$SET_SCROLLING (SCROLL_FLAG);
;  2421          2668  2
;  2422          2669  2     IF .SCROLL_FLAG EQL 1
;  2423          2670  2     THEN                            ! we're at the last line in the display
;  2424          2671  3         BEGIN                       ! and display is full
;  2425          2672  3         IF NOT NULLPARAMETER (LINE_ADV)
;  2426          2673  3         THEN
;  2427          2674  4             BEGIN       ! Arg present
;  2428          2675  4             IF ..LINE_ADV GTR 0
;  2429          2676  4             THEN                            ! positive line advancing
;  2430          2677  4
;  2431          2678  4                 $SCROLL_UP (..LINE_ADV)
;  2432          2679  4
;  2433          2680  4             ELSE
;  2434          2681  5                 BEGIN   ! Negative line advance specified
;  2435          2682  5
;  2436          2683  5                 IF ..LINE_ADV LSS 0
;  2437          2684  5                 THEN
;  2438          2685  5                     RETURN (SMG$_INVARG);
;  2439          2686  5
;  2440          2687  5                 END     ! Negative line advance specified
;  2441          2688  4             END         ! Arg present
;  2442          2689  4
;  2443          2690  3         ELSE    ! Arg not present, default to 1 line
;  2444          2691  3
;  2445          2692  3             $SCROLL_UP (1)
;  2446          2693  3
;  2447          2694  2         END;                        ! we're at the last line in the display
;  2448          2695  2
;  2449          2696  2 !+
;  2450          2697  2 ! Blank out the line before writing new text.
;  2451          2698  2 !-
;  2452          2699  2
;  2453          2700  3     BEGIN
;  2454          2701  3     LOCAL
;  2455          2702  3         START_INDEX;
;  2456          2703  3
;  2457          2704  3     START_INDEX = $SMG$LINEAR (.DCB [DCB_W_CURSOR_ROW], .DCB [DCB_W_CURSOR_COL]);
;  2458        P 2705  3     $SMG$BLANK_FILL_DCB ((.DCB [DCB_W_NO_COLS] - .DCB [DCB_W_CURSOR_COL] + 1),
;  2459          2706  3                          .START_INDEX);
;  2460          2707  3
```

```
; 2461          2708  2     END;
; 2462          2709  2
; 2463          2710  2 !+
; 2464          2711  2 ! Reset the line characteristics in case the line was previously
; 2465          2712  2 ! double high or wide.
; 2466          2713  2 !-
; 2467          2714  2
; 2468          2715  3     BEGIN
; 2469          2716  3     BIND
; 2470          2717  3         LINE_CHAR = .DCB [DCB_A_LINE_CHAR];
; 2471          2718  3     MAP
; 2472          2719  3         LINE_CHAR : VECTOR [,BYTE];
; 2473          2720  3     LINE_CHAR [.DCB [DCB_W_CURSOR_ROW]] = 0;
; 2474          2721  2     END;
; 2475          2722  2
; 2476          2723  2 !+
; 2477          2724  2 ! Move the text string into our virtual display buffer.
; 2478          2725  2 ! Note the cursor position is already correct for the call.
; 2479          2726  2 !-
; 2480          2727  2
; 2481          2728  3     IF NOT ( STATUS = SMG$$PUT_TEXT_TO_BUFFER ( .DCB,
; 2482          2729  3                                                 .REND_CODE,
; 2483          2730  3                                                 .STR_LEN,
; 2484          2731  3                                                 .STR_ADDR,
; 2485          2732  3                                                 IF NOT NULLPARAMETER (CHAR_SET)
; 2486          2733  3                                                     THEN ..CHAR_SET
; 2487          2734  3                                                     ELSE SMG$C_ASCII,
; 2488          2735  3                                                 WRAPPED_CHARS))
; 2489          2736  2     THEN
; 2490          2737  2         RETURN (.STATUS);
; 2491          2738  2
; 2492          2739  2
; 2493          2740  2 !+
; 2494          2741  2 ! If all went well so far, we need to enter the <CR>,<LF> to form the
; 2495          2742  2 ! end of line.
; 2496          2743  2 !-
; 2497          2744  2
; 2498          2745  2     DCB [DCB_W_CURSOR_COL] = 1;          ! Effect of <CR>
; 2499          2746  2
; 2500          2747  2 !+
; 2501          2748  2 ! Default to advancing one line if LINE_ADV is omitted.
; 2502          2749  2 !-
; 2503          2750  2
; 2504          2751  2     IF NULLPARAMETER (LINE_ADV)
; 2505          2752  2     THEN
; 2506          2753  3         BEGIN                        ! line adv omitted
; 2507          2754  3         IF .DCB [DCB_W_CURSOR_ROW] + 1 LEQ .DCB [DCB_W_BOTTOM_OF_SCRREG]
; 2508          2755  3         THEN
; 2509          2756  3             ! Just advance cursor row to next line
; 2510          2757  3             DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_CURSOR_ROW] + 1
; 2511          2758  3         ELSE
; 2512          2759  4             BEGIN
; 2513          2760  4             $SMG$SET_SCROLLING (SCROLL_FLAG);
; 2514          2761  4             IF .SCROLL_FLAG EQL 1
; 2515          2762  4             THEN
; 2516          2763  4                 IF .DCB [DCB_W_CURSOR_ROW] NEQ .DCB [DCB_W_BOTTOM_OF_SCRREG]
; 2517          2764  4                 THEN
```

```
; 2518          2765  4                     $SCROLL_UP (1);     ! scroll if within scrolling region
; 2519          2766  4                 IF .DCB [DCB_W_CURSOR_ROW] LEQ .DCB [DCB_W_BOTTOM_OF_SCRREG]
; 2520          2767  4                 THEN
; 2521          2768  4                     DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_BOTTOM_OF_SCRREG];
; 2522          2769  4                 DCB [DCB_V_FOL[] = 1; . remember that we used last line
; 2523          2770  3                 END;
; 2524          2771  3
; 2525          2772  3             END                         ! line_adv omitted
; 2526          2773  3
; 2527          2774  3 !+
; 2528          2775  3 ! Take care of the requested line advancing.
; 2529          2776  3 !-
; 2530          2777  3
; 2531          2778  2     ELSE
; 2532          2779  3         BEGIN                           !  line_adv specified
; 2533          2780  3         IF ..LINE_ADV GTR 0
; 2534          2781  3         THEN
; 253,          2782  4             BEGIN                       ! upspacing requested
; 2536          2783  4             IF .DCB [DCB_W_CURSOR_ROW] + ..LINE_ADV LEQ
; 2537          2784  4                .DCB [DCB_W_BOTTOM_OF_SCRREG]
; 2538          2785  4             THEN
; 2539          2786  4                 ! Just advance cursor row number
; 2540          2787  4                 DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_CURSOR_ROW]
; 2541          2788  4                                        + ..LINE_ADV
; 2542          2789  4             ELSE
; 2543          2790  5                 BEGIN               ! scrolling up
; 2544          2791  5                 $SMG$SET_SCROLLING (SCROLL_FLAG);
; 2545          2792  5                 IF .SCROLL_FLAG EQL 1
; 2546          2793  5                 THEN
; 2547          2794  5                     IF .DCB [DCB_W_CURSOR_ROW] NEQ .DCB [DCB_W_BOTTOM_OF_SCRREG]
; 2548          2795  5                     THEN
; 2549          2796  5                         $SCROLL_UP (..LINE_ADV); ! scroll if w/in scroll region
; 2550          2797  5                 IF .DCB [DCB_W_CURSOR_ROW] LEQ .DCB [DCB_W_BOTTOM_OF_SCRREG]
; 2551          2798  5                 THEN
; 2552          2799  5                     DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_BOTTOM_OF_SCRREG];
; 2553          2800  5                 DCB [DCB_V_FOL[] = 1; ! remember that we used last line
; 2554          2801  5                 END                 ! scrolling up
; 2555          2802  3             END;                    ! upspace action
; 2556          2803  3
; 2557          2804  2         END;                        ! line_adv specified
; 2558          2805  2
; 2559          2806  2     !+
; 2560          2807  2     ! If wrapping was requested and some characters overflowed the line,
; 2561          2808  2     ! call ourself again with the remainder of the characters.
; 2562          2809  2     !-
; 2563          2810  2
; 2564          2811  2     IF .WRAPPED_CHARS NEQ 0 AND
; 2565          2812  3        ((NOT NULLPARAMETER (LINE_ADV) AND ..LINE_ADV GTR 0) OR
; 2566          2813  3        NULLPARAMETER (LINE_ADV))
; 2567          2814  2     THEN
; 2568          2815  3         BEGIN                           ! overflow chars
; 2569          2816  4         IF (NOT NULLPARAMETER (WRAP_FLAG) AND
; 2570          2817  4           ..WRAP_FLAG NEQ 0)
; 2571          2818  3         THEN
; 2572          2819  4             BEGIN                       ! wrap set - recurse w/overflo-
; 2573          2820  4             LOCAL
; 2574          2821  4                 STR_DESC : BLOCK [8, BYTE],
```

```
; 2575          2822  4                   C_SET;
; 2576          2823  4
; 2577          2824  4               STR_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
; 2578          2825  4               STR_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
; 2579          2826  4               STR_DESC [DSC$W_LENGTH] = .STR_LEN - .DCB [DCB_W_NO_COLS];
; 2580          2827  4               STR_DESC [DSC$A_POINTER] = .STR_ADDR + .DCB [DCB_W_NO_COLS];
; 2581          2828  4
; 2582          2829  4               C_SET = SMG$C_ASCII;
; 2583          2830  4               IF NOT NULLPARAMETER( CHAR_SET )
; 2584          2831  4               THEN
; 2585          2832  4                   C_SET = ..CHAR_SET;
; 2586          2833  4
; 2587          2834  4               SMG$PUT_LINE (.DISPLAY_ID,
; 2588          2835  4                             STR_DESC,
; 2589          2836  4                             .LINE_ADV,
; 2590          2837  4                             .RENDITION_SET,
; 2591          2838  4                             .RENDITION_COMPLEMENT,
; 2592          2839  4                             .WRAP_FLAG,
; 2593          2840  4                             C_SET);
; 2594          2841  4               DONE = 0;
; 2595          2842  4               RETURN 1;             ! to keep Bliss happy
; 2596          2843  4               END                   ! wrap set - recurse w/overflow
; 2597          2844  3           ELSE
; 2598          2845  4               BEGIN                 ! wrap not set - truncation
; 2599          2846  4               !+
; 2600          2847  4               ! Wrap was not requested but there were overflow characters.
; 2601          2848  4               ! Put out diamond in last position to show truncation.
; 2602          2849  4               !-
; 2603          2850  4               IF .DCB [DCB_V_TRUNC_ICON]
; 2604          2851  4               THEN
; 2605          2852  5                   BEGIN
; 2606          2853  5                   IF NOT .DCB [DCB_V_FULL]
; 2607          2854  5                   THEN
; 2608          2855  5                       DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_CURSOR_ROW] - 1;
; 2609          2856  5                   DCB [DCB_W_CURSOR_COL] = .DCB [DCB_W_NO_COLS];
; 2610          2857  5
; 2611          2858  5                   SMG$$PUT_TEXT_TO_BUFFER (.DCB,
; 2612          2859  5                                            .REND_CODE + ATTR_M_USER_GRAPHIC,
; 2613          2860  5                                            1, UPLIT (BYTE (DIAMOND)),
; 2614          2861  5                                            SMG$C_ASCII);
; 2615          2862  5
; 2616          2863  5                   IF NOT .DCB [DCB_V_FULL]
; 2617          2864  5                   THEN
; 2618          2865  5                       DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_CURSOR_ROW] + 1;
; 2619          2866  5                                            ! restore for next call
; 2620          2867  5                   DCB [DCB_W_CURSOR_COL] = 1;
; 2621          2868  4                   END;
; 2622          2869  4
; 2623          2870  4               DONE = 1;
; 2624          2871  3               END;                  ! wrap not set - truncation
; 2625          2872  3           END
; 2626          2873  2       ELSE
; 2627          2874  2           DONE = 1;                 ! no wrap chars
; 2628          2875  2
; 2629          2876  2       !+
; 2630          2877  2       ! See if this change should be reflected on the screen.
; 2631          2878  2       ! Even if we call SMG$PUT_LINE again, SMG$$CHECK_FOR_OUTPUT_DCB
```

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


                                        60  00CC0 P.AAB:  .BYTE   96                                                      ;

                                      0FFC 00000         .ENTRY  SMG$PUT_LINE, Save R2,R3,R4,R5,R6,R7,R8,R9,-          ; 2481
                                                                 R10,R11                                                 ;
                              5E              24  C2 00002       SUBL2   #36, SP                                         ;
                                              7E  D4 00005       CLRL    STR_LEN                                         ; 2601
                                          18  AE  D4 00007       CLRL    WRAPPED_CHARS                                   ;
                                              7E  D4 0000A       CLRL    SCROLL_FLAG                                     ;
                      50              6C      02  83 0000C       SUBB3   #2, (AP), DIFF                                  ; 2633
                                      05      50  91 00010       CMPB    DIFF, #5                                        ;
                                              08  1B 00013       BLEQU   1$                                              ;
                                      50 00000000G    8F  D0 00015       MOVL    #SMG$_WRONUMARG, R0                     ;
                                                  04  00021       RET                                                     ;
                                      50      04      BC  D0 00022 1$:    MOVL    @DISPLAY_ID, R0                         ; 2635
                                04    BC      38      A0  D1 00026       CMPL    56(R0), @DISPLAY_ID                     ;
                                              06  12 0002B       BNEQ    2$                                              ;
                                      11      44      A0  91 0002D       CMPB    68(R0), #17                             ;
                                              08  13 00031       BEQL    3$                                              ;
                                      50 00000000G    8F  D0 00033 2$:    MOVL    #SMG$_INVDIS_ID, R0                     ;
                                                  04  00038       RET                                                     ;
                                      56      04      BC  D0 00039 3$:    MOVL    @DISPLAY_ID, DCB                        ;
                                      03              6C  91 0003F       CMPB    (AP), #3                                ; 2638
                                              0A  1F 00042       BLSSU   4$                                              ;
                                          0C  AC  D5 00044       TSTL    12(AP)                                          ;
                                              05  13 00047       BEQL    4$                                              ;
                                          0C  BC  D5 00049       TSTL    @LINE_ADV                                       ; 2639
                                              7C  19 0004C       BLSS    11$                                             ;
                                14    AE    2E    A6  9A 0004E 4$:    MOVZBL  46(DCB), REND_CODE                      ; 2648
                                      04              6C  91 00053       CMPB    (AP), #4                                ;
                                              0A  1F 00056       BLSSU   5$                                              ;
                                          10  AC  D5 00058       TSTL    16(AP)                                          ;
                                              05  13 0005B       BEQL    5$                                              ;
                                14    AE        10  BC  C8 0005D       BISL2   @RENDITION_SET, REND_CODE                   ;
```

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

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

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

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

```
                                        56  DD 00332          PUSHL   DCB                                     ;
                          00000000G     00  03  FB 00334          CALLS   #3, SMG$$CHECK_FOR_OUTPUT_DCB       ;
                                            04  00033B          RET                                         ; 2895
                                    50      01  D0 00033C 45$:    MOVL    #1, R0                              ;
                                            04  00033F          RET                                         ; 2897

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

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

```
;  2653          2899  1 %SBTTL 'SMG$PUT_WITH_SCROLL - Scroll virtual display and output'
;  2654          2900  1 GLOBAL ROUTINE SMG$PUT_WITH_SCROLL (
;  2655          2901  1                             DISPLAY_ID,
;  2656          2902  1                             STRING,
;  2657          2903  1                             DIRECTION,
;  2658          2904  1                             RENDITION_SET,
;  2659          2905  1                             RENDITION_COMPLEMENT,
;  2660          2906  1                             WRAP_FLAG,
;  2661          2907  1                             CHAR_SET
;  2662          2908  1                                     ) =
;  2663          2909  1 !++
;  2664          2910  1 ! FUNCTIONAL DESCRIPTION:
;  2665          2911  1 !
;  2666          2912  1 !       This routine scrolls a virtual display, if necessary,  and deposits
;  2667          2913  1 !       the string in the opened area.  If no string is specified, the opened
;  2668          2914  1 !       line is left blank.  This routine imitates a physical scrolling region
;  2669          2915  1 !       in a virtual display;  scrolling occurs when all the lines in the
;  2670          2916  1 !       virtual display have been filled.  A new line is opened at the bottom
;  2671          2917  1 !       of the virtual display for upscrolling, and at the top of the virtual
;  2672          2918  1 !       display for downscrolling.
;  2673          2919  1 !
;  2674          2920  1 !       Treatment of text which exceeds the rightmost boundary of the display
;  2675          2921  1 !       depends on the setting of WRAP_FLAG.  If WRAP_FLAG is on, another line
;  2676          2922  1 !       scroll occurs and overflow characters are written in the new space.
;  2677          2923  1 !       If WRAP_FLAG is off, overflow characters are lost.
;  2678          2924  1 !
;  2679          2925  1 !       The internal display cursor position is left at the character
;  2680          2926  1 !       position following the last character written.
;  2681          2927  1 !
;  2682          2928  1 !       See SMG$INSERT_LINE for inserting lines at locations other than the
;  2683          2929  1 !       first or last line of a virtual display.
;  2684          2930  1 !
;  2685          2931  1 ! CALLING SEQUENCE:
;  2686          2932  1 !
;  2687          2933  1 !       ret_status.wlc.v = SMG$PUT_WITH_SCROLL (
;  2688          2934  1 !                                  DISPLAY_ID.rl.r
;  2689          2935  1 !                                  [,STRING.rt.dx]
;  2690          2936  1 !                                  [,DIRECTION.rlu.r]
;  2691          2937  1 !                                  [,RENDITION_SET.rl.r]
;  2692          2938  1 !                                  [,RENDITION_COMPLEMENT.rl.r]
;  2693          2939  1 !                                  [,WRAP_FLAG.rl.r]
;  2694          2940  1 !                                  [,CHAR_SET.rl.r])
;  2695          2941  1 !
;  2696          2942  1 ! FORMAL PARAMETERS:
;  2697          2943  1 !
;  2698          2944  1 !       DISPLAY_ID.rl.r         Display id of desired virtual display.
;  2699          2945  1 !
;  2700          2946  1 !       STRING.rt.dx            Optional.  String to be deposited in the opened
;  2701          2947  1 !                               line.  If omitted, a blank line is the default.
;  2702          2948  1 !
;  2703          2949  1 !       DIRECTION.rlu.r         Optional.  Direction to scroll.  Values:
;  2704          2950  1 !                                       SMG$M_UP   (up)   Default
;  2705          2951  1 !                                       SMG$M_DOWN (down)
;  2706          2952  1 !
;  2707          2953  1 !       RENDITION_SET.rl.r      Optional.  Each 1 bit attribute in this
;  2708          2954  1 !                               parameter causes the corresponding attribute
;  2709          2955  1 !                               to be set in the display.  (See below for
```

```
;  2710          2956  1 !                                   list of settable attributes.)
;  2711          2957  1 !
;  2712          2958  1 !       RENDITION_COMPLEMENT.rl.r
;  2713          2959  1 !                                   Optional.  Each 1 bit attribute in this
;  2714          2960  1 !                                   parameter causes the corresponding attribute
;  2715          2961  1 !                                   to be complemented in the display.  (See
;  2716          2962  1 !                                   below for list of complementable
;  2717          2963  1 !                                   attributes.)
;  2718          2964  1 !
;  2719          2965  1 !       If the same bit is specified in both the RENDITION_SET parameter
;  2720          2966  1 !       and in the RENDITION_COMPLEMENT parameter, the application is
;  2721          2967  1 !       RENDITION_SET followed by RENDITION complement.  Using these two
;  2722          2968  1 !       parameters together the caller can exercise arbitrary and
;  2723          2969  1 !       independent control over each attribute on a single call.  On an
;  2724          2970  1 !       attribute by attribute basis he can cause the following
;  2725          2971  1 !       transformations:
;  2726          2972  1 !
;  2727          2973  1 !               SET     COMPLEMENT      Action
;  2728          2974  1 !               ---     ----------      ------
;  2729          2975  1 !               0       0               Attribute unchanged.
;  2730          2976  1 !               1       0               Attribute set to "on"
;  2731          2977  1 !               0       1               Attribute set to complement of
;  2732          2978  1 !                                       current setting.
;  2733          2979  1 !               1       1               Attribute set to "off".
;  2734          2980  1 !
;  2735          2981  1 !
;  2736          2982  1 !       Attributes which can be manipulated in this manner are:
;  2737          2983  1 !
;  2738          2984  1 !       SMG$M_BLINK  displays characters blinking.
;  2739          2985  1 !       SMG$M_BOLD  displays characters in higher-than-normal
;  2740          2986  1 !                   intensity.
;  2741          2987  1 !       SMG$M_REVERSE  displays characters in reverse video -- that is,
;  2742          2988  1 !                   using the opposite default rendition of the
;  2743          2989  1 !                   virtual display.
;  2744          2990  1 !       SMG$M_UNDERLINE  displays characters underlined.
;  2745          2991  1 !
;  2746          2992  1 !
;  2747          2993  1 !       WRAP_FLAG.rl.r  Optional.
;  2748          2994  1 !                       = 0 means no wrap
;  2749          2995  1 !                       = 1 means wrap
;  2750          2996  1 !                       If omitted, no wrap is the default.
;  2751          2997  1 !
;  2752          2998  1 !       CHAR_SET.rl.r   Optional.  Character set to use.
;  2753          2999  1 !                       Choices are:
;  2754          3000  1 !                               SMG$C_UNITED_KINGDOM
;  2755          3001  1 !                               SMG$C_ASCII (default)
;  2756          3002  1 !                               SMG$C_SPEC_GRAPHICS
;  2757          3003  1 !                               SMG$C_ALT_CHAR
;  2758          3004  1 !                               SMG$C_ALT_GRAPHICS
;  2759          3005  1 !
;  2760          3006  1 ! IMPLICIT INPUTS:
;  2761          3007  1 !
;  2762          3008  1 !       NONE
;  2763          3009  1 !
;  2764          3010  1 ! IMPLICIT OUTPUTS:
;  2765          3011  1 !
;  2766          3012  1 !       NONE
```

```
;2767            ;013  1 !
;2768            3014  1 ! COMPLETION STATUS:
;2769            3015  1 !
;2770            3016  1 !       SS$_NORMAL      Normal successful completion
;2771            3017  1 !       LIB$_INVSTRDES  Invalid string descriptor
;2772            3018  1 !       SMG$_INVDIS_ID  Invalid display id.
;2773            3019  1 !       SMG$_WRONUMARG  Wrong number of arguments.
;2774            3020  1 !                       or ACTION is given but RENDITION is not.
;2775            3021  1 !       SMG$_INVARG     Unrecognized action code
;2776            3022  1 !                       or Unrecognized rendition code
;2777            3023  1 !
;2778            3024  1 ! SIDE EFFECTS:
;2779            3025  1 !
;2780            3026  1 !       NONE
;2781            3027  1 !--
;2782            3028  2     BEGIN
;2783            3029  2
;2784          M 3030  2     MACRO $SCROLL_UP (COUNT) =
;2785          M 3031  2             SMG$$SCROLL_AREA (.DCB,
;2786          M 3032  2                               .DCB [DCB_W_TOP_OF_SCRREG],
;2787          M 3033  2                               .DCB [DCB_W_COL_START],
;2788          M 3034  2                               (.DCB [DCB_W_BOTTOM_OF_SCRREG] -
;2789          M 3035  2                                .DCB [DCB_W_TOP_OF_SCRREG] + 1),
;2790          M 3036  2                               .DCB [DCB_W_NO_COLS],
;2791            3037  2                               SMG$M_UP, COUNT) %;
;2792            3038  2
;2793          M 3039  2     MACRO $SCROLL_DOWN (COUNT) =
;2794          M 3040  2             SMG$$SCROLL_AREA (.DCB,
;2795          M 3041  2                               .DCB [DCB_W_TOP_OF_SCRREG],
;2796          M 3042  2                               .DCB [DCB_W_COL_START],
;2797          M 3043  2                               (.DCB [DCB_W_BOTTOM_OF_SCRREG] -
;2798          M 3044  2                                .DCB [DCB_W_TOP_OF_SCRREG] + 1),
;2799          M 3045  2                               .DCB [DCB_W_NO_COLS],
;2800            3046  2                               SMG$M_DOWN, COUNT) %;
;2801            3047  2
;2802            3048  2     BUILTIN
;2803            3049  2         NULLPARAMETER;
;2804            3050  2
;2805            3051  2     LITERAL
;2806            3052  2         K_STRING_ARG = 2,
;2807            3053  2         K_SET_ARG = 4,
;2808            3054  2         K_COMP_ARG = 5;
;2809            3055  2
;2810            3056  2     LOCAL
;2811            3057  2         DONE,                               ! Done flag
;2812            3058  2         STATUS,                             ! Status of subroutine calls
;2813            3059  2         DCB : REF $DCB_DECL,                ! Addr of display control block
;2814            3060  2         STR_LEN : INITIAL (0),              ! Length of string
;2815            3061  2         STR_ADDR,                           ! Addr of string
;2816            3062  2         WORK_DIR,                           ! Working direction
;2817            3063  2         WRAP_CHARS : INITIAL (0),           ! No. of overflow characters
;2818            3064  2         SCROLL_FLAG,                        ! Flag to scroll up, down, or neither
;2819            3065  2         REND_CODE;                          ! Rendition code from params
;2820            3066  2
;2821            3067  2     $SMG$VALIDATE_ARGCOUNT (1, 7);
;2822            3068  2
;2823            3069  2     $SMG$GET_DCB ( .DISPLAY_ID, DCB);   ! Get address of display control
```

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

```
; 2881          3127  2          IF .SCROLL_FLAG EQL 2 AND                                                                                  ; Rc
; 2882          3128  2             .WORK_DIR EQL SMG$M_DOWN
; 2883          3129  2          THEN
; 2884          3130  2              $SCROLL_DOWN (1);   ! on top line
; 2885          3131  2
; 2886          3132  2      !+
; 2887          3133  2      ! Reset the line characteristics in case the line was previously                                            ; 32
; 2888          3134  2      ! double high or wide.
; 2889          3135  2      !-
; 2890          3136  2
; 2891          3137  3          BEGIN
; 2892          3138  3          BIND
; 2893          3139  3              LINE_CHAR = .DCB [DCB_A_LINE_CHAR];
; 2894          3140  3          MAP
; 2895          3141  3              LINE_CHAR : VECTOR [,BYTE];
; 2896          3142  3          LINE_CHAR [.DCB [DCB_W_CURSOR_ROW]] = 0;
; 2897          3143  2          END;
; 2898          3144  2
; 2899          3145  2      !+
; 2900          3146  2      ! Write text into the space.  SMG$$PUT_TEXT_TO_BUFFER checks for
; 2901          3147  2      ! funny characters (tabs, etc) and also updates the cursor position.
; 2902          3148  2      !-
; 2903          3149  2
; 2904          3150  2          IF .STR_LEN NEQ 0
; 2905          3151  2          THEN
; 2906          3152  3              BEGIN
; 2907          3153  3              $SMG$SET_REND_CODE (K_SET_ARG, K_COMP_ARG);
; 2908          3154  3
; 2909          3155  3              DCB [DCB_W_CURSOR_COL] = .DCB [DCB_W_COL_START];
; 2910          3156  3                                       ! set cursor for PUT_TEXT
; 2911          3157  3              SMG$$PUT_TEXT_TO_BUFFER (.DCB, .REND_CODE,
; 2912          3158  3                                       .STR_LEN,
; 2913          3159  3                                       .STR_ADDR,
; 2914          3160  4                                       (IF NULLPARAMETER( CHAR_SET )
; 2915          3161  4                                            THEN SMG$C_ASCII
; 2916          3162  3                                            ELSE ..CHAR_SET),
; 2917          3163  3                                       WRAP_CHARS);
; 2918          3164  2              END;
; 2919          3165  2
; 2920          3166  2      !+
; 2921          3167  2      ! Update the cursor position.  Set bits in the DCB which may affect
; 2922          3168  2      ! future scrolling.
; 2923          3169  2      !-
; 2924          3170  2
; 2925          3171  2          DCB [DCB_W_CURSOR_COL] = 1;
; 2926          3172  2
; 2927          3173  2          CASE .WORK_DIR FROM SMG$M_UP TO SMG$M_DOWN OF
; 2928          3174  2          SET
; 2929          3175  2              [SMG$M_UP]:
; 2930          3176  3                  BEGIN
; 2931          3177  3                  IF .DCB [DCB_W_CURSOR_ROW] + 1 LEQ .DCB [DCB_W_BOTTOM_OF_SCRREG]
; 2932          3178  3                  THEN
; 2933          3179  3                      DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_CURSOR_ROW] + 1
; 2934          3180  3                  ELSE
; 2935          3181  3                      DCB [DCB_V_FULL] = 1;
; 2936          3182  2                  END;
; 2937          3183  2
```

```
; 2938     3184  2          [SMG$M_DOWN]:
; 2939     3185  3              BEGIN
; 2940     3186  3              IF .DCB [DCB_W_CURSOR_ROW] - 1 GEQ .DCB [DCB_W_TOP_OF_SCRREG]
; 2941     3187  3              THEN
; 2942     3188  3                  DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_CURSOR_ROW] - 1
; 2943     3189  3              ELSE
; 2944     3190  3                  DCB [DCB_V_FULL] = 1;
; 2945     3191  2              END;
; 2946     3192  2      TES;
; 2947     3193  2
; 2948     3194  2  !+
; 2949     3195  2  ! Check for overflow characters which should cause us to scroll 2 lines
; 2950     3196  2  ! in order to output everything (if wrap requested).
; 2951     3197  2  !-
; 2952     3198  2
; 2953     3199  2      IF .WRAP_CHARS NEQ 0
; 2954     3200  2      THEN
; 2955     3201  3          BEGIN
; 2956     3202  3          IF NOT NULLPARAMETER( WRAP_FLAG ) AND
; 2957     3203  3              ..WRAP_FLAG NEQ 0               ! wrap requested
; 2958     3204  3          THEN
; 2959     3205  4              BEGIN                           ! recurse until whole string output
; 2960     3206  4              LOCAL
; 2961     3207  4                  TEMP_DESC : BLOCK [8, BYTE],
; 2962     3208  4                  C_SET;
; 2963     3209  4
; 2964     3210  4              TEMP_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
; 2965     3211  4              TEMP_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
; 2966     3212  4              TEMP_DESC [DSC$W_LENGTH] = .STR_LEN - .DCB [DCB_W_NO_COLS];
; 2967     3213  4              TEMP_DESC [DSC$A_POINTER] = .STR_ADDR +
; 2968     3214  4                                          .DCB [DCB_W_NO_COLS];
; 2969     3215  4
; 2970     3216  4              C_SET = SMG$C_ASCII;
; 2971     3217  4              IF NOT NULLPARAMETER( CHAR_SET )
; 2972     3218  4              THEN
; 2973     3219  4                  C_SET = ..CHAR_SET;
; 2974     3220  4
; 2975     3221  4              !+
; 2976     3222  4              ! Notice that we always upscroll for wrapped text.  This is
; 2977     3223  4              ! because downscrolling would cause the last part of the line
; 2978     3224  4              ! to be displayed before the first part.  This approach may not
; 2979     3225  4              ! be perfect but it at least lines appear in a readable manner.
; 2980     3226  4              !-
; 2981     3227  4              IF .WORK_DIR EQL SMG$M_DOWN
; 2982     3228  4              THEN
; 2983     3229  5                  BEGIN
; 2984     3230  5                  LOCAL
; 2985     3231  5                      OVERFLOW_LINE;
; 2986     3232  5                  OVERFLOW_LINE = .DCB [DCB_W_CURSOR_ROW] + 1;
; 2987     3233  5                  SMG$INSERT_LINE (.DISPLAY_ID,
; 2988     3234  5                                   OVERFLOW_LINE,
; 2989     3235  5                                   TEMP_DESC, %REF (SMG$M_DOWN),
; 2990     3236  6                                   (IF NOT NULLPARAMETER (RENDITION_SET)
; 2991     3237  6                                        THEN .RENDITION_SET
; 2992     3238  5                                        ELSE 0),
; 2993     3239  6                                   (IF NOT NULLPARAMETER (RENDITION_COMPLEMENT)
; 2994     3240  6                                        THEN .RENDITION_COMPLEMENT
```

```
;  2995     3241  5                                        ELSE 0),
;  2996     3242  5                                    .WRAP_FLAG, C_SET);
;  2997     3243  5                DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_ROW_START];
;  2998     3244  5                END
;  2999     3245  4            ELSE
;  3000     3246  4                SMG$PUT_WITH_SCROLL (.DISPLAY_ID,
;  3001     3247  4                                     TEMP_DESC, %REF (SMG$M_UP),
;  3002     3248  5                                     (IF NOT NULLPARAMETER (RENDITION_SET)
;  3003     3249  5                                         THEN .RENDITION_SET
;  3004     3250  4                                         ELSE 0),
;  3005     3251  5                                     (IF NOT NULLPARAMETER (RENDITION_COMPLEMENT)
;  3006     3252  5                                         THEN .RENDITION_COMPLEMENT
;  3007     3253  4                                         ELSE 0),
;  3008     3254  4                                     .WRAP_FLAG,
;  3009     3255  4                                     C_SET);
;  3010     3256  4            DONE = 0;
;  3011     3257  4            RETURN 1;                   ! to keep Bliss happy
;  3012     3258  4            END
;  3013     3259  3        ELSE
;  3014     3260  4            BEGIN
;  3015     3261  4            !+
;  3016     3262  4            ! Wrap was not requested but there were overflow characters.
;  3017     3263  4            ! Put out diamond in last position to show truncation.
;  3018     3264  4            !-
;  3019     3265  4            IF .DCB [DCB_V_TRUNC_ICON]
;  3020     3266  4            THEN                                                          ; Re
;  3021     3267  5                BEGIN
;  3022     3268  5                IF NOT .DCB [DCB_V_FULL]
;  3023     3269  5                THEN
;  3024     3270  5                    DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_CURSOR_ROW] - 1;
;  3025     3271  5                DCB [DCB_W_CURSOR_COL] = .DCB [DCB_W_NO_COLS];
;  3026     3272  5
;  3027     3273  5                SMG$$PUT_TEXT_TO_BUFFER (.DCB,
;  3028     3274  5                                         .REND_CODE + ATTR_M_USER_GRAPHIC,
;  3029     3275  5                                         1, UPLIT (BYTE (DIAMOND)),
;  3030     3276  5                                         SMG$C_ASCII);
;  3031     3277  5
;  3032     3278  5                IF NOT .DCB [DCB_V_FULL]
;  3033     3279  5                THEN
;  3034     3280  5                    DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_CURSOR_ROW] + 1;
;  3035     3281  5                                         ! restore for next call
;  3036     3282  4                END;
;  3037     3283  4
;  3038     3284  4                DONE = 1;
;  3039     3285  3                END;
;  3040     3286  3            END
;  3041     3287  2        ELSE
;  3042     3288  2            DONE = 1;                   ! no wrap chars
;  3043     3289  2
;  3044     3290  2    !+
;  3045     3291  2    ! See if this change should be reflected on the screen.
;  3046     3292  2    ! Even if SMG$PUT_WITH_SCROLL is called multiple times, call
;  3047     3293  2    ! SMG$$CHECK_FOR_OUTPUT_DCB only once.
;  3048     3294  2    !-
;  3049     3295  2
;  3050     3296  2        IF .DONE
;  3051     3297  2        THEN
```

```
;  3052          3298  3          RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
;  3053          3299  3                                             SMG$C_PUT_WITH_SCROLL))
;  3054          3300  2      ELSE
;  3055          3301  2          RETURN 1;
;  3056          3302  2
;  3057          3303  1      END;                    ! End of routine SMG$PUT_WITH_SCROLL


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

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

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

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

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

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

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

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

```
; 3060          3305  1 %SBTTL 'SMG$READ_FROM_DISPLAY - Read text from virtual display'
; 3061          3306  1 GLOBAL ROUTINE SMG$READ_FROM_DISPLAY (
; 3062          3307  1                                       DISPLAY_ID,
; 3063          3308  1                                       RET_STRING,
; 3064          3309  1                                       SEARCH_STRING
; 3065          3310  1                                      ) =
; 3066          3311  1 !++
; 3067          3312  1 ! FUNCTIONAL DESCRIPTION:
; 3068          3313  1 !
; 3069          3314  1 !       This procedure returns back to the caller a text string which
; 3070          3315  1 !       represents the current contents of a portion of a virtual
; 3071          3316  1 !       display line.  The current cursor position row determines which
; 3072          3317  1 !       line is to act as the source of the text.
; 3073          3318  1 !       If the SEARCH_STRING parameter is omitted, the contents of
; 3074          3319  1 !       the current line, from the current cursor column to the
; 3075          3320  1 !       right-most column, is returned in RET_STRING.
; 3076          3321  1 !
; 3077          3322  1 !       If SEARCH_STRING is provided, it is a list of character codes
; 3078          3323  1 !       that are to act as terminators in "back searching" -- the
; 3079          3324  1 !       process of determining the first character position to be
; 3080          3325  1 !       returned.  Each character in this string serves as a potential
; 3081          3326  1 !       terminator.  If no terminator is encountered during "back
; 3082          3327  1 !       searching", the search is terminated by column 1 of the
; 3083          3328  1 !       virtual display.
; 3084          3329  1 !
; 3085          3330  1 !       Some examples:
; 3086          3331  1 !               Assume the line of text which contains the current
; 3087          3332  1 !       cursor contains the text:
; 3088          3333  1 !
; 3089          3334  1 !       "$ COPY/LOG FOO1 FOO2     "
; 3090          3335  1 !                 ^
; 3091          3336  1 !
; 3092          3337  1 !       Assume the cursor is positioned at the "G".
; 3093          3338  1 !
; 3094          3339  1 !       SMG$READ_FROM_DISPLAY (Display_id, OUTSTRING )
; 3095          3340  1 !       will return "G FOO1 FOO2      " since no terminator was supplied,
; 3096          3341  1 !       and hence no "back-searching" is performed.
; 3097          3342  1 !
; 3098          3343  1 !       SMG$READ_FROM_DISPLAY (Display_id, OUTSTRING, '/' )
; 3099          3344  1 !       will return "7LOG FOO1 FOO2     " since the specified terminator                                        ; R
; 3100          3345  1 !       was found prior to encountering column 1.
; 3101          3346  1 !
; 3102          3347  1 !       SMG$READ_FROM_DISPLAY (Display_id, OUTSTRING, '' )                                                      ; 3
; 3103          3348  1 !       will return "$ COPY/LOG FOO1 FOO2     " since the specified
; 3104          3349  1 !       terminator was not found prior to encountering column 1.
; 3105          3350  1 !
; 3106          3351  1 !       SMG$READ_FROM_DISPLAY (Display_id, OUTSTRING, '#' )
; 3107          3352  1 !       will return "$ COPY/LOG FOO1 FOO2     " since the specified
; 3108          3353  1 !       terminator was not seen before column 1 was encountered.
; 3109          3354  1 !
; 3110          3355  1 !       SMG$READ_FROM_DISPLAY (Display_id, OUTSTRING, "Y/" )
; 3111          3356  1 !       will return "7LOG FOO1 FOO2     " since, of the specified
; 3112          3357  1 !       terminators, the "/" was encountered before column 1 or any of
; 3113          3358  1 !       the other terminators.
; 3114          3359  1 !
; 3115          3360  1 !       NOTE: The "back-searching" begins at the current cursor column.
; 3116          3361  1 !       Hence,  if the text string is:
```

```
;  3117          3362  1 !       "$COPY/LOG F001_F002     " and cursor is positioned
;  3118          3363  1 !
;  3119          3364  1 !
;  3120          3365  1 !       SMG$READ_FROM_DISPLAY (Display_id, OUTSTRING, ' ' ) will
;  3121          3366  1 !       return "F002     ".
;  3122          3367  1 !
;  3123          3368  1 ! CALLING SEQUENCE:
;  3124          3369  1 !
;  3125          3370  1 !       ret_status.wlc.v = SMG$READ_FROM_DISPLAY (
;  3126          3371  1 !                                   DISPLAY_ID.rl.r,
;  3127          3372  1 !                                   RET_STRING.wt.dx
;  3128          3373  1 !                                   [,SEARCH_STRING.rt.dx])
;  3129          3374  1 !
;  3130          3375  1 ! FORMAL PARAMETERS:
;  3131          3376  1 !
;  3132          3377  1 !       DISPLAY_ID.rl.r         Address of a display id.
;  3133          3378  1 !
;  3134          3379  1 !       RET_STRING.wt.dx        Address of a string descriptor,
;  3135          3380  1 !                               describing the output string to be
;  3136          3381  1 !                               produced.
;  3137          3382  1 !
;  3138          3383  1 !       SEARCH_STRING.rt.dx     [Optional].  If specified, the address
;  3139          3384  1 !                               of a string descriptor, describing a
;  3140          3385  1 !                               list of terminators which is to stop the
;  3141          3386  1 !                               backward search in determining the
;  3142          3387  1 !                               starting position of the string to be
;  3143          3388  1 !                               returned.
;  3144          3389  1 !                               If omitted, the returned string starts
;  3145          3390  1 !                               at the current cursor column position --
;  3146          3391  1 !                               no back searching is performed.
;  3147          3392  1 !
;  3148          3393  1 ! IMPLICIT INPUTS:
;  3149          3394  1 !
;  3150          3395  1 !       NONE
;  3151          3396  1 !
;  3152          3397  1 ! IMPLICIT OUTPUTS:
;  3153          3398  1 !
;  3154          3399  1 !       NONE
;  3155          3400  1 !
;  3156          3401  1 ! COMPLETION STATUS:
;  3157          3402  1 !
;  3158          3403  1 !       SS$_NORMAL      Normal successful completion
;  3159          3404  1 !       LIB$_INVSTRDES  Invalid string descriptor
;  3160          3405  1 !       LIB$_INSVIRMEM  Insufficient virtual memory
;  3161          3406  1 !
;  3162          3407  1 ! SIDE EFFECTS:
;  3163          3408  1 !
;  3164          3409  1 !       NONE
;  3165          3410  1 !--
;  3166          3411  1
;  3167          3412  2     BEGIN
;  3168          3413  2     BUILTIN
;  3169          3414  2         NULLPARAMETER;
;  3170          3415  2
;  3171          3416  2     LOCAL
;  3172          3417  2         STATUS,                 ! Status of subroutine calls
;  3173          3418  2
```

```
;  3174          3419  2          START_COL,                  ! Computed column number where source-
;  3175          3420  2                                      ! to-be-returned starts.
;  3176          3421  2
;  3177          3422  2          START_LOC : REF VECTOR [1, BYTE], ! Address of text byte
;  3178          3423  2                                      ! currently being inspected
;  3179          3424  2
;  3180          3425  2          DCB : REF $DCB_DECL;        ! Address of Display Control Block
;  3181          3426  2
;  3182          3427  2      !+
;  3183          3428  2      ! Validate number of arguments and extract corresponding DCB address
;  3184          3429  2      !-
;  3185          3430  2          $SMG$VALIDATE_ARGCOUNT (1, 3);
;  3186          3431  2          $SMG$GET_DCB (.DISPLAY_ID, DCB)
;  3187          3432  2
;  3188          3433  2          IF NOT NULLPARAMETER (SEARCH_STRING)
;  3189          3434  2          THEN
;  3190          3435  3              BEGIN   ! Terminator string supplied
;  3191          3436  3              LOCAL
;  3192          3437  3                  T_LEN,              ! Length of terminator string
;  3193          3438  3
;  3194          3439  3                  T_ADDR : REF VECTOR [,BYTE];        ! Address of 1st byte
;  3195          3440  3                                                      ! of terminator string
;  3196          3441  3
;  3197          3442  3              !+
;  3198          3443  3              ! Extract length and address of 1st byte of terminator string.
;  3199          3444  3              !-
;  3200          3445  4              IF NOT (STATUS = LIB$ANALYZE_SDESC_R2 ( .SEARCH_STRING ;
;  3201          3446  4                                                      T_LEN, T_ADDR))
;  3202          3447  3              THEN
;  3203          3448  3                  RETURN (.STATUS);   ! Can't access terminator string
;  3204          3449  3
;  3205          3450  3              !+
;  3206          3451  3              ! Initialize pointer for searching.
;  3207          3452  3              !-
;  3208          3453  3              START_LOC =
;  3209          3454  3                      (.DCB [DCB_W_CURSOR_ROW] - 1) * .DCB [DCB_W_NO_COLS] +
;  3210          3455  3                       .DCB [DCB_W_CURSOR_COL] - 1 + .DCB [DCB_A_TEXT_BUF];
;  3211          3456  3
;  3212          3457  3              START_COL = .DCB [DCB_W_CURSOR_COL];
;  3213          3458  3              WHILE .START_COL NEQ 1
;  3214          3459  3              DO
;  3215          3460  4                  BEGIN       ! Back search
;  3216          3461  4                  !+
;  3217          3462  4                  ! If text byte currently under inspection is one of the
;  3218          3463  4                  ! the terminator bytes, exit from loop.
;  3219          3464  4                  !-
;  3220          3465  4                  INCR I FROM 0 TO .T_LEN -1
;  3221          3466  4                  DO
;  3222          3467  5                      BEGIN   ! Try to match terminator loop
;  3223          3468  5                      IF .START_LOC [0] EQL .T_ADDR [.I]
;  3224          3469  5                      THEN
;  3225          3470  6                          BEGIN       ! Terminator found exit path
;  3226          3471  7                          RETURN (LIB$SCOPY_R_DX6 (
;  3227          3472  7                              .DCB [DCB_W_NO_COLS] - .START_COL + 1, ! # bytes
;  3228          3473  7                              .START_LOC,                            ! from
;  3229          3474  6                              .RET_STRING));                         ! To desc
;  3230          3475  5                          END;        ! Terminator found exit path
```

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


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

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

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

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

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

```
;  3274          3518  1 %SBTTL 'SMG$RETURN_CURSOR_POS - Return current cursor position'
;  3275          3519  1 GLOBAL ROUTINE SMG$RETURN_CURSOR_POS (
;  3276          3520  1                                       DISPLAY_ID,
;  3277          3521  1                                       ROW_NUM,
;  3278          3522  1                                       COL_NUM
;  3279          3523  1                                      ) =
;  3280          3524  1 !++
;  3281          3525  1 ! FUNCTIONAL DESCRIPTION:
;  3282          3526  1 !
;  3283          3527  1 !       Returns the current cursor row and column position in the
;  3284          3528  1 !       specified virtual display.
;  3285          3529  1 !
;  3286          3530  1 ! CALLING SEQUENCE:
;  3287          3531  1 !
;  3288          3532  1 !       ret_status.wlc.v = SMG$RETURN_CURSOR_POS (
;  3289          3533  1 !                                                 DISPLAY_ID.rl.r,
;  3290          3534  1 !                                                 ROW_NUM.wl.r,
;  3291          3535  1 !                                                 COL_NUM.wl.r)
;  3292          3536  1 !
;  3293          3537  1 ! FORMAL PARAMETERS:
;  3294          3538  1 !
;  3295          3539  1 !       DISPLAY_ID.rl.r Display id of desired display.
;  3296          3540  1 !
;  3297          3541  1 !       ROW_NUM.wl.r    Returned row position.
;  3298          3542  1 !
;  3299          3543  1 !       COL_NUM.w.r     Returned column position.
;  3300          3544  1 !
;  3301          3545  1 ! IMPLICIT INPUTS:
;  3302          3546  1 !
;  3303          3547  1 !       NONE
;  3304          3548  1 !
;  3305          3549  1 ! IMPLICIT OUTPUTS:
;  3306          3550  1 !
;  3307          3551  1 !       NONE
;  3308          3552  1 !
;  3309          3553  1 ! COMPLETION STATUS:
;  3310          3554  1 !
;  3311          3555  1 !       SS$_NORMAL      Normal successful completion
;  3312          3556  1 !       SMG$_WRONUMARG  Wrong number arguments.
;  3313          3557  1 !       SMG$_INVDIS_ID  Invalid display id.
;  3314          3558  1 !
;  3315          3559  1 ! SIDE EFFECTS:
;  3316          3560  1 !
;  3317          3561  1 !       NONE
;  3318          3562  1 !--
;  3319          3563  2     BEGIN
;  3320          3564  2     LOCAL
;  3321          3565  2         DCB : REF $DCB_DECL;            ! Addr. of display control block
;  3322          3566  2                                                                          ; R
;  3323          3567  2     $SMG$VALIDATE_ARGCOUNT (3,3);       ! Check for right no. of args
;  3324          3568  2
;  3325          3569  2     $SMG$GET_DCB ( .DISPLAY_ID, DCB);   ! Get address of display control   ; 3
;  3326          3570  2                                         ! block
;  3327          3571  2
;  3328          3572  2     .ROW_NUM = .DCB [DCB_W_CURSOR_ROW];
;  3329          3573  2     .COL_NUM = .DCB [DCB_W_CURSOR_COL];
;  3330          3574  2
```

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


                                                  0000 00000         .ENTRY  SMG$RETURN_CURSOR_POS, Save nothing          ; 3519
                                        03        6C   91 00002         CMPB    (AP), #3                                     ; 3567
                                                  08   13 00005         BEQL    1$                                           ;
                                        50 00000000G    8F   DD 00007         PUSHL   #SMG$_WRONUMARG                      ;
                                                        04   DD 00009         PUSHL   #1                                   ;
                                                        04   FB 0000B         CALLS   #1, G^LIB$SIGNAL                     ;
                                                        04   0000E         RET                                          ;
                                        50        04    BC   D0 0000F 1$:     MOVL    @DISPLAY_ID, R0                      ; 3569
                                  04    BC        38    A0   D1 00013         CMPL    56(R0), @DISPLAY_ID                  ;
                                                        06   12 00018         BNEQ    2$                                   ;
                                        11        44    A0   91 0001A         CMPB    68(R0), #17                          ;
                                                        08   13 0001E         BEQL    3$                                   ;
                                        50 00000000G    8F   D0 00020 2$:     MOVL    #SMG$_INVDIS_ID, R0                  ;
                                                        04   00027         RET                                          ;
                                        50        04    BC   D0 00028 3$:     MOVL    @DISPLAY_ID, DCB                     ;
                                  08    BC        28    A0   3C 0002C         MOVZWL  40(DCB), @ROW_NUM                    ; 3572
                                  0C    BC        2A    A0   3C 00031         MOVZWL  42(DCB), @COL_NUM                    ; 3573
                                        50              01   D0 00036         MOVL    #1, R0                               ; 3575
                                                        04   00039         RET                                          ; 3576

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

```
;  3334          3577  1 %SBTTL 'SMG$SCROLL_DISPLAY_AREA - Scroll rectangular area in a virtual display'
;  3335          3578  1 GLOBAL ROUTINE SMG$SCROLL_DISPLAY_AREA (
;  3336          3579  1                                       DISPLAY_ID,
;  3337          3580  1                                       START_ROW,
;  3338          3581  1                                       START_COL,
;  3339          3582  1                                       HEIGHT,
;  3340          3583  1                                       WIDTH,
;  3341          3584  1                                       DIRECTION,
;  3342          3585  1                                       COUNT
;  3343          3586  1                                      ) =
;  3344          3587  1 !++
;  3345          3588  1 ! FUNCTIONAL DESCRIPTION:
;  3346          3589  1 !
;  3347          3590  1 !       This routine scrolls the contents of a rectangular area in a virtual
;  3348          3591  1 !       display in the specified direction for the specified count.  By default
;  3349          3592  1 !       the scrolling rectangle is the 'scrolling region' in the virtual
;  3350          3593  1 !       display.
;  3351          3594  1 !
;  3352          3595  1 !       The default direction is 'up' and the default number of units is 1.
;  3353          3596  1 !
;  3354          3597  1 !       The scrolling rectangle must lie entirely
;  3355          3598  1 !       within the virtual display, else SMG$_INVARG is returned.
;  3356          3599  1 !
;  3357          3600  1 !       If the number of units to be scrolled exceeds the size of the rectangle
;  3358          3601  1 !       in that direction the text within the rectangle simply "disappears"
;  3359          3602  1 !       at the rectangle boundary.  Locations emptied by this operation are
;  3360          3603  1 !       set to spaces.
;  3361          3604  1 !
;  3362          3605  1 !       The cursor position is left at START_ROW, START_COL.
;  3363          3606  1 !
;  3364          3607  1 ! CALLING SEQUENCE:
;  3365          3608  1 !
;  3366          3609  1 !       ret_status.wlc.v = SMG$SCROLL_DISPLAY_AREA (
;  3367          3610  1 !                                                   DISPLAY_ID.rl.r
;  3368          3611  1 !                                                   [,START_ROW.rl.r, START_COL_NUM.rl.r]
;  3369          3612  1 !                                                   [,HEIGHT.rl.r]
;  3370          3613  1 !                                                   [,WIDTH.rl.r]
;  3371          3614  1 !                                                   [,DIRECTION.rl.r]
;  3372          3615  1 !                                                   [,COUNT.rl.r]
;  3373          3616  1 !
;  3374          3617  1 ! FORMAL PARAMETERS:
;  3375          3618  1 !
;  3376          3619  1 !       DISPLAY_ID.rl.r Display id of desired display.
;  3377          3620  1 !
;  3378          3621  1 !       START_ROW.rl.r  Optional.  Row in which scrolling region starts.
;  3379          3622  1 !                       Defaults to row 1.
;  3380          3623  1 !
;  3381          3624  1 !       START_COL.rl.r  Optional.  Column in which scrolling region starts.
;  3382          3625  1 !                       Defaults to column 1.
;  3383          3626  1 !
;  3384          3627  1 !       HEIGHT.rl.r     Optional.  Height of scrolling region.  Defaults to
;  3385          3628  1 !                       the height of the scrolling region in the virtual
;  3386          3629  1 !                       display.
;  3387          3630  1 !
;  3388          3631  1 !       WIDTH.rl.r      Optional.  Width of scrolling region.  Defaults to
;  3389          3632  1 !                       the width of the scrolling region in the virtual
;  3390          3633  1 !                       display.
```

```
;  3391     3634  1 !
;  3392     3635  1 !       DIRECTION.rl.r  Optional.  Direction in which to scroll.  Defaults
;  3393     3636  1 !                       to SMG$M_UP.  Choices are:      SMG$M_UP
;  3394     3637  1 !                                                       SMG$M_DOWN
;  3395     3638  1 !                                                       SMG$M_RIGHT
;  3396     3639  1 !                                                       SMG$M_LEFT
;  3397     3640  1 !
;  3398     3641  1 !       COUNT.rl.r      Optional.  Number of positions to scroll.  Defaults
;  3399     3642  1 !                       to 1 line.
;  3400     3643  1 !
;  3401     3644  1 ! IMPLICIT INPUTS:
;  3402     3645  1 !
;  3403     3646  1 !       NONE
;  3404     3647  1 !
;  3405     3648  1 ! IMPLICIT OUTPUTS:
;  3406     3649  1 !
;  3407     3650  1 !       NONE
;  3408     3651  1 !
;  3409     3652  1 ! COMPLETION STATUS:
;  3410     3653  1 !
;  3411     3654  1 !       SS$_NORMAL      Normal successful completion
;  3412     3655  1 !       SMG$_WRONUMARG  Wrong number arguments.
;  3413     3656  1 !       SMG$_INVDIS_ID  Invalid display id.
;  3414     3657  1 !
;  3415     3658  1 ! SIDE EFFECTS:
;  3416     3659  1 !
;  3417     3660  1 !       NONE
;  3418     3661  1 !--
;  3419     3662  2     BEGIN
;  3420     3663  2     LOCAL
;  3421     3664  2         DCB : REF $DCB_DECL,            ! Addr. of display control block
;  3422     3665  2         LOCAL_ROW,                      ! working start row
;  3423     3666  2         LOCAL_COL,                      ! working start column
;  3424     3667  2         LOCAL_HEIGHT,                   ! working height
;  3425     3668  2         LOCAL_WIDTH,                    ! working width
;  3426     3669  2         LOCAL_DIR,                      ! working direction
;  3427     3670  2         LOCAL_COUNT,                    ! working count
   3428     3671  2         STATUS;                         ! status of called routine
    429     3672  2
   ,430     3673  2     BUILTIN
;  3431     3674  2         NULLPARAMETER;
;  3432     3675  2
    433     3676  2     $SMG$VALIDATE_ARGCOUNT (1,7);
;  3434     3677  2
;  3435     3678  2     $SMG$GET_DCB ( .DISPLAY_ID, DCB);   ! Get address of display control
     36     3679  2                                         ! block
;  3437     3680  2
;  3438     3681  2 !+
;  3439     3682  2 ! Validate arguments and fill in local variables.
;  3440     3683  2 !-
;  3441     3684  2
;  3442     3685  2     IF NOT NULLPARAMETER (START_ROW) AND
;  3443     3686  2        NOT NULLPARAMETER (START_COL)
;  3444     3687  2     THEN
;  3445     3688  3         BEGIN                           ! use caller's row, col
;  3446     3689  3         $SMG$VALIDATE_ROW_COL (..START_ROW, ..START_COL);
;  3447     3690  3         LOCAL_ROW = ..START_ROW;
```

```
;  3448          3691  3          LOCAL_COL = ..START_COL;
;  3449          3692  3          END
;  3450          3693  2      ELSE
;  3451          3694  3          BEGIN                       ! use default row, col
;  3452          3695  3          LOCAL_ROW = .DCB [DCB_W_TOP_OF_SCRREG];
;  3453          3696  3          LOCAL_COL = 1;
;  3454          3697  2          END;
;  3455          3698  2
;  3456          3699  2      LOCAL_HEIGHT = .DCB [DCB_W_BOTTOM_OF_SCRREG] -
;  3457          3700  2                     .LOCAL_ROW + 1; ! init to default
;  3458          3701  2
;  3459          3702  2      IF NOT NULLPARAMETER (HEIGHT)
;  3460          3703  2      THEN
;  3461          3704  3          BEGIN                       ! check caller's heigth
;  3462          3705  3          IF ..HEIGHT + .LOCAL_ROW - 1 LSS .DCB [DCB_W_ROW_START] OR
;  3463          3706  3             ..HEIGHT + .LOCAL_ROW - 1 GTR .DCB [DCB_W_NO_ROWS]
;  3464          3707  3          THEN
;  3465          3708  4              RETURN (SMG$_INVARG)
;  3466          3709  3          ELSE
;  3467          3710  3              LOCAL_HEIGHT = ..HEIGHT;
;  3468          3711  2          END;
;  3469          3712  2
;  3470          3713  2      LOCAL_WIDTH = .DCB [DCB_W_NO_COLS]; ! init to default
;  3471          3714  2
;  3472          3715  2      IF NOT NULLPARAMETER (WIDTH)
;  3473          3716  2      THEN
;  3474          3717  3          BEGIN                       ! check caller's width
;  3475          3718  3          IF ..WIDTH + .LOCAL_COL -1 LSS .DCB [DCB_W_COL_START] OR
;  3476          3719  3             ..WIDTH + .LOCAL_COL -1 GTR .DCB [DCB_W_NO_COLS]
;  3477          3720  3          THEN
;  3478          3721  4              RETURN (SMG$_INVARG)
;  3479          3722  3          ELSE
;  3480          3723  3              LOCAL_WIDTH = ..WIDTH;
;  3481          3724  2          END;
;  3482          3725  2
;  3483          3726  2      LOCAL_DIR = SMG$M_UP;
;  3484          3727  2
;  3485          3728  2      IF NOT NULLPARAMETER (DIRECTION)
;  3486          3729  2      THEN
;  3487          3730  3          BEGIN                       ! check caller's direction
;  3488          3731  3          CASE ..DIRECTION FROM SMG$M_UP TO SMG$M_LEFT OF
;  3489          3732  3          SET
;  3490          3733  3              [SMG$M_UP, SMG$M_DOWN, SMG$M_RIGHT, SMG$M_LEFT]:
;  3491          3734  3                  LOCAL_DIR = ..DIRECTION;
;  3492          3735  3
;  3493          3736  3              [INRANGE,OUTRANGE]:
;  3494          3737  3                  RETURN (SMG$_INVARG);
;  3495          3738  3          TES;
;  3496          3739  2          END;
;  3497          3740  2
;  3498          3741  2      LOCAL_COUNT = 1;                ! init to default
;  3499          3742  2
;  3500          3743  2      IF NOT NULLPARAMETER (COUNT)
;  3501          3744  2      THEN
;  3502          3745  3          BEGIN                       ! check caller's count
;  3503          3746  3          IF ..COUNT GTR 0
;  3504          3747  3          THEN
```

```
;                       3748  3              LOCAL_COUNT = ..COUNT
;                       3749  3          ELSE
;                       3750  3              RETURN (SMG$_INVARG);
;                       3751  2          END;
;                       3752  2
;                       3753  2 !+
;                       3754  2 ! Now all the arguments are set up.  Call an internal routine to do
;                       3755  2 ! the real work.
;                       3756  2 !-
;                       3757  2
;                       3758  2      STATUS = SMG$$SCROLL_AREA (.DCB,
;                       3759  2                                 .LOCAL_ROW, .LOCAL_COL,
;                       3760  2                                 .LOCAL_HEIGHT, .LOCAL_WIDTH,
;                       3761  2                                 .LOCAL_DIR, .LOCAL_COUNT);
;                       3762  2      IF NOT (.STATUS) THEN RETURN (.STATUS);
;                       3763  2
;                       3764  2 !+
;                       3765  2 ! The internal routine did not reset the cursor position.  Do that here.
;                       3766  2 !-
;                       3767  2
;                       3768  2      DCB [DCB_W_CURSOR_ROW] = .LOCAL_ROW;
;                       3769  2      DCB [DCB_W_CURSOR_COL] = .LOCAL_COL;
;                       3770  2
;                       3771  2 !+
;                       3772  2 ! Turn off scrolling since we have moved the cursor.
;                       3773  2 !-
;                       3774  2
;                       3775  2      DCB [DCB_V_FULL] = 0;
;                       3776  2
;                       3777  2 !+
;                       3778  2 ! See if this change should be reflected on the screen immediately.
;                       3779  2 !-
;                       3780  2
;                       3781  3      RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
;                       3782  2                                         SMG$C_SCROLL_DISPLAY_AREA));
;                       3783  1      END;                            ! End of routine SMG$SCROLL_DISPLAY_AREA


                                                01FC 00000          .ENTRY  SMG$SCROLL_DISPLAY_AREA, Save R2,R3,R4,R5,- ; 3578
                                                                            R6,R7,R8
                                 78 00000000G   00   9E 00002       MOVAB   SMG$_INVARG, R8
                              50 6C             01   83 00009       SUBB3   #1, (AP), DIFF                               ; 3676
                                 06             50   91 0000D       CMPB    DIFF, #6
                                                08   1B 00010       BLEQU   1$
                                 50 00000000G   8F   D0 00012       MOVL    #SMG$_WRONUMARG, R0
                                                04   00019       RET
                                 50       04    BC   D0 0001A 1$:   MOVL    @DISPLAY_ID, R0                              ; 3678
                           04    BC       38    A0   D1 0001E       CMPL    56(R0), @DISPLAY_ID
                                                06   12 00023       BNEQ    2$
                                 11       44    AC   91 00025       CMPB    68(R0), #17
                                                08   13 00029       BEQL    3$
                                 50 00000000G   8F   D0 0002B 2$:   MOVL    #SMG$_INVDIS_ID, R0
                                                04   00032       RET
                                 52       04    BC   D0 00033 3$:   MOVL    @DISPLAY_ID, DCB
```

```
                                    02            6C  91 00037          CMPB    (AP), #2                                ; 3685
                                                  43  1F 0003A          BLSSU   8$                                      ;
                                          08      AC  D5 0003C          TSTL    8(AP)                                   ;
                                                  3E  13 0003F          BEQL    8$                                      ;
                                    03            6C  91 00041          CMPB    (AP), #3                                ; 3686
                                                  39  1F 00044          BLSSU   8$                                      ;
                                          0C      AC  D5 00046          TSTL    12(AP)                                  ;
                                                  34  13 00049          BEQL    8$                                      ;
                                    51    08      BC  D0 0004B          MOVL    @START_ROW, R1                          ; 3689
                                                  08  15 0004F          BLEQ    4$                                      ;
          51        02    A2        10            00  ED 00051          CMPZV   #0, #16, 2(DCB), R1                     ;
                                                  08  18 00057          BGEQ    5$                                      ;
                                    50 00000000G  8F  D0 00059 4$:      MOVL    #SMG$_INVROW, R0                        ;
                                                  04  00060          RET                                             ;
                                    50    0C      BC  D0 00061 5$:      MOVL    @START_COL, R0                          ;
                                                  08  15 00065          BLEQ    6$                                      ;
          50        06    A2        10            00  ED 00067          CMPZV   #0, #16, 6(DCB), R0                     ;
                                                  08  18 0006D          BGEQ    7$                                      ;
                                    50 00000000G  8F  D0 0006F 6$:      MOVL    #SMG$_INVCOL, R0                        ;
                                                  04  00076          RET                                             ;
                                    56            51  D0 00077 7$:      MOVL    R1, LOCAL_ROW                           ; 3690
                                    57            50  D0 0007A          MOVL    R0, LOCAL_COL                           ; 3691
                                                  07  11 0007D          BRB     9$                                      ; 3685
                                    56    48      A2  3C 0007F 8$:      MOVZWL  72(DCB), LOCAL_ROW                      ; 3695
                                    57            01  D0 00083          MOVL    #1, LOCAL_COL                           ; 3696
                                    50    4A      A2  3C 00086 9$:      MOVZWL  74(DCB), R0                             ; 3700
                                    50            56  C2 0008A          SUBL2   LOCAL_ROW, R0                           ;
                                                  50  D6 0008D          INCL    LOCAL_HEIGHT                            ;
                                    04            6C  91 0008F          CMPB    (AP), #4                                ; 3702
                                                  20  1F 00092          BLSSU   10$                                     ;
                                          10      AC  D5 00094          TSTL    16(AP)                                  ;
                                                  1B  13 00097          BEQL    10$                                     ;
                                    51    10      BC  D0 00099          MOVL    @HEIGHT, R1                             ; 3705
                                    53    FF A641 9E  00009D          MOVAB   -1(LOCAL_ROW)[R1], R3                   ;
          53            62        10            00  ED 000A2          CMPZV   #0, #16, (DCB), R3                      ;
                                                  32  14 000A7          BGTR    11$                                     ;
          53        02  A2        10            00  ED 000A9          CMPZV   #0, #16, 2(DCB), R3                     ; 3706
                                                  2A  19 000AF          BLSS    11$                                     ;
                                    50            51  D0 000B1          MOVL    R1, LOCAL_HEIGHT                        ; 3710
                                    55    06      A2  3C 000B4 10$:     MOVZWL  6(DCB), LOCAL_WIDTH                     ; 3713
                                    05            6C  91 000B8          CMPB    (AP), #5                                ; 3715
                                                  28  1F 000BB          BLSSU   13$                                     ;
                                          14      AC  D5 000BD          TSTL    20(AP)                                  ;
                                                  23  13 000C0          BEQL    13$                                     ;
                                    51    14      BC  D0 000C2          MOVL    @WIDTH, R1                              ; 3718
                                    53    FF A741 9E  000C6          MOVAB   -1(LOCAL_COL)[R1], R3                   ;
          53        04  A2        10            00  ED 000CB          CMPZV   #0, #16, 4(DCB), R3                     ;
                                                  08  14 000D1          BGTR    11$                                     ;
          53        06  A2        10            00  ED 000D3          CMPZV   #0, #16, 6(DCB), R3                     ; 3719
                                                  07  18 000D9          BGEQ    12$                                     ;
                                    53            68  9E 000DB 11$:     MOVAB   SMG$_INVARG, R3                         ; 3721
                                    50            53  D0 000DE          MOVL    R3, R0                                  ;
                                                  04  000E1          RET                                             ;
                                    55            51  D0 000E2 12$:     MOVL    R1, LOCAL_WIDTH                         ; 3723
                                    54            01  D0 000E5 13$:     MOVL    #1, LOCAL_DIR                           ; 3726
                                    06            6C  91 000E8          CMPB    (AP), #6                                ; 3728
                                                  20  1F 000EB          BLSSU   16$                                     ;
```

```
                                                18   AC  D5 000ED          TSTL    4(AP)                                          ;
                                                     *B  13 000F0          BEQL    16$                                            ;
                      07              01        18   BC  CF 000F2          CASEL   @DIRECTION, #1, #7                             ; 3731
          0012        002E            0012           0012     000F7 14$:   .WORD   15$-14$,-                                      ;
          0012        002E            002E           002E     000FF               15$-14$,-                                      ;
                                                                                  17$-14$,-                                      ;
                                                                                  15$-14$,-                                      ;
                                                                                  17$-14$,-                                      ;
                                                                                  17$-14$,-                                      ;
                                                                                  17$-14$,-                                      ;
                                                                                  15$-14$                                        ;
                                                     1C  11 00107          BRB     17$                                            ; 3737
                                54              18   BC  D0 00109 15$:     MOVL    @DIRECTION, LOCAL_DIR                          ; 3734
                                53                   01  D0 0010D 16$:     MOVL    #1, LOCAL_COUNT                                ; 3741
                                07                   6C  91 00110          CMPB    (AP), #7                                       ; 3743
                                                     17  1F 00113          BLSSU   18$                                            ;
                                                1C   AC  D5 00115          TSTL    28(AP)                                         ;
                                                     12  13 00118          BEQL    18$                                            ;
                                                1C   BC  D5 0011A          TSTL    @COUNT                                         ; 3746
                                                     06  15 0011D          BLEQ    17$                                            ;
                                53              1C   BC  D0 0011F          MOVL    @COUNT, LOCAL_COUNT                            ; 3748
                                                     07  11 00123          BRB     18$                                            ;
                                51                   68  9E 00125 17$:     MOVAB   SMG$_INVARG, R1                                ; 3750
                                50                   51  D0 00128          MOVL    R1, R0                                         ;
                                                     04  00128            RET                                                    ;
                                                     53  DD 0012C 18$:     PUSHL   LOCAL_COUNT                                    ; 3761
                                                     54  DD 0012E          PUSHL   LOCAL_DIR                                      ;
                                                     21  BB 00130          PUSHR   #^M<R0,R5>                                     ; 3760
                                          00C4       8F  BB 00132          PUSHR   #^M<R2,R6,R7>                                  ; 3758
                    00000000G   00                   07  FB 00136          CALLS   #7, SMG$$SCROLL_AREA                           ;
                                17                   50  E9 0013D          BLBC    STATUS, 19$                                    ; 3762
                          28    A2                   56  B0 00140          MOVW    LOCAL_ROW, 40(DCB)                             ; 3768
                          2A    A2                   57  B0 00144          MOVW    LOCAL_COL, 42(DCB)                             ; 3769
                          34    A2                   01  8A 00148          BICB2   #1, 52(DCB)                                    ; 3775
                                                     1A  DD 0014C          PUSHL   #26                                            ; 3781
                                                     52  DD 0014E          PUSHL   DCB                                            ;
                    00000000G   00                   02  FB 00150          CALLS   #2, SMG$$CHECK_FOR_OUTPUT_DCB                  ;
                                                         04 00157 19$:     RET                                                    ; 3783

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

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

```
;  3543     3785  1 %SBTTL 'SMG$SET_CURSOR_ABS - Set absolute cursor position'
;  3544     3786  1 GLOBAL ROUTINE SMG$SET_CURSOR_ABS (
;  3545     3787  1                                    DISPLAY_ID,
;  3546     3788  1                                    ROW_NUM,
;  3547     3789  1                                    COL_NUM
;  3548     3790  1                                   ) =
;  3549     3791  1 !++
;  3550     3792  1 ! FUNCTIONAL DESCRIPTION:
;  3551     3793  1 !
;  3552     3794  1 !       Position cursor to absolute row and column specified.
;  3553     3795  1 !       Return SMG$_INVROW or SMG$_INVCOL if either position is outside
;  3554     3796  1 !       of the virtual display area.
;  3555     3797  1 !
;  3556     3798  1 ! CALLING SEQUENCE:
;  3557     3799  1 !
;  3558     3800  1 !       ret_status.wlc.v = SMG$SET_CURSOR_ABS (
;  3559     3801  1 !                                              DISPLAY_ID.rl.r
;  3560     3802  1 !                                              [,ROW_NUM.rl.r]
;  3561     3803  1 !                                              [,COL_NUM.rl.r])
;  3562     3804  1 !
;  3563     3805  1 ! FORMAL PARAMETERS:
;  3564     3806  1 !
;  3565     3807  1 !       DISPLAY_ID.rl.r Display id of desired display.
;  3566     3808  1 !
;  3567     3809  1 !       ROW_NUM.rl.r    Row position.  If not supplied, row number is
;  3568     3810  1 !                       not altered from current position.
;  3569     3811  1 !
;  3570     3812  1 !       COL_NUM.rl.r    Column position.  If not supplied, column number
;  3571     3813  1 !                       is not altered from current position.
;  3572     3814  1 !
;  3573     3815  1 ! IMPLICIT INPUTS:
;  3574     3816  1 !
;  3575     3817  1 !       NONE
;  3576     3818  1 !
;  3577     3819  1 ! IMPLICIT OUTPUTS:
;  3578     3820  1 !
;  3579     3821  1 !       NONE
;  3580     3822  1 !
;  3581     3823  1 ! COMPLETION STATUS:
;  3582     3824  1 !
;  3583     3825  1 !       SS$_NORMAL      Normal successful completion
;  3584     3826  1 !       SMG$_INVROW     Invalid row number
;  3585     3827  1 !       SMG$_INVCOL     Invalid column number
;  3586     3828  1 !       SMG$_WRONUMARG  Wrong number arguments.
;  3587     3829  1 !       SMG$_INVDIS_ID  Invalid display id.
;  3588     3830  1 !
;  3589     3831  1 ! SIDE EFFECTS:
;  3590     3832  1 !
;  3591     3833  1 !       NONE
;  3592     3834  1 !--
;  3593     3835  2     BEGIN
;  3594     3836  2     BUILTIN
;  3595     3837  2         NULLPARAMETER;
;  3596     3838  2
;  3597     3839  2     LOCAL
;  3598     3840  2         STATUS,                     ! Status of subroutine calls
;  3599     3841  2         ROW,                        ! Working row number
```

```
;  3600          3842  2          COL,                                ! Working column number
;  3601          3843  2          DCB : REF $DCB_DECL;                ! Addr. of display control block
;  3602          3844  2
;  3603          3845  2      $SMG$VALIDATE_ARGCOUNT (1,3);
;  3604          3846  2
;  3605          3847  2      $SMG$GET_DCB ( .DISPLAY_ID, DCB);   ! Get address of display control
;  3606          3848  2                                          ! block
;  3607          3849  2
;  3608          3850  2
;  3609          3851  3      ROW = (IF NOT NULLPARAMETER (ROW_NUM) THEN ..ROW_NUM
;  3610          3852  2                                          ELSE .DCB [DCB_W_CURSOR_ROW]);
;  3611          3853  2
;  3612          3854  3      COL = (IF NOT NULLPARAMETER (COL_NUM) THEN ..COL_NUM
;  3613          3855  2                                          ELSE .DCB [DCB_W_CURSOR_COL]);
;  3614          3856  2
;  3615          3857  2  !+
;  3616          3858  2  ! Make sure resulting position is still within the virtual display.
;  3617          3859  2  !-
;  3618          3860  2  !+
;  3619          3861  2  ! Validity check row and column specified
;  3620          3862  2  !-
;  3621          3863  2      $SMG$VALIDATE_ROW_COL ( .ROW, .COL);
;  3622          3864  2
;  3623          3865  2  !+
;  3624          3866  2  ! Ok, set them.
;  3625          3867  2  !-
;  3626          3868  2      DCB [DCB_W_CURSOR_ROW] = .ROW;
;  3627          3869  2      DCB [DCB_W_CURSOR_COL] = .COL;
;  3628          3870  2
;  3629          3871  2  !+
;  3630          3872  2  ! We have altered the cursor position so obviously sequential output
;  3631          3873  2  ! has been interupted.  Turn off the scrolling bit.
;  3632          3874  2  !-
;  3633          3875  2
;  3634          3876  2      DCB [DCB_V_FULL] = 0;
;  3635          3877  2
;  3636          3878  3      RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
;  3637          3879  2                                         SMG$C_SET_CURSOR_ABS));
;  3638          3880  1      END;                            ! End of routine SMG$SET_CURSOR_ABS



                                        0004 00000          .ENTRY  SMG$SET_CURSOR_ABS, Save R2                 ;  3786
                50              6C      01   83 00002          SUBB3   #1, (AP), DIFF                              ;  3845
                                02      50   91 00006          CMPB    DIFF, #2                                    ;
                                        08   1B 00009          BLEQU   1$                                          ;
                                50 00000000G    8F   D0 0000B          MOVL    #SMG$_WRONUMARG, R0                 ;
                                                04   00012          RET                                         ;
                                50      C4      BC   D0 00013 1$:      MOVL    @DISPLAY_ID, R0                     ;  3847
                          04    BC      38      A0   D1 00017          CMPL    56(R0), @DISPLAY_ID                 ;
                                                06   12 0001C          BNEQ    2$                                  ;
                                11      44      A0   91 0001E          CMPB    68(R0), #17                         ;
                                                08   13 00022          BEQL    3$                                  ;
                                50 00000000G    8F   D0 00024 2$:      MOVL    #SMG$_INVDIS_ID, R0                 ;
                                                04   0002B          RET                                         ;
```

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

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

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

```
;  3641          3882  1 %SBTTL 'SMG$SET_CURSOR_REL - Set relative cursor position'
;  3642          3883  1 GLOBAL ROUTINE SMG$SET_CURSOR_REL (
;  3643          3884  1                                    DISPLAY_ID,
;  3644          3885  1                                    DELTA_ROW,
;  3645          3886  1                                    DELTA_COLUMN
;  3646          3887  1                                   ) =
;  3647          3888  1 !++
;  3648          3889  1 ! FUNCTIONAL DESCRIPTION:
;  3649          3890  1 !
;  3650          3891  1 !       Position cursor to relative row and column specified --
;  3651          3892  1 !       relative to current position.
;  3652          3893  1 !       If adding DELTA_ROW or DELTA_COLUMN takes us outside of the
;  3653          3894  1 !       virtual display, return SMG$_INVROW or SMG$_INVCOL respectively.
;  3654          3895  1 !
;  3655          3896  1 !
;  3656          3897  1 ! CALLING SEQUENCE:
;  3657          3898  1 !
;  3658          3899  1 !       ret_status.wlc.v = SMG$SET_CURSOR_REL (
;  3659          3900  1 !                                              DISPLAY_ID.rl.r
;  3660          3901  1 !                                              [,DELTA_ROW.rl.r]
;  3661          3902  1 !                                              [,DELTA_COLUMN.rl.r])
;  3662          3903  1 !
;  3663          3904  1 ! FORMAL PARAMETERS:
;  3664          3905  1 !
;  3665          3906  1 !       DISPLAY_ID.rl.r         Display id of desired display.
;  3666          3907  1 !
;  3667          3908  1 !       DELTA_ROW.rl.r          Delta row position (-delta, +delta)
;  3668          3909  1 !                               If not supplied, row number is not
;  3669          3910  1 !                               altered from current position.
;  3670          3911  1 !
;  3671          3912  1 !       DELTA_COLUMN.rl.r       Delta_column position (-delta, +delta)
;  3672          3913  1 !                               If not supplied, column number is not
;  3673          3914  1 !                               altered from current position.
;  3674          3915  1 !
;  3675          3916  1 !
;  3676          3917  1 ! IMPLICIT INPUTS:
;  3677          3918  1 !
;  3678          3919  1 !       NONE
;  3679          3920  1 !
;  3680          3921  1 ! IMPLICIT OUTPUTS:
;  3681          3922  1 !
;  3682          3923  1 !       NONE
;  3683          3924  1 !
;  3684          3925  1 ! COMPLETION STATUS:
;  3685          3926  1 !
;  3686          3927  1 !       SS$_NORMAL      Normal successful completion
;  3687          3928  1 !       SMG$_INVROW     When delta_column is added to current column
;  3688          3929  1 !       SMG$_INVCOL     and/or delta_row is added to currert row,
;  3689          3930  1 !                       resulting row or column is outside virtual
;  3690          3931  1 !                       display.
;  3691          3932  1 !       SMG$_WRONUMARG  Wrong number arguments.
;  3692          3933  1 !
;  3693          3934  1 ! SIDE EFFECTS:
;  3694          3935  1 !
;  3695          3936  1 !       NONE
;  3696          3937  1 !--
;  3697          3938  2     BEGIN
```

```
;  3698          3939  2     BUILTIN
;  3699          3940  2         NULLPARAMETER;
;  3700          3941  2
;  3701          3942  2     LOCAL
;  3702          3943  2         STATUS,             ! Status of subroutine calls
;  3703          3944  2         D_ROW,              ! Working delta row
;  3704          3945  2         D_COL,              ! Working delta column
;  3705          3946  2         DCB : REF $DCB_DECL;    ! Addr. of display control block
;  3706          3947  2
;  3707          3948  2     $SMG$VALIDATE_ARGCOUNT (1,3);
;  3708          3949  2
;  3709          3950  2     $SMG$GET_DCB ( .DISPLAY_ID, DCB);   ! Get address of display control
;  3710          3951  2                                         ! block
;  3711          3952  2
;  3712          3953  2
;  3713          3954  3     D_ROW = (IF NOT NULLPARAMETER (DELTA_ROW) THEN ..DELTA_ROW
;  3714          3955  2                                               ELSE 0);
;  3715          3956  2
;  3716          3957  3     D_COL = (IF NOT NULLPARAMETER (DELTA_COLUMN) THEN ..DELTA_COLUMN
;  3717          3958  2                                                 ELSE 0);
;  3718          3959  2
;  3719          3960  2 !+
;  3720          3961  2 ! Make sure resulting position is still within the virtual display.
;  3721          3962  2 !-
;  3722          3963  2     IF .DCB [DCB_W_CURSOR_ROW] + .D_ROW LSS 1  OR
;  3723          3964  2        .DCB [DCB_W_CURSOR_ROW] + .D_ROW GTR .DCB [DCB_W_NO_ROWS]
;  3724          3965  2     THEN
;  3725          3966  2         RETURN (SMG$_INVROW);
;  3726          3967  2
;  3727          3968  2     IF .DCB [DCB_W_CURSOR_COL] + .D_COL LSS 1  OR
;  3728          3969  2        .DCB [DCB_W_CURSOR_COL] + .D_COL GTR .DCB [DCB_W_NO_COLS]
;  3729          3970  2     THEN
;  3730          3971  2         RETURN (SMG$_INVCOL);
;  3731          3972  2
;  3732          3973  2 !+
;  3733          3974  2 ! Ok, set them.
;  3734          3975  2 !-
;  3735          3976  2     DCB [DCB_W_CURSOR_ROW] = .DCB [DCB_W_CURSOR_ROW] + .D_ROW;
;  3736          3977  2     DCB [DCB_W_CURSOR_COL] = .DCB [DCB_W_CURSOR_COL] + .D_COL;
;  3737          3978  2
;  3738          3979  2 !+
;  3739          3980  2 ! Turn off scrolling.
;  3740          3981  2 !-
;  3741          3982  2
;  3742          3983  2     DCB [DCB_V_FULL] = 0;
;  3743          3984  2
;  3744          3985  3     RETURN (SMG$$CHECK_FOR_OUTPUT_DCB (.DCB,
;  3745          3986  2                                        SMG$C_SET_CURSOR_REL));
;  3746          3987  1     END;                            ! End of routine SMG$SET_CURSOR_REL


                                    0000C 00000          .ENTRY  SMG$SET_CURSOR_REL, Save R2,R3                  ; 3883
                50          6C      01   83 00002          SUBB3   #1, (AP), DIFF                                  ; 3948
                            02      50   91 00006          CMPB    DIFF, #2                                        ;
```

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

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


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

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

```
                                                    .EXTRN  LIB$STOP
```

```
;                               PSECT SUMMARY

;       Name                    Bytes                   Attributes

;   _SMG$CODE                     5686  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
```

```
;                           Library Statistics

;                                        -------- Symbols --------      Pages       Processing
;       File                             Total    Loaded   Percent      Mapped      Time

;   _$255$DUA28:[SYSLIB]STARLET.L32;1     9776        41         0         581       00:01.0
;   _$255$DUA28:[SMGRTL.OBJ]RTLLIB.L32;1    36         0         0           8       00:00.1
;   _$255$DUA28:[SMGRTL.OBJ]SMGLIB.L32;1   469        31         6          38       00:00.4
```

```
;                                   COMMAND QUALIFIERS

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

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

SMGDATSTR
REQ

SMGLNK
REQ

SMGKQB
SDL

SMGBLDTRM
LIS

SMGDISCHA
LIS

SMGKTH
SDL

SMGMACROS
REQ

SMGPROLOG
REQ

SMGTERM
REQ

SMGTRMSTR
R32

SMGTRMPTR
SDL

SMGSCRMAC
REQ

SMGALLESC
LIS

SMGTPACTL
REQ

SMGSCRTCB
REQ

SMGLIB
REQ

SMGTPALIB
REQ

SMGDEFKEY
LIS

SMGTABDEF
REQ

SMGTRMMAC
REQ

SMGBOOTAB
LIS

SMGDISDRW
LIS

SMGDISLIN
LIS

SMGDISINP
LIS

SMGDISDHW
LIS