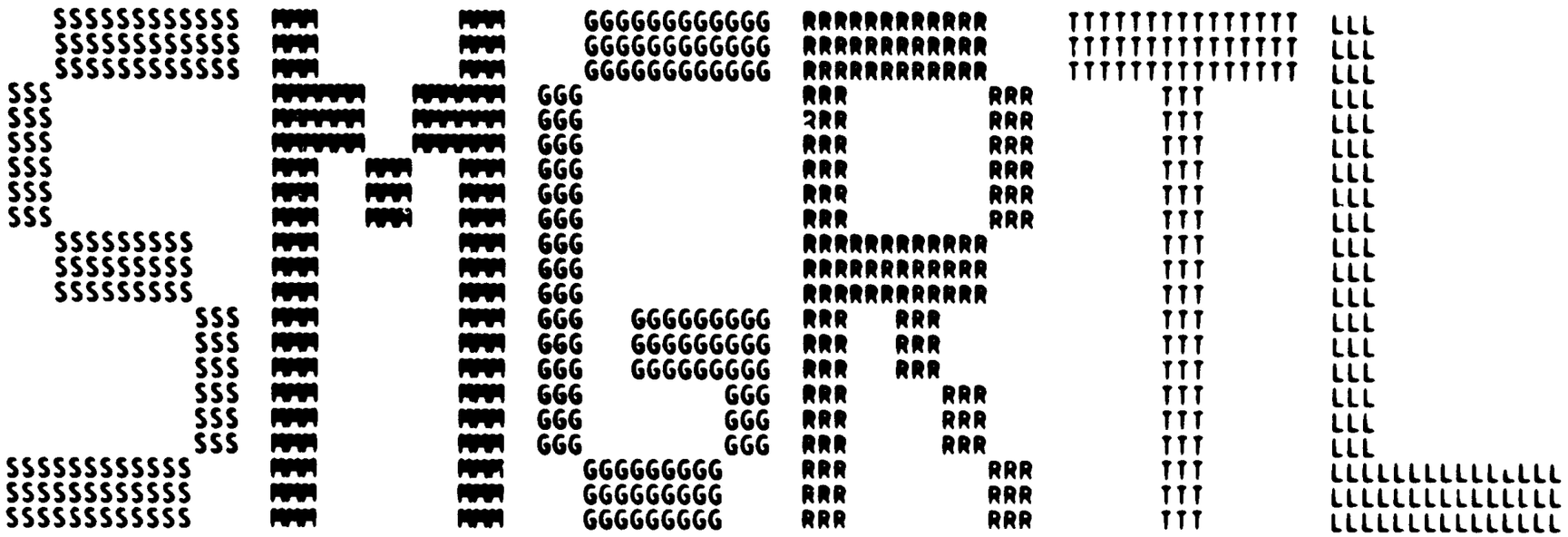
SSSSSSSSSSS MMM MMM GGGGGGGGGGG RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
SSSSSSSSSSS MMM MMM GGGGGGGGGGG RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
SSSSSSSSSSS MMM MMM GGGGGGGGGGG RRRRRRRRRRR TTTTTTTTTTTTTTT LLL
SSS MMMMMMM MMMMMMM GGG RRR RRR TTT LLL
SSS MMMMMMM MMMMMMM GGG RRR RRR TTT LLL
SSS MMMMMMM MMMMMMM GGG RRR RRR TTT LLL
SSS MMMM MMM MMMM GGG RRR RRR TTT LLL
SSS MMMM MMM MMMM GGG RRR RRR TTT LLL
SSS MMMM MMM MMMM GGG RRR RRR TTT LLL
SSSSSSSSS MMMM MMMM GGG RRRRRRRRRRRR TTT LLL
SSSSSSSSS MMMM MMMM GGG RRRRRRRRRRRR TTT LLL
SSSSSSSSS MMMM MMMM GGG RRRRRRRRRRRR TTT LLL
SSS MMMM MMMM GGG GGGGGGGGG RRR RRR TTT LLL
SSS MMMM MMMM GGG GGGGGGGGG RRR RRR TTT LLL
SSS MMMM MMMM GGG GGGGGGGGG RRR RRR TTT LLL
SSS MMMM MMMM GGG GGG RRR RRR TTT LLL
SSS MMMM MMMM GGG GGG RRR RRR TTT LLL
SSS MMMM MMMM GGG GGG RRR RRR TTT LLL
SSSSSSSSSSS MMMM MMMM GGGGGGGGG RRR RRR TTT LLLLLLLLLLLLLLL
SSSSSSSSSSS MMMM MMMM GGGGGGGGG RRR RRR TTT LLLLLLLLLLLLLLL
SSSSSSSSSSS MMMM MMMM GGGGGGGGG RRR RRR TTT LLLLLLLLLLLLLLL

FILEID**SMGMINUPD

```
;     1          0001  0 %TITLE 'SMGSSMINIMUM_UPDATE - Minimum update calculation and output'
;     2          0002  0 MODULE SMGSSMINIMUM_UPDATE (
;.001 STAN1046   0003  0                 IDENT = '1-046' ! File: SMGMINUPD.B32 Edit: STAN1046
;     4-1        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 !
```

```
;    31            0030  1 ;++
;    32            0031  1 ; FACILITY:     Screen Management
;    33            0032  1 ;
;    34            0033  1 ; ABSTRACT:
;    35            0034  1 ;
;    36            0035  1 ;       This module contains routines which inspect two screen
;    37            0036  1 ;       representations and calculate the near-minimal sequence of
;    38            0037  1 ;       terminal commands to change the current contents of the screen
;    39            0038  1 ;       to the new representation of the screen.
;    40            0039  1 ;       Also contained herein are routines pertaining to buffering.
;    41            0040  1 ;
;    42            0041  1 ; ENVIRONMENT:  User mode, Shared library routines.
;    43            0042  1 ;
;    44            0043  1 ; AUTHOR: R. Reichert, CREATION DATE: 15-APR-1983
;    45            0044  1 ;
;    46            0045  1 ; MODIFIED BY:
;    47            0046  1 ;
;001 STAN1046      0047  1 ; 1-046 - STAN 21-Oct-1984. Don't reset attributes if none were set.
;002 STAN1046      0048  1 ;-----------+
;003 STAN1046      0049  1 ; VMS V4.0  |
;004 STAN1046      0050  1 ;-----------+
;    48            0051  1 ; 1-045 - STAN 17-Apr-1984. Store unknown terminal type correctly in PBCB.
;    49            0052  1 ; 1-044 - STAN  7-Apr-1984. Minor change for unsolicit input.
;    50            0053  1 ; 1-043 - STAN 31-Mar-1984. Fix dot bug in SET_ATTRIBUTES_OFF.
;    51            0054  1 ; 1-042 - STAN 21-Mar-1984. Fixed bug with border vector.
;    52            0055  1 ; 1-041 - STAN 18-Mar-1984. Renove use of %ASCID that causes PSECTS
;    53            0056  1 ;         to be read/write thus making their use impractical for
;    54            0057  1 ;         shared images.
;    55            0058  1 ;         Home cursor before erasing screen.
;    56            0059  1 ;         Change test for unknown terminal.
;    57            0060  1 ; 1-040 - STAN 14-Mar-1984. Ensure final cursor position doesn't change
;    58            0061  1 ;         after removing any scrolling region in the exit handler.
;    59            0062  1 ;         Change END_BOLD capability to BEGIN_NORMAL_RENDITION.
;    60            0063  1 ;         Handle unknown terminals.
;    61            0064  1 ;         Make truncation icon work again; also other control displays.
;    62            0065  1 ;         Write two new routines, SET_ATTRIBUTES_ON and SET_ATTRIBUTES_OFF.
;    63            0066  1 ; 1-039 - STAN 23-Feb-1984. Bug fix.
;    64            0067  1 ;         Initialize characteristics from terminal characteristics
;    65            0068  1 ;         not from termtable capabilities.
;    66            0069  1 ;         Allow long sequences for border vector.
;    67            0070  1 ;         Add temporary SET_ATTRIBUTES_ONLY.
;    68            0071  1 ; 1-038 - STAN 21-Feb-1984. Bug fixes.
;    69            0072  1 ; 1-037 - STAN 21-Feb-1984. Store BS bit in PBCB.
;    70            0073  1 ; 1-036 - STAN 13-Feb-1984. Install Pam's fix for VT52s.
;    71            0074  1 ;         Bug fix in exit handler.
;    72            0075  1 ; 1-035 - STAN  7-Feb-1984. Allow positive terminal codes.
;    73            0076  1 ; 1-034 - STAN 15-Jan-1983. Use TERMTABLE.
;    74            0077  1 ;         Fix charset bug.
;    75            0078  1 ; 1-033 - STAN 14-Dec-1983. Fix dot bug in edit 32.
;    76            0079  1 ; 1-032 - RKR   2-Dec-1983. Add SMG$$ERASE_PASTEBOARD.  This inner routine
;    77            0080  1 ;         goes directly to SMG$$FLUSH_BUFFER rather than SMG$FLUSH_BUFFER.
;    78            0081  1 ;         Redirect current calls to SMG$ERASE_PASTEBOARD to call
;    79            0082  1 ;         SMG$$ERASE_PASTEBOARD instead.
;    80            0083  1 ; 1-031 - STAN  2-Nov-1983. Restore terminal width on exit.
;    81            0084  1 ; 1-030 - STAN 14-Oct-1983. Invalidate screen on CTRL/O.
;    82            0085  1 ; 1-029 - STAN 13-Oct-1983. Bug fix for scrolling wide lines.
;    83            0086  1 ; 1-028 - Handle DIAMOND and control character displays. STAN 5-Oct-1983.
```

```
;    84          0087  1  ;  1-027 - Handle user graphics. STAN 19-sep-1983.
;    85          0088  1  ;          Clear screen on exit if so requested.
;    86          0089  1  ;  1-026 - Add SMG$$AUTOB_OUTPUT so the autobende. routines can
;    87          0090  1  ;          output directly to the pb without knowing the pb addr.
;    88          0091  1  ;          PLL 9-Sep-1983
;    89          0092  1  ;  1-025 - Add SMG$ERASE_PASTEBOARD. STAN 25-Aug-1983
;    90          0093  1  ;  1-024 - Changes to SMG$$CHECK_HDWR_SCROLL and SMG$$MIN_UPD to support
;    91          0094  1  ;          double-wide/double high text.  RKR 17-AUG-1983.
;    92          0095  1  ;  1-023 - Add some preprocessing to SMG$$MIN_UPD to refine the range
;    93          0096  1  ;          of lines that actually changed.  RKR 12-AUG-1983.
;    94          0097  1  ;  1-022 - Modify CHECK_HDWR_SCROLL to bypass situation where in fact
;    95          0098  1  ;          the virtual display contains line-by-line identical contents
;    96          0099  1  ;          except for the top or bottom line.
;    97          0100  1  ;          For example, if you fill up (except the last line ) a
;    98          0101  1  ;          virtual display with the text
;    99          0102  1  ;                COMMAND:
;   100          0103  1  ;                COMMAND:
;   101          0104  1  ;                COMMAND:
;   102          0105  1  ;
;   103          0106  1  ;          As you write the last line to COMMAND:, the current logic
;   104          0107  1  ;          will downscroll the virtual display and repaint the top line.
;   105          0108  1  ;          This will produce the right result but looks ugly.  Right
;   106          0109  1  ;          now this will also happen when you clear a display to all
;   107          0110  1  ;          spaces since it falls into the upscroll logic.
;   108          0111  1  ;          This fix intercepts the cases where the virtual display
;   109          0112  1  ;          has changed only in the 1st or last line, avoids scrolling,
;   110          0113  1  ;          and lets the rest of minimal update repaint just the last line
;   111          0114  1  ;  1-021 - RKR Remove temporary fix to scrolling problem.  Compensating
;   112          0115  1  ;          code in SMG$$FIND_MIN_CURSOR_POS and SMG$$SET_PHYSICAL_CURSOR
;   113          0116  1  ;          should now take care of the problem.
;   114          0117  1  ;  1-020 - RKR 3-AUG-1983. Consolidate lines of code pertaining to
;   115          0118  1  ;          actually setting the physical scrolling region into a new
;   116          0119  1  ;          subroutine SMG$$FORCE_SCROLL_REG.
;   117          0120  1  ;  1-019 - RKR 1-AUG-1983. Modify SMG$$CHECK_HDWR_SCROLL to provide
;   118          0121  1  ;          downscrolling as well as upscrolling.
;   119          0122  1  ;  1-018 - STAN 28-Jul-1983. Temporary fix to remove scrolling
;   120          0123  1  ;                             region after use.
;   121          0124  1  ;  1-017 - RKR  15-JUL-1983. Fix bug found by J. Burrows.
;   122          0125  1  ;  1-016 - RKR  14-JUL-1983. Minor code improvements and better comments
;   123          0126  1  ;          to newly-added code.
;   124          0127  1  ;
;   125          0128  1  ;  1-015 - RKR  12-Jul-1983. Add SMG$$CHECK_HDWR_SCROLL.
;   126          0129  1  ;  1-014 - STAN 21-Jun-1983. Temporary fix.
;   127          0130  1  ;  1-013 - STAN 18-Jun-1983. File output.
;   128          0131  1  ;  1-012 - RKR 20-May-1983 Remove external references to DD_ structures
;   129          0132  1  ;                             and counts -- no longer needed (or available).
;   130          0133  1  ;  1-011 - STAN 16-May-1983 Pasteboard batching
;   131          0134  1  ;  1-010 - STAN 11-May-1983
;   132          0135  1  ;          Use shift out and shift in.
;   133          0136  1  ;  1-009 - STAN 10-May-1983
;   134          0137  1  ;          Temporary fix for rendition attribute.
;   135          0138  1  ;  1-008 - STAN 8-May-1983
;   136          0139  1  ;          Flush buffer only on success exit.
;   137          0140  1  ;  1-007 - STAN 2-May-1983
;   138          0141  1  ;          Fixed bug in buffering.
;   139          0142  1  ;          Handle border rendition.
;   140          0143  1  ;          Don't flush buffer on CLI forced exit.
```

```
;  141          0144  1 ! 1-006 - STAN 1-May-1983
;  142          0145  1 !           SMG$$PUT_OUTPUT,
;  143          0146  1 !           SMG$$OUTPUT,
;  144          0147  1 !           Finished SMG$$FLUSH_BUFFER.
;  145          0148  1 ! 1-005 - One additional tweak. RKR 26-APR-1983.
;  146          0149  1 ! 1-004 - Minor optimization.  RKR 26-APR-1983.
;  147          0150  1 ! 1-003 - Fix video attribute production.  RKR 21-APR-1983.
;  148          0151  1 ! 1-002 - Minor temp speed up.  RKR 18-APR-1983
;  149          0152  1 ! 1-001 - Shell for further development.  RKR 15-APR-1983.
;  150          0153  1 !--
```

```
;  152          0154  1 %SBTTL 'Declarations'
;  153          0155  1 !
;  154          0156  1 ! TABLE OF CONTENTS:
;  155          0157  1 !
;  156          0158  1
;  157          0159  1 FORWARD ROUTINE
;  158          0160  1
;  159          0161  1 !       Public entry points
;  160          0162  1
;  161          0163  1         SMG$ERASE_PASTEBOARD,           ! Clears screen
;  162          0164  1
;  163          0165  1         SMG$FLUSH_BUFFER,               ! Flush remaining buffered
;  164          0166  1                                         ! output to screen by display id.
;  165          0167  1
;  166          0168  1         SMG$PUT_PASTEBOARD,             ! Output pasteboard via user routine
;  167          0169  1
;  168          0170  1         SMG$SNAPSHOT,                   ! Take an RMS snapshot
;  169          0171  1
;  170          0172  1 !       Private entry points
;  171          0173  1
;  172          0174  1         SMG$$CHECK_HDWR_SCROLL,         ! Check to see if harware scroll
;  173          0175  1                                         ! will help min. update
;  174          0176  1
;  175          0177  1         SMG$$ERASE_PASTEBOARD,          ! Inner ERASE_PASTEBOARD routine
;  176          0178  1
;  177          0179  1         SMG$$PBCB_EXIT_HANDLER,         ! Exit handler to flush pasteboard
;  178          0180  1
;  179          0181  1         SMG$$SETUP_TERMINAL_TYPE,       ! Find out type of terminal.
;  180          0182  1
;  181          0183  1         SMG$$FLUSH_BUFFER,              ! Flush remaining buffered
;  182          0184  1                                         ! output to screen by PBCB.
;  183          0185  1
;  184          0186  1         SMG$$FORCE_SCROLL_REG,          ! Force physical scrolling
;  185          0187  1                                         ! region to that specified.
;  186          0188  1
;  187          0189  1         SMG$$PUT_SCREEN,                ! Put text to screen with rendition
;  188          0190  1                                         ! and cursor positioning
;  189          0191  1
;  190          0192  1         SMG$$AUTOB_OUTPUT,              ! Autobended entry to SMG$$OUTPUT
;  191          0193  1
;  192          0194  1         SMG$$SET_ATTRIBUTES_ON,
;  193          0195  1         SMG$$SET_ATTRIBUTES_OFF,
;  194          0196  1         SMG$$OUTPUT,                    ! Raw outputter
;  195          0197  1         SMG$$MIN_UPD,                   ! force compatibility *** temp
;  196          0198  1         SMG$$OUTPUT_PASTEBOARD,         ! Output pasteboard (use minimal
;  197          0199  1                                         ! update if this mode is enabled)
;  198          0200  1
;  199          0201  1 !       Local entry points
;  200          0202  1
;  201          0203  1         ESTABLISH_BORDER_VECTOR : NOVALUE,      ! Create border vector
;  202          0204  1         RMS_RTN,                        ! Output record with RMS
;  203          0205  1         OUTPUT;                         ! Low level output routine
;  204          0206  1
;  205          0207  1 !
;  206          0208  1 ! SWITCHES:
;  207          0209  1 !
;  208          0210  1 !       in include files
```

```
;  209          0211  1
;  210          0212  1 !
;  211          0213  1 ! LINKAGES:
;  212          0214  1 !
;  213          0215  1 !       in include files
;  214          0216  1
;  215          0217  1 !
;  216          0218  1 ! INCLUDE FILES
;  217          0219  1 !
;  218          0220  1
;  219          0221  1 REQUIRE 'RTLIN:SMGPROLOG';              ! defines psects, macros,
;  220          0299  1                                         ! structures, & terminal symbols
;  221          0300  1 REQUIRE 'RTLIN:STRLNK.REQ';             ! JSB linkages
;  222          0485  1
;  223          0486  1 !
;  224          0487  1 ! EXTERNAL REFERENCES
;  225          0488  1 !
;  226          0489  1
;  227          0490  1 EXTERNAL
;  228          0491  1
;  229          0492  1         PBD_L_COUNT,    ! No. of pasteboards we currently have
;  230          0493  1
;  231          0494  1         PBD_A_PBCB : VECTOR [PBD_K_MAX_PB, LONG],
;  232          0495  1                         ! Table of addresses of PBCB's
;  233          0496  1
;  234          0497  1         PBD_V_PB_AVAIL : BITVECTOR [PBD_K_MAX_PB];
;  235          0498  1                         ! Bit vector or pasteboard id numbers in use.
;  236          0499  1
;  237          0500  1 EXTERNAL LITERAL
;  238          0501  1
;  239          0502  1         SMG$_WILUSERMS, ! RMS will be used later to perform output
;  240          0503  1         SMG$_INVPAS_ID; ! Invalid pasteboard id
;  241          0504  1
;  242          0505  1 EXTERNAL ROUTINE
;  243          0506  1
;  244          0507  1         LIB$GET_EF,
;  245          0508  1         LIB$GET_VM,
;  246          0509  1         SMG$$FIND_MIN_CURSOR_POS,
;  247          0510  1         SMG$BEGIN_PASTEBOARD_UPDATE,
;  248          0511  1         SMG$END_PASTEBOARD_UPDATE,
;  249          0512  1         SMG$$OUTPUT_MINIMAL_UPDATE;
;  250          0513  1
;  251          0514  1 OWN
;  252          0515  1
;  253          0516  1         FIRST_TIME_FLAG : INITIAL (0);  !***** Kludge -- ignore ***
```

```
;  255          0517  1 %SBTTL 'SMG$ERASE_PASTEBOARD- Clear Screen'
;  256          0518  1 GLOBAL ROUTINE SMG$ERASE_PASTEBOARD ( PASTEBOARD_ID ) =
;  257          0519  1 !++
;  258          0520  1 ! FUNCTIONAL DESCRIPTION:
;  259          0521  1 !
;  260          0522  1 !       This routine erases the entire pasteboard.
;  261          0523  1 !       The physical cursor is left at (1,1).
;  262          0524  1 !
;  263          0525  1 ! CALLING SEQUENCE:
;  264          0526  1 !
;  265          0527  1 !       ret_status.wlc.v = SMG$ERASE_PASTEBOARD ( PASTEBOARD_ID.rl.r )
;  266          0528  1 !
;  267          0529  1 ! FORMAL PARAMETERS:
;  268          0530  1 !
;  269          0531  1 !       PASTEBOARD_ID.rl.r      The id of the PASTEBOARD which is to be cleared.
;  270          0532  1 !
;  271          0533  1 ! IMPLICIT INPUTS:
;  272          0534  1 !
;  273          0535  1 !       None
;  274          0536  1 !
;  275          0537  1 ! IMPLICIT OUTPUTS:
;  276          0538  1 !
;  277          0539  1 !       None
;  278          0540  1 !
;  279          0541  1 ! COMPLETION STATUS:
;  280          0542  1 !
;  281          0543  1 !       SS$_NORMAL      Normal successful completion
;  282          0544  1 !       SMG$_WRONUMARG  Wrong number of arguments.
;  283          0545  1 !       SMG$_INVPAS_ID  Invalid pasteboard id.
;  284          0546  1 !
;  285          0547  1 ! SIDE EFFECTS:
;  286          0548  1 !
;  287          0549  1 !       NONE
;  288          0550  1 !--
;  289          0551  2     BEGIN
;  290          0552  2
;  291          0553  2     LOCAL
;  292          0554  2         PBCB    : REF $PBCB_DECL;       ! Address of pasteboard control block.
;  293          0555  2
;  294          0556  2
;  295          0557  2     $SMG$VALIDATE_ARGCOUNT(1,1);
;  296          0558  2
;  297          0559  2 !+
;  298          0560  2 ! Isolate pasteboard control block.
;  299          0561  2 !-
;  300          0562  2
;  301          0563  2     $SMG$GET_PBCB(.PASTEBOARD_ID,PBCB); ! Get address of PBCB
;  302          0564  2
;  303          0565  2     RETURN (SMG$$ERASE_PASTEBOARD (.PBCB));
;  304          0566  2
;  305          0567  1 END;                ! routine SMG$ERASE_PASTEBOARD


;                                                                       .TITLE  SMG$$MINIMUM_UPDATE SMG$$MINIMUM_UPDATE - Minim
                                                                                um update calculatio
                                                                        .IDENT  \1-046\
```

```
                                                        .PSECT  _SMG$DATA,NOEXE,  PIC,2

                                        00000000  00000 FIRST_TIME_FLAG:
                                                        .LONG   0                                       ;

                                                        .EXTRN  PBD_L_COUNT, PBD_A_PBCB
                                                        .EXTRN  PBD_V_PB_AVAIL, SMG$_WILUSERMS
                                                        .EXTRN  SMG$_INVPAS_ID, LIB$GET_EF
                                                        .EXTRN  LIB$GET_VM, SMG$$FIND_MIN_CURSOR_POS
                                                        .EXTRN  SMG$BEGIN_PASTEBOARD_UPDATE
                                                        .EXTRN  SMG$END_PASTEBOARD_UPDATE
                                                        .EXTRN  SMG$$OUTPUT_MINIMAL_UPDATE
                                                        .EXTRN  SMG$_WRONUMARG

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

                                           0000 00000           .ENTRY  SMG$ERASE_PASTEBOARD, Save nothing      : 0518
                                    01              6C   91 00002        CMPB    (AP), #1                        : 0557
                                                    08   13 00005        BEQL    1$                              :
                                    50 00000000G    8F   D0 00007        MOVL    #SMG$_WRONUMARG, R0             :
                                                         04 0000E        RET                                     :
                                    50        04    BC   D0 0000F 1$:     MOVL    @PASTEBOARD_ID, R0              : 0563
                                                    11   19 00013        BLSS    2$                              :
                          00000000G 00              50   D1 00015        CMPL    R0, PBD_L_COUNT                 :
                                                    08   14 0001C        BGTR    2$                              :
                      08 00000000G 00               50   E0 0001E        BBS     R0, PBD_V_PB_AVAIL, 3$          :
                                    50 00000000G    8F   D0 00026 2$:     MOVL    #SMG$_INVPAS_ID, R0             :
                                                         04 0002D        RET                                     :
                                    50 00000000G0040    D0 0002E 3$:     MOVL    PBD_A_PBCB[R0], PBCB            :
                                                    50   DD 00036        PUSHL   PBCB                            : 056
                                0000V CF            01   FB 00038        CALLS   #1, SMG$$ERASE_PASTEBOARD       :
                                                         04 0003D        RET                                     : 0

; Routine Size:  62 bytes,    Routine Base:  _SMG$CODE + 0000
```

```
;  307          0568  1 %SBTTL 'SMG$$ERASE_PASTEBOARD- Clear Screen'
;  308          0569  1 GLOBAL ROUTINE SMG$$ERASE_PASTEBOARD ( PBCB : REF $PBCB_DECL ) =
;  309          0570  1 !++
;  310          0571  1 ! FUNCTIONAL DESCRIPTION:
;  311          0572  1 !
;  312          0573  1 !       This routine erases the entire pasteboard.
;  313          0574  1 !       The physical cursor is left at (1,1).
;  314          0575  1 !
;  315          0576  1 ! CALLING SEQUENCE:
;  316          0577  1 !
;  317          0578  1 !       ret_status.wlc.v = SMG$$ERASE_PASTEBOARD ( PBCB.rab.r)
;  318          0579  1 !
;  319          0580  1 ! FORMAL PARAMETERS:
;  320          0581  1 !
;  321          0582  1 !       PBCB.rab.r      Address of pasteboard control block
;  322          0583  1 !
;  323          0584  1 ! IMPLICIT INPUTS:
;  324          0585  1 !
;  325          0586  1 !       None
;  326          0587  1 !
;  327          0588  1 ! IMPLICIT OUTPUTS:
;  328          0589  1 !
;  329          0590  1 !       None
;  330          0591  1 !
;  331          0592  1 ! COMPLETION STATUS:
;  332          0593  1 !
;  333          0594  1 !       SS$_NORMAL      Normal successful completion
;  334          0595  1 !
;  335          0596  1 ! SIDE EFFECTS:
;  336          0597  1 !
;  337          0598  1 !       NONE
;  338          0599  1 !--
```

```
;   340          0600  2 BEGIN
;   341          0601  2
;   342          0602  2 LOCAL
;   343          0603  2
;   344          0604  2         STATUS,
;   345          0605  2         WCB     : REF $WCB_DECL;        ! Address of window control block.
;   346          0606  2
;   347          0607  2 !+
;   348          0608  2 ! Flush out our buffers.
;   349          0609  2 !-
;   350          0610  2
;   351          0611  2     STATUS=SMG$$FLUSH_BUFFER(.PBCB);
;   352          0612  2     IF NOT .STATUS THEN SIGNAL(.STATUS);
;   353          0613  2
;   354          0614  2 !+
;   355          0615  2 ! Home the cursor. (erase_whole_display doesn't necessarily do that).
;   356          0616  2 !-
;   357          0617  2
;   358          0618  2     $SMG$GET_TERM_DATA(HOME);
;   359          0619  2     STATUS=OUTPUT(.PBCB,.PBCB[PBCB_L_CAP_LENGTH],.PBCB[PBCB_A_CAP_BUFFER]);
;   360          0620  2     IF NOT .STATUS THEN RETURN .STATUS;
;   361          0621  2
;   362          0622  2 !+
;   363          0623  2 ! Physically clear the screen with an escape sequence.
;   364          0624  2 !-
;   365          0625  2
;   366          0626  2     $SMG$GET_TERM_DATA(ERASE_WHOLE_DISPLAY);
;   367          0627  2
;   368          0628  2 !+
;   369          0629  2 ! Make sure it happens immediately by calling OUTPUT rather than SMG$$OUTPUT.
;   370          0630  2 ! This way it won't get buffered.
;   371          0631  2 !-
;   372          0632  2
;   373          0633  2     STATUS=OUTPUT(.PBCB,.PBCB[PBCB_L_CAP_LENGTH],.PBCB[PBCB_A_CAP_BUFFER]);
;   374          0634  2     IF NOT .STATUS THEN RETURN .STATUS;
;   375          0635  2
;   376          0636  2 !+
;   377          0637  2 ! Set the screen buffers to all blanks.
;   378          0638  2 !-
;   379          0639  2
;   380          0640  2     WCB=.PBCB[PBCB_A_WCB];
;   381          0641  2     CH$FILL(%C' ',.WCB[WCB_L_BUFSIZE],.WCB[WCB_A_TEXT_BUF]);
;   382          0642  2     CH$FILL(%C' ',.WCB[WCB_L_BUFSIZE],.WCB[WCB_A_SCR_TEXT_BUF]);
;   383          0643  2     CH$FILL(0,    .WCB[WCB_L_BUFSIZE],.WCB[WCB_A_ATTR_BUF]);
;   384          0644  2     CH$FILL(0,    .WCB[WCB_L_BUFSIZE],.WCB[WCB_A_SCR_ATTR_BUF]);
;   385          0645  2     IF .WCB[WCB_A_CHAR_SET_BUF] NEQ 0
;   386          0646  2     THEN
;   387          0647  3         BEGIN
;   388          0648  3         CH$FILL(0,.WCB[WCB_L_BUFSIZE],.WCB[WCB_A_CHAR_SET_BUF]);
;   389          0649  2         END;
;   390          0650  2
;   391          0651  2     IF .WCB[WCB_A_SCR_CHAR_SET_BUF] NEQ 0
;   392          0652  2     THEN
;   393          0653  3         BEGIN
;   394          0654  3         CH$FILL(0,.WCB[WCB_L_BUFSIZE],.WCB[WCB_A_SCR_CHAR_SET_BUF]);
;   395          0655  2         END;
;   396          0656  2
```

```
;  397          0657  2 !+
;  398          0658  2 ! The physical cursor moves to (1,1).
;  399          0659  2 !-
;  400          0660  2
;  401          0661  2     WCB[WCB_W_CURR_CUR_ROW]=1;
;  402          0662  2     WCB[WCB_W_OLD_CUR_ROW] =1;
;  403          0663  2     WCB[WCB_W_CURR_CUR_COL]=1;
;  404          0664  2     WCB[WCB_W_OLD_CUR_COL] =1;
;  405          0665  2
;  406          0666  2 !+
;  407          0667  2 ! The line characteristics get set back to 0.
;  408          0668  2 !-
;  409          0669  2
;  410          0670  2     CH$FILL(0,.WCB[WCB_W_NO_ROWS]+1,.WCB[WCB_A_LINE_CHAR]);
;  411          0671  2     CH$FILL(0,.WCB[WCB_W_NO_ROWS]+1,.WCB[WCB_A_SCR_LINE_CHAR]);
;  412          0672  2
;  413          0673  2     RETURN      SS$_NORMAL
;  414          0674  2
;  415          0675  1 END;             ! routine SMG$$ERASE_PASTEBOARD


                                                        .EXTRN  SMG$GET_TERM_DATA

                                        01FC 00000      .ENTRY  SMG$$ERASE_PASTEBOARD, Save R2,R3,R4,R5,R6,-  ; 0569
                                                                R7,R8
                                58 00000000G    00  9E 00002  MOVAB   SMG$GET_TERM_DATA, R8                  ;
                                5E              10  C2 00009  SUBL2   #16, SP                               ;
                                52        04    AC  D0 0000C  MOVL    PBCB, R2                               ; 0611
                                                52  DD 00010  PUSHL   R2                                     ;
                          0000V CF              01  FB 00012  CALLS   #1, SMG$$FLUSH_BUFFER                  ;
                                56              50  D0 00017  MOVL    R0, STATUS                             ;
                                09              56  E8 0001A  BLBS    STATUS, 1$                             ; 0612
                                                56  DD 0001D  PUSHL   STATUS                                  ;
                   00000000G    00              01  FB 0001F  CALLS   #1, LIB$SIGNAL                         ;
                                53        0108  C2  9E 00026 1$:  MOVAB   264(R2), R3                        ; 0618
                                55        00FC  C2  9E 0002B  MOVAB   252(R2), R5                            ;
                                                65  D5 00030  TSTL    (R5)                                   ;
                                                04  12 00032  BNEQ    2$                                     ;
                                                63  D4 00034  CLRL    (R3)                                   ;
                                                21  11 00036  BRB     3$                                     ;
                                          04    AE  D4 00038 2$:  CLRL    INPUT_ARGS                         ;
                                          04    AE  9F 0003B  PUSHAB  INPUT_ARGS                             ;
                                        0104    C2  DD 0003E  PUSHL   260(R2)                                ;
                                                53  DD 00042  PUSHL   R3                                    ;
                                        0100    C2  9F 00044  PUSHAB  256(R2)                                ;
                                10      AE  01DC  8F  3C 00048  MOVZWL  #476, 16(SP)                         ;
                                          10    AE  9F 0004E  PUSHAB  16(SP)                                 ;
                                                55  DD 00051  PUSHL   R5                                    ;
                                68              06  FB 00053  CALLS   #6, SMG$GET_TERM_DATA                 ;
                                3A              50  E9 00056  BLBC    STATUS, 5$                             ;
                                54        0104  C2  9E 00059 3$:  MOVAB   260(R2), R4                        ; 0619
                                                64  DD 0005E  PUSHL   (R4)                                   ;
                                                63  DD 00060  PUSHL   (R3)                                    ;
                                                52  DD 00062  PUSHL   R2                                      ;
                          0000V CF              03  FB 00064  CALLS   #3, OUTPUT                            ;
                                56              50  D0 00069  MOVL    R0, STATUS                             ;
```

```
                          38        56  E9 0006C        BLBC    STATUS, 7$                          : 0620
                                    65  D5 0006F        TSTL    (R5)                                : 0626
                                    04  12 00071        BNEQ    4$
                                    63  D4 00073        CLRL    (R3)
                                    1F  11 00075        BRB     6$
                              04    AE  D4 00077 4$:    CLRL    INPUT_ARGS
                              04    AE  9F 0007A        PUSHAB  INPUT_ARGS
                                    64  DD 0007D        PUSHL   (R4)
                                    53  DD 0007F        PUSHL   R3
                            0100    C2  9F 00081        PUSHAB  256(R2)
                 10       AE  01DA  8F  3C 00085        MOVZWL  #474, 16(SP)
                              10    AE  9F 00088        PUSHAB  16(SP)
                                    55  DD 0008E        PUSHL   R5
                          68        06  FB 00090        CALLS   #6, SMG$GET_TERM_DATA
                          78        50  E9 00093 5$     BLBC    STATUS, 11$
                                    64  DD 00096 6$     PUSHL   (R4)                                : 0633
                                    63  DD 00098        PUSHL   (R3)
                                    52  DD 0009A        PUSHL   R2
                      0000V CF      03  FB 0009C        CALLS   #3, OUTPUT
                          56        50  D0 000A1        MOVL    R0, STATUS
                          04        56  E8 000A4        BLBS    STATUS, 8$                          : 0634
                          50        56  D0 000A7 7$:    MOVL    STATUS, R0
                                        04 000AA        RET
                          56  08    A2  D0 000AB 8$:    MOVL    8(R2), WCB                          : 0640
                          57  28    A6  D0 000AF        MOVL    40(WCB), R7                         : 0641
       57        20       6E        00  2C 000B3        MOVC5   #0, (SP), #32, R7, @8(WCB)
                              08    B6     000B8
       57        20       6E        00  2C 000BA        MOVC5   #0, (SP), #32, R7, @20(WCB)         : 0642
                              14    B6     000BF
       57        00       6E        00  2C 000C1        MOVC5   #0, (SP), #0, R7, @12(WCB)          : 0643
                              0C    B6     000C6
       57        00       6E        00  2C 000C8        MOVC5   #0, (SP), #0, R7, @24(WCB)          : 0644
                              18    B6     000CD
                              10    A6  D5 000CF        TSTL    16(WCB)                             : 0645
                                    07  13 000D2        BEQL    9$
       57        00       6E        00  2C 000D4        MOVC5   #0, (SP), #0, R7, @16(WCB)          : 0648
                              10    B6     000D9
                              1C    A6  D5 000DB 9$:    TSTL    28(WCB)                             : 0651
                                    07  13 000DE        BEQL    10$
       57        00       6E        00  2C 000E0        MOVC5   #0, (SP), #0, R7, @28(WCB)          : 0654
                              1C    B6     000E5
                 24   A6 00010001   8F  D0 000E7 10$:   MOVL    #65537, 36(WCB)                     : 0662
                 20   A6 00010001   8F  D0 000EF        MOVL    #65537, 32(WCB)                     : 0661
                          57  02    A6  3C 000F7        MOVZWL  2(WCB), R7                          : 0670
                                    57  D6 000FB        INCL    R7
       57        00       6E        00  2C 000FD        MOVC5   #0, (SP), #0, R7, @44(WCB)
                              2C    B6     00102
       57        00       6E        00  2C 00104        MOVC5   #0, (SP), #0, R7, @48(WCB)          : 0671
                              30    B6     00109
                          50        01  D0 0010B        MOVL    #1, R0                              : 0673
                                        04 0010E 11$    RET                                         : 0675

; Routine Size:  271 bytes,    Routine Base  _SMG$CODE + 003E
```

```
;  417          0676  1 %SBTTL 'SMG$FLUSH_BUFFER - Flush all buffered output to terminal'
;  418          0677  1 GLOBAL ROUTINE SMG$FLUSH_BUFFER (
;  419          0678  1                                  PASTEBOARD_ID
;  420          0679  1                                ) =
;  421          0680  1 !++
;  422          0681  1 !  FUNCTIONAL DESCRIPTION:
;  423          0682  1 !
;  424          0683  1 !       This routine causes all output which has been buffered up but
;  425          0684  1 !       not yet sent to the terminal, to be output at once.
;  426          0685  1 !       It does not matter if our caller is also buffering output.
;  427          0686  1 !       When a user requests a flush, we FLUSH. And NOW.
;  428          0687  1 !
;  429          0688  1 !  CALLING SEQUENCE:
;  430          0689  1 !
;  431          0690  1 !       ret_status.wlc.v = SMG$FLUSH_BUFFER ( PASTEBOARD_ID.rl.r )
;  432          0691  1 !
;  433          0692  1 !  FORMAL PARAMETERS:
;  434          0693  1 !
;  435          0694  1 !       PASTEBOARD_ID.rl.r      The id of the PASTEBOARD for which the
;  436          0695  1 !                               flushing action is to take place.
;  437          0696  1 !
;  438          0697  1 !  IMPLICIT INPUTS:
;  439          0698  1 !
;  440          0699  1 !       None
;  441          0700  1 !
;  442          0701  1 !  IMPLICIT OUTPUTS:
;  443          0702  1 !
;  444          0703  1 !       None
;  445          0704  1 !
;  446          0705  1 !  COMPLETION STATUS:
;  447          0706  1 !
;  448          0707  1 !       SS$_NORMAL          Normal successful completion
;  449          0708  1 !       SMG$_WRONUMARG      Wrong number of arguments.
;  450          0709  1 !       SMG$_INVPAS_ID      Invalid pasteboard id.
;  451          0710  1 !
;  452          0711  1 !  SIDE EFFECTS:
;  453          0712  1 !
;  454          0713  1 !       NONE
;  455          0714  1 !--
```

```
;  457          0715  2 BEGIN
;  458          0716  2
;  459          0717  2 LOCAL
;  460          0718  2
;  461          0719  2         PBCB;           ! Address of associated
;  462          0720  2                         ! pasteboard control block.
;  463          0721  2
;  464          0722  2 !+
;  465          0723  2 ! Isolate pasteboard control block and call inner routine to do the
;  466          0724  2 ! work.
;  467          0725  2 !-
;  468          0726  2
;  469          0727  2 $SMGSGET_PBCB(.PASTEBOARD_ID,PBCB);      ! Get address of PBCB
;  470          0728  2
;  471          0729  2 RETURN SMGSSFLUSH_BUFFER(.PBCB)
;  472          0730  2
;  473          0731  1 END;                    ! Routine SMGSFLUSH_BUFFER



                                    0000 00000          .ENTRY  SMGSFLUSH_BUFFER, Save nothing          ; 0677
                        50      04  BC  D0 00002        MOVL    @PASTEBOARD_ID, R0                      ; 0727
                                    11  19 00006        BLSS    1$
                00000000G   00      50  D1 00008        CMPL    R0, PBD_L_COUNT
                                    08  14 0000F        BGTR    1$
            08 00000000G    00      50  E0 00011        BBS     R0, PBD_V_PB_AVAIL, 2$
                        50 00000000G   8F  D0 00019 1$:  MOVL    #SMG$_INVPAS_ID, R0
                                       04 00020         RET
                        50 00000000G0040   D0 00021 2$:  MOVL    PBD_A_PBCB[R0], PBCB
                                       50  DD 00029      PUSHL   PBCB                                    ; 0729
                    0000V   CF         01  FB 0002B     CALLS   #1, SMGSSFLUSH_BUFFER
                                       04 00030         RET                                             ; 0731

; Routine Size:  49 bytes,    Routine Base:  _SMG$CODE + 014D
```

```
;  475          0732  1 %SBTTL 'SMG$$CHECK_HDWR_SCROLL - Check to see if use of hardware scroll will help'
;  476          0733  1 GLOBAL ROUTINE SMG$$CHECK_HDWR_SCROLL (
;  477          0734  1                                       PBCB : REF $PBCB_DECL
;  478          0735  1                                       ) =
;  479          0736  1 !++
;  480          0737  1 ! FUNCTIONAL DESCRIPTION:
;  481          0738  1 !
;  482          0739  1 !       This routine checks to see if the WCB text buffer has changed
;  483          0740  1 !       in such a way that we can optimize the output by using
;  484          0741  1 !       hardware scrolling regions and letting the terminal scroll.
;  485          0742  1 !
;  486          0743  1 !       Screen Text             Text
;  487          0744  1 !       Buffer                  Buffer
;  488          0745  1 !
;  489          0746  1 !       +---------------+       +---------------+
;  490          0747  1 !       |               |       |               |
;  491          0748  1 !       |               |       |               |
;  492          0749  1 ! N --->|               |       |               | <--- N
;  493          0750  1 !       |               |       |               |
;  494          0751  1 !       |               |       |               |
;  495          0752  1 ! M --->|               |       |               | <--- M
;  496          0753  1 !       |               |       |               |
;  497          0754  1 !       |               |       |               |
;  498          0755  1 !       +---------------+       +---------------+
;  499          0756  1 !
;  500          0757  1 !       If information in the PBCB tells us that lines N through M
;  501          0758  1 !       of the text buffer have been changed, we check to see if
;  502          0759  1 !
;  503          0760  1 !               Line N   of Text Buffer = Line N+1 of Screen Text Buf.
;  504          0761  1 !               Line N+1 of Text Buffer = Line N+2 of Screen Text Buf.
;  505          0762  1 !               .
;  506          0763  1 !               .
;  507          0764  1 !               .
;  508          0765  1 !
;  509          0766  1 !               Line M-1 of Text Buffer = Line M of Screen Text Buf.
;  510          0767  1 !
;  511          0768  1 !       This can be done with a single compare instruction since the
;  512          0769  1 !       areas are continguous.
;  513          0770  1 !
;  514          0771  1 !       If these areas are the same, the probability is very high that
;  515          0772  1 !       the text buffer was changed by scrolling line N through M
;  516          0773  1 !       upward by one line and inserting a new line (M) into the buffer.
;  517          0774  1 !       If we determine that this is the case, we use the hardware to
;  518          0775  1 !       accomplish the scroll for us, update the screen text buffer to
;  519          0776  1 !       reflect the effects of the scroll, and then fall into the
;  520          0777  1 !       normal minimal update logic to patch up minor differences, e.g.,
;  521          0778  1 !       attribute information.
;  522          0779  1 !
;  523          0780  1 !       In an analogous fashion, we also check to see if the change
;  524          0781  1 !       represents a downscroll of one line.
;  525          0782  1 !
;  526          0783  1 !       This routine is called only when it has already been established
;  527          0784  1 !       that we are dealing with a device that has settable scrolling
;  528          0785  1 !       regions and at least 2 consecutive lines have changed.
;  529          0786  1 !
;  530          0787  1 ! CALLING SEQUENCE:
;  531          0788  1 !
```

```
;  532          0789  1 !       ret_status.wlc.v = SMG$$CHECK_HDWR_SCROLL ( PBCB.rl.r)
;  533          0790  1 !
;  534          0791  1 ! FORMAL PARAMETERS:
;  535          0792  1 !
;  536          0793  1 !       PBCB.rl.r       Address of a Pasteboard Control Block
;  537          0794  1 !
;  538          0795  1 ! IMPLICIT INPUTS.
;  539          0796  1 !
;  540          0797  1 !       NONE
;  541          0798  1 !
;  542          0799  1 ! IMPLICIT OUTPUTS:
;  543          0800  1 !
;  544          0801  1 !       NONE
;  545          0802  1 !
;  546          0803  1 ! COMPLETION STATUS:
;  547          0804  1 !
;  548          0805  1 !       SS$_NORMAL      Normal Successful Completion
;  549          0806  1 !
;  550          0807  1 ! SIDE EFFECTS:
;  551          0808  1 !
;  552          0809  1 !       NONE
;  553          0810  1 !--
```

```
;  555          0811  2 BEGIN
;  556          0812  2
;  557          0813  2 LOCAL
;  558          0814  2
;  559          0815  2         STATUS,                 ! Status of subroutine calls
;  560          0816  2
;  561          0817  2         WCB : REF $WCB_DECL,    ! Address of associated Window Control
;  562          0818  2                                 ! Block
;  563          0819  2
;  564          0820  2         TB,                     ! Index of 1st byte in WCB text buffer
;  565          0821  2                                 ! that may have been changed.
;  566          0822  2
;  567          0823  2         STB,                    ! Index into WCB Screen Text buffer
;  568          0824  2                                 ! of starting byte position which
;  569          0825  2                                 ! should be the same if changes were
;  570          0826  2                                 ! made via a single-line scroll
;  571          0827  2                                 ! operation.
;  572          0828  2
;  573          0829  2         WIDTH,                  ! Longword counterpart of
;  574          0830  2                                 ! .WCB [WCB_W_NO_COLS] -- extracted to
;  575          0831  2                                 ! yield better code.
;  576          0832  2
;  577          0833  2         BTC,                    ! Number of bytes that need to be
;  578          0834  2                                 ! compared.
;  579          0835  2
;  580          0836  2         LCS : REF VECTOR [,BYTE],       ! Address of line
;  581          0837  2                                         ! characteristics vector assoc.
;  582          0838  2                                         ! with WCB [WCB_A_SCR_TEXT_BUF].
;  583          0839  2
;  584          0840  2         SR,     ! Top or bottom line of scrolling region.
;  585          0841  2                 ! =.LCR if scrolling up
;  586          0842  2                 ! =.FCR if scrolling down
;  587          0843  2
;  588          0844  2         FCR,    ! First changed row = .PBCB [PBCB_W_FIRST_CHANGED_ROW]
;  589          0845  2         LCR,    ! Last  changed row = .PBCB [PBCB_W_LAST_CHANGED_ROW]
;  590          0846  2                 ! Extracting the fields above gives better code
;  591          0847  2
;  592          0848  2         DL;     ! Delta number of lines (-1) that changed, = .LCR - .FCR
```

```
;  594          0849  2 WCB = .PBCB [PBCB_A_WCB];
;  595          0850  2
;  596          0851  2 !+
;  597          0852  2 ! Extract following fields for better code generation.
;  598          0853  2 !-
;  599          0854  2
;  600          0855  2 WIDTH = .WCB [WCB_W_NO_COLS];
;  601          0856  2 FCR = .PBCB [PBCB_W_FIRST_CHANGED_ROW];
;  602          0857  2 LCR = .PBCB [PBCB_W_LAST_CHANGED_ROW];
;  603          0858  2
;  604          0859  2 DL = .LCR - .FCR;        ! Known to be 1 or greater
;  605          0860  2
;  606          0861  2 !+
;  607          0862  2 ! Calc. the starting byte position in the text buffer that could have
;  608          0863  2 ! changed.
;  609          0864  2 !-
;  610          0865  2
;  611          0866  2 TB = (.FCR - 1) * .WIDTH;
;  612          0867  2
;  613          0868  2 !+
;  614          0869  2 ! Calc. the corresponding byte position in the screen text buffer that
;  615          0870  2 ! should match if change was brought about by an upward scroll of one
;  616          0871  2 ! line -- a common phenomena.  This will be one line further down in the
;  617          0872  2 ! buffer.
;  618          0873  2 !-
;  619          0874  2
;  620          0875  2 STB = .TB + .WIDTH;
;  621          0876  2
;  622          0877  2 !+
;  623          0878  2 ! Calc. how many byte positions in the text buffer should match the
;  624          0879  2 ! given slot in the screen text buffer.
;  625          0880  2 !-
;  626          0881  2
;  627          0882  2 BTC = ( .DL ) * .WIDTH;
;  628          0883  2
;  629          0884  2 !+
;  630          0885  2 ! Check to see if an upscroll or downscroll of one line accounts for
;  631          0886  2 ! the differences between the text and screen buffers.
;  632          0887  2 !-
;  633          0888  3 IF (CH$EQL ( .BTC, .WCB [WCB_A_TEXT_BUF]     + .TB,
;  634          0889  3                .BTC, .WCB [WCB_A_SCR_TEXT_BUF] + .STB))
;  635          0890  2 THEN
;  636          0891  2     SR = .LCR   ! Will be upscrolling
;  637          0892  2 ELSE
;  638          0893  3     BEGIN       ! Check for downscrolling
;  639          0894  3     TB = .FCR * .WIDTH ; ! Line N+1
;  640          0895  3     STB = .TB - .WIDTH ; ! Line N
;  641          0896  4     IF (CH$EQL ( .BTC, .WCB [WCB_A_TEXT_BUF]     + .TB,
;  642          0897  4                  .BTC, .WCB [WCB_A_SCR_TEXT_BUF] + .STB))
;  643          0898  3     THEN
;  644          0899  3         SR = .FCR       ! Will be downscrolling
;  645          0900  3     ELSE
;  646          0901  3         RETURN (SS$_NORMAL);    ! Quit -- neither upscroll or downscroll
;  647          0902  3                                 ! of 1 line will do it.
;  648          0903  3     END;        ! Check for downscrolling
;  649          0904  2
;  650          0905  2 !+
```

```
;  651          0906  2 ! If we reach here, we have a candidate for scrolling.
;  652          0907  2 ! Check to see if physical scrolling region on the terminal matches
;  653          0908  2 ! the area we want to scroll.  If not, set it to the desired region
;  654          0909  2 ! and record where we left it.
;  655          0910  2 !-
;  656          0911  2
;  657          0912  2 IF .FCR  NEQ .PBCB [PBCB_W_TOP_SCROLL_LINE]     OR
;  658          0913  2    .LCR  NEQ .PBCB [PBCB_W_BOT_SCROLL_LINE]
;  659          0914  2 THEN
;  660          0915  3         BEGIN   ! Not where we want it, reset
;  661          0916  4         IF NOT (STATUS = SMG$$FORCE_SCROLL_REG ( .PBCB, .FCR, .LCR))
;  662          0917  4         THEN
;  663          0918  4             RETURN .STATUS;
;  664          0919  3
;  665          0920  3         END;    ! Not where we want it, reset
;  666          0921  2
;  667          0922  2 !+
;  668          0923  2 ! Set physical cursor to either top  r bottom line of scroll region.
;  669          0924  2 !-
;  670          0925  2
;  671          0926  2    SMG$$FIND_MIN_CURSOR_POS ( .PBCB,
;  672          0927  2                               .WCB [WCB_W_OLD_CUR_ROW], ! Current
;  673          0928  2                               .WCB [WCB_W_OLD_CUR_COL], ! Current
;  674          0929  2                               .SR,                      ! Desired
;  675          0930  2                               i);                       ! Desired
;  676          0931  2
;  677          0932  2 !+
;  678          0933  2 ! Update screen image with respect to current cursor positioning.
;  679          0934  2 !-
;  680          0935  2
;  681          0936  2     WCB [WCB_W_OLD_CUR_ROW] = .SR;
;  682          0937  2     WCB [WCB_W_OLD_CUR_COL] = i;
;  683          0938  2     WCB [WCB_W_CURR_CUR_ROW] = .SR;
;  684          0939  2     WCB [WCB_W_CURR_CUR_COL] = i;
;  685          0940  2
;  686          0941  2 !+
;  687          0942  2 ! Set up base of line characteristics vector for what is currently
;  688          0943  2 ! on the screen.  This vector will have to have its entries shuffled
;  689          0944  2 ! up or down.
;  690          0945  2 !-
;  691          0946  2     LCS = .WCB [WCB_A_SCR_LINE_CHAR];
;  692          0947  2
;  693          0948  2 !+
;  694          0949  2 ! Write a line-feed into the bottom line of the scrolling region or
;  695          0950  2 ! perform a down_scroll in the top line of the scrolling region,
;  696          0951  2 ! causing current lines N through M to scroll either down or up.
;  697          0952  2 !       ***NOTE:  This is not the best solution.  Writing a <LF> will
;  698          0953  2 !                 cause a blank line to be written with video attributes
;  699          0954  2 !                 of "normal".  This line should really be line M, with
;  700          0955  2 !                 all its video attributes in all their glory.
;  701          0956  2 !                 That takes too long to compute.  We compromise with
;  702          0957  2 !                 a line of normal blanks and let the rest of Min Upd
;  703          0958  2 !                 straigten it out later, even though the line will get
;  704          0959  2 !                 written twice and will flicker at low baud rates.
;  705          0960  2 !-
;  706          0961  2
;  707          0962  2 IF .SR EQL .LCR
```

```
;  708     0963  2 THEN
;  709     0964  2     BEGIN       ! Upscroll action
;  710     0965  2     !+
;  711     0966  2     ! Upscroll by outputting a <LF> in last line of scrolling region.
;  712     0967  2     !-
;  713     0968  2
;  714     0969  2     !    $SMG$GET_TERM_DATA(SCROLL_FORWARD);
;  715     0970  2     !    IF .PBCB[PBCB_L_CAP_LENGTH] EQL 0
;  716     0971  2     !    THEN
;  717     0972  2     !       RETURN 1;
;  718     0973  2     !
;  719     0974  2     !    STATUS = SMG$$OUTPUT (.PBCB, .PBCB[PBCB_L_CAP_LENGTH],
;  720     0975  2     !                                 .PBCB[PBCB_A_CAP_BUFFER]);
;  721     0976  2     !    IF NOT .STATUS THEN RETURN .STATUS;
;  722     0977  2
;  723     0978  2     STATUS = SMG$$OUTPUT (.PBCB, 1, UPLIT BYTE(10));
;  724     0979  2     IF NOT .STATUS THEN RETURN .STATUS;
;  725     0980  2
;  726     0981  2     !+
;  727     0982  2     ! Slide screen line characteristics vector up by one to correspond
;  728     0983  2     ! to lines that got scrolled up.
;  729     0984  2     !-
;  730     0985  2     CH$MOVE ( .DL, LCS [.FCR+1], LCS [.FCR]);
;  731     0986  2     LCS[.LCR]=0;
;  732     0987  2
;  733     0988  2     END         ! Upscroll action
;  734     0989  2
;  735     0990  2 ELSE
;  736     0991  2
;  737     0992  2     BEGIN       ! Downscroll action
;  738     0993  2
;  739     0994  2     !+
;  740     0995  2     ! Downscroll by emitting a reverse index or a down-scrol' escape sequence.
;  741     0996  2     !-
;  742     0997  2
;  743     0998  2     $SMG$GET_TERM_DATA(REVERSE_INDEX);
;  744     0999  2     IF .PBCB[PBCB_L_CAP_LENGTH] EQL 0
;  745     1000  2     THEN
;  746     1001  3         BEGIN
;  747     1002  3         $SMG$GET_TERM_DATA(SCROLL_REVERSE,1)
;  748     1003  3         IF .PBCB[PBCB_L_CAP_LENGTH] EQL 0
;  749     1004  3         THEN
;  750     1005  3             RETURN 1;
;  751     1006  3         END;
;  752     1007  2
;  753     1008  2     STATUS = SMG$$OUTPUT (.PBCB, .PBCB[PBCB_L_CAP_LENGTH],
;  754     1009  2                                  .PBCB[PBCB_A_CAP_BUFFER]);
;  755     1010  2     IF NOT .STATUS THEN RETURN .STATUS;
;  756     1011  2
;  757     1012  2     !+
;  758     1013  2     ! Slide screen line characteristics vector down by one to correspond
;  759     1014  2     ! to lines that got scrolled down.
;  760     1015  2     !-
;  761     1016  2     CH$MOVE ( .DL, LCS [.FCR], LCS [.FCR+1]);
;  762     1017  2     LCS[.FCR]=0;
;  763     1018  2
;  764     1019  2     END;        ! Downscroll action
```

```
;  765          1020  2
;  766          1021  2
;  767          102?  2 !+
;  768          10 .5 2 ! Update screen buffer to reflect what scrolling operation should have
;  769          ' 24  2 ! done to the screen.
;  770           025  2 !-
;  771          1026  2     ! Text that got scrolled, move screen text buffer by 1 line
;  772          1027  2     CH$MOVE ( .BTC,
;  773          1028  2               .WCB [WCB_A_SCR_TEXT_BUF] + .STB,
;  774          1029  2               .WCB [WCB_A_SCR_TEXT_BUF] + .TB);
;  775          1030  2
;  776          1031  2     ! Attributes that go along with text that scrolled
;  777          1032  2     CH$MOVE ( .BTC,
;  778          1033  2               .WCB [WCB_A_SCR_ATTR_BUF] + .STB,
;  77?          1034  2               .WCB [WCB_A_SCR_ATTR_BUF] + .TB);
;  7?           1035  2
;  7 !          1036  2     ! Blank line introduced by scroll operation
;   82          1037  2     CH$FILL ( %C' ',                                  ! Fill
;  783          1038  2               .WIDTH,                                 ! No. of chars
;  784          1039  2               .WCB [WCB_A_SCR_TEXT_BUF] + (.SR -1) * .WIDTH);
;  785          1040  2
;  786          1041  2     ! Attributes for blank line introduced by scroll
;  787          1042  2     !   NOTE: See note above.  This line of code is related.
;  788          1043  2
;  789          1044  2     CH$FILL ( 0,                                                ! Fill
;  790          1045  2               .WIDTH,                                 ! No. of chars
;  791          1046  2               .WCB [WCB_A_SCR_ATTR_BUF] + (.SR -1) * .WIDTH);
;  792          1047  2
;  793          1048  2 RETURN  SS$_NORMAL
;  794          1049  2
;  795          1050  1 END;                       ! Routine SMG$$CHECK_HDWR_SCROLL


                                        0A  0017E P.AAA:  .BYTE   10                                                    ;

                                            0FFC 00000    .ENTRY  SMG$$CHECK_HDWR_SCROLL, Save R2,R3,R4,R5,-          ; 0733
                                                          R6,R7,R8,R9,R10,R11
                                    5E          14  C2 00002    SUBL2   #20, SP                                       ;
                                    5A      04  AC  D0 00005    MOVL    PBCB, R10                                     ; 0849
                                    58      08  AA  D0 00009    MOVL    8(R10), WCB                                   ;
                                    7E      06  A8  3C 00000    MOVZWL  6(WCB), WIDTH                                 ; 0855
                                    5?    00A8  CA  32 00011    CVTWL   168(R10), FCR                                 ; 085?
                                04  AE    00AA  CA  32 00016    CVTWL   170(R10), LCR                                 ; 0857
                             7E 04  AE          56  C3 0001C    SUBL3   FCR, LCR, DL                                  ; 0859
                                    50      FF  A6  9E 00021    MOVAB   -1(R6), R0                                    ; 0866
                             5B     50      04  AE  C5 00025    MULL3   WIDTH, R0, TB                                 ;
                                        04 BE4B  9F 0002A    PUSHAB  @WIDTH[TB]                                    ; 0875
                             7E 04  AE      08  AE  C5 0002E    MULL3   WIDTH, DL, BTC                                ; 0882
                                    54      04  AE  D0 00034    MOVL    STB, R4                                       ; 0888
                          14 B844   08 B84B      6E  29 00038    CMPC3   BTC, @8(WCB)[TB], @20(WCB)[R4]                ;
                                            06  12 00040    BNEQ    1$                                            ;
                                    57      10  AE  D0 00042    MOVL    LCR, SR                                       ; 0891
                                            1F  11 00046    BRB     3$                                            ;
                             5B     56      0C  AE  C5 00048 1$:  MULL3   WIDTH, FCR, TB                                ; 0894
```

```
                04   AE            5B      0C   AE  C3 0004D          SUBL3   WIDTH, TB, STB                                  ; 0895
                                   54      04   AE  D0 00053          MOVL    STB, R4                                         ; 0896
                14 B844       08 B84B                 6E  29 00057    CMPC3   BTC, @8(WCB)[TB], @20(WCB)[R4]
                                                      03  13 0005F    BEQL    2$
                                                  0146  31 00061    BRW     18$
                                   57                 56  D0 00064 2$: MOVL    FCR, SR                                         ; 0899
          56    00F4   CA        10                 00  ED 00067 3$: CMPZV   #0, #16, 244(R10), FCR                          ; 0912
                                                      0A  12 ^^06E    BNEQ    4$
    10    AE    00F6   CA        10                 00  ED ^^070      CMPZV   #0, #16, 246(R10), LCR                          ; 0913
                                                      14  13 00078    BEQL    5$
                                   10      AE         DD 0007A 4$:    PUSHL   LCR                                             ; 0916
                                   56                 DD 0007D        PUSHL   FCR
                                   5A                 DD 0007F        PUSHL   R10
                          0000V    CF                 03  FB 00081    CALLS   #3, SMG$$FORCE_SCROLL_REG
                             14    AE                 50  D0 00086    MOVL    R0, STATUS
                                   40      14         AE  E9 0008A    BLBC    STATUS, 6$
                                                      01  DD 0008E 5$: PUSHL   #1                                             ; 0926
                                   57                 DD  00090       PUSHL   SR                                              ; 0929
                                   7E      26         A8  32 00092    CVTWL   38(WCB), -(SP)                                  ; 0929
                                   7E      24         A8  32 00096    CVTWL   36(WCB), -(SP)                                  ; 0927
                                   5A                 DD  0009A       PUSHL   R10                                             ; 0926
                    00000000G      00                 05  FB 0009C    CALLS   #5, SMG$$FIND_MIN_CURSOR_POS
                             24    A8                 57  B0 000A3    MOVW    SR, 36(WCB)                                     ; 0936
                             26    A8                 01  B0 000A7    MOVW    #1, 38(WCB)                                     ; 0937
                             20    A8                 57  B0 000AB    MOVW    SR, 32(WCB)                                     ; 0938
                             22    A8                 01  B0 000AF    MOVW    #1, 34(WCB)                                     ; 0939
                                   59      30         A8  D0 000B3    MOVL    48(WCB), LCS                                    ; 0946
                             10    AE                 57  D1 000B7    CMPL    SR, LCR                                         ; 0962
                                                      27  12 000BB    BNEQ    8$
                                         FF3E         CF  9F 000BD    PUSHAB  P.AAA                                           ; 0978
                                                      01  DD 000C1    PUSHL   #1
                                                      5A  DD 000C3    PUSHL   R10
                          0000V    CF                 03  FB 000C5    CALLS   #3, SMG$$OUTPUT
                             14    AE                 50  D0 000CA    MOVL    R0, STATUS
                                   03        14       AE  E8 000CE 6$: BLBS    STATUS, 7$                                      ; 0974
                                             0095     31  00002       BRW     15$
          6649      01 A649        08       AE  28 000D5 7$: MOVC3   DL, 1(FCR)[LCS], (FCR)[LCS]                            ; 0985
                                   10 BE49        94  000DD       CLRB    @LCR[LCS]                                       ; 0986
                                              0096 31  000E1       BRW     17$                                             ; 0962
                                   52      0108   CA  9E 000E4 8$: MOVAB   264(R10), R2                                     ; 0998
                                   53      00FC   CA  9E 000E9    MOVAB   252(R10), R3
                                                      63  D5 000EE    TSTL    (R3)
                                                      04  12 000F0    BNEQ    9$
                                                      62  D4 000F2    CLRL    (R2)
                                                      25  11 000F4    BRB     10$
                                             18       AE  D4 000F6 9$: CLRL    INPUT_ARGS
                                             18       AE  9F 000F9    PUSHAB  INPUT_ARGS
                                         0104   CA      DD 000FC    PUSHL   260(R10)
                                                      52  DD 00100    PUSHL   R2
                                         0100   CA      9F 00102    PUSHAB  256(R10)
                                   20    AE    0252     8F 3C 00106    MOVZWL  #594, 32(SP)
                                               20       AE  9F 0010C    PUSHAB  32(SP)
                                                      53  DD 0010F    PUSHL   R3
                    00000000G      00                 06  FB 00111    CALLS   #6, SMG$GET_TERM_DATA
                                   33                 50  E9 00118    BLBC    STATUS, 12$
                                                      62  D5 0011B 10$: TSTL    (R2)                                            ; 0999
                                                      36  12 0011D    BNEQ    14$
```

```
                                                 63  D5 0011F          TSTL    (R3)                                                    ; 1002
                                                 04  12 00121          BNEQ    11$
                                                 62  D4 00123          CLRL    (R2)
                                                 2A  11 00125          BRB     13$
                                      18    AE   01  D0 00127 11$:     MOVL    #1, INPUT_ARGS
                                      1C    AE   01  D0 0012B          MOVL    #1, INPUT_ARGS+4
                                           18    AE  9F 0012F          PUSHAB  INPUT_ARGS
                                         0104    CA  DD 00132          PUSHL   260(R10)
                                                 52  DD 00136          PUSHL   R2
                                         0100    CA  9F 00138          PUSHAB  256(R10)
                                      20    AE 0232  8F 3C 0013C      MOVZWL  #562, 32(SP)
                                           20    AE  9F 00142          PUSHAB  32(SP)
                                                 53  DD 00145          PUSHL   R3
                                00000000G   00       06  FB 00147     CALLS   #6, SMG$GET_TERM_DATA
                                            5C       50  E9 0014E 12$: BLBC    STATUS, 19$
                                                     62  D5 00151 13$: TSTL    (R2)                                                    ; 1005
                                                     55  13 00153      BEQL    18$
                                                 0104    CA  DD 00155 14$: PUSHL   260(R10)                                            ; 1009
                                                     62  DD 00159      PUSHL   (R2)                                                    ; 1008
                                                     5A  DD 0015B      PUSHL   R10
                                      0000V         CF   03  FB 0015D  CALLS   #3, SMG$$OUTPUT
                                        14          AE   50  D0 00162  MOVL    R0, STATUS
                                                    05       14    AE  E8 00166  BLBS    STATUS, 16$                                     ; 1010
                                                    50       14    AE  D0 0016A 15$: MOVL    STATUS, R0
                                                                       04 0016E      RET
                                01 A649                6649    08    AE  28 0016F 16$: MOVC3   DL, (FCR)[LCS], 1(FCR)[LCS]            ; 1016
                                                                   6649    94 00177      CLRB    (FCR)[LCS]                              ; 1017
                                                           56      04    AE  D0 0017A 17$: MOVL    STB, R6                               ; 1029
                                14 B84B                14 B846          6E  28 0017E      MOVC3   BTC, @20(WCB)[R6], @20(WCB)[TB]
                                                           56      04    AE  D0 00186      MOVL    STB, R6                               ; 1034
                                18 B84B                18 B846          6E  28 0018A      MOVC3   BTC, @24(WCB)[R6], @24(WCB)[TB]
                                                                           57  D7 00192      DECL    R7                                    ; 1039
                                                        57      0C    AE  C4 00194      MULL2   WIDTH, R7
                          0C    A.                  20        6E      00  2C 00198      MOVC5   #0, (SP), #32, WIDTH, @2C(WCB)[R7]
                                                                   14 B847      0019E
                          0C    AE                  00        6E      00  2C 001A1      MOVC5   #0, (SP), #0, WIDTH, @24(WCB)[R7]       ; 1046
                                                                   18 B847      001A7
                                                        50          01  D0 001AA 18$: MOVL    #1, R0                                  ; 1048
                                                                       04 001AD 19$: RET                                             ; 1050

; Routine Size:  430 by es,    Routine Base:  _SMG$CODE + 017F
```

```
;  797          1051  1 %SBTTL 'SMG$$FLUSH_BUFFER - Flush all buffered output to terminal'
;  798          1052  1 GLOBAL ROUTINE SMG$$FLUSH_BUFFER ( P_PBCB ) =
;  799          1053  1 !++
;  800          1054  1 ! FUNCTIONAL DESCRIPTION:
;  801          1055  1 !
;  802          1056  1 !       This routine causes all output which has been buffered up but
;  803          1057  1 !       not yet sent to the terminal, to be output at once.
;  804          1058  1 !
;  805          1059  1 ! CALLING SEQUENCE:
;  806          1060  1 !
;  807          1061  1 !       ret_status.wlc.v = SMG$$FLUSH_BUFFER ( P_PBCB.rab.r )
;  808          1062  1 !
;  809          1063  1 ! FORMAL PARAMETERS:
;  810          1064  1 !
;  811          1065  1 !       P_PBCB.rab.r    The pasteboard control block address for which
;  812          1066  1 !                       the flushing action is to take place.
;  813          1067  1 !
;  814          1068  1 ! IMPLICIT INPUTS:
;  815          1069  1 !
;  816          1070  1 !       PBCB[PBCB_W_OUTPUT_BUFLEN]      number of characters in buffer
;  817          1071  1 !       PBCB[PBCB_W_OUTPUT_BUFFER]      address of buffer
;  818          1072  1 !
;  819          1073  1 ! IMPLICIT OUTPUTS:
;  820          1074  1 !
;  821          1075  1 !       PBCB[PBCB_W_OUTPUT_BUFLEN]      set to 0 (indicating buffer empty)
;  822          1076  1 !
;  823          1077  1 ! COMPLETION STATUS:
;  824          1078  1 !
;  825          1079  1 !       SS$_NORMAL      Normal successful completion
;  826          1080  1 !       SS$_xyz         errors from SMG$$OUTPUT.
;  827          1081  1 !
;  828          1082  1 ! SIDE EFFECTS:
;  829          1083  1 !
;  830          1084  1 !       NONE
;  831          1085  1 !--
```

```
;  833          1086  2 BEGIN
;  834          1087  2
;  835          1088  2 BIND
;  836          1089  2
;  837          1090  2         PBCB    = .P_PBCB       : $PBCB_DECL    ! Pasteboard control block
;  838          1091  2         OUTBUF  = .PBCB[PBCB_A_OUTPUT_BUFFER]   : VECTOR,
;  839          1092  2         OUTLEN  =  PBCB[PBCB_W_OUTPUT_BUFLEN]   : WORD;
;  840          1093  2
;  841          1094  2 LOCAL
;  842          1095  2
;  843          1096  2         STATUS;
;  844          1097  2
;  845          1098  2 !+
;  846          1099  2 ! Do nothing if the buffer is empty.
;  847          1100  2 !-
;  848          1101  2
;  849          1102  2 IF .OUTLEN EQL 0
;  850          1103  2   THEN  RETURN  SS$_NORMAL;
;  851          1104  2
;  852          1105  2 !+
;  853          1106  2 ! Output the buffer now.
;  854          1107  2 ! Save time by calling OUTPUT directly rather than SMGS$OUTPUT.
;  855          1108  2 ! (SMGS$OUTPUT would try to buffer the text up anyhow.)
;  856          1109  2 !-
;  857          1110  2
;  858          1111  2 STATUS=OUTPUT(PBCB,.OUTLEN,OUTBUF);
;  859          1112  2 IF NOT .STATUS THEN RETURN .STATUS;
;  860          1113  2
;  861          1114  2 !+
;  862          1115  2 ! Note that the buffer is now empty.
;  863          1116  2 !-
;  864          1117  2
;  865          1118  2 OUTLEN=0;
;  866          1119  2
;  867          1120  2 RETURN  SS$_NORMAL
;  868          1121  2
;  869          1122  1 END;                       ! Routine SMGS$FLUSH_BUFFER



                                        0004 00000         .ENTRY  SMGS$FLUSH_BUFFER, Save R2                      ;  1052
                              52    04  AC  D0 00002         MOVL    P_PBCB, R2                                      ;  1090
                                    72  A2  B5 00006         TSTW    114(R2)                                         ;  1103
                                        14  13 00009         BEQL    1$                                              ;
                                    6C  A2  DD 00008         PUSHL   108(R2)                                         ;  1111
                              7E    72  A2  3C 0000C         MOVZWL  114(R2), -(SP)                                  ;
                                        52  DD 00012         PUSHL   R2                                              ;
                        0000V CF                03  FB 00014         CALLS   #3, OUTPUT                              ;
                              06                50  E9 00019         BLBC    STATUS, 2$                              ;  1112
                                    72  A2  B4 0001C         CLRW    114(R2)                                         ;  1118
                              50                01  D0 0001F 1$:      MOVL    #1, R0                                  ;  1120
                                                04  00022 2$:      RET                                             ;  1122

; Routine Size:  35 bytes,    Routine Base:  _SMG$CODE + 032D
```

```
;  871          1123  1 %SBTTL 'SMG$$FORCE_SCROLL_REG - Force physical scrolling regions'
;  872          1124  1 GLOBAL ROUTINE SMG$$FORCE_SCROLL_REG (
;  873          1125  1                                       PBCB : REF $PBCB_DECL,
;  874          1126  1                                       TOP_LINE,
;  875          1127  1                                       BOT_LINE
;  876          1128  1                                     ) =
;  877          1129  1 !++
;  878          1130  1 ! FUNCTIONAL DESCRIPTION:
;  879          1131  1 !
;  880          1132  1 !       This routine performs three actions:
;  881          1133  1 !               a). Construct escape sequence needed to set scroll
;  882          1134  1 !                   region.
;  883          1135  1 !               b). Output this sequence to terminal.
;  884          1136  1 !               c). Update PBCB to reflect new position of scroll
;  885          1137  1 !                   region.
;  886          1138  1 !
;  887          1139  1 !       The physical cursor is left in first row of scrolling region,
;  888          1140  1 !       COLUMN 1.
;  889          1141  1 !
;  890          1142  1 ! CALLING SEQUENCE:
;  891          1143  1 !
;  892          1144  1 !       ret_status.wlc.v = SMG$$FORCE_SCROLL_REG (
;  893          1145  1 !                                                 PBCB.rab.r,
;  894          1146  1 !                                                 TOP_LINE.rl.v,
;  895          1147  1 !                                                 BOT_LINE.rl.v)
;  896          1148  1 !
;  897          1149  1 ! FORMAL PARAMETERS:
;  898          1150  1 !
;  899          1151  1 !       PBCB.rab.r      Address of Pasteboard Control Block
;  900          1152  1 !       TOP_LINE.rl.v   Top line of physical scroll region desired.
;  901          1153  1 !       BOT_LINE.rl.v   Bottom line of physica scroll region desired.
;  902          1154  1 !
;  903          1155  1 ! IMPLICIT INPUTS:
;  904          1156  1 !
;  905          1157  1 !       NONE
;  906          1158  1 !
;  907          1159  1 ! IMPLICIT OUTPUTS:
;  908          1160  1 !
;  909          1161  1 !       NONE
;  910          1162  1 !
;  911          1163  1 ! COMPLETION STATUS:
;  912          1164  1 !
;  913          1165  1 !       SS$_NORMAL      Normal successful completion
;  914          1166  1 !       SS$_xyz         errors from SMG$$OUTPUT.
;  915          1167  1 !
;  916          1168  1 ! SIDE EFFECTS:
;  917          1169  1 !
;  918          1170  1 !       Physical scrolling region changed.
;  919          1171  1 !--
```

```
;  921          1172  2 BEGIN
;  922          1173  2 LOCAL
;  923          1174  2
;  924          1175  2     WCB         : REF $WCB_DECL,
;  925          1176  2     STATUS;                         ! Status of subroutine calls
;  926          1177  2
;  927          1178  2 WCB=.PBCB[PBCB_A_WCB];
;  928          1179  2
;  929          1180  2 !+
;  930          1181  2 ! Create escape sequence needed into capability buffer.
;  931          1182  2 !-
;  932          1183  2
;  933          1184  2 $SMG$GET_TERM_DATA(SET_SCROLL_REGION,.TOP_LINE,.BOT_LINE);
;  934          1185  2
;  935          1186  2 !+
;  936          1187  2 ! Output BUFFER.
;  937          1188  2 !-
;  938          1189  3 IF NOT (STATUS = SMG$$OUTPUT ( .PBCB, .PBCB[PBCB_L_CAP_LENGTH],
;  939          1190  3                                 .PBCB[PBCB_A_CAP_BUFFER]))
;  940          1191  2 THEN
;  941          1192  2     RETURN .STATUS;
;  942          1193  2
;  943          1194  2 !+
;  944          1195  2 ! Record where scrolling region now is.
;  945          1196  2 !-
;  946          1197  2
;  947          1198  2 PBCB [PBCB_W_TOP_SCROLL_LINE] = .TOP_LINE;
;  948          1199  2 PBCB [PBCB_W_BOT_SCROLL_LINE] = .BOT_LINE;
;  949          1200  2
;  950          1201  2 !+
;  951          1202  2 ! Move the cursor to the first row of the scrolling region, column 1.
;  952          1203  2 !-
;  953          1204  2
;  954          1205  2 $SMG$GET_TERM_DATA(SET_CURSOR_ABS,.TOP_LINE,1);
;  955          1206  2
;  956          1207  2 !+
;  957          1208  2 ! Output BUFFER.
;  958          1209  2 !-
;  959          1210  2
;  960          1211  2 IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
;  961          1212  3   THEN  BEGIN
;  962          1213  4         IF NOT (STATUS = SMG$$OUTPUT ( .PBCB, .PBCB[PBCB_L_CAP_LENGTH],
;  963          1214  4                                        .PBCB[PBCB_A_CAP_BUFFER]))
;  964          1215  4         THEN
;  965          1216  4             RETURN .STATUS;
;  966          1217  4
;  967          1218  4         !+
;  968          1219  4         ! Record where the cursor is now.
;  969          1220  4         !-
;  970          1221  4
;  971          1222  4         WCB[WCB_W_CURR_CUR_ROW]=.TOP_LINE;
;  972          1223  4         WCB[WCB_W_CURR_CUR_COL]=1;
;  973          1224  4         WCB[WCB_W_OLD_CUR_ROW]=.TOP_LINE;
;  974          1225  4         WCB[WCB_W_OLD_CUR_COL]=1;
;  975          1226  4
;  976          1227  2         END;
;  977          1228  2
```

```
;  978          1229  2 RETURN SS$_NORMAL
;  979          1230  2
;  980          1231  1 END;                ! Routine SMG$$FORCE_SCROLL_REG



                                    01FC 00000          .ENTRY  SMG$$FORCE_SCROLL_REG, Save R2,R3,R4,R5,R6,-; 1124
                                                                R7,R8
                          58 00000000G  00  9E 00002    MOVAB   SMG$GET_TERM_DATA, R8
                          5E            10  C2 00009    SUBL2   #16, SP
                          52      04    AC  D0 0000C    MOVL    PBCB, R2                                      1178
                          53      08    A2  D0 00010    MOVL    8(R2), WCB
                          56    0108    C2  9E 00014    MOVAB   264(R2), R6                                   1184
                          54    00FC    C2  9E 00019    MOVAB   252(R2), R4
                                        64  D5 0001E    TSTL    (R4)
                                        04  12 00020    BNEQ    1$
                                        66  D4 00022    CLRL    (R6)
                                        27  11 00024    BRB     2$
                    04    AE            02  D0 00026 1$: MOVL    #2, INPUT_ARGS
                    08    AE      08    AC  7D 0002A    MOVQ    TOP_LINE, INPUT_ARGS+4
                                  04    AE  9F 0002F    PUSHAB  INPUT_ARGS
                                0104    C2  DD 00032    PUSHL   260(R2)
                                        56  DD 00036    PUSHL   R6
                                0100    C2  9F 00038    PUSHAB  256(R2)
                    10    AE    023C    8F  3C 0003C    MOVZWL  #572, 16(SP)
                                  10    AE  9F 00042    PUSHAB  16(SP)
                                        54  DD 00045    PUSHL   R4
                          68            06  FB 00047    CALLS   #6, SMG$GET_TERM_DATA
                          50            50  E9 0004A    BLBC    STATUS, 4$
                          55    0104    C2  9E 0004D 2$: MOVAB   260(R2), R5                                   1190
                                        65  DD 00052    PUSHL   (R5)
                                        66  DD 00054    PUSHL   (R6)                                          1189
                                        52  DD 00056    PUSHL   R2
                0000V     CF            03  FB 00058    CALLS   #3, SMG$$OUTPUT
                          57            50  D0 0005D    MOVL    R0, STATUS
                          52            57  E9 00060    BLBC    STATUS, 6$
                00F4      C2      08    AC  B0 00063    MOVW    TOP_LINE, 244(R2)                             1198
                00F6      C2      0C    AC  B0 00069    MOVW    BOT_LINE, 246(R2)                             1199
                                        64  D5 0006F    TSTL    (R4)                                          1205
                                        04  12 00071    BNEQ    3$
                                        66  D4 00073    CLRL    (R6)
                                        29  11 00075    BRB     5$
                    04    AE            02  D0 00077 3$: MOVL    #2, INPUT_ARGS
                    08    AE      08    AC  D0 0007B    MOVL    TOP_LINE, INPUT_ARGS+4
                    0C    AE            01  D0 00080    MOVL    #1, INPUT_ARGS+8
                                  04    AE  9F 00084    PUSHAB  INPUT_ARGS
                                        65  DD 00087    PUSHL   (R5)
                                        56  DD 00089    PUSHL   R6
                                0100    C2  9F 0008B    PUSHAB  256(R2)
                    10    AE    023A    8F  3C 0008F    MOVZWL  #570, 16(SP)
                                  10    AE  9F 00095    PUSHAB  16(SP)
                                        54  DD 00098    PUSHL   R4
                          68            06  FB 0009A    CALLS   #6, SMG$GET_TERM_DATA
                          2E            50  E9 0009D 4$: BLBC    STATUS, 9$
                                        66  D5 000A0 5$: TSTL    (R6)                                          1211
```

```
                                                27  13 000A2          BEQL    8$                                      ;
                                                65  DD 000A4          PUSHL   (R5)                                    ; 1214
                                                66  DD 000A6          PUSHL   (R6)                                    ; 1213
                                                52  DD 000A8          PUSHL   R2                                      ;
                      0000V  CF                 03  FB 000AA          CALLS   #3, SMG$$OUTPUT                         ;
                             57                 50  D0 000AF          MOVL    R0, STATUS                              ;
                             04                 57  E8 000B2          BLBS    STATUS, 7$                              ;
                             50                 57  D0 000B5 6$:      MOVL    STATUS, R0                              ; 1216
                                                04 000B8          RET                                                 ;
                        20   A3       08       AC  B0 000B9 7$:      MOVW    TOP_LINE, 32(WCB)                       ; 1222
                        22   A3                 01  B0 000BE          MOVW    #1, 34(WCB)                             ; 1223
                        24   A3       08       AC  B0 000C2          MOVW    TOP_LINE, 36(WCB)                       ; 1224
                        26   A3                 01  B0 000C7          MOVW    #1, 38(WCB)                             ; 1225
                             50                 01  D0 000CB 8$:      MOVL    #1, R0                                  ; 1229
                                                04 000CE 9$:      RET                                                 ; 1231

; Routine Size:  207 bytes,    Routine Base:  _SMG$CODE + 0350
```

```
;  982          1232  1 %SBTTL 'SMG$$PBCB_EXIT_HANDLER - Exit handler'
;  983          1233  1 GLOBAL ROUTINE SMG$$PBCB_EXIT_HANDLER ( P_REASON, P_PBCB ) =
;  984          1234  1 !++
;  985          1235  1 ! FUNCTIONAL DESCRIPTION:
;  986          1236  1 !
;  987          1237  1 !       This routine gets called on image exit once for
;  988          1238  1 !       each active pasteboard.  It flushes the output
;  989          1239  1 !       on that device.  No flush occurs, however, if
;  990          1240  1 !       the CLI forced the exit, as in the user typed
;  991          1241  1 !       CTRL/Y then EXIT.
;  992          1242  1 !
;  993          1243  1 !       If device is a terminal, reset the physcial scrolling region to
;  994          1244  1 !       full screen. If the user doesn't request the screen to be cleared,
;  995          1245  1 !       then leave the cursor alone (unless the width needs to be reset).
;  996          1246  1 !
;  997          1247  1 ! CALLING SEQUENCE:
;  998          1248  1 !
;  999          1249  1 !       ret_status.wlc.v = SMG$$PBCB_EXIT_HANDLER ( P_REASON.rl.r,
; 1000          1250  1 !                                                   P_PBCB.rab.r )
; 1001          1251  1 !
; 1002          1252  1 ! FORMAL PARAMETERS:
; 1003          1253  1 !
; 1004          1254  1 !       P_REASON        Address of word that contains exit reason.
; 1005          1255  1 !                       Should be PBCB[PBCB_L_EXIT_REASON].
; 1006          1256  1 !
; 1007          1257  1 !       P_PBCB.rab.r    The pasteboard control block address for which
; 1008          1258  1 !                       the flushing action is to take place.
; 1009          1259  1 !
; 1010          1260  1 ! IMPLICIT INPUTS:
; 1011          1261  1 !
; 1012          1262  1 !       contents of PBCB
; 1013          1263  1 !
; 1014          1264  1 ! IMPLICIT OUTPUTS:
; 1015          1265  1 !
; 1016          1266  1 !       PBCB[PBCB_W_OUTPUT_BUFLEN]      set to 0 (indicating buffer empty)
; 1017          1267  1 !
; 1018          1268  1 ! COMPLETION STATUS:
; 1019          1269  1 !
; 1020          1270  1 !       SS$_NORMAL      Normal successful completion
; 1021          1271  1 !
; 1022          1272  1 ! SIDE EFFECTS:
; 1023          1273  1 !
; 1024          1274  1 !       NONE
; 1025          1275  1 !--
```

```
;  1027          1276  2 BEGIN
;  1028          1277  2
;  1029          1278  2 BIND
;  1030          1279  2
;  1031          1280  2         PBCB    = .P_PBCB       : $PBCB_DECL;
;  1032          1281  2
;  1033          1282  2 LOCAL
;  1034          1283  2         STATUS,
;  1035          1284  2         WCB : REF $WCB_DECL;    ! Address of window control block
;  1036          1285  2
;  1037          1286  2 EXTERNAL ROUTINE
;  1038          1287  2
;  1039          1288  2         SMG$CHANGE_PBD_CHARACTERISTICS;
```

```
;  1041          1289  2 WCB = .PBCB [PBCB_A_WCB];
;  1042          1290  2
;  1043          1291  2 !+
;  1044          1292  2 ! If a scrolling region is set (other than the full screen),
;  1045          1293  2 ! then reset it now, being careful to leave the cursor alone
;  1046          1294  2 ! even though SET SCROLLING REGION may move it.
;  1047          1295  2 ! Note that if we never established any scrolling regions,
;  1048          1296  2 ! the TOP_SCROLL line will be 0.
;  1049          1297  2 !-
;  1050          1298  2
;  1051          1299  2 IF    .PBCB[PBCB_W_TOP_SCROLL_LINE] NEQ 0
;  1052          1300  2 AND  (.PBCB[PBCB_W_TOP_SCROLL_LINE] NEQ 1   OR
;  1053          1301  2       .PBCB[PBCB_W_BOT_SCROLL_LINE] NEQ .WCB[WCB_W_NO_ROWS])
;  1054          1302  2 THEN
;  1055          1303  3     BEGIN       ! Remove scrolling regions
;  1056          1304  3
;  1057          1305  3     LOCAL
;  1058          1306  3
;                1307  3         FINAL_ROW,      ! Final cursor row
;  1060          1308  3         FINAL_COL;      ! Final cursor column
;  1061          1309  3
;  1062          1310  3     !+
;  1063          1311  3     ! Construct escape sequence (possibly null if not a supporting terminal)
;  1064          1312  3     ! to set the hardware scroll region to the full height of the screen.
;  1065          1313  3     !-
;  1066          1314  3
;  1067        P 1315  3     $SMG$GET_TERM_DATA(SET_SCROLL_REGION,
;  1068        P 1316  3                        1,
;  1069          1317  3                        .WCB [WCB_W_NO_ROWS]);
;  1070          1318  3
;  1071          1319  3     !+
;  1072          1320  3     ! Output BUFFER.
;  1073          1321  3     !-
;  1074          1322  3
;  1075          1323  3     IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
;  1076          1324  3     THEN
;  1077          1325  4         BEGIN   ! Issue the reset
;  1078          1326  4
;   079          1327  4         !+
;  .080          1328  4         ! Remember where the user left the physical cursor, since
;  1081          1329  4         ! changing scrolling regions might upset this.
;  1082          1330  4         !-
;  1083          1331  4
;  1084          1332  4         FINAL_ROW=.WCB[WCB_W_CURR_CUR_ROW];
;  1085          1333  4         FINAL_COL=.WCB[WCB_W_CURR_CUR_COL];
;  1086          1334  4
;  1087          1335  4         STATUS = SMG$$OUTPUT(PBCB,.PBCB[PBCB_L_CAP_LENGTH],
;  1088          1336  4                              .PBCB[PBCB_A_CAP_BUFFER]);
;  1089          1337  4         IF NOT .STATUS THEN RETURN .STATUS;
;  1090          1338  4
;  1091          1339  4         !+
;  1092          1340  4         ! Move the cursor back to where it was.
;  1093          1341  4         ! (No need to do this if the screen will be cleared anyhow.)
;  1094          1342  4         !-
;  1095          1343  4
;  1096          1344  4         IF NOT .PBCB[PBCB_V_CLEAR_SCREEN]
;  1097          1345  5           THEN  BEGIN   ! Restore final cursor position
```

```
; 1098          1346  5
; 1099          1347  5                   $SMG$GET_TERM_DATA(SET_CURSOR_ABS,.FINAL_ROW,.FINAL_COL);
; 1100          1348  5
; 1101          1349  5                   STATUS = SMG$$OUTPUT(PBCB,.PBCB[PBCB_L_CAP_LENGTH],
; 1102          1350  5                                        .PBCB[PBCB_A_CAP_BUFFER]);
; 1103          1351  5                   IF NOT .STATUS THEN RETURN .STATUS
; 1104          1352  5
; 1105          1353  5                   END     ! Restore final cursor position
; 1106          1354  5
; 1107          1355  4               END     ! Issue the reset
; 1108          1356  4
; 1109          1357  2           END;        ! Remove scrolling regions
; 1110          1358  2
; 1111          1359  2       !+
; 1112          1360  2       ! Flush the buffer associated with this pasteboard if the exit
; 1113          1361  2       ! was successful.
; 1114          1362  2       ! This prevents us from flushing the buffer on things like
; 1115          1363  2       ! CTRL/Y (SS$_CLIFRCEXT).
; 1116          1364  2       ! Ignore any errors.
; 1117          1365  2       !-
; 1118          1366  2
; 1119          1367  2       IF .PBCB[PBCB_L_EXIT_REASON]
; 1120          1368  3         THEN  BEGIN
; 1121          1369  3               !+
; 1122          1370  3               ! If output is being controlled by RMS, then
; 1123          1371  3               ! do a final (or only) snapshot.
; 1124          1372  3               ! Otherwise, merely flush the buffer.
; 1125          1373  3               !-
; 1126          1374  3               IF .PBCB[PBCB_V_RMS]
; 1127          1375  3                 THEN  SMG$SNAPSHOT(PBCB[PBCB_L_PBID])
; 1128          1376  3                 ELSE  SMG$$FLUSH_BUFFER(PBCB);
; 1129          1377  2               END;
; 1130          1378  2
; 1131          1379  2       !+
; 1132          1380  2       ! Change the terminal width back to what it used to be.
; 1133          1381  2       !-
; 1134          1382  2       IF .PBCB[PBCB_W_WIDTH] NEQ .PBCB[PBCB_W_ORIG_WIDTH]
; 1135          1383  3         THEN  BEGIN   ! Change physical width
; 1136          1384  3
; 1137          1385  3               LOCAL
; 1138          1386  3
; 1139          1387  3                       DESIRED_WIDTH,
; 1140          1388  3                       NORMAL_WIDTH,
; 1141          1389  3                       WIDE_WIDTH,
; 1142          1390  3                       WIDTH_SEQUENCE;
; 1143          1391  3
; 1144          1392  3               DESIRED_WIDTH=.PBCB[PBCB_W_ORIG_WIDTH];
; 1145          1393  3
; 1146          1394  3               !+
; 1147          1395  3               ! First, clear the screen.
; 1148          1396  3               !-
; 1149          1397  3
; 1150          1398  3               $SMG$GET_TERM_DATA(HOME);
; 1151          1399  3               STATUS=OUTPUT.PBCB,.PBCB[PBCB_L_CAP_LENGTH],.PBCB[PBCB_A_CAP_BUFFER]);
; 1152          1400  3               IF NOT .STATUS THEN RETURN .STATUS;
; 1153          1401  3
; 1154          1402  3               $SMG$GET_TERM_DATA(ERASE_WHOLE_DISPLAY);
```

```
;  1155          1403  3          IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
;  1156          1404  4            THEN  BEGIN
;  1157          1405  4                  STATUS = SMG$$OUTPUT(PBCB,.PBCB[PBCB_L_CAP_LENGTH],
;  1158          1406  4                                      .PBCB[PBCB_A_CAP_BUFFER]);
;  1159          1407  4                  IF NOT .STATUS THEN RETURN .STATUS
;  1160          1408  4                  END;
;  1161          1409
;  1162          1410          !+
;  1163          1411          ! Second, get the normal size.
;  1164          1412          !-
;  1165          1413
;  1166          1414          $SMG$GET_TERM_DATA(COLUMNS);
;  1167          1415          IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
;  1168          1416  4         THEN  BEGIN
;  1169          1417  4               BIND RESULT=.PBCB[PBCB_A_CAP_BUFFER];
;  1170          1418  4               STATUS = SMG$$OUTPUT(PBCB,.PBCB[PBCB_L_CAP_LENGTH],
;  1171          1419  4                                    .PBCB[PBCB_A_CAP_BUFFER]);
;  1172          1420  4               IF NOT .STATUS THEN RETURN .STATUS;
;  1173          1421  4               NORMAL_WIDTH=.RESULT
;  1174          1422  4               END
;  1175          1423  4         ELSE  NORMAL_WIDTH=80;
;  1176          1424
;  1177          1425          !+
;  1178          1426          ! Third, get the wide size.
;  1179          1427          !-
;  1180          1428
;  1181          1429          $SMG$GET_TERM_DATA(WIDTH_WIDE);
;  1182          1430          IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
;  1183          1431  4         THEN  BEGIN
;  1184          1432  4               BIND RESULT=.PBCB[PBCB_A_CAP_BUFFER];
;  1185          1433  4               STATUS = SMG$$OUTPUT(PBCB,.PBCB[PBCB_L_CAP_LENGTH],
;  1186          1434  4                                    .PBCB[PBCB_A_CAP_BUFFER]);
;  1187          1435  4               IF NOT .STATUS THEN RETURN .STATUS;
;  1188          1436  4               WIDE_WIDTH=.RESULT
;  1189          1437  4               END
;  1190          1438  3         ELSE  WIDE_WIDTH=80;
;  1191          1439  3
;  1192          1440  3          !+
;  1193          1441  3          ! Decide which sequence to send.
;  1194          1442  3          !-
;  1195          1443  3
;  1196          1444  3          IF .DESIRED_WIDTH GTR .NORMAL_WIDTH
;  1197          1445  4            THEN  $SMG$GET_TERM_DATA(WIDTH_NARROW)
;  1198          1446  3            ELSE  $SMG$GET_TERM_DATA(WIDTH_WIDE);
;  1199          1447  3          IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
;  1200          1448  4            THEN  BEGIN
;  1201          1449  4                  STATUS = SMG$$OUTPUT(PBCB,.PBCB[PBCB_L_CAP_LENGTH],
;  1202          1450  4                                       .PBCB[PBCB_A_CAP_BUFFER]);
;  1203          1451  4                  IF NOT .STATUS THEN RETURN .STATUS;
;  1204          1452  3                  END;
;  1205          1453  3
;  1206          1454  2          END;     ! Change physical width
;  1207          1455  2
;  1208          1456  2  !+
;  1209          1457  2  ! Clear the screen if the user asked us to.
;  1210          1458  2  !-
;  1211          1459  2
```

```
;  1212          1460  2 IF .PBCB[PBCB_V_CLEAR_SCREEN]
;  1213          1461  2   THEN  SMG$$ERASE_PASTEBOARD(PBCB);
;  1214          1462  2
;  1215          1463  2 SMG$$FLUSH_BUFFER(PBCB);
;  1216          1464  2
;  1217          1465  2 ! ***    I don't know whether or not an exit routine is supposed
;  1218          1466  2 !        to return a value; so I'm returning SS$_NORMAL for now.
;  1219          1467  2
;  1220          1468  2 RETURN  SS$_NORMAL
;  1221          1469  2
;  1222          1470  1 END;                        ! Routine SMG$$PBCB_EXIT_HANDLER


                                                        .EXTRN  SMG$CHANGE_PBD_CHARACTERISTICS

                                        0FFC 00000      .ENTRY  SMG$$PBCB_EXIT_HANDLER, Save R2,R3,R4,R5,-      ; 1233
                                                                R6,R7,R8,R9,R10,R11
                                5B      0000V   CF  9E 00002    MOVAB   SMG$$OUTPUT, R11
                                5A 00000000G    00  9E 00007    MOVAB   SMG$GET_TERM_DATA, R10
                                5E              10  C2 0000E    SUBL2   #16, SP
                                52        08    AC  D0 00011    MOVL    P_PBCB, R2                              ; 1280
                                54        08    A2  D0 00015    MOVL    8(R2), WCB                              ; 1289
                                50      00F4    C2  3C 00019    MOVZWL  244(R2), R0                             ; 1299
                                                51  13 0001E    BEQL    4$
                                01              50  B1 00020    CMPW    R0, #1                                  ; 1300
                                                08  12 00023    BNEQ    1$
                            02  A4      00F6    C2  B1 00025    CMPW    246(R2), 2(WCB)                         ; 1301
                                                44  13 0002B    BEQL    4$
                                56      00FC    C2  9E 0002D 1$: MOVAB   252(R2), R6                             ; 131
                                                66  D5 00032    TSTL    (R6)
                                                09  12 00034    BNEQ    2$
                                53      0108    C2  9E 00036    MOVAB   264(R2), R3
                                                63  D4 0003B    CLRL    (R3)
                                                30  11 0003D    BRB     3$
                        04      AE              02  D0 0003F 2$: MOVL    #2, INPUT_ARGS
                        08      AE              01  D0 00043    MOVL    #1, INPUT_ARGS+4
                        0C      AE        02    A4  3C 00047    MOVZWL  2(WCB), INPUT_ARGS+8
                                  04      AE  9F 0004C    PUSHAB  INPUT_ARGS
                                0104      C2  DD 0004F    PUSHL   260(R2)
                                53      0108  C2  9E 00053    MOVAB   264(R2), R3
                                              53  DD 00058    PUSHL   R3
                                0100      C2  9F 0005A    PUSHAB  256(R2)
                        10      AE      023C  8F  3C 0005E    MOVZWL  #572, 16(SP)
                                  10      AE  9F 00064    PUSHAB  16(SP)
                                              56  DD 00067    PUSHL   R6
                                6A            06  FB 00069    CALLS   #6, SMG$GET_TERM_DATA
                                52            50  E9 0006C    BLBC    STATUS, 6$
                                              63  D5 0006F 3$: TSTL    (R3)                                    ; 1323
                                              60  13 00071 4$: BEQL    9$
                                58        20  A4  32 00073    CVTWL   32(WCB), FINAL_ROW                      ; 1332
                                57        22  A4  32 00077    CVTWL   34(WCB), FINAL_COL                      ; 1333
                                55      0104  C2  9E 0007B    MOVAB   260(R2), R5                             ; 1336
                                              65  DD 00080    PUSHL   (R5)
                                              63  DD 00082    PUSHL   (R3)                                    ; 1335
                                              52  DD 00084    PUSHL   R2
                                6B            03  FB 00086    CALLS   #3, SMG$$OUTPUT                         ;
```

```
                                54              50  D0 00089          MOVL    R0, STATUS                                      ;
                                41              54  E9 0008C          BLBC    STATUS, 8$                                      ;  1337
                          3F    0C    A2        02  E0 0008F          BBS     #2, 12(R2), 9$                                  ;  1344
                                                66  D5 00094          TSTL    (R6)                                            ;  1347
                                                04  12 00096          BNEQ    5$                                              ;
                                                63  D4 00098          CLRL    (R3)                                            ;
                                                28  11 0009A          BRB     7$                                              ;
                                04    AE        02  D0 0009C  5$:     MOVL    #2, INPUT_ARGS                                  ;
                                08    AE        58  D0 000A0          MOVL    FINAL_ROW, INPUT_ARGS+4                         ;
                                0C    AE        57  D0 000A4          MOVL    FINAL_COL, INPUT_ARGS+8                         ;
                                          04    AE  9F 000A8          PUSHAB  INPUT_ARGS                                      ;
                                                65  DD 000AB          PUSHL   (R5)                                            ;
                                                53  DD 000AD          PUSHL   R3                                              ;
                                          0100  C2  9F 000AF          PUSHAB  256(R2)                                         ;
                                10    AE  023A  8F  3C 000B3          MOVZWL  #570, 16(SP)                                    ;
                                            10  AE  9F 000B9          PUSHAB  16(SP)                                          ;
                                                56  DD 000BC          PUSHL   R6                                              ;
                                6A              06  FB 000BE          CALLS   #6, SMG$GET_TERM_DATA                           ;
                                70              50  E9 000C1  6$:     BLBC    STATUS, 14$                                     ;
                                                65  DD 000C4  7$:     PUSHL   (R5)                                            ;  1350
                                                63  DD 000C6          PUSHL   (R3)                                            ;  1349
                                                52  DD 000C8          PUSHL   R2                                              ;
                                6B              03  FB 000CA          CALLS   #3, SMG$$OUTPUT                                 ;
                                54              50  D0 000CD          MOVL    R0, STATUS                                      ;
                                77              54  E9 000D0  8$:     BLBC    STATUS, 16$                                     ;  1351
                                17        0088  C2  E9 000D3  9$:     BLBC    136(R2), 11$                                    ;  1367
                          0A    00D0    C2        03  E1 000D8          BBC     #3, 208(R2), 10$                                ;  1373
                                            14  A2  9F 000DE          PUSHAB  20(R2)                                          ;  1374
                                0000V   CF        01  FB 000E1          CALLS   #1, SMG$SNAPSHOT                                ;
                                                07  11 000E6          BRB     11$                                             ;
                                                52  DD 000[illegible]          PUSHL   R2                                      ;  1375
                          FE1F  CF              01  FB 00[illegible]          CALLS   #1, SMG$$FLUSH_BUFFER                   ;
                          00E6  C2        5A    A2  B1 000EF  11$:     CMPW    90(R2), 230(R2)                                 ;  1382
                                                03  1[illegible] 000F5          BNEQ    12$                                     ;
                                          0175  31  00[illegible] 000F7          BRW     39$                                     ;
                                59        00E6  C2  [illegible]C 000FA  12$:     MOVZWL  230(R2), DESIRED_WIDTH                  ;  1392
                                56        00FC  C2  9E 000FF          MOVAB   252(R2), R6                                     ;  1398
                                                66  D5 00104          TSTL    (R6)                                            ;
                                                09  12 00106          BNEQ    13$                                             ;
                                53        0108  C2  9E 00108          MOVAB   264(R2), R3                                     ;
                                                63  D4 0010D          CLRL    (R3)                                            ;
                                                26  11 0010F          BRB     15$                                             ;
                                            04  AE  D4 00111  13$:     CLRL    INPUT_ARGS                                      ;
                                            04  AE  9F 00114          PUSHAB  INPUT_ARGS                                      ;
                                          0104  C2  DD 00117          PUSHL   260(R2)                                         ;
                                53        0108  C2  9E 0011B          MOVAB   264(R2), R3                                     ;
                                                53  DD 00120          PUSHL   R3                                              ;
                                          0100  C2  9F 00122          PUSHAB  256(R2)                                         ;
                          10    AE        01DC  8F  3C 00126          MOVZWL  #476, 16(SP)                                    ;
                                            10  AE  9F 0012C          PUSHAB  16(SP)                                          ;
                                                56  DD 0012F          PUSHL   R6                                              ;
                                6A              06  FB 00131          CALLS   #6, SMG$GET_TERM_DATA                           ;
                                73              50  E9 00134  14$:     BLBC    STATUS, 21$                                     ;
                                55        0104  C2  9E 00137  15$:     MOVAB   260(R2), R5                                     ;  1399
                                                65  DD 0013C          PUSHL   (R5)                                            ;
                                                63  DD 0013E          PUSHL   (R3)                                            ;
                                                62  DD 00140          PUSHL   (R2)                                            ;
```

```
                 0000V  CF           03  FB  00142          CALLS   #3, OUTPUT
                        54           50  DO  00147          MOVL    R0, STATUS
                        73           54  E9  0014A  16$     BLBC    STATUS, 23$                     1400
                                     66  D5  0014D          TSTL    (R6)                            1402
                                     04  12  0014F          BNEQ    17$
                                     63  D4  00151          CLRL    (R3)
                                     1F  11  00153          BRB     18$
                                 C4  AE  D4  00155  17$:     CLRL    INPUT_ARGS
                                 C4  AE  9F  00158          PUSHAB  INPUT_ARGS
                                     65  DD  00158          PUSHL   (R5)
                                     53  DD  0015D          PUSHL   R3
                               0100  C2  9F  0015F          PUSHAB  256(R2)
                         10    AE  01DA  8F  3C  00163      MOVZWL  #474, 16(SP)
                                 10  AE  9F  00169          PUSHAB  16(SP)
                                     56  DD  0016C          PUSHL   R6
                                 6A      06  FB  0016E      CALLS   #6, SMGSGET_TERM_DATA
                                 7C      50  E9  00171      BLBC    STATUS, 27$
                                     63  D5  00174  18$:     TSTL    (R3)                            1405
                                     0F  13  00176          BEQL    19$
                                     65  DD  00178          PUSHL   (R5)                            1406
                                     63  DD  0017A          PUSHL   (R3)                            1408
                                     52  DD  0017C          PUSHL   R2
                                 68      03  FB  0017E      CALLS   #3, SMGSSOUTPUT
                                 54      50  DO  00181      MOVL    R0, STATUS
                                 7F      54  E9  00184      BLBC    STATUS, 29$                     1407
                                     66  D5  00187  19$:     TSTL    (R6)                            1414
                                     04  12  00189          BNEQ    20$
                                     63  D4  0018B          CLRL    (R3)
                                     1E  11  0018D          BRB     22$
                                 04  AE  D4  0018F  20$:     CLRL    INPUT_ARGS
                                 04  AE  9F  00192          PUSHAB  INPUT_ARGS
                                     65  DD  00195          PUSHL   (R5)
                                     53  DD  00197          PUSHL   R3
                               0100  C2  9F  00199          PUSHAB  256(R2)
                         10    AE      DD  8F  9A  00190    MOVZBL  #221, 16(SP)
                                 10  AE  9F  001A2          PUSHAB  16(SP)
                                     56  DD  001A5          PUSHL   R6
                                 6A      0E  FB  001A7      CALLS   #6, SMGSGET_TERM_DATA
                                 43      50  E9  001AA  21$:     BLBC    STATUS, 27$
                                     63  D5  001AD  22$:     TSTL    (R3)                            1415
                                     17  13  001AF          BEQL    24$
                                 57      65  D0  001B1      MOVL    (R5), R7                        1417
                                     65  DD  001B4          PUSHL   (R5)                            1419
                                     63  DD  001B6          PUSHL   (R3)                            1418
                                     52  DD  001B8          PUSHL   R2
                                 68      03  FB  001BA      CALLS   #3, SMGSSOUTPUT
                                 54      50  D0  001BD      MOVL    R0, STATUS
                                 43      54  E9  001C0  23$:     BLBC    STATUS, 29$                 1420
                                 58      67  D0  001C3      MOVL    (R7), NORMAL_WIDTH              1421
                                         04  11  001C6      BRB     25$
                                 58  50  8F  9A  001C8  24$:     MOVZBL  #80, NORMAL_WIDTH           1423
                                         66  D5  001CC  25$:     TSTL    (R6)                        1429
                                         04  12  001CE          BNEQ    26$
                                         63  D4  001D0          CLRL    (R3)
                                         1F  11  001D2          BRB     28$
                                     04  AE  D4  001D4  26$:     CLRL    INPUT_ARGS
                                     04  AE  9F  001D7          PUSHAB  INPUT_ARGS
```

```
                                65  DD 001DA          PUSHL   (R5)                                    ;
                                53  DD 001DC          PUSHL   R3                                      ;
                          0100  C2  9F 001DE          PUSHAB  256(R2)                                 ;
                    10    AE  0246  8F  3C 001E2      MOVZWL  #582, 16(SP)                            ;
                          10  AE  9F 001E8          PUSHAB  16(SP)                                  ;
                                56  DD 001EB          PUSHL   R6                                      ;
                          6A        06  FB 001ED      CALLS   #6, SMGSGET_TERM_DATA                   ;
                          62        50  E9 001F0 27$:  BLBC    STATUS, 36$                             ;
                                63  D5 001F3 28$:  TSTL    (R3)                                    ; 1430
                                17  13 001F5          BEQL    30$                                     ;
                          57        65  D0 001F7      MOVL    (R5), R7                                ; 1432
                                65  DD 001FA          PUSHL   (R5)                                    ; 1434
                                63  DD 001FC          PUSHL   (R3)                                    ; 143?
                                52  DD 001FE          PUSHL   R2                                      ;
                          6B        03  FB 00200      CALLS   #3, SMGSSOUTPUT                         ;
                          54        50  D0 00203      MOVL    R0, STATUS                              ;
                          62        54  E9 00206 29$:  BLBC    STATUS, 38$                             ; 1435
                          50        67  D0 00209      MOVL    (R7), WIDE_WIDTH                        ; 1436
                                04  11 0020C          BRB     31$                                     ;
                          50    50  8F  9A 0020E 30$:  MOVZBL  #80, WIDE_WIDTH                       ; 1438
                          58        59  D1 00212 31$:  CMPL    DESIRED_WIDTH, NORMAL_WIDTH             ; 1444
                                1A  15 00215          BLEQ    32$                                     ;
                                66  D5 00217          TSTL    (R6)                                    ; 1445
                                1A  13 00219          BEQL    33$                                     ;
                          04    AE  D4 0021B          CLRL    INPUT_ARGS                              ;
                          04    AE  9F 0021E          PUSHAB  INPUT_ARGS                              ;
                                65  DD 00221          PUSHL   (R5)                                    ;
                                53  DD 00223          PUSHL   R3                                      ;
                          0100  C2  9F 00225          PUSHAB  256(R2)                                 ;
                    10    AE  0245  8F  3C 00229      MOVZWL  #581, 16(SP)                            ;
                                1C  11 0022F          BRB     35$                                     ;
                                66  D5 00231 32$:  TSTL    (R6)                                    ; 1446
                                04  12 00233          BNEQ    34$                                     ;
                                63  D4 00235 33$:  CLRL    (R3)                                    ;
                                1F  11 00237          BRB     37$                                     ;
                          04    AE  D4 00239 34$:  CLRL    INPUT_ARGS                              ;
                          04    AE  9F 0023C          PUSHAB  INPUT_ARGS                              ;
                                65  DD 0023F          PUSHL   (R5)                                    ;
                                53  DD 00241          PUSHL   R3                                      ;
                          0100  C2  9F 00243          PUSHAB  256(R2)                                 ;
                    10    AE  0246  8F  3C 00247      MOVZWL  #582, 16(SP)                            ;
                          1C  AE  9F 0024D 35$:  PUSHAB  16(SP)                                  ;
                                56  DD 00250          PUSHL   R6                                      ;
                          6A        06  FB 00252      CALLS   #6, SMGSGET_TERM_DATA                   ;
                          2D        50  E9 00255 36$:  BLBC    STATUS, 41$                             ;
                                63  D5 00258 37$:  TSTL    (R3)                                    ; 1447
                                13  13 0025A          BEQL    39$                                     ;
                                65  DD 0025C          PUSHL   (R5)                                    ; 1450
                                63  DD 0025E          PUSHL   (R3)                                    ; 1449
                                52  DD 00260          PUSHL   R2                                      ;
                          68        03  FB 00262      CALLS   #3, SMGSSOUTPUT                         ;
                          54        50  D0 00265      MOVL    R0, STATUS                              ;
                          04        54  E8 00268      BLBS    STATUS, 39$                             ; 1451
                          50        54  D0 0026B 38$:  MOVL    STATUS, R0                              ;
                                04  0026E          RET                                             ;
07        0C    A2                02  E1 0026F 39$:  BBC     #2, 12(R2), 40$                         ; 1460
                                52  DD 00274          PUSHL   R2                                      ; 1461
```

```
                F9A4    CF              01  FB 00276                CALLS   #1, SMG$$ERASE_PASTEBOARD               ;
                                        52  DD 00278 40$:           PUSHL   R2                                      ; 1463
                FC8C    CF              01  FB 0027D                CALLS   #1, SMG$$FLUSH_BUFFER                   ;
                        50              01  D0 00282                MOVL    #1, R0                                  ; 1468
                                            04 00285 41$:           RET                                             ; 1470

; Routine Size:  646 bytes,    Routine Base:  _SMG$CODE + 041F
```

```
;  1224          1471  1 %SBTTL 'SMG$$SETUP_TERMINAL_TYPE - Setup terminal type for SMG$$ routines'
;  1225          1472  1 GLOBAL ROUTINE SMG$$SETUP_TERMINAL_TYPE (
;  1226          1473  1                  FILE_NAME,
;  1227          1474  1                  NAME_LEN,
;  1228          1475  1                  P_TERM_TYPE,
;  1229          1476  1                  PBCB_ADR
;  1230          1477  1                  ) =
;  1231          1478  1 !++
;  1232          1479  1 ! FUNCTIONAL DESCRIPTION:
;  1233          1480  1 !
;  1234          1481  1 !       This routine uses the specified file name to determine device
;  1235          1482  1 !       characteristics and assign a terminal type code which is understood
;  1236          1483  1 !       by other SMG$$ routines.  SMG$$ routines use the terminal type to
;  1237          1484  1 !       determine the correct escape sequence for a given function (ex. set
;  1238          1485  1 !       cursor).
;  1239          1486  1 !
;  1240          1487  1 ! CALLING SEQUENCE:
;  1241          1488  1 !
;  1242          1489  1 !       ret_status.wlc.v = SMG$$SETUP_TERM_TYPE (FILE_NAME.rt.r,
;  1243          1490  1 !                                                NAME_LEN.rl.v,
;  1244          1491  1 !                                                P_TERM_TYPE.wl.r
;  1245          1492  1 !                                                [,PBCB_ADR.wl.r])
;  1246          1493  1 !
;  1247          1494  1 ! FORMAL PARAMETERS:
;  1248          1495  1 !
;  1249          1496  1 !       FILE_NAME.rt.r          addr of file name text
;  1250          1497  1 !       NAME_LEN.rl.            length of file name text
;  1251          1498  1 !       P_TERM_TYPE.wl.r        terminal type code, one of the following:
;  1252          1499  1 !                                   unknown
;  1253          1500  1 !                                   vt05        (unused)
;  1254          1501  1 !                                   vt52        (unused)
;  1255          1502  1 !                                   vt100       (unused)
;  1256          1503  1 !                                   vtforeign
;  1257          1504  1 !                                   hardcopy
;  1258          1505  1 !
;  1259          1506  1 !       PBCB_ADR.wl.r           Address of longword to receive address
;  1260          1507  1 !                                of the pasteboard control block.
;  1261          1508  1 !                                If 0 or omitted, no PBCB gets allocated.
;  1262          1509  1 !
;  1263          1510  1 ! IMPLICIT INPUTS:
;  1264          1511  1 !
;  1265          1512  1 !       NONE
;  1266          1513  1 !
;  1267          1514  1 ! IMPLICIT OUTPUTS:
;  1268          1515  1 !
;  1269          1516  1 !       PBCB fields get filled in.
;  1270          1517  1 !
;  1271          1518  1 ! COMPLETION STATUS:
;  1272          1519  1 !
;  1273          1520  1 !
;  1274          1521  1 ! SIDE EFFECTS:
;  1275          1522  1 !
;  1276          1523  1 !       NONE
;  1277          1524  1 !--
```

```
;  1279     1525  2      BEGIN
;  1280     1526  2
;  1281     1527  2      BIND
;  1282     1528  2
;  1283     1529  2          TERM_TYPE       = .P_TERM_TYPE;         ! Address to get terminal type
;  1284     1530  2
;  1285     1531  2      BUILTIN
;  1286     1532  2
;  1287     1533  2          NULLPARAMETER;
;  1288     1534  2
;  1289     1535  2      LOCAL
;  1290     1536  2
;  1291     1537  2          SMGFAB          : $FAB_DECL,
;  1292     1538  2          SMGNAM          : $NAM_DECL,
;  1293     1539  2          DEVNAM_DSC : BLOCK [8, BYTE],               ! dsc for name
;  1294     1540  2          DVI_ITMLST : VECTOR [6*3 + 1] INITIAL       ! item list for $GETDVI
;  1295     1541  2                  (DVI$_DEVTYPE     ^ 16 + 4, 0, 0,!  device type  (DT$_xyz)
;  1296     1542  2                   DVI$_DEVDEPEND   ^ 16 + 4, 0, 0,!  device dependent bits (1)
;  1297     1543  2                   DVI$_DEVDEPEND2  ^ 16 + 4, 0, 0,!  device dependent bits (2)
;  1298     1544  2                   DVI$_DEVBUFSIZ   ^ 16 + 4, 0, 0,!  terminal width
;  1299     1545  2                   DVI$_DEVCLASS    ^ 16 + 4, 0, 0,!  device class (DC$_xyz)
;  1300     1546  2                   DVI$_DEVNAM      ^ 16 +64, 0, 0,!  result name string
;  1301     1547  2                   0),                            !  terminater
;  1302     1548  2
;  1303     1549  2          DVI_EFN,                                   ! event flag for $GETDVI,
;  1304     1550  2          DVI_IOSB        : VECTOR [4, WORD],        ! I/O Status block for $GETDVI
;  1305     1551  2          STATUS,                                    ! status retd by called routines
;  1306     1552  2          DEV_TYPE        : VOLATILE,                ! storage for $GETDVI value
;  1307     1553  2          DEV_DEPEND      : VOLATILE BLOCK [4, BYTE],     ! device dependent bits (1)
;  1308     1554  2          DEV_DEPEND2     : VOLATILE BLOCK [4, BYTE],     ! device dependent bits (2)
;  1309     1555  2          DEV_BUFSIZ      : VOLATILE,                ! storage for $GETDVI value
;  1310     1556  2          DEV_CLASS       : VOLATILE,                ! storage fpr $GETDVI value
;  1311     1557  2
;  1312     1558  2          DEV_PAGSIZ,                                ! gets the number of rows of device
;  1313     1559  2
;  1314     1560  2          DEV_DEVNAM : VECTOR [64, BYTE],            ! Buffer for result name
;  1315     1561  2                                                     ! string
;  1316     1562  2
;  1317     1563  2          DEV_NAMLEN : VOLATILE WORD,                ! Length of returned
;  1318     1564  2                                                     ! resultant name string
;  1319     1565  2          TERMTABLE;                                 ! Address of terminal table
;  1320     1566  2
;  1321     1567  2      BIND
;  1322     1568  2          DVI_TYPE     = DVI_ITMLST + 4,             ! make it easy to reference
;  1323     1569  2          DVI_DEPEND   = DVI_ITMLST + 16,            !
;  1324     1570  2          DVI_DEPEND2  = DVI_ITMLST + 28,            !  items retd by $GETDVI
;  1325     1571  2          DVI_BUFSIZ   = DVI_ITMLST + 40,            !
;  1326     1572  2          DVI_CLASS    = DVI_ITMLST + 52,            !
;  1327     1573  2          DVI_DEVNAM   = DVI_ITMLST + 64,            !
;  1328     1574  2          DVI_NAMLEN   = DVI_ITMLST + 68;            !
;  1329     1575  2
;  1330     1576  2      BIND
;  1331     1577  2
;  1332     1578  2          FABDEV          = SMGFAB[FAB$L_DEV]     : BLOCK[,BYTE], ! Device characteristcs
;  1333     1579  2          DVI_NAME_LEN    = SMGNAM[NAM$T_DVI]     : BYTE,
;  1334     1580  2          DVI_NAME        = SMGNAM[NAM$T_DVI]+1   : VECTOR[,BYTE];
;  1335     1581  2
```

```
; 1336          1582  2     OWN
; 1337          1583  2
; 1338          1584  2         GENERIC_ANSI_CRT_BUF    : VECTOR[16,BYTE]
; 1339          1585  2
; 1340          1586  2                 INITIAL(BYTE('GENERIC_ANSI_CRT')),
; 1341          1587  2
; 1342          1588  2         GENERIC_DEC_CRT_BUF     : VECTOR[15,BYTE]
; 1343          1589  2
; 1344          1590  2                 INITIAL(BYTE('GENERIC_DEC_CRT')),
; 1345          1591  2
; 1346          1592  2         GENERIC_ANSI_CRT_DESC   : VECTOR[2],
; 1347          1593  2         GENERIC_DEC_CRT_DESC    : VECTOR[2];
; 1348          1594  2
; 1349          1595  2     EXTERNAL ROUTINE
; 1350          1596  2
; 1351          1597  2         SMG$INIT_TERM_TABLE_BY_TYPE,        ! Initialize terminal table by type
; 1352          1598  2         SMG$INIT_TERM_TABLE,                ! Initialize terminal table by name
; 1353          1599  2         SMG$$CREATE_PASTEBOARD,             ! Create a PBCB.
; 1354          1600  2         LIB$LP_LINES;                       ! Get number of rows for lpt
```

```
;  1356          1601  2      !+
;  1357          1602  2      ! Use RMS to parse the device name.
;  1358          1603  2      ! This will give us a 1-15 character physical device name
;  1359          1604  2      ! in the DVI field in the NAM block.
;  1360          1605  2      ! (If we just use $GETDVI, it may return a 63-character hidden
;  1361          1606  2      ! device name.)
;  1362          1607  2      ! The main reason we call $PARSE is so that we can allow filenames.
;  1363          1608  2      ! If the user specifies "TTB5:" as his device, he gets a terminal.
;  1364          1609  2      ! If the user specifies "TTB5" as his output, he gets the file
;  1365          1610  2      ! TTB5.LIS on his default disk and directory.
;  1366          1611  2      !-
;  1367          1612  2
;  1368          1613  2      !+
;  1369          1614  2      ! Initialize the FAB and NAM blocks.
;  1370          1615  2      !-
;  1371          1616  2
;  1372        P 1617  2      $FAB_INIT(  FAB     = SMGFAB,
;  1373        P 1618  2                  DNM     = 'SMGOUTPUT.LIS',
;  1374        P 1619  2                  NAM     = SMGNAM,
;  1375        P 1620  2                  FNA     = .FILE_NAME,
;  1376          1621  2                  FNS     = .NAME_LEN);
;  1377          1622  2
;  1378          1623  2      $NAM_INIT(NAM=SMGNAM);
;  1379          1624  2
;  1380          1625  2      STATUS=$PARSE(FAB=SMGFAB);
;  1381          1626  2      IF NOT .STATUS THEN RETURN .STATUS;
;  1382          1627  2
;  1383          1628  2      !+
;  1384          1629  2      ! The device name is now a counted string in the NAM block
;  1385          1630  2      ! beginning at offset NAM$T_DVI.
;  1386          1631  2      ! There is an obscure case though that can occur.  If the output device
;  1387          1632  2      ! is on another node, then RMS cannot figure out the device name
;  1388          1633  2      ! so the DVI field is empty.  This can happen if an SMG job is
;  1389          1634  2      ! run as a TASK with SYS$OUTPUT defined to be SYS$NET.
;  1390          1635  2      ! This happens when you use the "TASK=FOO" kind of filespecification
;  1391          1636  2      ! to RMS.
;  1392          1637  2      ! To allow this to work, we check to see if the device characteristic
;  1393          1638  2      ! of the pasteboard device is DEV$M_NET.  If so, we bypass the call
;  1394          1639  2      ! to $GETDVI, and we fill in the fields the best we can.
;  1395          1640  2      !-
;  1396          1641  2
;  1397          1642  2      IF .FABDEV[DEV$V_NET]
;  1398          1643  2      THEN
;  1399          1644  3          BEGIN           ! Network device
;  1400          1645  3
;  1401          1646  3          !+
;  1402          1647  3          ! Fudge the items to reasonable values.
;  1403          1648  3          !-
;  1404          1649  3
;  1405          1650  3          DEV_TYPE        = DT$_MBX;
;  1406          1651  3          DEV_DEPEND      = 0;
;  1407          1652  3          DEV_DEPEND2     = 0;
;  1408          1653  3          DEV_CLASS       = DC$_MAILBOX;
;  1409          1654  3          DEV_BUFSIZ      = 80;
;  1410          1655  3          DEV_DEVNAM      = .FILE_NAME;
;  1411          1656  3          DEV_NAMLEN      = .NAME_LEN;
;  1412          1657  3          END             ! Network device
```

```
;  1413          1658  2     ELSE
;  1414          1659  2         BEGIN           ! Normal device
;  1415          1660  2
;  1416          1661  2         DVI_TYPE    = DEV_TYPE;                 ! fill in rest of itmlst
;  1417          1662  2         DVI_DEPEND  = DEV_DEPEND;
;  1418          1663  2         DVI_DEPEND2 = DEV_DEPEND2;
;  1419          1664  2         DVI_CLASS   = DEV_CLASS;
;  1420          1665  2         DVI_BUFSIZ  = DEV_BUFSIZ;
;  1421          1666  2         DVI_DEVNAM  = DEV_DEVNAM;
;  1422          1667  2         DVI_NAMLEN  = DEV_NAMLEN;
;  1423          1668  2
;  1424          1669  2         IF NOT (STATUS = LIB$GET_EF (DVI_EFN))
;  1425          1670  2         THEN RETURN (.STATUS);          ! get unique event flag number
;  1426          1671  2
;  1427          1672  2         !+
;  1428          1673  2         ! Create a descriptor for use by $GETDVI.
;  1429          1674  2         !-
;  1430          1675  2
;  1431          1676  2         DEVNAM_DSC [DSC$B_DTYPE]   =  DSC$K_DTYPE_T;
;  1432          1677  2         DEVNAM_DSC [DSC$B_CLASS]   =  DSC$K_CLASS_S;
;  1433          1678  2         DEVNAM_DSC [DSC$W_LENGTH]  = .DVI_NAME_LEN;
;  1434          1679  2         DEVNAM_DSC [DSC$A_POINTER] =  DVI_NAME;
;  1435          1680  2
;  1436        P 1681  2         STATUS = $GETDVI(       EFN     = .DVI_EFN,
;  1437        P 1682  2                                 IOSB    =  DVI_IOSB,
;  1438        P 1683  2                                 DEVNAM  =  DEVNAM_DSC,
;  1439          1684  2                                 ITMLST  =  DVI_ITMLST);
;  1440          1685  2         IF NOT .STATUS THEN RETURN (.STATUS);
;  1441          1686  2
;  1442          1687  2         ! *** Possible bug: Should we deallocate the event flag if the
;  1443          1688  2         !         $getdvi fails?  Otherwise, enough calls to this routine
;  1444          1689  2         !         that return errors will use up all the event flags.
;  1445          1690  2
;  1446          1691  3         STATUS=$WAITFR (EFN = .DVI_EFN);                ! make $GETDVI synchronous
;  1447          1692  3         IF NOT .STATUS THEN RETURN (.STATUS);
;  1448          1693  3
;  1449          1694  3         !+
;  1450          1695  3         ! When the operation completes, the final status
;  1451          1696  3         ! is left in the first word of the I/O status block.
;  1452          1697  3         !-
;  1453          1698  3
;  1454          1699  3         IF NOT .DVI_IOSB [0] THEN RETURN (.DVI_IOSB [0]);
;  1455          1700  3
;  1456          1701  2         END;    ! Normal device
;  1457          1702  2
;  1458          1703  2     !+
;  1459          1704  2     ! Calculate the number of rows and columns for our pasteboard.
;  1460          1705  2     ! If the device is a terminal, then the number of rows
;  1461          1706  2     ! is the high byte in DEVDEPEND.
;  1462          1707  2     ! If the device is not a terminal, then the number of rows
;  1463          1708  2     ! is hereby declared to be that returned by LIB$LP_LINES.
;  1464          1709  2     ! (We don't let this get bigger than 511 however, since we
;  1465          1710  2     ! store the result in a byte.)
;  1466          1711  2     ! The number of columns is the device buffer size, whether or not
;  1467          1712  2     ! the device is a terminal.
;  1468          1713  2     ! If the device is not aterminal, then we assume it will eventually
;  1469          1714  2     ! be printed on a terminal or a lineprinter, so we minimize
```

```
: 1470          1715  2      ! the width with 132.  We might wish to reconsider this idea
: 1471          1716  2      ! in the future if we ever start producing wider terminals.
: 1472          1717  2      !-
: 1473          1718  2
: 1474          1719  2      IF .DEV_CLASS EQL DC$_TERM
: 1475          1720  2      THEN
: 1476          1721  2          DEV_PAGSIZ = .DEV_DEPEND[TT$V_PAGE]
: 1477          1722  2      ELSE
: 1478          1723  2          BEGIN
: 1479          1724  3          DEV_PAGSIZ = MIN(LIB$LP_LINES(),511);
: 1480          1725  3          DEV_BUFSIZ = MIN(.DEV_BUFSIZ,132)
: 1481          1726  2          END;
: 1482          1727  2
: 1483          1728  2      !+
: 1484          1729  2      ! Allocate a pasteboard control block (PBCB).
: 1485          1730  2      !-
: 1486          1731  2
: 1487          1732  2      IF NOT NULLPARAMETER(PBCB_ADR)
: 1488          1733  2      THEN
: 1489          1734  3          BEGIN
: 1490          1735  3          STATUS = SMG$$CREATE_PASTEBOARD ( DEV_PAGSIZ, DEV_BUFSIZ, .PBCB_ADR );
: 1491          1736  3          IF NOT .STATUS THEN RETURN (.STATUS);
: 1492          1737  2          END;
: 1493          1738  2
: 1494          1739  2      !+
: 1495          1740  2      ! If the device is a terminal, we get information about it from
: 1496          1741  2      ! TERMTABLE.EXE.  We set TERM_TYPE to VTTERMTABLE.
: 1497          1742  2      !-
: 1498          1743  2
: 1499          1744  2      TERMTABLE=0;
: 1500          1745  2
: 1501          1746  2      IF .DEV_CLASS EQL DC$_TERM
: 1502          1747  2      THEN
: 1503          1748  3          BEGIN    ! Get info from TERMTABLE
: 1504          1749  3          IF .DEV_TYPE EQL TT$_UNKNOWN
: 1505          1750  4            THEN  BEGIN    ! TERMTABLE never heard of it
: 1506          1751  4                  LOCAL GENERIC_NAME;
: 1507          1752  4                  !+
: 1508          1753  4                  ! Initialize our descriptors.
: 1509          1754  4                  ! We couldn't do this with an INITIAL clause
: 1510          1755  4                  ! because that would generate a .ADDRESS directive
: 1511          1756  4                  ! which would require fixup vectors
: 1512          1757  4                  ! which would make the PSECT read/write
: 1513          1758  4                  ! which would prevent sharing in our shared image.
: 1514          1759  4                  !-
: 1515          1760  4
: 1516          1761  4                  GENERIC_ANSI_CRT_DESC[0]=%ALLOCATION(GENERIC_ANSI_CRT_BUF);
: 1517          1762  4                  GENERIC_ANSI_CRT_DESC[1]=GENERIC_ANSI_CRT_BUF;
: 1518          1763  4                  GENERIC_DEC_CRT_DESC[0]=%ALLOCATION(GENERIC_DEC_CRT_BUF);
: 1519          1764  4                  GENERIC_DEC_CRT_DESC[1]=GENERIC_DEC_CRT_BUF;
: 1520          1765  4
: 1521          1766  4                  !+
: 1522          1767  4                  ! Well, if it's either an ANSI_CRT or a DEC_CRT,
: 1523          1768  4                  ! we can handle it. DEC_CRT has priority over ANSI.
: 1524          1769  4                  !-
: 1525          1770  4                  GENERIC_NAME=0;
: 1526          1771  4                  IF .DEV_DEPEND2[TT2$V_ANSICRT]
```

```
;  1527          1772  4                   THEN  GENERIC_NAME=GENERIC_ANSI_CRT_DESC;
;  1528          1773  4                 IF .DEV_DEPEND2[TT2$V_DECCRT]
;  1529          1774  4                   THEN  GENERIC_NAME=GENERIC_DEC_CRT_DESC;
;  1530          1775  4                 IF .GENERIC_NAME EQL 0
;  1531          1776  4                   THEN  TERM_TYPE=UNKNOWN
;  1532          1777  5                   ELSE  BEGIN   ! Use a generic terminal
;  1533          1778  5                         STATUS=SMG$INIT_TERM_TABLE(.GENERIC_NAME,TERMTABLE);
;  1534          1779  5                         IF NOT .STATUS THEN RETURN .STATUS;
;  1535          1780  5                         TERM_TYPE=VTTERMTABLE
;  1536          1781  4                         END;    ! Use a generic terminal
;  1537          1782  4                 END     ! TERMTABLE never heard of it
;  1538          1783  4             ELSE  BEGIN   ! Standard TERMTABLE terminal
;  1539          1784  4                   STATUS=SMG$INIT_TERM_TABLE_BY_TYPE(DEV_TYPE,TERMTABLE);
;  1540          1785  4                   IF NOT .STATUS THEN RETURN .STATUS;
;  1541          1786  4                   TERM_TYPE=VTTERMTABLE
;  154?          1787  4                   END     ! Standard TERMTABLE terminal
;  154'          1788  3           END     ! Get info from TERMTABLE
;  1544          1789  2       ELSE
;  1545          1790  2           TERM_TYPE=UNKNOWN;
;  1546          1791  2
;  1547          1792  2       !+
;  1548          1793  2       ! Store items in the PBCB if one was created.
;  1549          1794  2       !-
;  1550          1795  2
;  1551          1796  2       IF NOT NULLPARAMETER(4)
;  1552          1797  2       THEN
;  1553          1798  3           BEGIN   ! storing into PBCB
;  1554          1799  3
;  1555          1800  3           BIND    PBCB = ..PBCB_ADR : $PBCB_DECL ;
;  1556          1801  3
;  1557          1802  3           !+
;  1558          1803  3           ! We will need an event flag for many future operations on this
;  1559          1804  3           ! pasteboard, so we store away the event flag in the PBCB.
;  1560          1805  3           !-
;  1561          1806  3
;  1562          1807  3           PBCB [PBCB_B_EFN] = .DVI_EFN;
;  1563          1808  3
;  1564          1809  3           PBCB [PBCB_B_DEVTYPE]   = .TERM_TYPE;   ! Internal type
;  1565          1810  3
;  1566          1811  3           !+
;  1567          1812  3           ! Fill in the 12-byte device characteristics block in the PBCB.
;  1568          1813  3           ! Note that the DEVDEPEND field will not be valid if the device
;  1569          1814  3           ! is not a terminal because we replace the top byte of this
;  1570          1815  3           ! longword with the device page size (as it would be for a terminal).
;  1571          1816  3           !-
;  1572          1817  3
;  1573          1818  3           PBCB [PBCB_B_PHY_DEV_TYPE]= .DEV_TYPE;  ! Physical type.
;  1574          1819  3           PBCB [PBCB_B_CLASS]     = .DEV_CLASS;   ! Device class
;  1575          1820  3           PBCB [PBCB_W_WIDTH]     = .DEV_BUFSIZ;  ! Number of columns.
;  1576          1821  3           PBCB [PBCB_W_ORIG_WIDTH]= .DEV_BUFSIZ;  ! Number of columns (original value)
;  1577          1822  3           PBCB [PBCB_L_DEVDEPEND] = .DEV_DEPEND;  ! Implicitly sets overlapped
;  1578          1823  3                                                   ! field PBCB_B_ROWS also.
;  1579          1824  3           PBCB [PBCB_B_ROWS]      = .DEV_PAGSIZ;  ! Reset it again.
;  1580          1825  3           PBCB [PBCB_L_DEVDEPEND2]= .DEV_DEPEND2; ! Secondary characteristics.
;  1581          1826  3           PBCB [PBCB_L_TERMTABLE] = .TERMTABLE;   ! Terminal table
;  1582          1827  3           PBCB [PBCB_L_LONGEST_SEQUENCE]=SMG$K_LONGEST_SEQUENCE;
;  1583          1828  3
```

```
;  1584          1829  4          !+
;  1585          1830  4          ! Allocate a buffer to hold the longest possible sequence
;  1586          1831  4          ! that can be returned to us.
;  1587          1832  4          !-
;  1588          1833  4
;  1589          1834  4          STATUS=LIB$GET_VM(PBCB[PBCB_L_LONGEST_SEQUENCE],
;  1590          1835  4                            PBCB[PBCB_A_CAP_BUFFER]);
;  1591          1836  4          IF NOT .STATUS THEN RETURN .STATUS;
;  1592          1837  4
;  1593          1838  4          !+
;  1594          1839  4          ! Create the border vector.
;  1595          1840  4          !-
;  1596          1841  4
;  1597          1842  4          ESTABLISH_BORDER_VECTOR(PBCB);
;  1598          1843  4
;  1599          1844  4          !+
;  1600          1845  4          !       If terminal does not support direct cursor positioning,
;  1601          1846  4          !       then treat it as a hardcopy device.
;  1602          1847  4          !       Otherwise, treat it as a TERMTABLE scope terminal.
;  1603          1848  4          !-
   1604          1849  4
;  1605          1850  4          IF .TERM_TYPE EQL VTTERMTABLE
;  1606          1851  4            THEN  BEGIN
;  1607          1852  4                  $SMG$GET_TERM_DATA(SET_CURSOR_ABS, 1, 1);
;  1608          1853  4                  IF .PBCB[PBCB_L_CAP_LENGTH] EQL 0
;  1609          1854  5                    THEN  BEGIN
;  1610          1855  5                          TERM_TYPE=HARDCOPY;
;  1611          1856  5                          PBCB[PBCB_B_DEVTYPE]   = .TERM_TYPE;    ! Internal type
;  1612          1857  4                          END;
;  1613          1858  4                  $SMG$GET_TERM_DATA(SCOPE);
;  1614          1859  4                  IF .PBCB[PBCB_L_CAP_LENGTH] EQL 0
;  1615          1860  5                    THEN  BEGIN
;  1616          1861  5                          TERM_TYPE=HARDCOPY;
;  1617          1862  5                          PBCB[PBCB_B_DEVTYPE]   = .TERM_TYPE;    ! Internal type
;  1618          1863  5                          END
;  1619          1864  5                    ELSE  BEGIN
;  1620          1865  5                          BIND BOOL = .PBCB[PBCB_A_CAP_BUFFER];
;  1621          1866  5                          IF NOT .BOOL
;  1622          1867  6                            THEN  BEGIN
;  1623          1868  6                                  TERM_TYPE=HARDCOPY;
;  1624          1869  6                                  PBCB[PBCB_B_DEVTYPE]   = .TERM_TYPE;    ' Internal type
;  1625          1870  5                                  END;
;  1626          1871  4                          END;
;  1627          1872  3                  END;
;  1628          1873  3
;  1629          1874  3          !+
;  1630          1875  3          ! Find out if this terminal supports high and/or wide lines,
;  1631          1876  3          ! and if it has physical tabs and backspaces.
;  1632          1877  3          !-
;  1633          1878  3
;  1634          1879  3          IF .TERMTABLE NEQ 0
;  1635          1880  4            THEN  BEGIN   ! Get info from TERMTABLE
;  1636          1881  4
;  1637          1882  4                  $SMG$GET_TERM_DATA(DOUBLE_WIDE);
;  1638          1883  4                  IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
;  1639          1884  4                    THEN  PBCB[PBCB_V_WIDE] = 1;  ! It handles wide lines
;  1640          1885  4
```

```
;  1641          ,886  4                $SMG$GET_TERM_DATA(DOUBLE_HIGH_TOP);
;  1642          1887  4                IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
;  1643          1888  4                  THEN  PBCB[PBCB_V_HIGH] = 1;  ! It handles double high lines
;  1644          1889  4
;  1645          1890  4                IF .DEV_DEPEND[TT$V_MECHTAB]
;  1646          1891  5                  THEN  BEGIN
;  1647          1892  5                        PBCB[PBCB_V_TABS] = 1;          ! It handles physical tabs
;  1648          1893  5                                                        ! We check the NOTABS bit
;  1649          1894  5                                                        ! elsewhere, because the user
;  1650          1895  5                                                        ! can dynamically change that
;  1651          1896  5                                                        ! at runtime.
;  1652          1897  4                        END;
;  1653          1898  4
;  165'          1899  4                $SMG$GET_TERM_DATA(BACKSPACE);
;  16' ,         1900  4                IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
;  1' ,6         1901  5                  THEN  BEGIN
;  ' ,57         1902  5                        BIND ANSWER = .PBCB[PBCB_A_CAP_BUFFER];
;  .658          1903  5                        IF .ANSWER
;  1659          1904  5                          THEN  PBCB[PBCB_V_BS] = 1;    ! It handles backspace
;  1660          1905  4                        END;
;  1661          1906  4
;  1662          1907  3                END;    ! Get info from TERMTABLE
;  1663          1908  3
;  1664          1909  3          !+
;  1665          1910  3          ! Fill in the device name.
;  1666          1911  3          !-
;  1667          1912  3
;  1668          1913  3          PBCB [PBCB_W_DEVNAM_LEN]= .DEV_NAMLEN;  ! Length of device name
;  1669          1914  3          CH$MOVE ( .DEV_NAMLEN, DEV_DEVNAM, PBCB[PBCB_T_DEVNAM]);
;  1670          1915  3
;  1671          1916  3          !+
;  1672          1917  3          ! Initially, we don't know what color the background is.
;  1673          1918  3          !-
;  1674          1919  3
;  1675          1920  3          PBCB [PBCB_B_BACKGROUND_COLOR] = SMG$C_COLOR_UNKNOWN;
;  1676          1921  3
;  1677          1922  3          !+
;  1678          1923  3          ! Output any initialization sequence now.
;  1679          1924  3          !-
;  1680          1925  3
;  1681          1926  3          $SMG$GET_TERM_DATA(INIT_STRING);
;  1682          1927  3
;  1683          1928  3          IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
;  1684          1929  4            THEN  BEGIN
;  1685          1930  4                  STATUS=SMG$$OUTPUT(PBCB,.PBCB[PBCB_L_CAP_LENGTH],
;  1686          1931  4                                    .PBCB[PBCB_A_CAP_BUFFER]);
;  1687          1932  4                  IF NOT .STATUS THEN RETURN .STATUS
;  1688          1933  3                  END;
;  1689          1934  3
;  1690          1935  2          END;    ! storing into PBCB
;  1691          1936  2
;  1692          1937  2      RETURN .STATUS
;  1693          1938  2
;  1694          1939  1      END;                                ! End of routine SMG$$SETUP_TERMINAL_TYPE
```

```
                                                                        .PSECT  _SMG$DATA,NOEXE,  PIC,2

52  43  5F  49  53  4E  41  5F  43  49  52  45  4E  45  47  00004 GENERIC_ANSI_CRT_BUF:
                                                                        .ASCII  \GENERIC_ANSI_CRT\                                      ;
                                                            54  00013                                                                   ;
54  52  43  5F  43  45  44  5F  43  49  52  45  4E  45  47  00014 GENERIC_DEC_CRT_BUF:
                                                                        .ASCII  \GENERIC_DEC_CRT\                                       ;
                                                            00023       .BLKB   1
                                                            00024 GENERIC_ANSI_CRT_DESC:
                                                                        .BLKB   8
                                                            0002C GENERIC_DEC_CRT_DESC:
                                                                        .BLKB   8

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

                                                            006A5       .BLKB   3
00000000  00000000  000A0004  00000000  00000000  00060004  006A8 P.AAB:  .LONG   393220, 0, 0, 655364, 0, 0, 1835012, 0, -              ;
00000000  00000000  00080004  00000000  00000000  001C0004  006C0                 0, 524292, 0, 0, 262148, 0, 0, 2097216, -           ;
00000000  00000000  00200040  00000000  00000000  00040004  006D8                 0, 0, )                                             ;
                                                  00000000  006F0                                                                   ;
        53  49  4C  2E  54  55  50  54  55  4F  47  4D  53  006F4 P.AAC:  .ASCII  \SMGOUTPUT.LIS\                                     ;

                                                                        .EXTRN  SMG$INIT_TERM_TABLE_BY_TYPE
                                                                        .EXTRN  SMG$INIT_TERM_TABLE
                                                                        .EXTRN  SMG$$CREATE_PASTEBOARD
                                                                        .EXTRN  LIB$LP_LINES, SYS$PARSE
                                                                        .EXTRN  SYS$GETDVI, SYS$WAITFR

                                                        0FFC 00000      .ENTRY  SMG$$SETUP_TERMINAL_TYPE, Save R2,R3,R4,R5,-            ; 1472
                                                                                R6,R7,R8,R9,R10,R11
                                                    5E  FE80  CE  9E 00002  MOVAB   -384(SP), SP                                    ;
                                                    5B    0C  AC  D0 00007  MOVL    P_TERM_TYPE, R11                              ; 1529
                                          7C  AE        96  AF  004C  8F  28 0000B  MOVC3   #76, P.AAB, DVI_ITMLST                  ; 1547
0050  8F                               00              6E        00  2C 00013  MOVC5   #0, (SP), #0, #80, $RMS_PTR                  ; 1621
                                                            B0  AD        0001A
                                                    B0  AD  5003  8F  B0 0001C  MOVW    #20483, $RMS_PTR                            ;
                                                    C6  AD        02  90 00022  MOVB    #2, $RMS_PTR+22                             ;
                                                    CF  AD        02  90 00026  MOVB    #2, $RMS_PTR+31                             ;
                                                    D8  AD  FF50  CD  9E 0002A  MOVAB   SMGNAM, $RMS_PTR+40                         ;
                                                    DC  AD    04  AC  D0 00030  MOVL    FILE_NAME, $RMS_PTR+44                      ;
                                                    E0  AD    BB  AF  9E 00035  MOVAB   P.AAC, $RMS_PTR+48                          ;
                                                    E4  AD    08  AC  90 0003A  MOVB    NAME_LEN, $RMS_PTR+52                       ;
                                                    E5  AD        0D  90 0003F  MOVB    #13, $RMS_PTR+53                            ;
0060  8F                               00              6E        00  2C 00043  MOVC5   #0, (SP), #0, #96, $RMS_PTR                  ; 1623
                                                        FF50  CD        0004A
                                                FF50  CD  6002  8F  B0 0004D  MOVW    #24578, $RMS_PTR                            ;
                                                      B0  AD  9F 00054  PUSHAB  SMGFAB                                          ; 1625
                                        00000000G     00        01  FB 00057  CALLS   #1, SYS$PARSE                               ;
                                                    59          50  D0 0005E  MOVL    R0, STATUS                                  ;
                                                    5D          59  E9 00061  BLBC    STATUS, 2$                                  ; 1626
                                              21    F1  AD      05  E1 00064  BBC     #5, FAB$DEV+1, 1$                           ; 1642
                                                    70  AE      01  D0 00069  MOVL    #1, DEV_TYPE                                ; 1650
                                                          6C  AE  D4 0006D  CLRL    DEV_DEPEND                                  ; 1651
                                                          68  AE  D4 00070  CLRL    DEV_DEPEND2                                 ; 1652
                                                    60  AE    A0  8F  9A 00073  MOVZBL  #160, DEV_CLASS                           ; 1653
                                                    64  AE    50  8F  9A 00078  MOVZBL  #80, DEV_BUFSIZ                           ; 1654
                                                    20  AE    04  AC  D0 0007D  MOVL    FILE_NAME, DEV_DEVNAM                       ; 1655
```

```
                1E   AE        08   AC  B0 00082          MOVW    NAME_LEN, DCV_NAMLEN                    ;  1656
                                   008A  31 00087          BRW     4$                                      ;  1642
              0080   CE        70   AE  9E 0008A  1$:     MOVAB   DEV_TYPE, DVI_TYPE                      ;  1661
              008C   CE        6C   AE  9E 00090          MOVAB   DEV_DEPEND, DVI_DEPEND                  ;  1662
              0098   CE        68   AE  9E 00096          MOVAB   DEV_DEPEND2, DVI_DEPEND2                ;  1663
              0080   CE        60   AE  9E 0009C          MOVAB   DEV_CLASS, DVI_CLASS                    ;  1664
              00A4   CE        64   AE  9E 000A2          MOVAB   DEV_BUFSIZ, DVI_BUFSIZ                  ;  1665
              008C   CE        40   AE  9E 000A8          MOVAB   DEV_DEVNAM, DVI_DEVNAM                  ;  1666
              FF40   CD        1E   AE  9E 000AE          MOVAB   DEV_NAM_EN, DVI_NAMLEN                  ;  1667
                               04   AE  9F 000B4          PUSHAB  DVI_EFN                                 ;  1669
  0C 0000000G   00                   01  FB 000B7          CALLS   #1, LIB$GET_EF
                59                   50  D0 000BE          MOVL    R0, STATUS
                44                   59  E9 000C1  2$:     BLBC    STATUS, 3$
              FF4A   CD      010E    8F  B0 000C4          MOVW    #270, DEVNAM_DSC+2                      ;  1671
              FF48   CD      FF64    CD  9B 000CB          MOVZBW  DVI_NAME_LEN, DEVNAM_DSC                ;  1672
              FF4C   CD      FF65    CD  9E 000D2          MOVAB   DVI_NAME, DEVNAM_DSC+4                  ;  1673
                                     7E  7C 000D9          CLRQ    -(SP)                                   ;  1684
                                     7E  D4 000DB          CLRL    -(SP)
                              0080   CE  9F 000DD          PUSHAB  DVI_IOSB
                              008C   CE  9F 000E1          PUSHAB  DVI_ITMLST
                              FF48   CD  9F 000E5          PUSHAB  DEVNAM_DSC
                                     7E  D4 000E9          CLRL    -(SP)
                                20   AE  DD 000EB          PUSHL   DVI_EFN
  00000000G   0C                     FB  FB 000EE          CALLS   #8, SYS$GETDVI
                59                   50  D0 000F5          MOVL    R0, STATUS
                75                   59  E9 000F8          BLBC    STATUS, 9$                              ;  1685
                                04   AE  DD 000FB          PUSHL   DVI_EFN                                 ;  1691
  00000000G   00                     01  FB 000FE          CALLS   #1, SYS$WAITFR
                59                   50  D0 00105          MOVL    R0, STATUS
                65                   59  E9 00108  3$:     BLBC    STATUS, 9$
                05              74   AE  E8 00108          BLBS    DVI_IOSB, 4$                            ;  1692
                50              74   AE  3C 0010F          MOVZWL  DVI_IOSB, R0                            ;  1699
                                     04  00113          RET
  00000042      8F              60   AE  D1 00114  4$:     CMPL    DEV_CLASS, #66                          ;  1719
                                     07  12 00119          BNEQ    5$
                08   AE         6F   AE  9A 0011E          MOVZBL  DEV_DEPEND+3, DEV_PAGSIZ                ;  1721
                                     2E  11 00123          BRB     8$
  00000000G     00                   00  FB 00125  5$:     CALLS   #0, LIB$LP_LINES                        ;  1724
  000001FF      8F                   50  D1 0012C          CMPL    R0, #511
                                     05  15 00133          BLEQ    6$
                50              01FF 8F  3C 00135          MOVZWL  #511, R0
                08   AE              50  D0 0013A  6$:     MOVL    R0, DEV_PAGSIZ
                50              64   AE  D0 0013E          MOVL    DEV_BUFSIZ, R0                          ;  1725
  00000084      8F                   50  D1 00142          CMPL    R0, #132
                                     04  15 00149          BLEQ    7$
                50              84   8F  9A 0014B          MOVZBL  #132, R0
                64   AE              50  D0 0014F  7$:     MOVL    R0, DEV_BUFSIZ
                04                   6C  91 00153  8$:     CMPB    (AP), #2                                ;  1732
                                     1B  1F 00156          BLSSU   10$
                              10     AC  D5 00158          TSTL    16(AP)
                                     16  13 0015B          BEQL    10$
                              10     AC  DD 0015D          PUSHL   PBCB_ADR                                ;  1735
                              68     AE  9F 00160          PUSHAB  DEV_BUFSIZ
                              10     AE  9F 00163          PUSHAB  DEV_PAGSIZ
  00000C00G     00                   03  FB 00166          CALLS   #3, SMG$$CREATE_PASTEBOARD
                59                   50  D0 00160          MOVL    R0, STATUS
                65                   59  E9 00170  9$      BLBC    STATUS, 14$                             ;  1736
```

```
                                  0C   AE  D4 00173 10$:     CLRL    TERMTABLE                                  : 1744
             00000042   8F        60   AE  D1 00176          CMPL    DEV_CLASS, #66                             : 1746
                                       6F  12 0017E          BNEQ    17$
                                  70   AE  D5 00180          TSTL    DEV_TYPE                                   : 1749
                                       56  12 00183          BNEQ    15$
             00000000'  EF             10  D0 00185          MOVL    #16, GENERIC_ANSI_CRT_DESC                 : 1761
             00000000'  EF  00000000'  EF  9E 0018C          MOVAB   GENERIC_ANSI_CRT_BUF, -                    : 1762
                                                                     GENERIC_ANSI_CRT_DESC+4
             00000000'  EF             0F  D0 00197          MOVL    #15, GENERIC_DEC_CRT_DESC                  : 1763
             00000000'  EF  00000000'  EF  9E 0019E          MOVAB   GENERIC_DEC_CRT_BUF, GENERIC_DEC_CRT_DESC+4 : 1764
                                  50   D4 001A9              CLRL    GENERIC_NAME                               : 1770
                        07        6B   AE  E9 001AB          BLBC    DEV_DEPEND2+3, 11$                         : 1771
                        50  00000000'  EF  9E 001AF          MOVAB   GENERIC_ANSI_CRT_DESC, GENERIC_NAME        : 1772
    07          6B      AE             05  E1 001B6 11$:     BBC     #5, DEV_DEPEND2+3, 12$                     : 1773
                        50  00000000'  EF  9E 001BB          MOVAB   GENERIC_DEC_CRT_DESC, GENERIC_NAME         : 1774
                                       50  D5 001C2 12$:     TSTL    GENERIC_NAME                               : 1775
                                       29  13 001C4          BEQL    17$
                                  0C   AE  9F 001C6          PUSHAB  TERMTABLE                                  : 1778
                                       50  DD 001C9          PUSHL   GENERIC_NAME
             00000000G  00             02  FB 001CB          CALLS   #2, SMG$INIT_TERM_TABLE
                        59             50  D0 001D2 13$:     MOVL    R0, STATUS
                        12             59  E8 001D5          BLBS    STATUS, 16$                                : 1779
                                020A   31  001D8 14$:        BRW     42$
                                  0C   AE  9F 001DB 15$:     PUSHAB  TERMTABLE                                  : 1784
                                  74   AE  9F 001DE          PUSHAB  DEV_TYPE
             00000000G  00             02  FB 001E1          CALLS   #2, SMG$INIT_TERM_TABLE_BY_TYPE
                                       E8  11 001E8          BRB     13$
                        6B             06  D0 001EA 16$:     MOVL    #6, (R11)                                  : 1786
                                       02  11 001ED          BRB     18$                                        : 1748
                                       6B  D4 001EF 17$:     CLRL    (R11)                                      : 1790
                        04             6C  91 001F1 18$:     CMPB    (AP), #4                                   : 1796
                                       E2  1F 001F4          BLSSU   14$
                                  10   AC  D5 001F6          TSTL    16(AP)
                                       DD  13 001F9          BEQL    14$
                        56        10   BC  D0 001FB          MOVL    @PBCB_ADR, R6                              : 1800
                   66   A6        04   AE  90 001FF          MOVB    DVI_EFN, 102(R6)                           : 1807
                   10   A6             6B  90 00204          MOVB    (R11), 16(R6)                              : 1809
                   59   A6        70   AE  90 00208          MOVB    DEV_TYPE, 89(R6)                           : 1818
                   58   A6        60   AE  90 0020D          MOVB    DEV_CLASS, 88(R6)                          : 1819
                   5A   A6        64   AE  B0 00212          MOVW    DEV_BUFSIZ, 90(R6)                         : 1820
                 00E6   C6        64   AE  B0 00217          MOVW    DEV_BUFSIZ, 230(R6)                        : 1821
                   5C   A6        6C   AE  D0 0021D          MOVL    DEV_DEPEND, 92(R6)                         : 1822
                   5F   A6        08   AE  90 00222          MOVB    DEV_PAGSIZ, 95(R6)                         : 1824
                   60   A6        68   AE  D0 00227          MOVL    DEV_DEPEND2, 96(R6)                        : 1825
                        58      00FC   C6  9E 0022C          MOVAB   252(R6), R8                                : 1826
                        68        0C   AE  D0 00231          MOVL    TERMTABLE, (R8)
                        5A      0100   C6  9E 00235          MOVAB   256(R6), R10                               : 1827
                        6A        FF   8F  9A 0023A          MOVZBL  #255, (R10)
                        57      0104   C6  9E 0023E          MOVAB   260(R6), R7                                : 1835
                                       57  DD 00243          PUSHL   R7
                                       5A  DD 00245          PUSHL   R10
             00000000G  00             02  FB 00247          CALLS   #2, LIB$GET_VM
                        59             50  D0 0024E          MOVL    R0, STATUS
                        84             59  E9 00251          BLBC    STATUS, 14$                                : 1836
                                       56  DD 00254          PUSHL   R6                                         : 1842
                0000V   CF             01  FB 00256          CALLS   #1, ESTABLISH_BORDER_VECTOR
                        06             6B  D1 0025B          CMPL    (R11), #6                                  : 1850
```

```
                                        7D  12 0025E          BNEQ    25$                                     ;
                                        68  D5 00260          TSTL    (R8)                                    ; 1852
                                        09  12 00262          BNEQ    19$                                     ;
                             52   0108  C6  9E 00264          MOVAB   264(R6), R2                             ;
                                        62  D4 00269          CLRL    (R2)                                    ;
                                        2F  11 0026B          BRB     20$                                     ;
                        10   AE         02  D0 0026D 19$:     MOVL    #2, INPUT_ARGS                          ;
                        14   AE         01  D0 00271          MOVL    #1, INPUT_ARGS+4                        ;
                        18   AE         01  D0 00275          MOVL    #1, INPUT_ARGS+8                        ;
                                  10   AE  9F 00279          PUSHAB  INPUT_ARGS                              ;
                                        67  DD 0027C          PUSHL   (R7)                                    ;
                             52   0108  C6  9E 0027E          MOVAB   264(R6), R2                             ;
                                        52  DD 00283          PUSHL   R2                                      ;
                                        5A  DD 00285          PUSHL   R10                                     ;
                        10   AE   023A  8F  3C 00287          MOVZWL  #570, 16(SP)                            ;
                                  10   AE  9F 0028D          PUSHAB  16(SP)                                  ;
                                        58  DD 00290          PUSHL   R8                                      ;
                00000000G    00         06  FB 00292          CALLS   #6, SMG$GET_TERM_DATA                   ;
                             79         50  E9 00299          BLBC    STATUS, 28$                             ;
                                        62  D5 0029C 20$:     TSTL    (R2)                                    ; 1853
                                        07  12 0029E          BNEQ    21$                                     ;
                             6B         05  D0 002A0          MOVL    #5, (R11)                               ; 1855
                        10   A6         6B  90 002A3          MOVB    (R11), 16(R6)                           ; 1856
                                        68  D5 002A6 21$:     TSTL    (R8)                                    ; 1858
                                        04  12 002A9          BNEQ    22$                                     ;
                                        62  D4 002AB          CLRL    (R2)                                    ;
                                        1F  11 002AD          BRB     23$                                     ;
                                  10   AE  D4 002AF 22$:     CLRL    INPUT_ARGS                              ;
                                  10   AE  9F 002B2          PUSHAB  INPUT_ARGS                              ;
                                        67  DD 002B5          PUSHL   (R7)                                    ;
                                        52  DD 002B7          PUSHL   R2                                      ;
                                        5A  DD 002B9          PUSHL   R10                                     ;
                        10   AE         12  D0 002BB          MOVL    #18, 16(SP)                             ;
                                  10   AE  9F 002BF          PUSHAB  16(SP)                                  ;
                                        58  DD 002C2          PUSHL   R8                                      ;
                00000000G    00         06  FB 002C4          CALLS   #6, SMG$GET_TERM_DATA                   ;
                             79         50  E9 002CB          BLBC    STATUS, 32$                             ;
                                        62  D5 002CE 23$:     TSTL    (R2)                                    ; 1859
                                        04  13 002D0          BEQL    24$                                     ;
                             07   00    B7  E8 002D2          BLBS    @0(R7), 25$                             ; 1866
                             6B         05  D0 002D6 24$:     MOVL    #5, (R11)                               ; 1868
                        10   A6         6B  90 002D9          MOVB    (R11), 16(R6)                           ; 1869
                                  0C   AE  D5 002DD 25$:     TSTL    TERMTABLE                               ; 1879
                                        03  12 002E0          BNEQ    26$                                     ;
                                      00AB  31 002E2          BRW     39$                                     ;
                                        68  D5 002E5 26$:     TSTL    (R8)                                    ; 1882
                                        09  12 002E7          BNEQ    27$                                     ;
                             52   0108  C6  9E 002E9          MOVAB   264(R6), R2                             ;
                                        62  D4 002EE          CLRL    (R2)                                    ;
                                        26  11 002F0          BRB     29$                                     ;
                                  10   AE  D4 002F2 27$:     CLRL    INPUT_ARGS                              ;
                                  10   AE  9F 002F5          PUSHAB  INPUT_ARGS                              ;
                                        67  DD 002F8          PUSHL   (R7)                                    ;
                             52   0108  C6  9E 002FA          MOVAB   264(R6), R2                             ;
                                        52  DD 002FF          PUSHL   R2                                      ;
                                        5A  DD 00301          PUSHL   R10                                     ;
                        10   AE   01CE  8F  3C 00303          MOVZWL  #462, 16(SP)                            ;
```

```
                                        10   AE  9F 00309          PUSHAB  16(SP)                                                    ;
                                             58  DD 0030C          PUSHL   R8                                                        ;
                          00000000G     00     06  FB 0030E          CALLS   #6, SMG$GET_TERM_DATA                                     ;
                                        68     50  E9 00315 28$:     BLBC    STATUS, 37$                                               ;
                                               62  D5 00318 29$:     TSTL    (R2)                                                      ; 1883
                                               05  13 0031A          BEQL    30$                                                       ;
                              00FA      C6     01  88 0031C          BISB2   #1, 250(R6)                                               ; 1884
                                               68  D5 00321 30$:     TSTL    (R8)                                                      ; 1886
                                               04  12 00323          BNEQ    31$                                                       ;
                                               62  D4 00325          CLRL    (R2)                                                      ;
                                               21  11 00327          BRB     33$                                                       ;
                                        10   AE  D4 00329 31$:     CLRL    INPUT_ARGS                                                ;
                                        10   AE  9F 0032C          PUSHAB  INPUT_ARGS                                                ;
                                             67  DD 0032F          PUSHL   (R7)                                                      ;
                                             52  DD 00331          PUSHL   R2                                                        ;
                                             5A  DD 00333          PUSHL   R10                                                       ;
                                10   AE  01CD  8F  3C 00335          MOVZWL  #461, 16(SP)                                              ;
                                        10   AE  9F 00338          PUSHAB  16(SP)                                                    ;
                                             58  DD 0033E          PUSHL   R8                                                        ;
                          00000000G     00     06  FB 00340          CALLS   #6, SMG$GET_TERM_DATA                                     ;
                                        36     50  E9 00347 32$:     BLBC    STATUS, 37$                                               ;
                                               62  D5 0034A 33$:     TSTL    (R2)                                                      ; 1887
                                               05  13 0034C          BEQL    34$                                                       ;
                              00FA      C6     02  88 0034E          BISB2   #2, 250(R6)                                               ; 1888
                                        05   6D  AE  E9 00353 34$:     BLBC    DEV_DEPEND+1, 35$                                         ; 1890
                              00FA      C6     04  88 00357          BISB2   #4, 250(R6)                                               ; 1892
                                               68  D5 0035C 35$:     TSTL    (R8)                                                      ; 1899
                                               04  12 0035E          BNEQ    36$                                                       ;
                                               62  D4 00360          CLRL    (R2)                                                      ;
                                               1F  11 00362          BRB     38$                                                       ;
                                        10   AE  D4 00364 36$:     CLRL    INPUT_ARGS                                                ;
                                        10   AE  9F 00367          PUSHAB  INPUT_ARGS                                                ;
                                             67  DD 0036A          PUSHL   (R7)                                                      ;
                                             52  DD 0036C          PUSHL   R2                                                        ;
                                             5A  DD 0036E          PUSHL   R10                                                       ;
                                10   AE          04  D0 00370          MOVL    #4, 16(SP)                                                ;
                                        10   AE  9F 00374          PUSHAB  16(SP)                                                    ;
                                             58  DD 00377          PUSHL   R8                                                        ;
                          00000000G     00     06  FB 00379          CALLS   #6, SMG$GET_TERM_DATA                                     ;
                                        65     50  E9 00380 37$:     BLBC    STATUS, 43$                                               ;
                                               62  D5 00383 38$:     TSTL    (R2)                                                      ; 1900
                                               09  13 00385          BEQL    39$                                                       ;
                                        05   00  B7  E9 00387          BLBC    @0(R7), 39$                                               ; 1903
                                10   00D1  C6     01  88 0038B          BISB2   #1, 209(R6)                                               ; 1904
                                12   A6          1E  AE  B0 00390 39$:     MOVW    DEV_NAMLEN, 18(R6)                                        ; 1917
18   A6         20   AE          1E  AE  28 00395          MOVC3   DEV_NAMLEN, DEV_DEVNAM, 24(R6)                                    ; 1914
                                     00F9  C6  94 0039C          CLRB    249(R6)                                                   ; 1920
                                               68  D5 003A0          TSTL    (R8)                                                      ; 1926
                                               09  12 003A2          BNEQ    40$                                                       ;
                                        52   0108  C6  9E 003A4          MOVAB   264(R6), R2                                               ;
                                               62  D4 003A9          CLRL    (R2)                                                      ;
                                               26  11 003AB          BRB     41$                                                       ;
                                        10   AE  D4 003AD 40$:     CLRL    INPUT_ARGS                                                ;
                                        10   AE  9F 003B0          PUSHAB  INPUT_ARGS                                                ;
                                             67  DD 003B3          PUSHL   (R7)                                                      ;
                                        52   0108  C6  9E 003B5          MOVAB   264(R6), R2                                               ;
                                             52  DD 003BA          PUSHL   R2                                                        ;
```

```
                                                5A  DD 003BC             PUSHL   R10
                                      10   AE     01DE  8F  3C 003BE             MOVZWL  #478, 16(SP)
                                                  10  AE  9F 003C4             PUSHAB  16(SP)
                                                      58  DD 003C7             PUSHL   R8
                                00000000G     00        06  FB 003C9             CALLS   #6, SMG$GET_TERM_DATA
                                              15        50  E9 0G3DO             BLBC    STATUS, 43$
                                                        62  D5 003D3 41$:        TSTL    (R2)                        ; 1928
                                                        0E  13 003D5             BEQL    42$
                                                        67  DD 003D7             PUSHL   (R7)                        ; 1931
                                                        62  DD 003D9             PUSHL   (R2)                        ; 1930
                                                        56  DD 003DB             PUSHL   R6
                                          0000V     CF        03  FB 003DD             CALLS   #3, SMG$$OUTPUT
                                                    59        50  D0 003E2             MOVL    R0, STATUS
                                                    50        59  D0 003E5 42$:        MOVL    STATUS, R0          ; 1937
                                                              04  003E8 43$:        RET                         ; 1939

; Routine Size:  1001 bytes,    Routine Base:  _SMG$CODE + 0701
```

```
;  1696          1940  1 %SBTTL 'ESTABLISH_BORDER_VECTOR'
;  1697          1941  1 ROUTINE ESTABLISH_BORDER_VECTOR( P_PBCB) : NOVALUE =
;  1698          1942  1 !++
;  1699          1943  1 ! FUNCTIONAL DESCRIPTION:
;  1700          1944  1 !
;  1701          1945  1 !       Creates a 16-longword vector used for translating border characters.
;  1702          1946  1 !       If the longword has a value less than 256, then it is a border
;  1703          1947  1 !       character.  If it is greater than 256, then it is the addresss of
;  1704          1948  1 !       a border character sequence.
;  1705          1949  1 !
;  1706          1950  1 ! CALLING SEQUENCE:
;  1707          1951  1 !
;  1708          1952  1 !       ret_status.wlc.v = ESTABLISH_BORDER_VECTOR(P_PBCB)
;  1709          1953  1 !
;  1710          1954  1 ! FORMAL PARAMETERS:
;  1711          1955  1 !
;  1712          1956  1 !       P_PBCB.rab.r            Address of PBCB
;  1713          1957  1 !
;  1714          1958  1 ! IMPLICIT INPUTS:
;  1715          1959  1 !
;  1716          1960  1 !       contents of PBCB
;  1717          1961  1 !
;  1718          1962  1 ! IMPLICIT OUTPUTS:
;  1719          1963  1 !
;  1720          1964  1 !       R_BORDER_VECTOR field in PBCB gets set up
;  1721          1965  1 !       V_COMPLEX_BORDER is set to 1 if some border element
;  1722          1966  1 !                       is longer than a byte
;  1723          1967  1 !
;  1724          1968  1 ! COMPLETION STATUS:
;  1725          1969  1 !
;  1726          1970  1 !       NONE
;  1727          1971  1 !
;  1728          1972  1 ! SIDE EFFECTS:
;  1729          1973  1 !
;  1730          1974  1 !       NONE
;  1731          1975  1 !--
```

```
;  1733          1976  2 BEGIN
;  1734          1977  2
;  1735          1978  2 BIND    PBCB    = .P_PBCB       : $PBCB_DECL;
;  1736          1979  2
;  1737          1980  2 LOCAL   STATUS;
;  1738          1981  2
;  1739          1982  2 !+
;  1740          1983  2 ! Macro to set code longword in border_vector if
;  1741          1984  2 ! corresponding capability is defined.
;  1742          1985  2 ! We use the default character if the capability doesn't exisit
;  1743          1986  2 ! or if it does exist but the terminal is a non-AVO ANSI CRT.
;  1744          1987  2 !-
;  1745          1988  2
;  1746          1989  2 MACRO
;  1747          1990  2
;  1748        M 1991  2     $VECTOR_SET(CODE,CAP,DEFAULT) =
;  1749        M 1992  2
;  1750        M 1993  2         BEGIN
;  1751        M 1994  2
;  1752        M 1995  2         BIND TT2 = PBCB[PBCB_L_DEVDEPEND2]      : $BBLOCK;
;  1753        M 1996  2
;  1754        M 1997  2         BIND VECT =  PBCB[PBCB_R_BORDER_VECTOR] : VECTOR[16];
;  1755        M 1998  2
;  1756        M 1999  2         $SMG$GET_TERM_DATA(CAP);
;  1757        M 2000  2
;  1758        M 2001  2         IF .PBCB[PBCB_L_CAP_LENGTH] EQL 0
;  1759        M 2002  2         OR (.TT2[TT2$V_ANSICRT] AND NOT .TT2[TT2$V_AVO])
;  1760        M 2003  2           THEN  BEGIN   ! Use default character
;  1761        M 2004  2                 VECT[CODE]=DEFAULT
;  1762        M 2005  2                 END     ! Use default character
;  1763        M 2006  2           ELSE  BEGIN   ! Use specified string
;  1764        M 2007  2                 IF .PBCB[PBCB_L_CAP_LENGTH] GTR 1
;  1765        M 2008  2                   THEN  BEGIN   ! It's a long string
;  1766        M 2009  2
;  1767        M 2010  2                 LOCAL   SIZE;
;  1768        M 2011  2
;  1769        M 2012  2                 !+
;  1770        M 2013  2                 ! Build a byte-counted string for this string.
;  1771        M 2014  2                 !-
;  1772        M 2015  2
;  1773        M 2016  2                 SIZE=.PBCB[PBCB_L_CAP_LENGTH]+1;
;  1774        M 2017  2
;  1775        M 2018  2                 !+
;  1776        M 2019  2                 ! Allocate virtual memory for the capability.
;  1777        M 2020  2                 ! Store it as a counted string.
;  1778        M 2021  2                 !-
;  1779        M 2022  2
;  1780        M 2023  2                 STATUS=LIB$GET_VM(SIZE,VECT[CODE]);
;  1781        M 2024  2                 IF NOT .STATUS THEN RETURN .STATUS;
;  1782        M 2025  2
;  1783        M 2026  2                 !+
;  1784        M 2027  2                 ! Copy the capability in.
;  1785        M 2028  2                 !-
;  1786        M 2029  2
;  1787        M 2030  2                 CH$MOVE(.PBCB[PBCB_L_CAP_LENGTH],
;  1788        M 2031  2                         .PBCB[PBCB_A_CAP_BUFFER],
;  1789        M 2032  2                         .VECT[CODE]+T);
```

```
; 1790          M 2033  2
; 1791          M 2034  2                           !+
; 1792          M 2035  2                           ! Set the byte count.
; 1793          M 2036  2                           !-
; 1794          M 2037  2
; 1795          M 2038  2                                   BEGIN
; 1796          M 2039  2                                   BIND COUNT = .VECT[CODE] : BYTE;
; 1797          M 2040  2                                   COUNT=.PBCB[PBCB_L_CAP_LENGTH]
; 1798          M 2041  2                                   END;
; 1799          M 2042  2
; 1800          M 2043  2                           PBCB[PBCB_V_COMPLEX_BORDER]=1
; 1801          M 2044  2
; 1802          M 2045  2                           END     ! It's a long string
; 1803          M 2046  2                     ELSE  BEGIN   ! It's a single character
; 1804          M 2047  2                           BIND CHAR = .PBCB[PBCB_A_CAP_BUFFER] : BYTE;
; 1805          M 2048  2                           VECT[CODE]=.CHAR
; 1806          M 2049  2                           END     ! It's a single character
; 1807          M 2050  2                   END;    ! Use specified string
; 1808          M 2051  2
; 1809          M 2052  2           END
; 1810          M 2053  2
; 1811            2054  2      %;
```

```
;1813     2055  2 !+
;1814     2056  2 ! The border vector is a 16-longword vector.
;1815     2057  2 ! The nth longword represents the character used to represent
;1816     2058  2 ! the nth border element.  It is the character itself (if <256,
;1817     2059  2 ! or the address of a (byte) counted string for the capability.
;1818     2060  2 ! These elements are described below:
;1819     2061  2 !
;1820     2062  2 ! code  description             default
;1821     2063  2 !
;1822     2064  2 ! 0     unused                  space
;1823     2065  2 ! 1     right                   -
;1824     2066  2 ! 2     up                      |
;1825     2067  2 ! 3     lower left corner       +
;1826     2068  2 ! 4     left                    -
;1827     2069  2 ! 5     horizontal              -
;1828     2070  2 ! 6     lower right corner      +
;1829     2071  2 ! 7     tup                     +
;1830     2072  2 ! 8     down                    |
;1831     2073  2 ! 9     upper left corner       +
;1832     2074  2 ! 10    vertical                |
;1833     2075  2 ! 11    tright                  +
;1834     2076  2 ! 12    upper right corner      +
;1835     2077  2 ! 13    tdown                   +
;1836     2078  2 ! 14    tleft                   +
;1837     2079  2 ! 15    cross                   +
;1838     2080  2 !-
;1839     2081  2
;1840     2082  2 !+
;1841     2083  2 ! Note: "tright" means a T with the stem pointing to the right.
;1842     2084  2 !       (This is called a left T on the VT100 manual.)
;1843     2085  2 !-
;1844     2086  2
;1845     2087  2 !+
;1846     2088  2 ! Note how the codes "or" together.
;1847     2089  2 !-
;1848     2090  2
;1849     2091  2 BIND
;1850     2092  2
;1851     2093  2         MAP_SEQUENCE    = UPLIT BYTE(' -|+--++|+|+++++') : VECTOR[16,BYTE];
;1852     2094  2
;1853     2095  2 !+
;1854     2096  2 ! Now replace each element, one at a time, with the appropriate special
;1855     2097  2 ! character, if one was specified in the TERMTABLE file.
;1856     2098  2 ! Otherwise, store in the default character.
;1857     2099  2 !-
;1858     2100  2
;1859     2101  2 $VECTOR_SET( 1,HORIZONTAL_BAR,%C'-');
;1860     2102  2 $VECTOR_SET( 2,VERTICAL_BAR,%C'|');
;1861     2103  2 $VECTOR_SET( 3,LOWER_LEFT_CORNER,%C'+');
;1862     2104  2 $VECTOR_SET( 4,HORIZONTAL_BAR,%C'-');
;1863     2105  2 $VECTOR_SET( 5,HORIZONTAL_BAR,%C'-');
;1864     2106  2 $VECTOR_SET( 6,LOWER_RIGHT_CORNER,%C'+');
;1865     2107  2 $VECTOR_SET( 7,BOTTOM_T_CHAR,%C'+');
;1866     2108  2 $VECTOR_SET( 8,VERTICAL_BAR,%C'|');
;1867     2109  2 $VECTOR_SET( 9,UPPER_LEFT_CORNER,%C'+');
;1868     2110  2 $VECTOR_SET(10,VERTICAL_BAR,%C'|');
;1869     2111  2 $VECTOR_SET(11,LEFT_T_CHAR,%C'+');
```

```
;  1870          2112  2 $VECTOR_SET(12,UPPER_RIGHT_CORNER,%C'+');
;  1871          2113  2 $VECTOR_SET(13,TOP_T_CHAR,%C'+');
;  1872          2114  2 $VECTOR_SET(14,RIGHT_T_CHAR,%C'+');
;  1873          2115  2 $VECTOR_SET(15,CROSS_CHAR,%C'+');
;  1874          2116  2
;  1875          2117  1 END;


2B  2B  2B  2B  7C  2B  7C  2B  2B  2D  2D  2B  7C  2D  20  00AEA P.AAD:  .ASCII  \ -|+--++'+|+++++\                          ;
                                                          2B  00AF9                                                          ;
                                                                                                                             ;
                                                                HARD_SEQUENCE=      P.AAD                                  ;


                                                          0FFC 00000 ESTABLISH_BORDER_VECTOR:                                ;
                                                                     .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11           ; 1941
                                                 5E       B0   AE  9E 00002          MOVAB   -80(SP), SP                    ;
                                                 56       04   AC  D0 00006          MOVL    P_PBCB, R6                     ; 1978
                                                 5B       60   A6  9E 0000A          MOVAB   96(R6), R11                    ; 2101
                                                 57     010C   C6  9E 0000E          MOVAB   268(R6), R7                    ;
                                                 5A     00FC   C6  9E 00013          MOVAB   252(R6), R10                   ;
                                                                6A  D5 00018          TSTL    (R10)                          ;
                                                                09  12 0001A          BNEQ    1$                             ;
                                                 58     0108   C6  9E 0001C          MOVAB   264(R6), R8                    ;
                                                                68  D4 00021          CLRL    (R8)                           ;
                                                                2A  11 00023          BRB     2$                             ;
                                                          44    AE  D4 00025 1$:      CLRL    INPUT_ARGS                     ;
                                                          44    AE  9F 00028          PUSHAB  INPUT_ARGS                     ;
                                                        0104    C6  DD 0002B          PUSHL   260(R6)                        ;
                                                 58     0108   C6  9E 0002F          MOVAB   264(R6), R8                    ;
                                                          58    DD 00034          PUSHL   R8                             ;
                                                        0100    C6  9F 00036          PUSHAB  256(R6)                        ;
                                                    14  AE      01DD  8F  3C 0003A    MOVZWL  #477, 20(SP)                   ;
                                                          14    AE  9F 00040          PUSHAB  20(SP)                         ;
                                                          5A    DD 00043          PUSHL   R10                            ;
                                          00000000G       00        FB 00045          CALLS   #6, SMG$GET_TERM_DATA          ;
                                                          76        E9 0004C          BLBC    STATUS, 8$                     ;
                                                                68  D5 0004F 2$:      TSTL    (R8)                           ;
                                                                08  13 00051          BEQL    3$                             ;
                                                    0A      03  AB  E9 00053          BLBC    3(R11), 4$                     ;
                                              06    6B          1B  E0 00057          BBS     #27, (R11), 4$                 ;
                                                    04  A7      2D  D0 0005B 3$:      MOVL    #45, 4(R7)                     ;
                                                                3A  11 0005F          BRB     6$                             ;
                                                    01          68  D1 00061 4$:      CMPL    (R8), #1                       ;
                                                                2F  15 00064          BLEQ    5$                             ;
                                        08  AE      68          01  C1 00066          ADDL3   #1, (R8), SIZE                 ;
                                                          04    A7  9F 0006B          PUSHAB  4(R7)                          ;
                                                          0C    AE  9F 0006E          PUSHAB  SIZE                           ;
                                          00000000G       00        FB 00071          CALLS   #2, LIB$GET_VM                 ;
                                                    04    AE        50  D0 00078          MOVL    R0, STATUS             ;
                                                          76        E9 0007C          BLBC    STATUS, 12$                    ;
                                                          59    04  A7  D0 00080          MOVL    4(R7), R9              ;
                                        01  A9      0104    D6        68  28 00084          MOVC3   (R8), @260(R6), 1(R9)  ;
                                                          69        68  90 0008B          MOVB    (R8), (R9)             ;
                                                    00D1  C6        02  88 0008E          BISB2   #2, 209(R6)            ;
                                                                06  11 00093          BRB     6$                             ;
```

```
                                04   A7      0104    D6  9A 00095 5$:     MOVZBL  @260(R6), 4(R7)                                  2102
                                                     6A  D5 00098 6$:     TSTL    (R10)
                                                     04  12 0009D        BNEQ    7$
                                                     68  D4 0009F        CLRL    (R8)
                                                     25  11 000A1        BRB     9$
                                          44    AE    D4 000A3 7$:     CLRL    INPUT_ARGS
                                          44    AE    9F 000A6         PUSHAB  INPUT_ARGS
                                        0104    C6    DD 000A9         PUSHL   260(R6)
                                                     58  DD 000AD        PUSHL   R8
                                        0100    C6    9F 000AF         PUSHAB  256(R6)
                                10   AE      0244    8F  3C 000B3     MOVZWL  #580, 16(SP)
                                          10    AE    9F 000B9         PUSHAB  16(SP)
                                                     5A  DD 000BC     PUSHL   R10
                        00000000G   00              06  FB 000BE     CALLS   #6, SMG$GET_TERM_DATA
                                    77              50  E9 000C5 8$:     BLBC    STATUS, 16$
                                                     68  D5 000C8 9$:     TSTL    (R8)
                                                     08  13 000CA        BEQL    10$
                                    0B      03      AB  E9 000CC         BLBC    3(R11), 11$
                            07      68              1B  E0 000D0         BBS     #27, (R11), 11$
                                08  A7      7C      8F  9A 000D4 10$.    MOVZBL  #124, 8(R7)
                                                     3A  11 000D9        BRB     14$
                                    01              68  D1 000DB 11$:     CMPL    (R8), #1
                                                     2F  15 000DE        BLEQ    13$
                        0C  AE              68      01  C1 000E0         ADDL3   #1, (R8), SIZE
                                          08    A7    9F 000E5         PUSHAB  8(R7)
                                          10    AE    9F 000E8         PUSHAB  SIZE
                        00000000G   00              02  FB 000EB     CALLS   #2, LIB$GET_VM
                                04  AE              50  D0 000F2     MOVL    R0, STATUS
                                    75      04  AE  E9 000F6 12$:     BLBC    STATUS, 20$
                                    59      08  A7  D0 000FA         MOVL    8(R7), R9
                        01  A9      0104    D6      28 000FE         MOVC3   (R8), @260(R6), 1(R9)
                                    69              90 00105         MOVB    (R8), (R9)
                            00D1    C6              88 00108         BISB2   #2, 209(R6)
                                                     11 0010D        BRB     14$
                                08  A7      0104    D6  9A 0010F 13$:     MOVZBL  @260(R6), 8(R7)                                  2103
                                                     6A  D5 00115 14$:     TSTL    (R10)
                                                     04  12 00117        BNEQ    15$
                                                     68  D4 00119        CLRL    (R8)
                                                     25  11 0011B        BRB     17$
                                          44    AE    D4 0011D 15$:     CLRL    INPUT_ARGS
                                          44    AE    9F 00120         PUSHAB  INPUT_ARGS
                                        0104    C6    DD 00123         PUSHL   260(R6)
                                                     58  DD 00127        PUSHL   R8
                                        0100    C6    9F 00129         PUSHAB  256(R6)
                                10   AE      0229    8F  3C 0012D     MOVZWL  #553, 16(SP)
                                          10    AE    9F 00133         PUSHAB  16(SP)
                                                     5A  DD 00136     PUSHL   R10
                        000000000G  00              06  FB 00138     CALLS   #6, SMG$GET_TERM_DATA
                                    76              50  E9 0013F 16$:     BLBC    STATUS, 24$
                                                     68  D5 00142 17$:     TSTL    (R8)
                                                     08  13 00144        BEQL    18$
                                    0A      03      AB  E9 00146         BLBC    3(R11), 19$
                        06          68              1B  E0 0014A         BBS     #27, (R11), 19$
                                0C  A7              2B  D0 0014E 18$:     MOVL    #43, 12(R7)
                                                     3A  11 00152        BRB     22$
                                    01              68  D1 00154 19$:     CMPL    (R8), #1
                                                     2F  15 00157        BLEQ    21$
```

```
                                                                M 7
                10    AE              68                01  C1 00159          ADDL3   #1, (R8), SIZE
                                         0C             A7  9F 0015E          PUSHAB  12(R7)
                                         14             AE  9F 00161          PUSHAB  SIZE
                          00000000G      00                 FB 00164          CALLS   #2, LIB$GET_VM
                                  04     AE                 D0 0016B          MOVL    R0, STATUS
                                         75             04  AE  E9 0016F 20$:  BLBC    STATUS, 28$
                                         59             0C  A7  D0 00173          MOVL    12(R7), R9
                01    A9          0104   D6                 68  28 00177          MOVC3   (R8), @260(R6), 1(R9)
                                         69                 68  90 0017E          MOVB    (R8), (R9)
                                  00D1   C6                 02  88 00181          BISB2   #2, 209(R6)
                                                            06  11 00186          BRB     22$
                                    0C   A7          0104   D6  9A 00188 21$:     MOVZBL  @260(R6), 12(R7)
                                                            6A  D5 0018E 22$:     TSTL    (R10)                                   2104
                                                            04  12 00190          BNEQ    23$
                                                            68  D4 00192          CLRL    (R8)
                                                            25  11 00194          BRB     25$
                                                        44  AE  D4 00196 23$:     CLRL    INPUT_ARGS
                                                        44  AE  9F 00199          PUSHAB  INPUT_ARGS
                                                      0104  C6  DD 0019C          PUSHL   260(R6)
                                                            58  DD 001A0          PUSHL   R8
                                                      0100  C6  9F 001A2          PUSHAB  256(R6)
                                    10   AE          01DD   8F  3C 001A6          MOVZWL  #477, 16(SP)
                                                        10  AE  9F 001AC          PUSHAB  16(SP)
                                                            5A  DD 001AF          PUSHL   R10
                          00000000G      00                 FB 001B2          CALLS   #6, SMG$GET_TERM_DATA
                                         76                 50  E9 001B8 24$:     BLBC    STATUS, 32$
                                                            68  D5 001BB 25$:     TSTL    (R8)
                                                            08  13 001BD          BEQL    26$
                                         0A             03  AB  E9 001BF          BLBC    3(R11), 27$
                      06                 6B                 1B  E0 001C3          BBS     #27, (R11), 27$
                                    10   A7                 2D  D0 001C7 26$:     MOVL    #45, 16(R7)
                                                            3A  11 001CB          BRB     30$
                                         01                 68  D1 001CD 27$:     CMPL    (R8), #1
                                                            2F  15 001D0          BLEQ    29$
                14    AE                68                 01  C1 001D2          ADDL3   #1, (R8), SIZE
                                                        10  A7  9F 001D7          PUSHAB  16(R7)
                                                        18  AE  9F 001DA          PUSHAB  SIZE
                          00000000G      00                 FB 001DD          CALLS   #2, LIB$GET_VM
                                  04     AE                 D0 001E4          MOVL    R0, STATUS
                                         75             04  AE  E9 001E8 28$:     BLBC    STATUS, 36$
                                         59             10  A7  D0 001EC          MOVL    16(R7), R9
                01    A9          0104   D6                 68  28 001F0          MOVC3   (R8), @260(R6), 1(R9)
                                         69                 68  90 001F7          MOVB    (R8), (R9)
                                  00D1   C6                 02  88 001FA          BISB2   #2, 209(R6)
                                                            06  11 001FF          BRB     30$
                                    10   A7          0104   D6  9A 00201 29$:     MOVZBL  @260(R6), 16(R7)
                                                            6A  D5 00207 30$:     TSTL    (R10)                                   2105
                                                            04  12 00209          BNEQ    31$
                                                            68  D4 0020B          CLRL    (R8)
                                                            25  11 0020D          BRB     33$
                                                        44  AE  D4 0020F 31$:     CLRL    INPUT_ARGS
                                                        44  AE  9F 00212          PUSHAB  INPUT_ARGS
                                                      0104  C6  DD 00215          PUSHL   260(R6)
                                                            58  DD 00219          PUSHL   R8
                                                      0100  C6  9F 0021B          PUSHAB  256(R6)
                                    10   AE          01DD   8F  3C 0021F          MOVZWL  #477, 16(SP)
                                                        10  AE  9F 00225          PUSHAB  16(SP)
```

```
                                                 5A  DD 00228          PUSHL   R10
                                                 06  FB 0022A          CALLS   #6, SMG$GET_TERM_DATA
                    00000000G  00                50  E9 00231 32$:     BLBC    STATUS, 40$
                               76                68  D5 00234 33$:     TSTL    (R8)
                                                 08  13 00236          BEQL    34$
                               0A          03    AB  E9 00238          BLBC    3(R11), 35$
                         06    6B                1B  E0 0023C          BBS     #27, (R11), 35$
                            14 A7                2D  D0 00240 34$:     MOVL    #45, 20(R7)
                                                 3A  11 00244          BRB     38$
                               01                68  D1 00246 35$:     CMPL    (R8), #1
                                                 2F  15 00249          BLEQ    37$
                   18    AE    68                01  C1 0024B          ADDL3   #1, (R8), SIZE
                                           14    A7  9F 00250          PUSHAB  20(R7)
                                           1C    AE  9F 00253          PUSHAB  SIZE
                    00000000G  00                02  FB 00256          CALLS   #2, LIB$GET_VM
                            04 AE                50  D0 0025D          MOVL    R0, STATUS
                               75          04    AE  E9 00261 36$:     BLBC    STATUS, 44$
                               59          14    A7  D0 00265          MOVL    20(R7), R9
                01    A9 0104 D6                68  28 00269          MOVC3   (R8), @260(R6), 1(R9)
                               69                68  90 00270          MOVB    (R8), (R9)
                         00D1  C6                02  88 00273          BISB2   #2, 209(R6)
                                                 06  11 00278          BRB     38$
                            14 A7          0104  D6  9A 0027A 37$:     MOVZBL  @260(R6), 20(R7)
                                                 6A  D5 00280 38$:     TSTL    (R10)                   2106
                                                 04  12 00282          BNEQ    39$
                                                 68  D4 00284          CLRL    (R8)
                                                 25  11 00286          BRB     41$
                                           44    AE  D4 00288 39$:     CLRL    INPUT_ARGS
                                           44    AE  9F 0028B          PUSHAB  INPUT_ARGS
                                         0104    C6  DD 0028E          PUSHL   260(R8)
                                                 58  DD 00292          PUSHL   R8
                                         0100    C6  9F 00294          PUSHAB  256(R6)
                               10    AE    022A  8F  3C 00298          MOVZWL  #554, 16(SP)
                                           10    AE  9F 0029E          PUSHAB  16(SP)
                                                 5A  DD 002A1          PUSHL   R10
                    00000000G  00                06  FB 002A3          CALLS   #6, SMG$GET_TERM_DATA
                               76                50  E9 002AA 40$:     BLBC    STATUS, 48$
                                                 68  D5 002AD 41$:     TSTL    (R8)
                                                 08  13 002AF          BEQL    42$
                               0A          03    AB  E9 002B1          BLBC    3(R11), 43$
                         06    6B                1B  E0 002B5          BBS     #27, (R11), 43$
                            18 A7                2B  D0 002B9 42$:     MOVL    #43, 24(R7)
                                                 3A  11 002BD          BRB     46$
                               01                68  D1 002BF 43$:     CMPL    (R8), #1
                                                 2F  15 002C2          BLEQ    45$
                   1C    AE    68                01  C1 002C4          ADDL3   #1, (R8), SIZE
                                           18    A7  9F 002C9          PUSHAB  24(R7)
                                           20    AE  9F 002CC          PUSHAB  SIZE
                    00000000G  00                02  FB 002CF          CALLS   #2, LIB$GET_VM
                            04 AE                50  D0 002D6          MOVL    R0, STATUS
                               75          04    AE  E9 002DA 44$:     BLBC    STATUS, 52$
                               59          18    A7  D0 002DE          MOVL    24(R7), R9
                01    A9 0104 D6                68  28 002E2          MOVC3   (R8), @260(R6), 1(R9)
                               69                68  90 002E9          MOVB    (R8), (R9)
                         00D1  C6                02  88 002EC          BISB2   #2, 209(R6)
                                                 06  11 002F1          BRB     46$
                            18 A7          0104  D6  9A 002F3 45$:     MOVZBL  @260(R6), 24(R7)
```

```
                                                 6A  D5 002F9 46$:    TSTL    (R10)                                   ;2107
                                                 04  12 002FB         BNEQ    47$
                                                 68  D4 002FD         CLRL    (R8)
                                                 25  11 002FF         BRB     49$
                                           44    AE  D4 00301 47$:    CLRL    INPUT_ARGS
                                           44    AE  9F 00304         PUSHAB  INPUT_ARGS
                                         0104    C6  DD 00307         PUSHL   260(R6)
                                                 58  DD 00308         PUSHL   R8
                                         0100    C6  9F 00300         PUSHAB  256(R6)
                                   10    AE    01C1  BF  3C 00311         MOVZWL  #449, 16(SP)
                                           10    AE  9F 00317         PUSHAB  16(SP)
                                                 5A  DD 0031A         PUSHL   R10
                              00000000G    00       06  FB 0031C         CALLS   #6, SMGSGET_TERM_DATA
                                           76       50  E9 00323 48$:    BLBC    STATUS, 56$
                                                    68  D5 00326 49$:    TSTL    (R8)
                                                    08  13 00328         BEQL    50$
                                           0A    03    A8  E9 0032A         BLBC    3(R11), 51$
                                   06      68          1B  E0 0032E         BBS     #27, (R11), 51$
                                   1C    A7            2B  D0 00332 50$:    MOVL    #43, 28(R7)
                                                       3A  11 00336         BRB     54$
                                           01          68  D1 00338 51$:    CMPL    (R8), #1
                                                       2F  15 00338         BLEQ    53$
                                 20    AE              01  C1 0033D         ADDL3   #1, (R8), SIZE
                                           1C    A7    24  AE  9F 00342         PUSHAB  28(R7)
                                           24    AE          9F 00345         PUSHAB  SIZE
                              00000000G    00               02  FB 00348         CALLS   #2, LIBSGET_VM
                                     04    AE              50  D0 0034F         MOVL    R0, STATUS
                                           76       04    AE  E9 00353 52$:    BLBC    STATUS, 60$
                                           59       1C    A7  DC 00357         MOVL    28(R7), R9
                                 01    A9      0104    D6      28 00358         MOVC3   (R8), @260(R6), 1(R9)
                                           69            68  90 00362         MOVB    (R8), (R9)
                                         00D1    C6          02  88 00365         BISB2   #2, 209(R6)
                                                             06  11 0036A         BRB     54$
                                           1C    A7    0104  D6  9A 0036C 53$:    MOVZBL  @260(R6), 28(R7)
                                                             6A  D5 00372 54$:    TSTL    (R10)                           ;2108
                                                             04  12 00374         BNEQ    55$
                                                             68  D4 00376         CLRL    (R8)
                                                             25  11 00378         BRB     57$
                                                       44    AE  D4 0037A 55$:    CLRL    INPUT_ARGS
                                                       44    AE  9F 0037D         PUSHAB  INPUT_ARGS
                                                     0104    C6  DD 00380         PUSHL   260(R6)
                                                             58  DD 00384         PUSHL   R8
                                                     0100    C6  9F 00386         PUSHAB  256(R6)
                                               10    AE    0244  8F  3C 0038A         MOVZWL  #580, 16(SP)
                                                       10    AE  9F 00390         PUSHAB  16(SP)
                                                             5A  DD 00393         PUSHL   R10
                                          00000000G    00       06  FB 00395         CALLS   #6, SMGSGET_TERM_DATA
                                                       77       50  E9 0039C 56$     BLBC    STATUS, 64$
                                                                68  D5 0039F 57$     TSTL    (R8)
                                                                08  13 003A1         BEQL    58$
                                                       08    03    AB  E9 003A3         BLBC    3(R11), 59$
                                             07        68          1B  E0 003A7         BBS     #27, (R11), 59$
                                               20    A7            7C  8F  9A 003AB 58$:    MOVZBL  #124, 32(R7)
                                                                   3A  11 003B0         BRB     62$
                                                       01          68  D1 003B2 59$     CMPL    (R8), #1
                                                                   2F  15 003B5         BLEQ    61$
                                     24    AE              68          01  C1 003B7         ADDL3   #1, (R8), SIZE
```

```
                                  20   A7  9F 003BC        PUSHAB  32(R7)
                                  28   AE  9F 003BF        PUSHAB  SIZE
                00000000G  00          02  FB 003C2        CALLS   #2, LIB$GET_VM
                       04  AE          50  D0 003C9        MOVL    R0, STATUS
                           75     04   AE  E9 003CD 60$:   BLBC    STATUS, 68$
                           59     20   A7  D0 003D1        MOVL    32(R7), R9
      01   A9       0104   D6          68  28 003D5        MOVC3   (R8), @260(R6), 1(R9)
                           69          68  90 003DC        MOVB    (R8), (R9)
                    00D1   C6          02  88 003DF        BISB2   #2, 209(R6)
                                       06  11 003E4        BRB     62$
                 20   A7        0104   D6  9A 003E6 61$:   MOVZBL  @260(R6), 32(R7)
                                       6A  D5 003EC 62$:   TSTL    (R10)                              2109
                                       04  12 003EE        BNEQ    63$
                                       68  D4 003F0        CLRL    (R8)
                                       25  11 003F2        BRB     65$
                                  44   AE  D4 003F4 63$:   CLRL    INPUT_ARGS
                                  44   AE  9F 003F7        PUSHAB  INPUT_ARGS
                                0104   C6  DD 003FA        PUSHL   260(R8)
                                       58  DD 003FE        PUSHL   R8
                                0100   C6  9F 00400        PUSHAB  256(R6)
                 10   AE        0242   8F  3C 00404        MOVZWL  #578, 16(SP)
                                  10   AE  9F 0040A        PUSHAB  16(SP)
                                       5A  DD 0040D        PUSHL   R10
                00000000G  00          06  FB 0040F        CALLS   #6, SMG$GET_TERM_DATA
                           76          50  E9 00416 64$:   BLBC    STATUS, 72$
                                       68  D5 00419 65$:   TSTL    (R8)
                                       08  13 0041B        BEQL    66$
                           0A     03   AB  E9 0041D        BLBC    3(R11), 67$
           06              6B          18  E0 00421        BBS     #27, (R11), 67$
                 24        A7          2B  D0 00425 66$:   MOVL    #43, 36(R7)
                                       3A  11 00429        BRB     70$
                           01          68  D1 0042B 67$:   CMPL    (R8), #1
                                       2F  15 0042E        BLEQ    69$
      28   AE              68          01  C1 00430        ADDL3   #1, (R8), SIZE
                                  24   A7  9F 00435        PUSHAB  36(R7)
                                  2C   AE  9F 00438        PUSHAB  SIZE
                00000000G  00          02  FB 0043B        CALLS   #2, LIB$GET_VM
                       04  AE          50  D0 00442        MOVL    R0, STATUS
                           76     04   AE  E9 00446 68$:   BLBC    STATUS, 76$
                           59     24   A7  D0 0044A        MOVL    36(R7), R9
      01   A9       0104   D6          68  28 0044E        MOVC3   (R8), @260(R6), 1(R9)
                           69          68  90 00455        MOVB    (R8), (R9)
                    00D1   C6          02  88 00458        BISB2   #2, 209(R6)
                                       06  11 0045D        BRB     70$
                 24   A7        0104   D6  9A 0045F 69$:   MOVZBL  @260(R6), 36(R7)
                                       6A  D5 00465 70$:   TSTL    (R10)                              2110
                                       04  12 00467        BNEQ    71$
                                       68  D4 00469        CLRL    (R8)
                                       25  11 0046B        BRB     73$
                                  44   AE  D4 0046D 71$:   CLRL    INPUT_ARGS
                                  44   AE  9F 00470        PUSHAB  INPUT_ARGS
                                0104   C6  DD 00473        PUSHL   260(R8)
                                       58  DD 00477        PUSHL   R8
                                0100   C6  9F 00479        PUSHAB  256(R6)
                 10   AE        0244   8F  3C 0047D        MOVZWL  #580, 16(SP)
                                  10   AE  9F 00483        PUSHAB  16(SP)
                                       5A  DD 00486        PUSHL   R10
```

```
                 00000000G  00         06  FB 00488       CALLS    #6, SMG$GET_TERM_DATA       :
                            77         50  E9 0048F 72$:  BLBC     STATUS, 80$                 :
                                       68  D5 00492 73$:  TSTL     (R8)                        :
                                       08  13 00494       BEQL     74$                         :
                            0B     03  AB  E9 00496       BLBC     3(R11), 75$                 :
             07             6B         1B  E0 0049A       BBS      #27, (R11), 75$             :
                     28     A7     7C  8F  9A 0049E 74$:  MOVZBL   #124, 40(R7)                :
                                       3A  11 004A3       BRB      78$                         :
                            01         68  D1 004A5 75$:  CMPL     (R8), #1                    :
                                       2F  15 004A8       BLEQ     77$                         :
             2C      AE     68         01  C1 004AA       ADDL3    #1, (R8),   ZE              :
                                   28  A7  9F 004AF       PUSHAB   40(R7)                      :
                                   30  AE  9F 004B2       PUSHAB   SIZE                        :
                 00000000G  00         02  FB 004B5       CALLS    #2, LIB$GET_VM              :
                        04  AE         50  D0 004BC       MOVL     R0, STATUS                  :
                            75     04  AE  E9 004C0 76$:  BLBC     STATUS, 84$                 :
                            59     28  A7  D0 004C4       MOVL     40(R7), R9                  :
             01      A9   0104  D6     68  28 004C8       MOVC3    (R8), @260(R6), 1(R9)       :
                            69         68  90 004CF       MOVB     (R8), (R9)                  :
                      00D1  C6         02  88 004D2       BISB2    #2, 209(R6)                 :
                                       06  11 004D7       BRB      78$                         :
                     28     A7   0104  D6  9A 004D9 77$:  MOVZBL   @260(R6), 40(R7)            :
                                       6A  D5 004DF 78$:  TSTL     (R10)                       : 2111
                                       04  12 004E1       BNEQ     79$                         :
                                       68  D4 004E3       CLRL     (R8)                        :
                                       25  11 004E5       BRB      81$                         :
                                   44  AE  D4 004E7 79$:  CLRL     INPUT_ARGS                  :
                                   44  AE  9F 004EA       PUSHAB   INPUT_ARGS                  :
                                 0104  C6  DD 004ED       PUSHL    260(R6)                     :
                                       58  DD 004F1       PUSHL    R8                          :
                                 0100  C6  9F 004F3       PUSHAB   256(R6)                     :
                     10     AE   0227  8F  3C 004F7       MOVZWL   #551, 16(SP)                :
                                   10  AE  9F 004FD       PUSHAB   16(SP)                      :
                                       5A  DD 00500       PUSHL    R10                         :
                 00000000G  00         06  FB 00502       CALLS    #6, SMG$GET_TERM_DATA       :
                            76         50  E9 00509 80$:  BLBC     STATUS, 88$                 :
                                       68  D5 0050C 81$:  TSTL     (R8)                        :
                                       08  13 0050E       BEQL     82$                         :
                            0A     03  AB  E9 00510       BLBC     3(R11), 83$                 :
             06             6B         1B  E0 00514       BBS      #27, (R11), 83$             :
                     2C     A7         2B  D0 00518 82$:  MOVL     #43, 44(R7)                 :
                                       3A  11 0051C       BRB      86$                         :
                            01         68  D1 0051E 83$:  CMPL     (R8), #1                    :
                                       2F  15 00521       BLEQ     85$                         :
             30      AE     68         01  C1 00523       ADDL3    #1, (R8), SIZE              :
                                   2C  A7  9F 00528       PUSHAB   44(R7)                      :
                                   34  AE  9F 0052B       PUSHAB   SIZE                        :
                 00000000G  00         02  FB 0052E       CALLS    #2, LIB$GET_VM              :
                        04  AE         50  D0 00535       MOVL     R0, STATUS                  :
                            75     04  AE  E9 00539 84$:  BLBC     STATUS, 92$                 :
                            59     2C  A7  D0 0053D       MOVL     44(R7), R9                  :
             01      A9   0104  D6     68  28 00541       MOVC3    (R8), @260(R6), 1(R9)       :
                            69         68  90 00548       MOVB     (R8), (R9)                  :
                      00D1  C6         02  88 0054B       BISB2    #2, 209(R6)                 :
                                       06  11 00550       BRB      86$                         :
                     2C     A7   0104  D6  9A 00552 85$:  MOVZBL   @260(R6), 44(R7)            :
                                       6A  D5 00558 86$:  TSTL     (R10)                       : 2112
```

```
                                                 04  12  0055A          BNEQ    87$
                                                 68  D4  0055C          CLRL    (R8)
                                                 25  11  0055E          BRB     89$
                                             44  AE  D4  00560 87$:     CLRL    INPUT_ARGS
                                             44  AE  9F  00563          PUSHAB  INPUT_ARGS
                                           0104  C6  DD  00566          PUSHL   260(R8)
                                                 58  DD  0056A          PUSHL   R8
                                           0100  C6  9F  0056C          PUSHAB  256(R6)
                                     10    AE  0243  8F  3C  00570      MOVZWL  #579, 16(SP)
                                             10  AE  9F  00576          PUSHAB  16(SP)
                                                 5A  DD  00579          PUSHL   R10
                              00000000G      00      06  FB  0057B      CALLS   #6, SMG$GET_TERM_DATA
                                             76      50  E9  00582 88$:  BLBC    STATUS, 96$
                                                     68  D5  00585 89$:  TSTL    (R8)
                                                     08  13  00587      BEQL    90$
                                         0A      03  AB  E9  00589      BLBC    3(R11), 91$
                                 06      6B          1B  E0  0058D      BBS     #27, (R11), 91$
                                     30  A7          2B  D0  00591 90$:  MOVL    #43, 48(R7)
                                                     3A  11  00595      BRB     94$
                                         01          68  D1  00597 91$:  CMPL    (R8), #1
                                                     2F  15  0059A      BLEQ    93$
                             34  AE          68      01  C1  0059C      ADDL3   #1, (R8), SIZE
                                             30  A7  9F  005A1          PUSHAB  48(R7)
                                             38  AE  9F  005A4          PUSHAB  SIZE
                              00000000G      00      02  FB  005A7      CALLS   #2, LIB$GET_VM
                                     04      AE      50  D0  005AE      MOVL    R0, STATUS
                                             75      04  AE  E9  005B2 92$:  BLBC    STATUS, 100$
                                             59      30  A7  D0  005B6      MOVL    48(R7), R9
                         01  A9      0104    D6          28  005BA      MOVC3   (R8), @260(R6), 1(R9)
                                             69          90  005C1      MOVB    (R8), (R9)
                                     00D1    C6          02  88  005C4      BISB2   #2, 209(R6)
                                                         06  11  005C9      BRB     94$
                                     30  A7      0104    D6  9A  005CB 93$:  MOVZBL  @260(R6), 48(R7)
                                                         6A  D5  005D1 94$:  TSTL    (R10)                    2113
                                                         04  12  005D3      BNEQ    95$
                                                         68  D4  005D5      CLRL    (R8)
                                                         25  11  005D7      BRB     97$
                                                     44  AE  D4  005D9 95$:  CLRL    INPUT_ARGS
                                                     44  AE  9F  005DC      PUSHAB  INPUT_ARGS
                                                   0104  C6  DD  005DF      PUSHL   260(R8)
                                                         58  DD  005E3      PUSHL   R8
                                                   0100  C6  9F  005E5      PUSHAB  256(R6)
                                             10    AE  0240  8F  3C  005E9  MOVZWL  #576, 16(SP)
                                                     10  AE  9F  005EF      PUSHAB  16(SP)
                                                         5A  DD  005F2      PUSHL   R10
                                      000000000G     00      06  FB  005F4  CALLS   #6, SMG$GET_TERM_DATA
                                                     76      50  E9  005FB 96$:  BLBC    STATUS, 104$
                                                             68  D5  005FE 97$:  TSTL    (R8)
                                                             08  13  00600      BEQL    98$
                                                 0A      03  AB  E9  00602      BLBC    3(R11), 99$
                                         06      6B          1B  E0  00606      BBS     #27, (R11), 99$
                                             34  A7          2B  D0  0060A 98$:  MOVL    #43, 52(R7)
                                                             3A  11  0060E      BRB     102$
                                                 01          68  D1  00610 99$:  CMPL    (R8), #1
                                                             2F  15  00613      BLEQ    101$
                                     38  AE          68      01  C1  00615      ADDL3   #1, (R8), SIZE
                                                     34  A7  9F  0061A          PUSHAB  52(R7)
```

```
                                        3C   AE  9F 0061D              PUSHAB  SIZE
                          00000000G     00       02 FB 00620              CALLS   #2, LIB$GET_VM
                                 04     AE       50 D0 00627              MOVL    R0, STATUS
                                        75       04 AE E9 0062B 100$:     BLBC    STATUS, 108$
                                        59       34 A7 D0 0062F           MOVL    52(R7), R9
                01    A9          0104  D6                 68 28 00633     MOVC3   (R8), @260(R6), 1(R9)
                                        69                 68 90 0063A     MOVB    (R8), (R9)
                                  00D1  C6                 02 88 0063D     BISB2   #2, 209(R6)
                                        06                 11 00642        BRB     102$
                                    34  A7       0104      D6 9A 00644 101$:    MOVZBL  @260(R6), 52(R7)
                                        6A                 D5 0064A 102$:    TSTL    (R10)                                    2114
                                        04                 12 0064C          BNEQ    103$
                                        68                 D4 0064E          CLRL    (R8)
                                        25                 11 00650          BRB     105$
                                        44       44        AE D4 00652 103$:    CLRL    INPUT_ARGS
                                        44       44        AE 9F 00655          PUSHAB  INPUT_ARGS
                                      0104       C6        DD 00658             PUSHL   260(R6)
                                        58                 DD 0065C             PUSHL   R8
                                      0100       C6        9F 0065E             PUSHAB  256(R6)
                                    10  AE       022F      8F 3C 00662          MOVZWL  #559, 16(SP)
                                        10       AE        9F 00668             PUSHAB  16(SP)
                                        5A                 DD 0066B             PUSHL   R10
                          00000000G     00                 06 FB 0066D          CALLS   #6, SMG$GET_TERM_DATA
                                        76                 50 E9 00674 104$:    BLBC    STATUS, 112$
                                        68                 D5 00677 105$:    TSTL    (R8)
                                        08                 13 00679          BEQL    106$
                                        0A       03        AB E9 0067B          BLBC    3(R11), 107$
                      06                6B                 1B E0 0067F          BBS     #27, (R11), 107$
                                    38  A7                 2B D0 00683 106$:    MOVL    #43, 56(R7)
                                        3A                 11 00687          BRB     110$
                                        01                 68 D1 00689 107$:    CMPL    (R8), #1
                                        2F                 15 0068C          BLEQ    109$
                3C    AE                68                 01 C1 0068E          ADDL3   #1, (R8), SIZE
                                                 38        A7 9F 00693          PUSHAB  56(R7)
                                                 40        AE 9F 00696          PUSHAB  SIZE
                                00000000G        00                 02 FB 00699          CALLS   #2, LIB$GET_VM
                                       04        AE                 50 D0 006A0          MOVL    R0, STATUS
                                                 74       04        AE E9 006A4 108$:    BLBC    STATUS, 116$
                                                 59       38        A7 D0 006A8          MOVL    56(R7), R9
                01    A9          0104          D6                 68 28 006AC          MOVC3   (R8), @260(R6), 1(R9)
                                                 69                 68 90 006B3          MOVB    (R8), (R9)
                                        00D1     C6                 02 88 006B6          BISB2   #2, 209(R6)
                                                 06                 11 006BB          BRB     110$
                                           38    A7       0104      D6 9A 006BD 109$:    MOVZBL  @260(R6), 56(R7)
                                                 6A                 D5 006C3 110$:    TSTL    (R10)                            211*
                                                 04                 12 006C5          BNEQ    111$
                                                 68                 D4 006C7          CLRL    (R8)
                                                 25                 11 006C9          BRB     113$
                                                 44       44        AE D4 006CB 111$:    CLRL    INPUT_ARGS
                                                 44       44        AE 9F 006CE          PUSHAB  INPUT_ARGS
                                               0104       C6        DD 006D1             PUSHL   260(R6)
                                                 58                 DD 006D5             PUSHL   R8
                                               0100       C6        9F 006D7             PUSHAB  256(R6)
                                             10  AE       01C3      8F 3C 006DB          MOVZWL  #451, 16(SP)
                                                 10       AE        9F 006E1             PUSHAB  16(SP)
                                                 5A                 DD 006E4             PUSHL   R10
                                00000000G        00                 06 FB 006E6          CALLS   #6, SMG$GET_TERM_DATA
```

```
                                    4A              50  E9 006ED 112$:   BLBC    STATUS, 118$
                                                    68  D5 006F0 113$:   TSTL    (R8)
                                                    08  13 006F2         BEQL    114$
                                    09      03      AB  E9 006F4         BLBC    3(R11), 115$
                          05        6B              1B  E0 006F8         BBS     #27, (R11), 115$
                             3C     A7              28  D0 006FC 114$:   MOVL    #43, 60(R7)
                                                    04  00700           RET
                                    01              68  D1 00701 115$:   CMPL    (R8), #1
                                                    2E  15 00704         BLEQ    117$
                    40    AE                68      01  C1 00706         ADDL3   #1, (R8), SIZE
                                                3C  A7  9F 0070B         PUSHAB  60(R7)
                                                44  AE  9F 0070E         PUSHAB  SIZE
                             00000000G          00      02  FB 00711     CALLS   #2, LIB$GET_VM
                                   04           AE      50  D0 00718     MOVL    R0, STATUS
                                                1A  04  AE  E9 0071C 116$:   BLBC    STATUS, 118$
                                                59  3C  A7  D0 00720     MOVL    60(R7), R9
                    01    A9        0104        D6      68  28 00724     MOVC3   (R8), @260(R6), 1(R9)
                                                69      68  90 0072B     MOVB    (R8), (R9)
                                   00D1         C6      02  88 0072E     BISB2   #2, 209(R6)
                                                        04  00733         RET
                                      3C        A7  0104    D6  9A 00734 117$:   MOVZBL  @260(R6), 60(R7)
                                                            04  0073A 118$:   RET                                    2117

; Routine Size:  1851 bytes,    Routine Base:  _SMG$CODE + 0AFA
```

```
;  1877          2118  1 %SBTTL 'SMG$PUT_PASTEBOARD - Output pasteboard via routine'
;  1878          2119  1 GLOBAL ROUTINE SMG$PUT_PASTEBOARD ( PASTEBOARD_ID, P_RTN, P_PRM, P_FF_FLAG ) =
;  1879          2120  1 !++
;  1880          2121  1 ! FUNCTIONAL DESCRIPTION:
;  1881          2122  1 !
;  1882          2123  1 !       This routine is used to get access to the contents of a pasteboard.
;  1883          2124  1 !       The caller specifies an action routine.  The action routine
;  1884          2125  1 !       will then get called once for each line in the pasteboard.
;  1885          2126  1 !       The action routine will be passed a descriptor for that line
;  1886          2127  1 !       followed by a user-specified parameter.
;  1887          2128  1 !
;  1888          2129  1 ! CALLING SEQUENCE:
;  1889          2130  1 !
;  1890          2131  1 !       ret_status.wlc.v = SMG$PUT_PASTEBOARD ( PASTEBOARD_ID.rl.r
;  1891          2132  1 !                                               ,P_RTN
;  1892          2133  1 !                                               [,P_PRM.rl.r]
;  1893          2134  1 !                                               [,P_FF_FLAG.rl.r])
;  1894          2135  1 !       ACTION ROUTINE:
;  1895          2136  1 !
;  1896          2137  1 !       ret_status.wlc.v = RTN(LINE.rt.dx,PRM.rl.v)
;  1897          2138  1 !
;  1898          2139  1 !               A false status return means stop sending lines.
;  1899          2140  1 !
;  1900          2141  1 ! FORMAL PARAMETERS:
;  1901          2142  1 !
;  1902          2143  1 !       PASTEBOARD_ID.rl.r      pasteboard id
;  1903          2144  1 !
;  1904          2145  1 !       P_RTN.rzem.r            Address of routine to be called.
;  1905          2146  1 !
;  1906          2147  1 !       P_PRM.rl.r              User-specified parameter to be passed
;  1907          2148  1 !                               along to the action routine
;  1908          2149  1 !                               If omitted, a 0 will be passed as
;  1909          2150  1 !                               the user parameter.
;  1910          2151  1 !
;  1911          2152  1 !       P_FF_FLAG.rl.r          A flag (0 or 1).  If 1, then the first
;  1912          2153  1 !                               line passed to the action routine
;  1913          2154  1 !                               will be prepended with a formfeed.
;  1914          2155  1 !                               (If the output device is a terminal
;  1915          2156  1 !                               and if the terminal does not have
;  1916          2157  1 !                               the MECHFORM characteristic, then
;  1917          2158  1 !                               a linefeed will be used instead.)
;  1918          2159  1 !                               If not specified, then no form feed
;  1919          2160  1 !                               will be prepended.
;  1920          2161  1 !
;  1921          2162  1 ! IMPLICIT INPUTS:
;  1922          2163  1 !
;  1923          2164  1 !       contents of PBCB
;  1924          2165  1 !
;  1925          2166  1 ! IMPLICIT OUTPUTS:
;  1926          2167  1 !
;  1927          2168  1 !       NONE
;  1928          2169  1 !
;  1929          2170  1 ! COMPLETION STATUS:
;  1930          2171  1 !
;  1931          2172  1 !       SS$_NORMAL      Normal successful completion
;  1932          2173  1 !
;  1933          2174  1 !       other           Error return passed back by an action routine
```

```
; 1934          2175  1 !
; 1935          2176  1 ! SIDE EFFECTS:
; 1936          2177  1 !
; 1937          2178  1 !       NONE
; 1938          2179  1 !--
```

```
;  1940          2180  2 BEGIN
;  1941          2181  2
;  1942          2182  2 BIND
;  1943          2183  2
;  1944          2184  2         PRM     = .P_PRM,
;  1945          2185  2         FF_FLAG = .P_FF_FLAG;
;  1946          2186  2
;  1947          2187  2 LOCAL
;  1948          2188  2
;  1949          2189  2         TEMP_BUF        : VECTOR[512,BYTE],     ! *** TEMP
;  1950          2190  2         STATUS,
;  1951          2191  2         BUF_DESC        : BLOCK[8,BYTE],        ! Descriptor for buffer to be
;  1952          2192  2                                                 ! passed to the user
;  1953          2193  2         ACTION_PRM,                             ! Value of action parameter
;  1954          2194  2         ACTION_FF_FLAG,
;  1955          2195  2         PBCB            : REF $PBCB_DECL,
;  1956          2196  2         WCB             : REF $WCB_DECL;
;  1957          2197  2
;  1958          2198  2 BUILTIN
;  1959          2199  2
;  1960          2200  2         NULLPARAMETER;
;  1961          2201  2
;  1962          2202  2 OWN
;  1963          2203  2
;  1964          2204  2         BORDER_TRANS    : VECTOR[16,BYTE]
;  1965          2205  2                           INITIAL (BYTE(' -¦+--++¦+¦+++++'));
```

```
;  1967          2206  2 $SMG$VALIDATE_ARGCOUNT(2,4);
;  1968          2207  2
;  1969          2208  2 !+
;  1970          2209  2 ! Get the pasteboard control block from the pasteboard id.
;  1971          2210  2 !-
;  1972          2211  2
;  1973          2212  2 $SMG$GET_PBCB(.PASTEBOARD_ID,PBCB);
;  1974          2213  2
;  1975          2214  2 !+
;  1976          2215  2 ! Get the value of the action routine parameter.
;  1977          2216  2 !-
;  1978          2217  2
;  1979          2218  2 IF NULLPARAMETER(3)
;  1980          2219  2   THEN  ACTION_PRM=0
;  1981          2220  2   ELSE  ACTION_PRM=.PRM;
;  1982          2221  2
;  1983          2222  2 !+
;  1984          2223  2 ! Get the value of the formfeed flag.
;  1985          2224  2 !-
;  1986          2225  2
;  1987          2226  2 IF NULLPARAMETER(4)
;  1988          2227  2   THEN  ACTION_FF_FLAG=0
;  1989          2228  2   ELSE  ACTION_FF_FLAG=.FF_FLAG;
;  1990          2229  2
;  1991          2230  2 !+
;  1992          2231  2 ! Set up the WCB reference.
;  1993          2232  2 !-
;  1994          2233  2
;  1995          2234  2 WCB=.PBCB[PBCB_A_WCB];
;  1996          2235  2
;  1997          2236  2 !+
;  1998          2237  2 ! Temporary fix: ****
;  1999          2238  2 !-
;  2000          2239  2
;  2001          2240  2 IF .PBCB[PBCB_A_RBF] EQL 0
;  2002          2241  2   THEN  PBCB[PBCB_A_RBF]=TEMP_BUF;
;  2003          2242  2
;  2004          2243  2 !+
;  2005          2244  2 ! Set up the descriptor for our buffer.
;  2006          2245  2 !-
;  2007          2246  2
;  2008          2247  2 BUF_DESC[DSC$W_LENGTH] =.WCB[WCB_W_NO_COLS];
;  2009          2248  2 BUF_DESC[DSC$B_CLASS]  = DSC$K_CLASS_S;
;  2010          2249  2 BUF_DESC[DSC$B_DTYPE]  = DSC$K_DTYPE_T;
;  2011          2250  2 BUF_DESC[DSC$A_POINTER]= .PBCB[PBCB_A_RBF];
;  2012          2251  2
;  2013          2252  2 !+
;  2014          2253  2 ! Call the action routine once for each line in the pasteboard.
;  2015          2254  2 !-
;  2016          2255  2
;  2017          2256  2 INCR ROW FROM 0 TO .PBCB[PBCB_B_ROWS]-1 DO
;  2018          2257  3         BEGIN   ! Output a row
;  2019          2258  3
;  2020          2259  3         BIND
;  2021          2260  3
;  2022          2261  3                 WIDTH   = WCB[WCB_W_NO_COLS]                : WORD,
;  2023          2262  3                 TEXT    = .WCB[WCB_A_TEXT_BUF]+.ROW*.WIDTH : VECTOR[,BYTE],
```

```
: 2024          2263  3                 ATTR    = .WCB[WCB_A_ATTR_BUF]+.ROW*.WIDTH : VECTOR[,BYTE],
: 2025          2264  3                 RBF     = .PBCB[PBCB_A_RBF]                 : VECTOR[,BYTE];
: 2026          2265  3
: 2027          2266  3         BIND ROUTINE
: 2028          2267  3
: 2029          2268  3                 ACTION_ROUTINE = .P_RTN;
: 2030          2269  3
: 2031          2270  3         !+
: 2032          2271  3         ! Copy the appropriate row into the record buffer.
: 2033          2272  3         !-
: 2034          2273  3
: 2035          2274  3         CH$MOVE(.WCB[WCB_W_NO_COLS],TEXT,RBF);
: 2036          2275  3
: 2037          2276  3         !+
: 2038          2277  3         ! Scan the attribute buffer looking for any border elements.
: 2039          2278  3         ! If we find one, change its representation in the record buffer
: 2040          2279  3         ! to something more reasonable.
: 2041          2280  3         !-
: 2042          2281  3
: 2043          2282  3         INCR COL FROM 0 TO .WCB[WCB_W_NO_COLS]-1 DO
: 2044          2283  4                 BEGIN   ! Scan record buffer
: 2045          2284  4                 BIND CHAR = RBF[.COL] : BYTE;
: 2046          2285  4                 LITERAL BORDER_MASK = ATTR_M_BORD_ELEM OR ATTR_M_USER_GRAPHIC;
: 2047          2286  4                 IF (.ATTR[.COL] AND BORDER_MASK) NEQ 0
: 2048          2287  5                   THEN  BEGIN
: 2049          2288  5                         IF .CHAR GTRU 16
: 2050          2289  5                           THEN  RBF[.COL]=%C'*'
: 2051          2290  5                           ELSE  RBF[.COL]=.BORDER_TRANS[.CHAR<0,4>];
: 2052          2291  5                         END
: 2053          2292  4                 END;    ! Scan record buffer
: 2054          2293  3
: 2055          2294  3         STATUS=ACTION_ROUTINE(BUF_DESC,.ACTION_PRM);
: 2056          2295  3         IF NOT .STATUS THEN RETURN .STATUS
: 2057          2296  3
: 2058          2297  2         END;    ! Output a row
: 2059          2298  2
: 2060          2299  2 RETURN  SS$_NORMAL
: 2061          2300  2
: 2062          2301  1 END;                   ! Routine SMG$PUT_PASTEBOARD


                                                        .PSECT  _SMG$DATA,NOEXE,  PIC,2

2B 2B 2B 2B 7C 2B 7C 2B 2B 2D 2D 2B 7C 2D 20  00034 BORDER_TRANS:
                                                        .ASCII  \ -¦+--++¦+¦+++++\
                                              2B  00043


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

                                           0FFC 00000   .ENTRY  SMG$PUT_PASTEBOARD, Save R2,R3,R4,R5,R6,R7,- ; 2119
                                                                R8,R9,R10,R11
                                     5E    FDF4   CE  9E 00002 MOVAB   -524(SP), SP
                           50        6C           02  83 00007 SUBB3   #2, (AP), DIFF                          ; 2206
                                     02           50  91 0000B CMPB    DIFF, #2
```

```
                                                08  1B 0000E                BLEQU   1$                                                  ;
                                     50 00000000G   8F  D0 00010                MOVL    #SMG$_WRONUMARG, R0                                 ;
                                                    04  00017                RET                                                         ;
                                     50        04   BC  D0 00018 1$:          MOVL    @PASTEBOARD_ID, R0                                  ; 2212
                                                    11  19 0001C                BLSS    2$                                                  ;
                                00000000G    00         50  D1 0001E          CMPL    R0, PBD_L_COUNT                                     ;
                                                    09  14 00025              BGTR    2$                                                  ;
                             08 00000000G    00         50  E0 00027          BBS     R0, PBD_V_PB_AVAIL, 3$                              ;
                                     50 00000000G   8F  D0 0002F 2$:          MOVL    #SMG$_INVPAS_ID, R0                                 ;
                                                    04  00036                RET                                                         ;
                                     50 00000000G0040  D0 00037 3$:          MOVL    PBD_A_PBCB[R0], PBCB                                ;
                                     03                 6C  91 0003F          CMPB    (AP), #3                                            ; 2218
                                                    05  1F 00042              BLSSU   4$                                                  ;
                                              0C    AC  D5 00044              TSTL    12(AP)                                              ;
                                                    04  12 00047              BNEQ    5$                                                  ;
                                                    6E  D4 00049 4$:          CLRL    ACTION_PRM                                          ; 2219
                                                    04  11 0004B              BRB     6$                                                  ;
                                     6E        0C    BC  D0 0004D 5$:          MOVL    @P_PRM, ACTION_PRM                                  ; 2220
                                     04                 6C  91 00051 6$:          CMPB    (AP), #4                                            ; 2226
                                                    05  1F 00054              BLSSU   7$                                                  ;
                                              10    AC  D5 00056              TSTL    16(AP)                                              ;
                                                    04  12 00059              BNEQ    8$                                                  ;
                                                    51  D4 0005B 7$:          CLRL    ACTION_FF_FLAG                                      ; 2227
                                                    04  11 0005D              BRB     9$                                                  ;
                                     51        10    BC  D0 0005F 8$:          MOVL    @P_FF_FLAG, ACTION_FF_FLAG                          ; 2228
                                     56        08    A0  D0 00063 9$:          MOVL    8(PBCB), WCB                                        ; 2234
                                     59      00F0    C0  9E 00067              MOVAB   240(PBCB), R9                                       ; 2240
                                                    69  D5 0006C              TSTL    (R9)                                                ;
                                                    04  12 0006E              BNEQ    10$                                                 ;
                                           69        0C   AE  9E 00070          MOVAB   TEMP_BUF, (R9)                                      ; 2241
                                     04    AE        06   A6  B0 00074 10$:     MOVW    6(WCB), BUF_DESC                                    ; 2247
                                     06    AE      010E   8F  B0 00079          MOVW    #270, BUF_DESC+2                                    ; 2249
                                     08    AE             69  D0 0007F          MOVL    (R9), BUF_DESC+4                                    ; 2250
                                           5A        5F   A0  9A 00083          MOVZBL  95(PBCB), R10                                       ; 2256
                                           57             01  CE 00087          MNEGL   #1, ROW                                             ; 2294
                                                    59  11 0008A              BRB     15$                                                 ;
                                           50        06   A6  3C 0008C 11$:     MOVZWL  6(WCB), R0                                          ; 2262
                                           50             57  C4 00090          MULL2   ROW, R0                                             ;
                                  51       50        08   A6  C1 00093          ADDL3   8(WCB), R0, R1                                      ;
                                  58       50        0C   A6  C1 00098          ADDL3   12(WCB), R0, R8                                     ; 2263
                              00  B9       61        06   A6  28 0009D          MOVC3   6(WCB), (R1), @0(R9)                                ; 2274
                                           53        06   A6  3C 000A3          MOVZWL  6(WCB), R3                                          ; 2282
                                           52             01  CE 000A7          MNEGL   #1, COL                                             ;
                                                    22  11 000AA              BRB     14$                                                 ;
                                  51       69             52  C1 000AC 12$:     ADDL3   COL, (R9), R1                                       ; 2287
                                     C0    8F           6248  93 000B0          BITB    (COL)[R8], #192                                     ; 2286
                                                    17  13 000B5              BEQL    14$                                                 ;
                                           10             61  91 000B7          CMPB    (R1), #16                                           ; 2288
                                                    05  1B 000BA              BLEQU   13$                                                 ;
                                           61             2A  90 000BC          MOVB    #42, (R1)                                           ; 2289
                                                    0D  11 000BF              BRB     14$                                                 ;
                           50              61        04   00  EF 000C1 13$:     EXTZV   #0, #4, (R1), R0                                    ; 2290
                                           61 00000000'EF40  90 000C6          MOVB    BORDER_TRANS[R0], (R1)                              ;
                                  DA       52             53  F2 000CE 14$:     AOBLSS  R3, COL, 12$                                        ; 2286
                                                    6E  DD 000D2              PUSHL   ACTION_PRM                                          ; 2294
                                              08    AE  9F 000D4              PUSHAB  BUF_DESC                                            ;
                                     08       BC             02  FB 000D7          CALLS   #2, @P_RTN                                          ;
```

```
                         5B      50  DO 000DB          MOVL    R0, STATUS
                         04      5B  E8 000DE          BLBS    STATUS, 15$                 ;  2295
                         50      5B  DO 000E1          MOVL    STATUS, R0
                                     04 000E4          RET
                     A3  57      5A  F2 000E5 15$:     AOBLSS  R10, ROW, 11$
                         50      01  DO 000E9          MOVL    #1, R0                      ;  2299
                                     04 000EC          RET                                 ;  2301

; Routine Size:  237 bytes,    Routine Base:  _SMG$CODE + 1235
```

```
;  2064          2302  1 %SBTTL 'SMG$$SNAPSHOT - Snapshot pasteboard into a file'
;  2065          2303  1 GLOBAL ROUTINE SMG$$SNAPSHOT ( PASTEBOARD_ID ) =
;  2066          2304  1 !++
;  2067          2305  1 ! FUNCTIONAL DESCRIPTION:
;  2068          2306  1 !
;  2069          2307  1 !       If the output device is being controlled by RMS
;  2070          2308  1 !       (i.e. it is a file or unknown terminal)
;  2071          2309  1 !       then calling this routine causes a snapshot
;  2072          2310  1 !       of the current pasteboard to be taken and output
;  2073          2311  1 !       to the output file.
;  2074          2312  1 !       Pasteboard batching has no affect on this routine.
;  2075          2313  1 !
;  2076          2314  1 ! CALLING SEQUENCE:
;  2077          2315  1 !
;  2078          2316  1 !       ret_status.wlc.v = SMG$$SNAPSHOT ( PASTEBOARD_ID.rl.r)
;  2079          2317  1 !
;  2080          2318  1 ! FORMAL PARAMETERS:
;  2081          2319  1 !
;  2082          2320  1 !       PASTEBOARD_ID.rl.r      pasteboard id
;  2083          2321  1 !
;  2084          2322  1 ! IMPLICIT INPUTS:
;  2085          2323  1 !
;  2086          2324  1 !       contents of PBCB
;  2087          2325  1 !
;  2088          2326  1 ! IMPLICIT OUTPUTS:
;  2089          2327  1 !
;  2090          2328  1 !       NONE
;  2091          2329  1 !
;  2092          2330  1 ! COMPLETION STATUS:
;  2093          2331  1 !
;  2094          2332  1 !       SS$_NORMAL      Normal successful completion
;  2095          2333  1 !       SMG$_NOTRMSOUT  (success) no action taken since output is
;  2096          2334  1 !                       not being controlled by RMS
;  2097          2335  1 !       RMS$_xyz        Errors from RMS
;  2098          2336  1 !
;  2099          2337  1 ! SIDE EFFECTS:
;  2100          2338  1 !
;  2101          2339  1 !       NONE
;  2102          2340  1 --
```

```
;  2104          2341  2 BEGIN
;  2105          2342  2
;  2106          2343  2 EXTERNAL LITERAL
;  2107          2344  2
;  2108          2345  2         SMG$_NOTRMSOUT;          ! Not RMS output
;  2109          2346  2
;  2110          2347  2 LOCAL
;  2111          2348  2
;  2112          2349  2         STATUS,
;  2113          2350  2         PBCB             : REF $PBCB_DECL;
;  2114          2351  2
;  2115          2352  2 $SMG$VALIDATE_ARGCOUNT(1,1);
;  2116          2353  2
;  2117          2354  2 !+
;  2118          2355  2 ! Get the pasteboard control block from the pasteboard id.
;  2119          2356  2 !-
;  2120          2357  2
;  2121          2358  2 $SMG$GET_PBCB(.PASTEBOARD_ID,PBCB);
;  2122          2359  2
;  2123          2360  2 !+
;  2124          2361  2 ! Do nothing if output is not being controlled by RMS.
;  2125          2362  2 !-
;  2126          2363  2
;  2127          2364  2 IF NOT .PBCB[PBCB_V_RMS]
;  2128          2365  2   THEN  RETURN  SMG$_NOTRMSOUT;
;  2129          2366  2
;  2130          2367  2 !+
;  2131          2368  2 ! Output this pasteboard using our special RMS output routine.
;  2132          2369  2 !-
;  2133          2370  2
;  2134          2371  2 STATUS=SMG$PUT_PASTEBOARD(.PASTEBOARD_ID,RMS_RTN,PBCB);
;  2135          2372  2 IF NOT .STATUS THEN RETURN .STATUS;
;  2136          2373  2
;  2137          2374  2 RETURN  SS$_NORMAL
;  2138          2375  2
;  2139          2376  1 END;                    ! Routine SMG$SNAPSHOT


                                                                .EXTRN  SMG$_NOTRMSOUT

                                                0000 00000      .ENTRY  SMG$SNAPSHOT, Save nothing                    ;  2303
                                          5E    04   C2 00002  SUBL2   #4, SP
                                          01    6C   91 00005  CMPB    (AP), #1                                      ;  2352
                                                08   13 00008  BEQL    1$
                                          50 00000000G  8F   D0 0000A  MOVL    #SMG$_WRONUMARG, R0
                                                04   11 0000E  RET
                                          50        04   BC   D0 00012 1$:    MOVL    @PASTEBOARD_ID, R0            ;  2358
                                                11   19 00016  BLSS    2$
                                    00000000G     00        50   D1 00018  CMPL    R0, PBD_L_COUNT
                                                08   14 0001F  BGTR    2$
                                08  00000000G     00        50   E0 00021  BBS     R0, PBD_V_PB_AVAIL, 3$
                                          50 00000000G  8F   D0 00029 2$:    MOVL    #SMG$_INVPAS_ID, R0
                                                04   00030  RET
                                          6E 00000000G00040  D0 00031 3$:    MOVL    PBD_A_PBCB[R0], PBCB
                                          50        6E   D0 00039  MOVL    PBCB, R0                              ;  2364
                                08      00D0    C0        03   E0 0003C  BBS     #3, 208(R0), 4$
```

```
                          50 00000000G   8F   DO 00042          MOVL    #SMG$_NOTRMSOUT, R0                    ; 2365
                                         04   00049          RET
                                         5E   DD 0004A 4$:     PUSHL   SP                                     ; 2371
                                   0000V CF   9F 0004C          PUSHAB  RMS_RTN
                                      04 AC   DD 00050          PUSHL   PASTEBOARD_ID
                              FEBB    CF   03 FB 00053          CALLS   #3, SMG$PUT_PASTEBOARD
                                      03   50 E9 00058          BLBC    STATUS, 5$                             ; 2372
                                      50   01 D0 0005B          MOVL    #1, R0                                 ; 2374
                                           04 0005E 5$:         RET                                            ; 2376

; Routine Size:  95 bytes,    Routine Base:  _SMG$CODE + 1322
```

```
;  2141          2377  1 %SBTTL 'SMG$$SET_ATTRIBUTES_ON'
;  2142          2378  1 GLOBAL ROUTINE SMG$$SET_ATTRIBUTES_ON (
;  2143          2379  1                         PBCB : REF $PBCB_DECL,
;  2144          2380  1                         FLAGS : BITVECTOR
;  2145          2381  1                       ) =
;  2146          2382  1 !++
;  2147          2383  1 ! FUNCTIONAL DESCRIPTION:
;  2148          2384  1 !
;  2149          2385  1 !       This routine generates the escape sequences turning on
;  2150          2386  1 !       attributes such as bolding and blinking.
;  2151          2387  1 !
;  2152          2388  1 ! CALLING SEQUENCE:
;  2153          2389  1 !
;  2154          2390  1 !       ret_status.wlc.v = SMG$$SET_ATTRIBUTES_ON (PBCB,
;  2155          2391  1 !                                                  FLAGS.r[.v)
;  2156          2392  1 !
;  2157          2393  1 ! FORMAL PARAMETERS:
;  2158          2394  1 !
;  2159          2395  1 !       PBCB
;  2160          2396  1 !       FLAGS.rl.v              flags specifying which attributes to turn on
;  2161          2397  1 !
;  2162          2398  1 ! IMPLICIT INPUTS:
;  2163          2399  1 !
;  2164          2400  1 !       NONE
;  2165          2401  1 !
;  2166          2402  1 ! IMPLICIT OUTPUTS.
;  2167          2403  1 !
;  2168          2404  1 !       NONE
;  2169          2405  1 !
;  2170          2406  1 ! COMPLETION STATUS:
;  2171          2407  1 !
;  2172          2408  1 !
;  2173          2409  1 ! SIDE EFFECTS:
;  2174          2410  1 !
;  2175          2411  1 !       NONE
;  2176          2412  1 !--
```

```
;  2178          2413  2 BEGIN
;  2179          2414  2
;  2180          2415  2 LOCAL
;  2181          2416  2
;  2182          2417  2         STATUS;
;  2183          2418  2
;  2184          2419  2 BIND    TT2 = PBCB[PBCB_L_DEVDEPEND2]   : $BBLOCK;
;  2185          2420  2
;  2186          2421  2 !+
;  2187          2422  2 ! Renditions requires that the AVO (ADVANCED_VIDEO) terminal
;  2188          2423  2 ! characteristic bit be set.  Even if the TERMTABLE entries
;  2189          2424  2 ! show that the terminal has the BEGIN_BOLD capability,
;  2190          2425  2 ! the terminal might not have the advanced video option.
;  2191          2426  2 !-
;  2192          2427  2
;  2193          2428  2 IF .FLAGS[ATTR_V_REND_GRAPHIC]
;  2194          2429  3 AND (.TT2[TT2$V_AVO] OR NOT .TT2[TT2$V_ANSICRT])
;  2195          2430  3   THEN  BEGIN
;  2196          2431  3         $SMG$GET_TERM_DATA(BEGIN_LINE_DRAWING_CHAR);
;  2197          2432  3
;  2198          2433  3         IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
;  2199          2434  4           THEN  BEGIN
;  2200          2435  4                 STATUS=SMG$$OUTPUT(.PBCB,.PBCB[PBCB_L_CAP_LENGTH],
;  2201          2436  4                                   .PBCB[PBCB_A_CAP_BUFFER]);
;  2202          2437  4                 IF NOT .STATUS THEN RETURN .STATUS
;  2203          2438  3                 END;
;  2204          2439  2         END;
;  2205          2440  2
;  2206          2441  2 !+
;  2207          2442  2 ! Get and output the string to set the correct attributes.
;  2208          2443  2 !-
;  2209          2444  2
;  2210          2445  2 IF .FLAGS[ATTR_V_REND_BOLD]
;  2211          2446  3 AND (.TT2[TT2$V_AVO] OR NOT .TT2[TT2$V_ANSICRT])
;  2212          2447  3   THEN  BEGIN
;  2213          2448  3         $SMG$GET_TERM_DATA(BEGIN_BOLD);
;  2214          2449  3
;  2215          2450  3         IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
;  2216          2451  4           THEN  BEGIN
;  2217          2452  4                 STATUS=SMG$$OUTPUT(.PBCB,.PBCB[PBCB_L_CAP_LENGTH],
;  2218          2453  4                                   .PBCB[PBCB_A_CAP_BUFFER]);
;  2219          2454  4                 IF NOT .STATUS THEN RETURN .STATUS;
;  2220          2455  3                 END;
;  2221          2456  2         END;
;  2222          2457  2
;  2223          2458  2 IF .FLAGS[ATTR_V_REND_BLINK]
;  2224          2459  3 AND (.TT2[TT2$V_AVO] OR NOT .TT2[TT2$V_ANSICRT])
;  2225          2460  3   THEN  BEGIN
;  2226          2461  3         $SMG$GET_TERM_DATA(BEGIN_BLINK);
;  2227          2462  3
;  2228          2463  3         IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
;  2229          2464  4           THEN  BEGIN
;  2230          2465  4                 STATUS=SMG$$OUTPUT(.PBCB,.PBCB[PBCB_L_CAP_LENGTH],
;  2231          2466  4                                   .PBCB[PBCB_A_CAP_BUFFER]);
;  2232          2467  4                 IF NOT .STATUS THEN RETURN .STATUS;
;  2233          2468  3                 END;
;  2234          2469  2         END;
```

```
;  2235          2470  2
;  2236          2471  2 IF .FLAGS[ATTR_V_REND_REV]
;  2237          2472  3 AND (.TT2[TT2$V_AVO] OR NOT .TT2[TT2$V_ANSICRT])
;  2238          2473  3   THEN  BEGIN
;  2239          2474  3         $SMG$GET_TERM_DATA(BEGIN_REVERSE);
;  2240          2475  3
;  2241          2476  3         IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
;  2242          2477  4           THEN  BEGIN
;  2243          2478  4                 STATUS=SMG$$OUTPUT(.PBCB,.PBCB[PBCB_L_CAP_LENGTH],
;  2244          2479  4                                   .PBCB[PBCB_A_CAP_BUFFER]);
;  2245          2480  4                 IF NOT .STATUS THEN RETURN .STATUS;
;  2246          2481  3                 END;
;  2247          2482  2         END;
;  2248          2483  2
;  2249          2484  2 IF .FLAGS[ATTR_V_REND_UNDER]
;  2250          2485  3 AND (.TT2[TT2$V_AVO] OR NOT .TT2[TT2$V_ANSICRT])
;  2251          2486  3   THEN  BEGIN
;  2252          2487  3         $SMG$GET_TERM_DATA(BEGIN_UNDERSCORE);
;  2253          2488  3
;  2254          2489  3         IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
;  2255          2490  4           THEN  BEGIN
;  2256          2491  4                 STATUS=SMG$$OUTPUT(.PBCB,.PBCB[PBCB_L_CAP_LENGTH],
;  2257          2492  4                                   .PBCB[PBCB_A_CAP_BUFFER]);
;  2258          2493  4                 IF NOT .STATUS THEN RETURN .STATUS;
;  2259          2494  3                 END;
;  2260          2495  2         END;
;  2261          2496  2
;  2262          2497  2 RETURN  SS$_NORMAL
;  2263          2498  2
;  2264          2499  1 END;                                ! End of routine SMG$$SET_ATTRIBUTES_ON


                                    00FC 00000          .ENTRY  SMG$$SET_ATTRIBUTES_ON, Save R2,R3,R4,R5,-      ; 2378
                                                                R6,R7
                              57  0000V CF  9E 00002            MOVAB   SMG$$OUTPUT, R7                          ;
                              56 00000000G 00  9E 00007         MOVAB   SMG$GET_TERM_DATA, R6                    ;
                              5E           10  C2 0000E         SUBL2   #16, SP                                  ;
                              52     04    AC  D0 00011         MOVL    PBCB, R2                                 ; 2419
                              54     60    A2  9E 00015         MOVAB   96(R2), R4                               ;
                        4F    08    AC        04  E1 00019      BBC     #4, FLAGS, 4$                            ; 2428
                        04          64        1B  E0 0001E      BBS     #27, (R4), 1$                            ; 2429
                              47     03    A4  E8 00022         BLBS    3(R4), 4$                                ;
                              53   0108    C2  9E 00026 1$:     MOVAB   264(R2), R3                              ; 2431
                                   00FC    C2  D5 0002B         TSTL    252(R2)                                  ;
                                           04  12 0002F         BNEQ    2$                                       ;
                                           63  D4 00031         CLRL    (R3)                                     ;
                                           23  11 00033         BRB     3$                                       ;
                                     04    AE  D4 00035 2$:     CLRL    INPUT_ARGS                               ;
                                     04    AE  9F 00038         PUSHAB  INPUT_ARGS                               ;
                                   0104    C2  DD 0003B         PUSHL   260(R2)                                  ;
                                           53  DD 0003F         PUSHL   R3                                       ;
                                   0100    C2  9F 00041         PUSHAB  256(R2)                                  ;
                              10   AE    01BE  8F 3C 00045      MOVZWL  #446, 16(SP)                             ;
                                     10    AE  9F 0004B         PUSHAB  16(SP)                                   ;
```

```
                                00FC   C2  9F  0004E          PUSHAB  252(R2)                                    ;
                             66        06  FB  00052          CALLS   #6, SMG$GET_TERM_DATA                      ;
                             50        50  E9  00055          BLBC    STATUS, 7$                                 ;
                                       63  D5  00058  3$:     TSTL    (R3)                                       ; 2433
                                       11  13  0005A          BEQL    4$                                         ;
                                0104   C2  DD  0005C          PUSHL   260(R2)                                    ; 2436
                                       63  DD  00060          PUSHL   (R3)                                       ; 2435
                                       52  DD  00062          PUSHL   R2                                         ;
                             67        03  FB  00064          CALLS   #3, SMG$$OUTPUT                            ;
                             55        50  D0  00067          MOVL    R0, STATUS                                 ;
                             50        55  E9  0006A          BLBC    STATUS, 9$                                 ; 2437
                             4F   08   AC  E9  0006D  4$:     BLBC    FLAGS, 10$                                 ; 2445
                        04   64        1B  E0  00071          BBS     #27, (R4), 5$                              ; 2446
                             47   03   A4  E8  00075          BLBS    3(R4), 10$                                 ;
                             53   0108   C2  9E  00079  5$:     MOVAB   264(R2), R3                                ; 2449
                                  00FC   C2  D5  0007E          TSTL    252(R2)                                    ;
                                         04  12  00082          BNEQ    6$                                         ;
                                         63  D4  00084          CLRL    (R3)                                       ;
                                         23  11  00086          BRB     8$                                         ;
                                     04  AE  D4  00088  6$:     CLRL    INPUT_ARGS                                 ;
                                     04  AE  9F  0008B          PUSHAB  INPUT_ARGS                                 ;
                                0104   C2  DD  0008E          PUSHL   260(R2)                                    ;
                                       53  DD  00092          PUSHL   R3                                         ;
                                0100   C2  9F  00094          PUSHAB  256(R2)                                    ;
                        10   AE   01BB   8F  3C  00098          MOVZWL  #443, 16(SP)                               ;
                                     10  AE  9F  0009E          PUSHAB  16(SP)                                     ;
                                00FC   C2  9F  000A1          PUSHAB  252(R2)                                    ;
                             66        06  FB  000A5          CALLS   #6, SMG$GET_TERM_DATA                      ;
                             51        50  E9  000A8  7$:     BLBC    STATUS, 13$                                ;
                                       63  D5  000AB  8$:     TSTL    (R3)                                       ; 2450
                                       11  13  000AD          BEQL    10$                                        ;
                                0104   C2  DD  000AF          PUSHL   260(R2)                                    ; 2453
                                       63  DD  000B3          PUSHL   (R3)                                       ; 2452
                                       52  DD  000B5          PUSHL   R2                                         ;
                             67        03  FB  000B7          CALLS   #3, SMG$$OUTPUT                            ;
                             55        50  D0  000BA          MOVL    R0, STATUS                                 ;
                             51        55  E9  000BD  9$:     BLBC    STATUS, 15$                                ; 2454
                        4F   08   AC        02  E1  000C0  10$:    BBC     #2, FLAGS, 16$                             ; 2458
                        04   64        1B  E0  000C5          BBS     #27, (R4), 11$                             ; 2459
                             47   03   A4  E8  000C9          BLBS    3(R4), 16$                                 ;
                             53   0108   C2  9E  000CD  11$:     MOVAB   264(R2), R3                                ; 2461
                                  00FC   C2  D5  000D2          TSTL    252(R2)                                    ;
                                         04  12  000D6          BNEQ    12$                                        ;
                                         63  D4  000D8          CLRL    (R3)                                       ;
                                         23  11  000DA          BRB     14$                                        ;
                                     04  AE  D4  000DC  12$:     CLRL    INPUT_ARGS                                 ;
                                     04  AE  9F  000DF          PUSHAB  INPUT_ARGS                                 ;
                                0104   C2  DD  000E2          PUSHL   260(R2)                                    ;
                                       53  DD  000E6          PUSHL   R3                                         ;
                                0100   C2  9F  000E8          PUSHAB  256(R2)                                    ;
                        10   AE   01BA   8F  3C  000EC          MOVZWL  #442, 16(SP)                               ;
                                     10  AE  9F  000F2          PUSHAB  16(SP)                                     ;
                                00FC   C2  9F  000F5          PUSHAB  252(R2)                                    ;
                             66        06  FB  000F9          CALLS   #6, SMG$GET_TERM_DATA                      ;
                             51        50  E9  000FC  13$:     BLBC    STATUS, 19$                                ;
                                       63  D5  000FF  14$:     TSTL    (R3)                                       ; 2463
                                       11  13  00101          BEQL    16$                                        ;
```

```
                        0104  C2  DD 00103          PUSHL   260(R2)                     ; 2466
                              63  DD 00107          PUSHL   (R3)                        ; 2465
                              52  DD 00109          PUSHL   R2
                  67          03  FB 0010B          CALLS   #3, SMG$$OUTPUT
                  55          50  D0 0010E          MOVL    R0, STATUS
                  51          55  E9 00111 15$:     BLBC    STATUS, 21$                 ; 2467
     4F      08   AC          01  E1 00114 16$:     BBC     #1, FLAGS, 22$              ; 2471
     04           64          1B  E0 00119          BBS     #27, (R4), 17$              ; 2472
                  47    03    A4  E8 0011D          BLBS    3(R4), 22$
                  53  0108    C2  9E 00121 17$:     MOVAB   264(R2), R3                 ; 2474
                      00FC    C2  D5 00126          TSTL    252(R2)
                              04  12 0012A          BNEQ    18$
                              63  D4 0012C          CLRL    (R3)
                              23  11 0012E          BRB     20$
                        04    AE  D4 00130 18$:     CLRL    INPUT_ARGS
                        04    AE  9F 00133          PUSHAB  INPUT_ARGS
                      0104    C2  DD 00136          PUSHL   260(R2)
                              53  DD 0013A          PUSHL   R3
                      0100    C2  9F 0013C          PUSHAB  256(R2)
             10   AE  01BF    8F  3C 00140          MOVZWL  #447, 16(SP)
                        10    AE  9F 00146          PUSHAB  16(SP)
                      00FC    C2  9F 00149          PUSHAB  252(R2)
                  66          06  FB 0014D          CALLS   #6, SMG$GET_TERM_DATA
                  70          50  E9 00150 19$:     BLBC    STATUS, 28$
                              63  D5 00153 20$:     TSTL    (R3)                        ; 2476
                              11  13 00155          BEQL    22$
                      0104    C2  DD 00157          PUSHL   260(R2)                     ; 2479
                              63  DD 0015B          PUSHL   (R3)                        ; 2478
                              52  DD 0015D          PUSHL   R2
                  67          03  FB 0015F          CALLS   #3, SMG$$OUTPUT
                  55          50  D0 00162          MOVL    R0, STATUS
                  54          55  E9 00165 21$:     BLBC    STATUS, 26$                 ; 2480
     53      08   AC          03  E1 00168 22$:     BBC     #3, FLAGS, 27$              ; 2484
     04           64          1B  E0 0016D          BBS     #27, (R4), 23$              ; 2485
                  4B    03    A4  E8 00171          BLBS    3(R4), 27$
                  53  0108    C2  9E 00175 23$:     MOVAB   264(R2), R3                 ; 2487
                      00FC    C2  D5 0017A          TSTL    252(R2)
                              04  12 0017E          BNEQ    24$
                              63  D4 00180          CLRL    (R3)
                              23  11 00182          BRB     25$
                        04    AE  D4 00184 24$:     CLRL    INPUT_ARGS
                        04    AE  9F 00187          PUSHAB  INPUT_ARGS
                      0104    C2  DD 0018A          PUSHL   260(R2)
                              53  DD 0018E          PUSHL   R3
                      0100    C2  9F 00190          PUSHAB  256(R2)
             10   AE  01C0    8F  3C 00194          MOVZWL  #448, 16(SP)
                        10    AE  9F 0019A          PUSHAB  16(SP)
                      00FC    C2  9F 0019D          PUSHAB  252(R2)
                  66          06  FB 001A1          CALLS   #6, SMG$GET_TERM_DATA
                  1C          50  E9 001A4          BLBC    STATUS, 28$
                              63  D5 001A7 25$:     TSTL    (R3)                        ; 2489
                              15  13 001A9          BEQL    27$
                      0104    C2  DD 001AB          PUSHL   260(R2)                     ; 2492
                              63  DD 001AF          PUSHL   (R3)                        ; 2491
                              52  DD 001B1          PUSHL   R2
                  67          03  FB 001B3          CALLS   #3, SMG$$OUTPUT
                  55          50  D0 001B6          MOVL    R0, STATUS
```

```
                                 04          55  E8 001B9          BLBS    STATUS, 27$                                    ; 2493
                                 50          55  D0 001BC 26$:     MOVL    STATUS, R0                                     ;
                                                 04 001BF          RET                                                    ;
                                 50          01  D0 001C0 27$:     MOVL    #1, R0                                         ; 2497
                                                 04 001C3 28$:     RET                                                    ; 2499

; Routine Size:  452 bytes,    Routine Base:  _SMG$CODE + 1381
```

```
;  2266          2500  1 %SBTTL 'SMG$$SET_ATTRIBUTES_OFF'
;  2267          2501  1 GLOBAL ROUTINE SMG$$SET_ATTRIBUTES_OFF (
;  2268          2502  1                       PBCB : REF $PBCB_DECL,
;  2269          2503  1                       FLAGS : BITVECTOR
;  2270          2504  1                     ) =
;  2271          2505  1 !++
;  2272          2506  1 ! FUNCTIONAL DESCRIPTION:
;  2273          2507  1 !
;  2274          2508  1 !       This routine generates the escape sequenc s turning on
;  2275          2509  1 !       attributes such as bolding and blinking.
;  2276          2510  1 !
;  2277          2511  1 ! CALLING SEQUENCE:
;  2278          2512  1 !
;  2279          2513  1 !       ret_status.wlc.v = SMG$$SET_ATTRIBUTES_C F (PBCB,
;  2280          2514  1 !                                                   LAGS.rl.v)
;  2281          2515  1 !
;  2282          2516  1 ! FORMAL PARAMETERS:
;  2283          2517  1 !
;  2284          2518  1 !       PBCB
;  2285          2519  1 !       FLAGS.rl.v              flags specifying which attributes to turn on
;  2286          2520  1 !
;  2287          2521  1 ! IMPLICIT INPUTS:
;  2288          2522  1 !
;  2289          2523  1 !       NONE
;  2290          2524  1 !
;  2291          2525  1 ! IMPLICIT UUTPUTS:
;  2292          2526  1 !
;  2293          2527  1 !       NONE
;  2294          2528  1 !
;  2295          2529  1 ! COMPLETION STATUS:
;  2296          2530  1 !
;  2297          2531  1 !
;  2298          2532  1 ! SIDE EFFECTS:
;  2299          2533  1 !
;  2300          2534  1 !       NONE
;  2301          2535  1 !--
```

```
;  2303          2536  2 BEGIN
;  2304          2537  2
;  2305          2538  2 LOCAL
;  2306          2539  2
;  2307          2540  2         STATUS;
;  2308          2541  2
;  2309          2542  2 BIND    TT2 = PBCB[PBCB_L_DEVDEPEND2]   : $BBLOCK;
;  2310          2543  2
;  2311          2544  2 !+
;  2312          2545  2 ! Renditions requires that the AVO (ADVANCED VIDEO) terminal
;  2313          2546  2 ! characteristic bit be set.  Even if the TERMTABLE entries
;  2314          2547  2 ! show that the terminal has the BEGIN_BOLD capability,
;  2315          2548  2 ! the terminal might not have the advanced video option.
;  2316          2549  2 !-
;  2317          2550  2
;  2318          2551  2 !+
;  2319          2552  2 ! Get and output the suffix string to reset attributes to normal.
;  2320          2553  2 ! We used to assume that END_BOLD brings back normal attributes.
;  2321          2554  2 ! Now we rely on BEGIN_NORMAL_RENDITION.
;  2322          2555  2 !-
;  2323          2556  2
;  2324          2557  2 IF .TT2[TT2$V_AVO] OR NOT .TT2[TT2$V_ANSICRT]
;  2325          2558  3   THEN  BEGIN
;  2326          2559  3
;  2327        L 2560  3         %IF %DECLARED( SMG$K_BEGIN_NORMAL_RENDITION )
;  2328          2561  3         %THEN
;  2329          2562  3                 $SMG$GET_TERM_DATA(BEGIN_NORMAL_RENDITION);
;  2330        U 2563  3         %ELSE
;  2331        U 2564  3                 $SMG$GET_TERM_DATA(END_BOLD);
;  2332          2565  3         %FI
;  2333          2566  3
;  2334          2567  3         IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
;  2335          2568  4           THEN  BEGIN
;  2336          2569  4                 STATUS=SMG$$OUTPUT(.PBCB,.PBCB[PBCB_L_CAP_LENGTH],
;  2337          2570  4                                    .PBCB[PBCB_A_CAP_BUFFER]);
;  2338          2571  4                 IF NOT .STATUS THEN RETURN .STATUS;
;  2339          2572  3                 END;
;  2340          2573  2         END;
;  2341          2574  2
;  2342          2575  2 RETURN  SS$_NORMAL
;  2343          2576  2
;  2344          2577  1 END;


                                        000C 00000          .ENTRY  SMG$$SET_ATTRIBUTES_OFF, Save R2,R3          ; 2501
                                    5E  10   C2 00002       SUBL2   #16, SP                                      ;
                                    52  04   AC D0 00005       MOVL    PBCB, R2                                 ; 2542
                              04    63   A2  03   E0 00009       BBS     #3, 99(R2), 1$                           ; 2557
                                    4A   63   A2  E8 0000E       BLBS    99(R2), 4$                               ;
                                    53 0108   C2  9E 00012 1$:    MOVAB   264(R2), R3                              ; 2562
                                       00FC   C2  D5 00017       TSTL    252(R2)                                  ;
                                              04  12 0001B       BNEQ    2$                                       ;
                                              63  D4 0001D       CLRL    (R3)                                     ;
                                              27  11 0001F       BRB     3$                                       ;
```

```
                                        04   AE  D4 00021 2$:     CLRL    INPUT_ARGS                                  ;
                                        04   AE  9F 00024         PUSHAB  INPUT_ARGS                                  ;
                                      0104   C2  DD 00027         PUSHL   260(R2)                                     ;
                                        53   DD  0002B           PUSHL   R3                                          ;
                                      0100   C2  9F 0002D         PUSHAB  256(R2)                                     ;
                                  10  AE   0253  8F  3C 00031     MOVZWL  #595, 16(SP)                                ;
                                        10   AE  9F 00037         PUSHAB  16(SP)                                      ;
                                      00FC   C2  9F 0003A         PUSHAB  252(R2)                                     ;
                          00000000G   00       06  FB 0003E     CALLS   #6, SMG$GET_TERM_DATA                       ;
                                      17       50  E9 00045     BLBC    STATUS, 5$                                  ;
                                               63  D5 00048 3$:  TSTL    (R3)                                        ; 2567
                                               10  13 0004A     BEQL    4$                                          ;
                                      0104     C2  DD 0004C     PUSHL   260(R2)                                     ; 2570
                                               63  DD 00050     PUSHL   (R3)                                        ; 2569
                                               52  DD 00052     PUSHL   R2                                          ;
                                0000V  CF            03  FB 00054     CALLS   #3, SMG$$OUTPUT                             ;
                                       03            50  E9 00059     BLBC    STATUS, 5$                                  ; 2571
                                       50            01  D0 0005C 4$:  MOVL    #1, R0                                      ; 2575
                                                     04  0005F 5$:  RET                                                 ; 2577

; Routine Size:  96 bytes,    Routine Base:  _SMG$CODE + 1545
```

```
;  2346          2578  1 %SBTTL 'RMS_RTN - Action routine used to output a line with RMS'
;  2347          2579  1 ROUTINE RMS_RTN ( P_LINE_DESC, P_PBCB ) =
;  2348          2580  1 !++
;  2349          2581  1 ! FUNCTIONAL DESCRIPTION:
;  2350          2582  1 !
;  2351          2583  1 !       Outputs a line to the output file using RMS.
;  2352          2584  1 !
;  2353          2585  1 ! CALLING SEQUENCE:
;  2354          2586  1 !
;  2355          2587  1 !       ret_status.wlc.v = RMS_RTN ( P_LINE_DESC.rt.ds, P_PBCB.rab.r)
;  2356          2588  1 !
;  2357          2589  1 ! FORMAL PARAMETERS:
;  2358          2590  1 !
;  2359          2591  1 !       P_LINE_DESC.rt.ds       Address of fixed length string descriptor
;  2360          2592  1 !                               for line to be output.
;  2361          2593  1 !
;  2362          2594  1 !       P_PBCB.rab.r            Address of pasteboard control block
;  2363          2595  1 !
;  2364          2596  1 ! IMPLICIT INPUTS:
;  2365          2597  1 !
;  2366          2598  1 !       contents of PBCB
;  2367          2599  1 !
;  2368          2600  1 ! IMPLICIT OUTPUTS:
;  2369          2601  1 !
;  2370          2602  1 !       NONE
;  2371          2603  1 !
;  2372          2604  1 ! COMPLETION STATUS:
;  2373          2605  1 !
;  2374          2606  1 !       RMS$_NORMAL     Normal successful completion
;  2375          2607  1 !       RMS$_xyz        Errors from RMS
;  2376          2608  1 !
;  2377          2609  1 ! SIDE EFFECTS:
;  2378          2610  1 !
;  2379          2611  1 !       NONE
;  2380          2612  1 !--
```

```
;  2332          2613  2 BEGIN
;  2383          2614  2
;  2384          2615  2 BIND
;  2385          2616  2
;  2386          2617  2         LINE_DESC       = .P_LINE_DESC          : BLOCK[8,BYTE],
;  2387          2618  2         PBCB            = .P_PBCB               : $PBCB_DECL,
;  2388          2619  2         SMGRAB          = .PBCB[PBCB_A_RAB]     : $RAB_DECL;
;  2389          2620  2
;  2390          2621  2 !+
;  2391          2622  2 ! Output this line using RMS.
;  2392          2623  2 !-
;  2393          2624  2
;  2394          2625  2 SMGRAB[RAB$W_RSZ]       = .LINE_DESC[DSC$W_LENGTH];
;  2395          2626  2 SMGRAB[RAB$L_RBF]       = .LINE_DESC[DSC$A_POINTER];
;  2396          2627  2
;  2397          2628  3 RETURN $PUT(RAB=SMGRAB)
;  2398          2629  3
;  2399          2630  1 END;                    ! Routine RMS_RTN


                                                        .EXTRN  SYS$PUT

                                              0000 00000 RMS_RTN:.WORD   Save nothing                          ; 2579
                                    51      04    AC   D0 00002        MOVL   P_LINE_DESC, R1                 ; 2617
                                    50      08    AC   D0 00006        MOVL   P_PBCB, R0                      ; 2618
                                    50    00EC    C0   D0 0000A        MOVL   236(R0), R0                     ; 2619
                                22  A0            61   B0 0000F        MOVW   (R1), 34(R0)                    ; 2625
                                28  A0      04    A1   D0 00013        MOVL   4(R1), 40(R0)                   ; 2626
                                               50   DD 00018        PUSHL  R0                              ; 2628
                          00000000G  00        01   FB 0001A        CALLS  #1, SYS$PUT
                                               04 00021             RET                                    ; 2630

; Routine Size:  34 bytes,    Routine Base:  _SMG$CODE + 15A5
```

```
;  2401          2631  1 %SBTTL 'SMG$$MIN_UPD - Calculate minimum update sequence and output'
;  2402          2632  1 GLOBAL ROUTINE SMG$$MIN_UPD (
;  2403          2633  1                              PBCB : REF $PBCB_DECL
;  2404          2634  1                             ) =
;  2405          2635  1 !++
;  2406          2636  1 ! FUNCTIONAL DESCRIPTION:
;  2407          2637  1 !
;  2408          2638  1 !       Obsolete.
;  2409          2639  1 !       SMG$$OUTPUT_PASTEBOARD should be called instead.
;  2410          2640  1 !       If this or that bombs out, you can use the old original temporary
;  2411          2641  1 !       routine that Rich wrote.  It's called SMG$$OLD_MIN_UPD.
;  2412          2642  1 !
;  2413          2643  1 ! CALLING SEQUENCE:
;  2414          2644  1 !
;  2415          2645  1 !       ret_status.wlc.v = SMG$$MIN_UPD ( PBCB.rab.r )
;  2416          2646  1 !
;  2417          2647  1 ! FORMAL PARAMETERS:
;  2418          2648  1 !
;  2419          2649  1 !       PBCB.rab.r              Address of pasteboard control block.
;  2420          2650  1 !
;  2421          2651  1 ! IMPLICIT INPUTS:
;  2422          2652  1 !
;  2423          2653  1 !       NONE
;  2424          2654  1 !
;  2425          2655  1 ! IMPLICIT OUTPUTS:
;  2426          2656  1 !
;  2427          2657  1 !       NONE
;  2428          2658  1 !
;  2429          2659  1 ! COMPLETION STATUS:
;  2430          2660  1 !
;  2431          2661  1 !       SS$_NORMAL      Normal successful completion
;  2432          2662  1 !
;  2433          2663  1 ! SIDE EFFECTS:
;  2434          2664  1 !
;  2435          2665  1 !       NONE
;  2436          2666  1 !--
```

```
;  2438          2667  2 BEGIN
;  2439          2668  2 LOCAL
;  2440          2669  2     WCB : REF $WCB_DECL;         ! Address of Window Control Block.
;  2441          2670  2
;  2442          2671  2 WCB = .PBCB [PBCB_A_WCB];
;  2443          2672  2 IF .PBCB [PBCB_W_LAST_CHANGED_ROW] NEQ 0
;  2444          2673  2 THEN
;  2445          2674  3     BEGIN       ! Normal case
;  2446          2675  3     LOCAL
;  2447          2676  3         LC  : REF VECTOR [,BYTE],        ! Addr of line characteristics
;  2448          2677  3                                          ! vector for text buffer.
;  2449          2678  3         LCS : REF VECTOR [,BYTE],        ! Addr of line characteristics
;  2450          2679  3                                          ! vector for screen text buffer.
;  2451          2680  3         B_OFFSET,        ! Byte offset to begining of line of interest
;  2452          2681  3         WIDTH;           ! Extracted copy of .WCB [WCB_W_NO_COLS]
;  2453          2682  3
;  2454          2683  3     WIDTH = .WCB [WCB_W_NO_COLS];
;  2455          2684  3     B_OFFSET = (.PBCB [PBCB_W_FIRST_CHANGED_ROW] - 1 ) * .WIDTH;
;  2456          2685  3     LC  = .WCB [WCB_A_LINE_CHAR];
;  2457          2686  3     LCS = .WCB [WCB_A_SCR_LINE_CHAR];
;  2458          2687  3
;  2459          2688  3     !+
;  2460          2689  3     ! Try to narrow the range of lines that claim to have been changed.
;  2461          2690  3     ! If we can collapse it to 1 or less, scrolling is not feasible.
;  2462          2691  3     ! As a by-product of doing these tests, the range of lines that
;  2463          2692  3     ! may have changed will possibly be narrowed, making minimum
;  2464          2693  3     ! update's work faster.
;  2465          2694  3     ! First try to refine the "first changed line" downwards.
;  2466          2695  3     !-
;  2467          2696  3     WHILE .PBCB [PBCB_W_FIRST_CHANGED_ROW] LSS
;  2468          2697  3           .PBCB [PBCB_W_LAST_CHANGED_ROW]
;  2469          2698  3     DO
;  2470          2699  4         BEGIN   ! Collapsing loop
;  2471          2700  4         !+
;  2472          2701  4         ! If this line is the same, with respect to the text buffer,
;  2473          2702  4         ! the attribute buffer, and the line characteristics vector,
;  2474          2703  4         ! then it is not changed -- drag down the first changed line by
;  2475          2704  4         ! 1.
;  2476          2705  4         !-
;  2477          2706  4         IF CH$EQL ( .WIDTH, .WCB [WCB_A_TEXT_BUF]     + .B_OFFSET,
;  2478          2707  4                     .WIDTH, .WCB [WCB_A_SCR_TEXT_BUF] + .B_OFFSET)
;  2479          2708  4
;  2480          2709  4            AND
;  2481          2710  4
;  2482          2711  4            CH$EQL ( .WIDTH, .WCB [WCB_A_ATTR_BUF]     + .B_OFFSET,
;  2483          2712  4                     .WIDTH, .WCB [WCB_A_SCR_ATTR_BUF] + .B_OFFSET)
;  2484          2713  4
;  2485          2714  4            AND
;  2486          2715  4
;  2487          2716  4            .LC  [.PBCB [PBCB_W_FIRST_CHANGED_ROW] ] EQL
;  2488          2717  4            .LCS [.PBCB [PBCB_W_FIRST_CHANGED_ROW] ]
;  2489          2718  4
;  2490          2719  4         THEN
;  2491          2720  5             BEGIN       ! Advance one row
;  2492          2721  5             PBCB [PBCB_W_FIRST_CHANGED_ROW] =
;  2493          2722  5                                 .PBCB [PBCB_W_FIRST_CHANGED_ROW] + 1;
;  2494          2723  5             B_OFFSET = .B_OFFSET + .WIDTH;
```

```
; 2495          2724  5              END ! Advance one row
; 2496          2725  4          ELSE
; 2497          2726  4              EXITLOOP;   ! 1st refined downward as far as possible
; 2498          2727  4          END;    ! Collapsing loop
; 2499          2728  4
; 2500          2729  4      !+
; 2501          2730  4      ! Now try to refine "last changed line" upward in a similar manner.
; 2502          2731  4      !-
; 2503          2732  4      B_OFFSET = (.PBCB [PBCB_W_LAST_CHANGED_ROW] - 1 ) * .WIDTH;
; 2504          2733  4
; 2505          2734  4      WHILE .PBCB [PBCB_W_FIRST_CHANGED_ROW] LSS
; 2506          2735  4            .PBCB [PBCB_W_LAST_CHANGED_ROW]
; 2507          2736  4      DO
; 2508          2737  4          BEGIN   ! Collapsing loop
; 2509          2738  4
; 2510          2739  4          !+
; 2511          2740  4          ! If this line is the same, with respect to the text buffer,
; 2512          2741  4          ! the attribute buffer, and the line characteristics vector,
; 2513          2742  4          ! then it is not changed -- drag up the last changed line by 1.
; 2514          2743  4          !-
; 2515          2744  4          IF CH$EQL ( .WIDTH, .WCB [WCB_A_TEXT_BUF]     + .B_OFFSET,
; 2516          2745  4                      .WIDTH, .WCB [WCB_A_SCR_TEXT_BUF] + .B_OFFSET)
; 2517          2746  4
; 2518          2747  4             AND
; 2519          2748  4
; 2520          2749  4             CH$EQL ( .WIDTH, .WCB [WCB_A_ATTR_BUF]     + .B_OFFSET,
; 2521          2750  4                      .WIDTH, .WCB [WCB_A_SCR_ATTR_BUF] + .B_OFFSET)
; 2522          2751  4
; 2523          2752  4             AND
; 2524          2753  4
; 2525          2754  4             .LC  [.PBCB [PBCB_W_LAST_CHANGED_ROW] ] EQL
; 2526          2755  4             .LCS [.PBCB [PBCB_W_LAST_CHANGED_ROW] ]
; 2527          2756  4
; 2528          2757  4          THEN
; 2529          2758  5              BEGIN       ! Back up one row
; 2530          2759  5              PBCB [PBCB_W_LAST_CHANGED_ROW] =
; 2531          2760  5                                  .PBCB [PBCB_W_LAST_CHANGED_ROW] - 1;
; 2532          2761  5              B_OFFSET = .B_OFFSET - .WIDTH;
; 2533          2762  5              END ! Back up one row
; 2534          2763  4          ELSE
; 2535          2764  4              EXITLOOP;   ! 1st refined downward as far as possible
; 2536          2765  3          END;    ! Collapsing loop
; 2537          2766  3      END         ! Normal case
; 2538          2767  3
; 2539          2768  2 ELSE
; 2540          2769  2
; 2541          2770  3      BEGIN       ! Range not set case
; 2542          2771  3      !+
; 2543          2772  3      ! It is possible, in some obscure cases, to reach here with the
; 2544          2773  3      ! the 1st changed row set to #rows+1 and last changed row set to 0.
; 2545          2774  3      ! In this case, set range to whole pasteboard.
; 2546          2775  3      !-
; 2547          2776  3      PBCB [PBCB_W_FIRST_CHANGED_ROW] = 1;
; 2548          2777  3      PBCB [PBCB_W_LAST_CHANGED_ROW]  = .WCB [WCB_W_NO_ROWS];
; 2549          2778  2      END;        ! Range not set case
; 2550          2779  2
; 2551          2780  2 !+
```

```
;  2552          2781  2 ! If we reach here, we have legitimate differences on the lines
;  2553          2782  2 ! between .PBCB [PBCB_FIRST_CHANGED_ROW] and
;  2554          2783  2 ! .PBCB [PBCB_LAST_CHANGED_ROW]
;  2555          2784  2 !-
;  2556          2785  2
;  2557          2786  2 !+
;  2558          2787  2 ! If terminal supports scrolling regions, check to see if minimal update can be helped
;  2559          2788  2 ! by using physical scrolling regions.
;  2560          2789  2 ! *** Actually, we could also do full screen scrolling if we wanted to.
;  2561          2790  2 !-
;  2562          2791  2
;  2563          2792  2 $SMG$GET_TERM_DATA(SET_SCROLL_REGION,1,2);
;  2564          2793  2
;  2565          2794  2 IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
;  2566          2795  2 THEN
;  2567          2796  3     BEGIN       ! Terminal supports scrolling regions
;  2568          2797  3     !+
;  2569          2798  3     ! Changed area must be at least 2 lines high for scrolling to be
;  2570          2799  3     ! useful.
;  2571          2800  3     !-
;  2572          2801  3     IF .PBCB [PBCB_W_LAST_CHANGED_ROW] -
;  2573          2802  3        .PBCB [PBCB_W_FIRST_CHANGED_ROW] GEQ 1
;  2574          2803  3     THEN
;  2575          2804  4         BEGIN   ! Scrolling may be possible
;  2576          2805  4
;  2577          2806  4         SMG$$CHECK_HDWR_SCROLL (.PBCB);
;  2578          2807  4
;  2579          2808  3         END;    ! Scrolling may be possible
;  2580          2809  2     END;        ! Terminal is a VT100
;  2581          2810  2
;  2582          2811  2 RETURN  SMG$$OUTPUT_PASTEBOARD(.PBCB)
;  2583          2812  2
;  2584          2813  1 END,                     ! End of routine SMG$$MIN_UPD


                                    0FFC 00000          .ENTRY  SMG$$MIN_UPD, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 2632
                                                                R10,R11
                                5E        10  C2 00002          SUBL2   #16, SP                               ;
                                56    04  AC  D0 00005          MOVL    PBCB, R6                              ; 2671
                                54    08  A6  D0 00009          MOVL    8(R6), WCB                            ;
                              00A8    C6  9F 0000D              PUSHAB  168(R6)                               ; 2684
                                5B  00AA  C6  9E 00011          MOVAB   170(R6), R11                          ; 2677
                                          6B  B5 00016          TSTW    (R11)                                 ;
                                          7A  13 00018          BEQL    4$                                    ;
                                57    06  A4  3C 0001A          MOVZWL  6(WCB), WIDTH                         ; 2683
                                50    00  BE  32 0001E          CVTWL   @0(SP), R0                            ; 2684
                                          50  D7 00022          DECL    R0                                    ;
                          55            50  57  C5 00024          MULL3   WIDTH, R0, B_OFFSET                   ;
                                59    2C  A4  D0 00028          MOVL    44(WCB), LC                           ; 2685
                                58    30  A4  D0 0002C          MOVL    48(WCB), LCS                          ; 2686
                                5A    00  BE  32 00030 1$:      CVTWL   @0(SP), R10                           ; 2696
                                5A        6B  B1 00034          CMPW    (R11), R10                            ; 2697
                                          25  15 00037          BLEQ    2$                                    ;
                     14 B445        08 B445     57  29 00039          CMPC3   WIDTH, @8(WCB)[B_OFFSET], @20(WCB)-   ; 2706
```

```
                                                      [B_OFFSET]                              ;
                              1B  12 00041       BNEQ   2$                                    ;
 18 B445       0C B445        57  29 00043       CMPC3  WIDTH, @12(WCB)[B_OFFSET], @24(WCB)-  ; 2711
                                                      [B_OFFSET]                              ;
                              11  12 0004B       BNEQ   2$                                    ;
                  6A48      6A49  91 0004D       CMPB   (R10)[LC], (R10)[LCS]                 ; 2717
                              0A  12 00052       BNEQ   2$                                    ;
 00    BE           5A        01  A1 00054       ADDW3  #1, R10, @0(SP)                       ; 2722
                    55        57  C0 00059       ADDL2  WIDTH, B_OFFSET                       ; 2723
                              D2  11 0005C       BRB    1$                                    ; 2706
                    50        6B  32 0005E 2$:   CVTWL  (R11), R0                             ; 2732
                              50  D7 00061       DECL   R0                                    ;
       55           50        57  C5 00063       MULL3  WIDTH, R0, B_OFFSET                   ;
                    5A        6B  32 00067 3$:   CVTWL  (R11), R10                            ; 2735
                    5A    00  BE  B1 0006A       CMPW   @0(SP), R10                           ;
                              2C  18 0006E       BGEQ   5$                                    ;
 14 B445       08 B445        57  29 00070       CMPC3  WIDTH, @8(WCB)[B_OFFSET], @20(WCB)-   ; 2744
                                                      [B_OFFSET]                              ;
                              22  12 00078       BNEQ   5$                                    ;
 18 B445       0C B445        57  29 0007A       CMPC3  WIDTH, @12(WCB)[B_OFFSET], @24(WCB)-  ; 2749
                                                      [B_OFFSET]                              ;
                              18  12 00082       BNEQ   5$                                    ;
                  6A48      6A49  91 00084       CMPB   (R10)[LC], (R10)[LCS]                 ; 2755
                              11  12 00089       BNEQ   5$                                    ;
       6B           5A        01  A3 0008B       SUBW3  #1, R10, (R11)                        ; 2760
                    55        57  C2 0008F       SUBL2  WIDTH, B_OFFSET                       ; 2761
                              D3  11 00092       BRB    3$                                    ; 2744
               00   BE        01  B0 00094 4$:   MOVW   #1, @0(SP)                            ; 2776
                    6B    02  A4  B0 00098       MOVW   2(WCB), (R11)                         ; 2777
                    52  0108  C6  9E 0009C 5$:   MOVAB  264(R6), R2                           ; 2792
                        00FC  C6  D5 000A1       TSTL   252(R6)                               ;
                              04  12 000A5       BNEQ   6$                                    ;
                              62  D4 000A7       CLRL   (R2)                                  ;
                              30  11 000A9       BRB    7$                                    ;
               08   AE        02  D0 000AB 6$:   MOVL   #2, INPUT_ARGS                        ;
               0C   AE        01  D0 000AF       MOVL   #1, INPUT_ARGS+4                      ;
               10   AE        02  D0 000B3       MOVL   #2, INPUT_ARGS+8                      ;
                          08  AE  9F 000B7       PUSHAB INPUT_ARGS                            ;
                        0104  C6  DD 000BA       PUSHL  260(R6)                               ;
                              52  DD 000BE       PUSHL  R2                                    ;
                        0100  C6  9F 000C0       PUSHAB 256(R6)                               ;
               14   AE  023C  8F  3C 000C4       MOVZWL #572, 20(SP)                          ;
                          14  AE  9F 000CA       PUSHAB 20(SP)                                ;
                        00FC  C6  9F 000CD       PUSHAB 252(R6)                               ;
          00000000G 00        06  FB 000D1       CALLS  #6, SMG$GET_TERM_DATA                 ;
                    1F        50  E9 000D8       BLBC   STATUS, 9$                            ;
                              62  D5 000DB 7$:   TSTL   (R2)                                  ; 2794
                              14  13 000DD       BEQL   8$                                    ;
                    50    00  BE  32 000DF       CVTWL  @0(SP), R0                            ; 2802
                              50  D6 000E3       INCL   R0                                    ;
 50         6B      10        00  EC 000E5       CMPV   #0, #16, (R11), R0                    ;
                              07  19 000EA       BLSS   8$                                    ;
                              56  DD 000EC       PUSHL  R6                                    ; 2806
              EAC5  CF        01  FB 000EE       CALLS  #1, SMG$$CHECK_HDWR_SCROLL            ;
                              56  DD 000F3 8$:   PUSHL  R6                                    ; 2811
              0000V CF        01  FB 000F5       CALLS  #1, SMG$$OUTPUT_PASTEBOARD            ;
                                  04 000FA 9$:   RET                                          ; 2813
```

; Routine Size: 251 bytes,   Routine Base: _SMG$CODE + 15C7

```
;  2586          2814  1 %SBTTL 'SMG$$OUTPUT_PASTEBOARD - bring pasteboard up-to-date'
;  2587          2815  1 GLOBAL ROUTINE SMG$$OUTPUT_PASTEBOARD ( P_PBCB ) =
;  2588          2816  1
;  2589          2817  1 !++
;  2590          2818  1 ! FUNCTIONAL DESCRIPTION:
;  2591          2819  1 !
   2592          2820  1 !       Brings the display associated with this pasteboard up-to-date.
;  2593          2821  1 !       It does this by either redrawing it in its entirety,
;  2594          2822  1 !       or by perfomring minimal update if that mode is enabled.
;  2595          2823  1 !
;  2596          2824  1 ! CALLING SEQUENCE:
;  2597          2825  1 !
;  2598          2826  1 !       ret_status.wlc.v = SMG$$OUTPUT_PASTEBOARD ( P_PBCB.rab.r )
;  2599          2827  1 !
;  2600          2828  1 ! FORMAL PARAMETERS:
;  2601          2829  1 !
;  2602          2830  1 !       P_PBCB.rab.r            Address of pasteboard control block.
;  2603          2831  1 !
;  2604          2832  1 ! IMPLICIT INPUTS:
;  2605          2833  1 !
;  2606          2834  1 !       contents of PBCB and its WCB
;  2607          2835  1 !
;  2608          2836  1 ! IMPLICIT OUTPUTS:
;  2609          2837  1 !
;  2610          2838  1 !       NONE
;  2611          2839  1 !
;  2612          2840  1 ! COMPLETION STATUS:
;  2613          2841  1 !
;  2614          2842  1 !       SMG$_BATWAS_ON  OK, but batching was on, so nothing happened
;  2615          2843  1 !       SS$_NORMAL      Normal successful completion
;  2616          2844  1 !
;  2617          2845  1 ! SIDE EFFECTS:
;  2618          2846  1 !
;  2619          2847  1 !       NONE
;  2620          2848  1 !--
```

```
;  2622          2849  2 BEGIN
;  2623          2850  2
;  2624          2851  2 BIND
;  2625          2852  2
;  2626          2853  2         PBCB            = .P_PBCB                : $PBCB_DECL,
;  2627          2854  2         WCB             = .PBCB[PBCB_A_WCB]      : $WCB_DECL,
;  2628          2855  2         TEXT_BUF        = .WCB[WCB_A_TEXT_BUF]   : VECTOR[,BYTE],
;  2629          2856  2         ATTR_BUF        = .WCB[WCB_A_ATTR_BUF]   : VECTOR[,BYTE],
;  2630          2857  2         ROWS            =  WCB[WCB_W_NO_ROWS]    : WORD,
;  2631          2858  2         COLS            =  WCB[WCB_W_NO_COLS]    : WORD;
;  2632          2859  2
;  2633          2860  2 LOCAL
;  2634          2861  2
;  2635          2862  2         STATUS,
;  2636          2863  2         SIZE;
;  2637          2864  2
;  2638          2865  2 EXTERNAL LITERAL
;  2639          2866  2
;  2640          2867  2         SMG$_BATWAS_ON;
```

```
;  2642          2868  2 !+
;  2643          2869  2 ! Do nothing if the output is being controlled by RMS.
;  2644          2870  2 !-
;  2645          2871  2
;  2646          2872  2 IF .PBCB[PBCB_V_RMS]
;  2647          2873  2   THEN  RETURN  SMG$_WILUSERMS;
;  2648          2874  2
;  2649          2875  2 !+
;  2650          2876  2 ! Do nothing if batching is in effect.
;  2651          2877  2 !-
;  2652          2878  2
;  2653          2879  2 IF .PBCB[PBCB_L_BATCH_LEVEL] NEQ 0
;  2654          2880  2   THEN  RETURN  SMG$_BATWAS_ON;
;  2655          2881  2
;  2656          2882  2 SIZE = .ROWS * .COLS;
;  2657          2883  2
;  2658          2884  2 !+
;  2659          2885  2 ! If minimal updating is in effect, then call
;  2660          2886  2 ! SMG$$OUTPUT_MINIMAL_UPDATE to output a minimal update sequence.
;  2661          2887  2 ! Then return.
;  2662          2888  2 !
;  2663          2889  2
;  2664          2890  2 IF .PBCB[PBCB_V_MINUPD]
;  2665          2891  2   THEN  RETURN SMG$$OUTPUT_MINIMAL_UPDATE(PBCB);
;  2666          2892  2
;  2667          2893  2 !+
;  2668          2894  2 ! Otherwise, do nothing (for now).
;  2669          2895  2 !-
;  2670          2896  2
;  2671          2897  2 RETURN  SS$_NORMAL
;  2672          2898  2
;  2673          2899  1 END;                       ! End of routine SMG$$OUTPUT_PASTEBOARD


                                                                .EXTRN  SMG$_BATWAS_ON

                                                    000C 00000          .ENTRY  SMG$$OUTPUT_PASTEBOARD, Save R2,R3      ;  2815
                                          50      04   AC   D0 00002    MOVL    P_PBCB, R0                              ;  2853
                                    52    08   A0        02   C1 00006    ADDL3   #2, 8(R0), R2                           ;  2857
                                    51    08   A0        06   C1 00008    ADDL3   #6, 8(R0), R1                           ;  2858
                                    08  00D0   C0        03   E1 00010    BBC     #3, 208(R0), 1$                         ;  2872
                                          50 00000000G  8F   D0 00016    MOVL    #SMG$_WILUSERMS, R0                     ;  2873
                                                              04 0001D    RET                                             ;
                                               00A4     C0   D5 0001E 1$:  TSTL    164(R0)                                 ;  2879
                                                        08   13 00022    BEQL    2$                                      ;
                                          50 00000000G  8F   D0 00024    MOVL    #SMG$_BATWAS_ON, R0                     ;  2880
                                                              04 0002B    RET                                             ;
                                          53            62   3C 0002C 2$:  MOVZWL  (R2), R3                               ;  2882
                                          51            61   3C 0002F    MOVZWL  (R1), SIZE                              ;
                                          51            53   C4 00032    MULL2   R3, SIZE                                ;
                                    0A    0C  A0        01   E1 00035    BBC     #1, 12(R0), 3$                          ;  2890
                                          50  DD        00   3A          PUSHL   R0                                      ;  2891
                                 00000000G    00        01   FB 0003C    CALLS   #1, SMG$$OUTPUT_MINIMAL_UPDATE          ;
                                                              04 00043    RET                                             ;
                                          50            01   D0 00044 3$:  MOVL    #1, R0                                  ;  2897
                                                              04 00047    RET                                             ;  2899
```

; Routine Size: 72 bytes,   Routine Base: _SMG$CODE + 16C2

```
;  2675          2900  1 %SBTTL 'SMG$$PUT_SCREEN - Output to screen'
;  2676          2901  1 GLOBAL ROUTINE SMG$$PUT_SCREEN (
;  2677          2902  1                  P_PBCB,
;  2678          2903  1                  TEXT_LEN,
;  2679          2904  1                  TEXT_ADR,
;  2680          2905  1                  ROW_NUM,
;  2681          2906  1                  COL_NUM,
;  2682          2907  1                  FLAGS : BITVECTOR[32] ) =
;  2683          2908  1 !++
;  2684          2909  1 ! FUNCTIONAL DESCRIPTION:
;  2685          2910  1 !
;  2686          2911  1 !       Logically outputs a string to the screen using calls
;  2687          2912  1 !       to SMG$$OUTPUT.  The text may be accomapnied by
;  2688          2913  1 !       renditions and cursor positioning.
;  2689          2914  1 !       Note that the output sequences generated may get
;  2690          2915  1 !       buffered up by SMG$$OUTPUT if buffering is enabled.
;  2691          2916  1 !
;  2692          2917  1 ! CALLING SEQUENCE:
;  2693          2918  1 !
;  2694          2919  1 !       ret_status.wlc.v = SMG$$PUT_SCREEN(
;  2695          2920  1 !                                          P_PBCB.rab.r,
;  2696          2921  1 !                                          TEXT_LEN.rl.v,
;  2697          2922  1 !                                          TEXT_ADR.rt.r
;  2698          2923  1 !                                          ROW_NUM.rl.v,
;  2699          2924  1 !                                          COL_NUM.rl.v
;  2700          2925  1 !                                          [,FLAGS.rl.v])
;  2701          2926  1 !
;  2702          2927  1 ! FORMAL PARAMETERS:
;  2703          2928  1 !
;  2704          2929  1 !       P_PBCB.rab.r            Address of pasteboard control block.
;  2705          2930  1 !
;  2706          2931  1 !       TEXT_LEN.rl.v           Number of characters in text string
;  2707          2932  1 !
;  2708          2933  1 !       TEXT_ADR.rt.r           Address of start of text string
;  2709          2934  1 !
;  2710          2935  1 !       ROW_NUM.rl.v            Row number
;  2711          2936  1 !
;  2712          2937  1 !       COL_NUM.rl.v            Column number
;  2713          2938  1 !
;  2714          2939  1 !       FLAGS.rl.v              Rendition codes. (bit encoded)
;001 STAN1046    2940  1 !                               Optional.  If omitted, normal rendition
;  2716-1        2941  1 !                               occurs.
;  2717          2942  1 !
;  2718          2943  1 ! IMPLICIT INPUTS:
;  2719          2944  1 !
;  2720          2945  1 !       NONE
;  2721          2946  1 !
;  2722          2947  1 ! IMPLICIT OUTPUTS:
;  2723          2948  1 !
;  2724          2949  1 !       NONE
;  2725          2950  1 !
;  2726          2951  1 ! COMPLETION STATUS:
;  2727          2952  1 !
;  2728          2953  1 !       SS$_NORMAL      Normal successful completion
;  2729          2954  1 !
;  2730          2955  1 ! SIDE EFFECTS:
;  2731          2956  1 !
```

```
; 2732           2957  1 !       NONE
; 2733           2958  1 !--
```

```
;  2735          2959  2 BEGIN
;  2736          2960  2
;  2737          2961  2 BIND
;  2738          2962  2
;  2739          2963  2         PBCB    = .P_PBCB       : $PBCB_DECL;   ! Pasteboard control block
;  2740          2964  2
;  2741          2965  2 LOCAL
;  2742          2966  2
;001 STAN1046    2967  2         RENDITION_CHANGED,      ! TRUE if we changed rendition
;  2743          2968  2         STATUS;
;  2744          2969  2
;  2745          2970  2 OWN
;  2746          2971  2
;  2747          2972  2         TRANSLATED_TEXT_DESC    : BLOCK[8,BYTE] ! reusable dynamic descriptor
;  2748          2973  2                                 ! must be OWN storage
;  2749          2974  2         PRESET( [DSC$B_DTYPE]   = DSC$K_DTYPE_T,
;  2750          2975  2                 [DSC$B_CLASS]   = DSC$K_CLASS_D,
;  2751          2976  2                 [DSC$W_LENGTH]  = 0,
;  2752          2977  2                 [DSC$A_POINTER] = 0);
;  2753          2978  2
;  2754          2979  2 BUILTIN
;  2755          2980  2
;  2756          2981  2         ACTUALCOUNT;
```

```
; 2758          2982  2 !+
; 2759          2983  2 ! Do nothing if the output is being controlled by RMS.
; 2760          2984  2 !-
; 2761          2985  2
; 2762          2986  2 IF .PBCB[PBCB_V_RMS]
; 2763          2987  2   THEN  RETURN SMG$_WILUSERMS;
; 2764          2988  2
; 2765          2989  2 !+
; 2766          2990  2 ! If a rendition was specified, then output it now.
; 2767          2991  2 !-
; 2768          2992  2
; 2769          2993  2 IF ACTUALCOUNT() GEQU 6
; 2770          2994  3   THEN  BEGIN   ! output text and renditions and cursor positioning
; 2771          2995  3
; 2772          2996  3         BIND    TT2 = PBCB[PBCB_L_DEVDEPEND2]   : $BBLOCK;
; 2773          2997  3
; 2774          2998  3         !+
;001 STAN1046  2999  3         ! Note that renditions haven't changed yet.
;002 STAN1046  3000  3         ! [Set to 1 if they have changed.]
;003 STAN1046  3001  3         !-
;004 STAN1046  3002  3
;005 STAN1046  3003  3         RENDITION_CHANGED=0;
;006 STAN1046  3004  3
;007 STAN1046  3005  3         !+
; 2775          3006  3         ! Renditions requires that the AVO (ADVANCED_VIDEO) terminal
; 2776          3007  3         ! characteristic bit be set.  Even if the TERMTABLE entries
; 2777          3008  3         ! show that the terminal has the BEGIN_BOLD capability,
; 2778          3009  3         ! the terminal might not have the advanced video option.
; 2779          3010  3         !-
; 2780          3011  3
; 2781          3012  3         IF .FLAGS[ATTR_V_REND_GRAPHIC]
; 2782          3013  4         AND (.TT2[TT2$V_AVO] OR NOT .TT2[TT2$V_ANSICRT])
; 2783          3014  4           THEN  BEGIN
; 2784          3015  4                 $SMG$GET_TERM_DATA(BEGIN_LINE_DRAWING_CHAR);
; 2785          3016  4
; 2786          3017  4                 IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
; 2787          3018  5                   THEN  BEGIN
; 2788          3019  5                         STATUS=SMG$$OUTPUT(PBCB,.PBCB[PBCB_L_CAP_LENGTH],
; 2789          3020  5                                            .PBCB[PBCB_A_CAP_BUFFER]);
; 2790          3021  5                         IF NOT .STATUS THEN RETURN .STATUS
; 2791          3022  4                         END;
; 2792          3023  3                 END;
; 2793          3024  3
; 2794          3025  3         !+
; 2795          3026  3         ! Get and output the string to set the correct attributes.
; 2796          3027  3         !-
; 2797          3028  3
; 2798          3029  3         IF .FLAGS[ATTR_V_REND_BOLD]
; 2799          3030  4         AND (.TT2[TT2$V_AVO] OR NOT .TT2[TT2$V_ANSICRT])
; 2800          3031  4           THEN  BEGIN
; 2801          3032  4                 $SMG$GET_TERM_DATA(BEGIN_BOLD);
; 2802          3033  4
; 2803          3034  4                 IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
; 2804          3035  5                   THEN  BEGIN
; 2805          3036  5                         STATUS=SMG$$OUTPUT(PBCB,.PBCB[PBCB_L_CAP_LENGTH],
; 2806          3037  5                                            .PBCB[PBCB_A_CAP_BUFFER]);
; 2807          3038  5                         IF NOT .STATUS THEN RETURN .STATUS;
```

```
;001 STAN1046    3039  5                          RENDITION_CHANGED=1
; 2808           3040  4                          END;
; 2809           3041  3                  END;
; 2810           3042  3
; 2811           3043  3          IF .FLAGS[ATTR_V_REND_BLINK]
; 2812           3044  4          AND (.TT2[TT2$V_AVO] OR NOT .TT2[TT2$V_ANSICRT])
; 2813           3045  4            THEN  BEGIN
; 2814           3046  4                  $SMG$GET_TERM_DATA(BEGIN_BLINK);
; 2815           3047  4
; 2816           3048  4                  IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
; 2817           3049  5                    THEN  BEGIN
; 2818           3050  5                          STATUS=SMG$$OUTPUT(PBCB,.PBCB[PBCB_L_CAP_LENGTH],
; 2819           3051  5                                             .PBCB[PBCB_A_CAP_BUFFER]);
; 2820           3052  5                          IF NOT .STATUS THEN RETURN .STATUS;
;001 STAN1046    3053  5                          RENDITION_CHANGED=1
; 2821           3054  4                          END;
; 2822           3055  3                  END;
; 2823           3056  3
; 2824           3057  3          IF .FLAGS[ATTR_V_REND_REV]
; 2825           3058  4          AND (.TT2[TT2$V_AVO] OR NOT .TT2[TT2$V_ANSICRT])
; 2826           3059  4            THEN  BEGIN
; 2827           3060  4                  $SMG$GET_TERM_DATA(BEGIN_REVERSE);
; 2828           3061  4
; 2829           3062  4                  IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
; 2830           3063  5                    THEN  BEGIN
; 2831           3064  5                          STATUS=SMG$$OUTPUT(PBCB,.PBCB[PBCB_L_CAP_LENGTH],
; 2832           3065  5                                             .PBCB[PBCB_A_CAP_BUFFER]);
; 2833           3066  5                          IF NOT .STATUS THEN RETURN .STATUS;
;001 STAN1046    3067  5                          RENDITION_CHANGED=1
; 2834           3068  4                          END;
; 2835           3069  3                  END;
; 2836           3070  3
; 2837           3071  3          IF .FLAGS[ATTR_V_REND_UNDER]
; 2838           3072  4          AND (.TT2[TT2$V_AVO] OR NOT .TT2[TT2$V_ANSICRT])
; 2839           3073  4            THEN  BEGIN
; 2840           3074  4                  $SMG$GET_TERM_DATA(BEGIN_UNDERSCORE);
; 2841           3075  4
; 2842           3076  4                  IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
; 2843           3077  5                    THEN  BEGIN
; 2844           3078  5                          STATUS=SMG$$OUTPUT(PBCB,.PBCB[PBCB_L_CAP_LENGTH],
; 2845           3079  5                                             .PBCB[PBCB_A_CAP_BUFFER]);
; 2846           3080  5                          IF NOT .STATUS THEN RETURN .STATUS;
;001 STAN1046    3081  5                          RENDITION_CHANGED=1
; 2847           3082  4                          END;
; 2848           3083  3                  END;
; 2849           3084  3
; 2850           3085  3          !+
; 2851           3086  3          ! Output the text if they are not border elements.
; 2852           3087  3          ! Output translated text for border elements.
; 2853           3088  3          !-
; 2854           3089  3
; 2855           3090  3          IF .FLAGS[ATTR_V_BORD_ELEM] OR .FLAGS[ATTR_V_USER_GRAPHIC]
; 2856           3091  4            THEN  BEGIN   ! handing border element
; 2857           3092  4
; 2858           3093  4                  LOCAL
; 2859           3094  4
; 2860           3095  4                          TEXT_DESC       : VECTOR[2];
```

```
;  2861          3096  4                    EXTERNAL ROUTINE
;  2862          3097  4                            LIB$SCOPY_DXDX;
;  2863          3098  4
;  2864          3099  4                    !+
;  2865          3100  4                    ! Build a fixed-length string descriptor for the
;  2866          3101  4                    ! source text.
;  2867          3102  4                    !-
;  2868          3103  4
;  2869          3104  4                    TEXT_DESC[0]=.TEXT_LEN;
;  2870          3105  4                    TEXT_DESC[1]=.TEXT_ADR;
;  2871          3106  4
;  2872          3107  4                    !+
;  2873          3108  4                    ! Get and output the prefix string to start borders.
;  2874          3109  4                    !-
;  2875          3110  4
;  2876          3111  4                    IF .TT2[TT2$V_AVO] OR NOT .TT2[TT2$V_ANSICRT]
;  2877          3112  4                      THEN  BEGIN
;  2878          3113  4                            $SMG$GET_TERM_DATA(BEGIN_LINE_DRAWING_CHAR);
;  2879          3114  5
;  2880          3115  5                            IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
;  2881          3116  5                              THEN  BEGIN
;  2882          3117  5                                    STATUS=SMG$$OUTPUT(PBCB,.PBCB[PBCB_L_CAP_LENGTH],
;  2883          3118  6                                                       .PBCB[PBCB_A_CAP_BUFFER]);
;  2884          3119  6                                    IF NOT .STATUS THEN RETURN .STATUS;
;  2885          3120  6                                    END;
;  2886          3121  6                            END;
;  2887          3122  5
;  2888          3123  4                    !+
;  2889          3124  4                    ! If the COMPLEX_BORDER bit is set, then we have to output
;  2890          3125  4                    ! the border characters one at a time.
;  2891          3126  4                    ! Otherwise, we can do a byte for byte translation.
;  2892          3127  4                    !-
;  2893          3128  4
;  2894          3129  4                    IF .PBCB[PBCB_V_COMPLEX_BORDER]
;  2895          3130  4                      THEN  BEGIN   ! Complex border
;  2896          3131  4
;  2897          3132  5                            INCR I FROM 0 TO .TRANSLATED_TEXT_DESC[DSC$W_LENGTH]-1 DO
;  2898          3133  5                                    BEGIN
;  2899          3134  5                                    LOCAL CHAR;
;  2900          3135  6                                    BIND    BUF = .TRANSLATED_TEXT_DESC[DSC$A_POINTER]
;  2901          3136  6                                            : VECTOR[,BYTE];
;  2902          3137  6                                    BIND    BORDER_VECTOR = PBCB[PBCB_R_BORDER_VECTOR]
;  2903          3138  6                                            : VECTOR[16];
;  2904          3139  6                                    CHAR=.BUF[.I];
;  2905          3140  6                                    IF .CHAR GEQU 16
;  2906          3141  6                                      THEN  BEGIN
;  2907          3142  6                                            CHAR=.CHAR/16;
;  2908          3143  7                                            !+
;  2909          3144  7                                            ! The following code is ridiculous.
;  2910          3145  7                                            ! We should really have a control_vector in the PBCB
;  2911          3146  7                                            ! so that these characters will be gotton just once.
;  2912          3147  7                                            ! However, we had no chance to do this before final code freeze.
;  2913          3148  7                                            !-
;  2914          3149  7
;  2915          3150  7                                            SELECTONE .CHAR OF
;  2916          3151  7
;  2917          3152  7
```

```
;  2918          3153  7
;  2919          3154  7                                                    SET
;  2920          3155  7
;  2921          3156  7                                                    [6]:    $SMG$GET_TERM_DATA(TRUNCATION_ICON);
;  2922          3157  7                                                    [9]:    $SMG$GET_TERM_DATA(HT_GRAPHIC);
;  2923          3158  7                                                    [10]:   $SMG$GET_TERM_DATA(LF_GRAPHIC);
;  2924          3159  7                                                    [11]:   $SMG$GET_TERM_DATA(VT_GRAPHIC);
;  2925          3160  7                                                    [12]:   $SMG$GET_TERM_DATA(FF_GRAPHIC);
;  2926          3161  7                                                    [13]:   $SMG$GET_TERM_DATA(CR_GRAPHIC);
;  2927          3162  8                                                    [OTHERWISE]:    $SMG$GET_TERM_DATA(TRUNCATION_ICON) ! Error characte
;  2928          3163  8
;  2929          3164  7                                                    TES;
;  2930          3165  7
;  2931          3166  7                                                    IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
;  2932          3167  8                                                      THEN  BEGIN
;  2933          3168  8                                                            STATUS=SMG$$OUTPUT(PBCB,.PBCB[PBCB_L_CAP_LENGTH],
;  2934          3169  8                                                                               .PBCB[PBCB_A_CAP_BUFFER]);
;  2935          3170  8                                                            IF NOT .STATUS THEN RETURN .STATUS
;  2936          3171  8                                                            END
;  2937          3172  8                                                      ELSE  BEGIN
;  2938          3173  8                                                            STATUS=SMG$$OUTPUT(PBCB,1,UPLIT BYTE('*'));
;  2939          3174  8                                                            IF NOT .STATUS THEN RETURN .STATUS
;  2940          3175  7                                                            END;
;  2941          3176  7
;  2942          3177  7                                                    END
;  2943          3178  7                                                ELSE  BEGIN
;  2944          3179  7                                                    BIND COUNT=.BORDER_VECTOR[.CHAR] : BYTE,
;  2945          3180  7                                                         STRING=COUNT+1;
;  2946          3181  7                                                    STATUS=SMG$$OUTPUT(PBCB,,COUNT,STRING);
;  2947          3182  7                                                    IF NOT .STATUS THEN RETURN .STATUS
;  2948          3183  7                                                    END
;  2949          3184  5                                            END;
;  2950          3185  5
;  2951          3186  5
;  2952          3187  5                                                END     ! Complex border
;  2953          3188  5                                            ELSE  BEGIN   ! Simple border
;  2954          3189  5
;  2955          3190  5                                                    !+
;  2956          3191  5                                                    ! Copy the input string to the output string.
;  2957          3192  5                                                    !-
;  2958          3193  5
;  2959          3194  5                                                    STATUS=LIB$SCOPY_DXDX(TEXT_DESC,TRANSLATED_TEXT_DESC);
;  2960          3195  5                                                    IF NOT .STATUS THEN RETURN .STATUS;
;  2961          3196  5
;  2962          3197  5                                                    !+
;  2963          3198  5                                                    ! Change the characters as per the border vector.
;  2964          3199  5                                                    !-
;  2965          3200  5
;  2966          3201  5                                                    INCR I FROM 0 TO .TRANSLATED_TEXT_DESC[DSC$W_LENGTH]-1 DO
;  2967          3202  6                                                            BEGIN
;  2968          3203  6                                                            LOCAL CHAR;
;  2969          3204  6                                                            BIND    BUF = .TRANSLATED_TEXT_DESC[DSC$A_POINTER]
;  2970          3205  6                                                                    : VECTOR[,BYTE];
;  2971          3206  6                                                             IND    BORDER_VECTOR = PBCB[PBCB_A_BORDER_VECTOR]
;  2972          3207  6                                                                    : VECTOR[16];
;  2973          3208  6                                                            !+
;  2974          3209  6                                                            ! If the character is larger than 15, then
```

```
;  2975          3210  6                                    ! the high-order nibble is a special
;  2976          3211  6                                    ! user-graphic character.  We could allow
;  2977          3212  6                                    ! for up to 15 characters here, things
;  2978          3213  6                                    ! like CR_GRAPHIC, etc., but for now
;  2979          3214  6                                    ! we only allow code 6 to mean truncation-icon.
;  2980          3215  6                                    ! Other codes represent the error character.
;  2981          3216  6                                    !-
;  2982          3217  6                                    CHAR=.BUF[.I];
;  2983          3218  6                                    IF .CHAR GEQU 16
;  2984          3219  7                                      THEN  BEGIN
;  2985          3220  7                                            CHAR=.CHAR/16;
;  2986          3221  7
;  2987          3222  7                                            !+
;  2988          3223  7                                            ! The following code is ridiculous.
;  2989          3224  7                                            ! We should really have a control_vector in the PBCB
;  2990          3225  7                                            ! so that these characters will be gotton just once.
;  2991          3226  7                                            ! However, we had no chance to do this before final code freeze.
;  2992          3227  7                                            !-
;  2993          3228  7
;  2994          3229  7                                            SELECTONE .CHAR OF
;  2995          3230  7
;  2996          3231  7                                            SET
;  2997          3232  7
;  2998          3233  7                                            [6]:    $SMG$GET_TERM_DATA(TRUNCATION_ICON);
;  2999          3234  7                                            [9]:    $SMG$GET_TERM_DATA(HT_GRAPHIC);
;  3000          3235  7                                            [10]:   $SMG$GET_TERM_DATA(LF_GRAPHIC);
;  3001          3236  7                                            [11]:   $SMG$GET_TERM_DATA(VT_GRAPHIC);
;  3002          3237  7                                            [12]:   $SMG$GET_TERM_DATA(FF_GRAPHIC);
;  3003          3238  7                                            [13]:   $SMG$GET_TERM_DATA(CR_GRAPHIC);
;  3004          3239  8                                            [OTHERWISE]:    $SMG$GET_TERM_DATA(TRUNCATION_ICON) ! Error characte
;  3005          3240  8
;  3006          3241  7                                            TES;
;  3007          3242  7
;  3008          3243  7                                            IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
;  3009          3244  8                                              THEN  BEGIN
;  3010          3245  8                                                    BIND CHAR=.PBCB[PBCB_A_CAP_BUFFER] : BYTE;
;  3011          3246  8                                                    BUF[.I]=.CHAR
;  3012          3247  8                                                    END
;  3013          3248  8                                              ELSE  BEGIN
;  3014          3249  8                                                    BUF[.I]=%C'*'
;  3015          3250  7                                                    END;
;  3016          3251  7
;  3017          3252  7                                            END
;  3018          3253  6                                      ELSE  BUF[.I]=.BORDER_VECTOR[.CHAR]
;  3019          3254  5                                    END;
;  3020          3255  5
;  3021          3256  5                                !+
;  3022          3257  5                                ! Output the translated characters.
;  3023          3258  5                                ! We do not free the dynamic string at this time
;  3024          3259  5                                ! as an optimization.  We are sure to need that
;  3025          3260  5                                ! space later.
;  3026          3261  5                                !-
;  3027          3262  5
;  3028          3263  5                                STATUS=SMG$$OUTPUT(PBCB,
;  3029          3264  5                                                   .TRANSLATED_TEXT_DESC[DSC$W_LENGTH],
;  3030          3265  5                                                   .TRANSLATED_TEXT_DESC[DSC$A_POINTER]);
;  3031          3266  5                                IF NOT .STATUS THEN RETURN .STATUS;
```

```
;  3032          3267  5
;  3033          3268  4                           END;    ! Simple border
;  3034          3269  4
;  3035          3270  4                   !+
;  3036          3271  4                   ! Get and output the suffix string to reset attributes to normal.
;  3037          3272  4                   !-
;  3038          3273  4
;  3039          3274  4                   IF .TT2[TT2$V_AVO] OR NOT .TT2[TT2$V_ANSICRT]
;  3040          3275  5                     THEN  BEGIN
;  3041          3276  5                           $SMG$GET_TERM_DATA(END_LINE_DRAWING_CHAR);
;  3042          3277  5
;  3043          3278  5                           IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
;  3044          3279  6                             THEN  BEGIN
;  3045          3280  6                                   STATUS=SMG$$OUTPUT(PBCB,.PBCB[PBCB_L_CAP_LENGTH],
;  3046          3281  6                                                      .PBCB[PBCB_A_CAP_BUFFER]);
;  3047          3282  6                                   IF NOT .STATUS THEN RETURN .STATUS;
;  3048          3283  5                                   END;
;  3049          3284  4                           END;
;  3050          3285  4
;  3051          3286  4                   END     ! handling border element
;  3052          3287  4             ELSE  BEGIN   ! handling normal text
;  3053          3288  4                   STATUS=SMG$$OUTPUT(PBCB,.TEXT_LEN,.TEXT_ADR);
;  3054          3289  4                   IF NOT .STATUS THEN RETURN .STATUS;
;  3055          3290  3                   END;    ! handling normal text
;  3056          3291  3
;  3057          3292  3           !+
;  3058          3293  3           ! Get and output the suffix string to reset attributes to normal.
;  3059          3294  3           ! We used to assume that END_BOLD brings back normal attributes.
;  3060          3295  3           ! Now we rely on BEGIN_NORMAL_RENDITION.
;001 STAN1046    3296  3           ! Only reset to normal if we had changed to some other rendition.
;002 STAN1046    3297  3           !-
;003 STAN1046    3298  3
;004 STAN1046    3299  4           IF .RENDITION_CHANGED AND (.TT2[TT2$V_AVO] OR NOT .TT2[TT2$V_ANSICRT])
;  3064-3        3300  4             THEN  BEGIN
;  3065          3301  4
;  3066        L 3302  4                   %IF %DECLARED( SMG$K_BEGIN_NORMAL_RENDITION )
;  3067          3303  4                   %THEN
;  3068          3304  4                         $SMG$GET_TERM_DATA(BEGIN_NORMAL_RENDITION);
;  3069        U 3305  4                   %ELSE
;  3070        U 3306  4                         $SMG$GET_TERM_DATA(END_BOLD);
;  3071          3307  4                   %FI
;  3072          3308  4
;  3073          3309  4                   IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
;  3074          3310  5                     THEN  BEGIN
;  3075          3311  5                           STATUS=SMG$$OUTPUT(PBCB,.PBCB[PBCB_L_CAP_LENGTH],
;  3076          3312  5                                              .PBCB[PBCB_A_CAP_BUFFER]);
;  3077          3313  5                           IF NOT .STATUS THEN RETURN .STATUS;
;  3078          3314  4                           END;
;  3079          3315  3                   END;
;  3080          3316  3
;  3081          3317  3           IF .FLAGS[ATTR_V_REND_GRAPHIC]
;  3082          3318  4           AND (.TT2[TT2$V_AVO] OR NOT .TT2[TT2$V_ANSICRT])
;  3083          3319  4             THEN  BEGIN
;  3084          3320  4                   $SMG$GET_TERM_DATA(END_LINE_DRAWING_CHAR);
;  3085          3321  4
;  3086          3322  4                   IF .PBCB[PBCB_L_CAP_LENGTH] NEQ 0
;  3087          3323  5                     THEN  BEGIN
```

```
;  3088          3324  5                        STATUS=SMG$$OUTPUT(PBCB,.PBCB[PBCB_L_CAP_LENGTH],
;  3089          3325  5                                          .PBCB[PBCB_A_CAP_BUFFER]);
;  3090          3326  5                        IF NOT .STATUS THEN RETURN .STATUS
;  3091          3327  4                        END;
;  3092          3328  3                    END;
;  3093          3329  3
;  3094          3330  3            END     ! output text and renditions
;  3095          3331  3      ELSE  BEGIN   ! output text only
;  3096          3332  3
;  3097          3333  3            STATUS=SMG$$OUTPUT(PBCB,.TEXT_LEN,.TEXT_ADR);
;  3098          3334  3            IF NOT .STATUS THEN RETURN .STATUS
;  3099          3335  3
;  3100          3336  2            END;    ! output text only
;  3101          3337  2
;  3102          3338  2 RETURN SS$_NORMAL
;  3103          3339  2
;  3104          3340  1 END;     ! routine SMG$$PUT_SCREEN


                                                        .PSECT  _SMG$DATA,NOEXE,  PIC,2

                                              0000  00044 TRANSLATED_TEXT_DESC:
                                                        .WORD   0                                                       ;
                                             02  0E  00046                                                           ;
                                                        .BYTE   14, 2                                                   ;
                                         00000000  00048                                                            ;
                                                        .LONG   0                                                       ;

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

                                                2A  0170A P.AAE:  .ASCII  \*\                                           ;

                                                        .EXTRN  LIB$SCOPY_DXDX

                                                        .ENTRY  SMG$$PUT_SCREEN, Save R2,R3,R4,R5,R6,R7,R8,-            ; 2901
                                                                R9,R10,R11                                              ;
                                    5B      0000V   CF  9E 00002                MOVAB   SMG$$OUTPUT, R11                ;
                                    5A  00000000'   EF  9E 00007                MOVAB   TRANSLATED_TEXT_DESC, R10       ;
                                    59  00000000G   00  9E 0000E                MOVAB   SMG$GET_TERM_DATA, R9           ;
                                    5E              18  C2 00015                SUBL2   #24, SP                         ;
                                    52        04    AC  D0 00018                MOVL    P_PBCB, R2                      ; 2963
                              08    00D0    C2          03  E1 0001C            BBC     #3, 208(R2), 1$                 ; 2986
                                    50  00000000G   8F  D0 00022                MOVL    #SMG$_WILUSERMS, R0             ; 2987
                                                        04 00029                RET                                     ;
                                    06              6C  91 0002A 1$:            CMPB    (AP), #6                        ; 2993
                                                    03  1E 0002D                BGEQU   2$                              ;
                                                05EA    31 0002F                BRW     102$                            ;
                                    54        60    A2  9E 00032 2$:            MOVAB   96(R2), R4                      ; 2996
                                                    57  D4 00036                CLRL    RENDITION_CHANGED               ; 3003
                              54        18    AC          04  E1 00038          BBC     #4, FLAGS, 6$                   ; 3012
                              04              64          1B  E0 0003D          BBS     #27, (R4), 3$                   ; 3013
                                    4C        03    A4  E8 00041                BLBS    3(R4), 6$                       ;
                                          00FC    C2  D5 00045 3$:            TSTL    252(R2)                         ; 3015
                                                    09  12 00049                BNEQ    4$                              ;
                                    53      0108    C2  9E 0004B                MOVAB   264(R2), R3                     ;
                                                    63  D4 00050                CLRL    (R3)                            ;
                                                    28  11 00052                BRB     5$                              ;
                                          0C    AE  D4 00054 4$:            CLRL    INPUT_ARGS                      ;
```

```
                                    0C   AE  9F 00057          PUSHAB  INPUT_ARGS                                    ;
                                  0104   C2  DD 0005A          PUSHL   260(R2)                                      ;
                                    53   0108  C2 9E 0005E     MOVAB   264(R2), R3                                  ;
                                                53 DD 00063     PUSHL   R3                                           ;
                                  0100   C2  9F 00065          PUSHAB  256(R2)                                      ;
                                    10   AE  01BE 8F 3C 00069 MOVZWL  #446, 16(SP)                                 ;
                                    10   AE  9F 0006F          PUSHAB  16(SP)                                       ;
                                  00FC   C2  9F 00072          PUSHAB  252(R2)                                      ;
                                    69       06 FB 00076      CALLS   #6, SMG$GET_TERM_DATA                        ;
                                    55       50 E9 00079      BLBC    STATUS, 9$                                   ;
                                             63 D5 0007C 5$:  TSTL    (R3)                                         ; 3017
                                             11 13 0007E      BEQL    6$                                           ;
                                  0104   C2  DD 00080          PUSHL   260(R2)                                      ; 3020
                                             63 DD 00084      PUSHL   (R3)                                         ; 3019
                                             52 DD 00086      PUSHL   R2                                           ;
                                    6B       03 FB 00088      CALLS   #3, SMG$$OUTPUT                              ;
                                    56       50 D0 0008B      MOVL    R0, STATUS                                   ;
                                    55       56 E9 0008E      BLBC    STATUS, 11$                                  ; 3021
                                    57   18  AC E9 00091 6$:   BLBC    FLAGS, 12$                                   ; 3029
                                04  64       1B E0 00095      BBS     #27, (R4), 7$                                ; 3030
                                    4F   03  A4 E8 00099      BLBS    3(R4), 12$                                   ;
                                  00FC   C2  D5 0009D 7$:     TSTL    252(R2)                                      ; 3032
                                             09 12 000A1      BNEQ    8$                                           ;
                                    53   0108  C2 9E 000A3     MOVAB   264(R2), R3                                  ;
                                             63 D4 000A8      CLRL    (R3)                                         ;
                                             28 11 000AA      BRB     10$                                          ;
                                    0C   AE  D4 000AC 8$:     CLRL    INPUT_ARGS                                   ;
                                    0C   AE  9F 000AF         PUSHAB  INPUT_ARGS                                   ;
                                  0104   C2  DD 000B2         PUSHL   260(R2)                                      ;
                                    53   0108  C2 9E 000B6     MOVAB   264(R2), R3                                  ;
                                                53 DD 000BB     PUSHL   R3                                           ;
                                  0100   C2  9F 000BD         PUSHAB  256(R2)                                      ;
                                10  AE   01BB 8F 3C 000C1     MOVZWL  #443, 16(SP)                                 ;
                                    10   AE  9F 000C7         PUSHAB  16(SP)                                       ;
                                  00FC   C2  9F 000CA         PUSHAB  252(R2)                                      ;
                                    69       06 FB 000CE      CALLS   #6, SMG$GET_TERM_DATA                        ;
                                    59       50 E9 000D1 9$:  BLBC    STATUS, 15$                                  ;
                                             63 D5 000D4 10$: TSTL    (R3)                                         ; 3034
                                             14 13 000D6      BEQL    12$                                          ;
                                  0104   C2  DD 000D8         PUSHL   260(R2)                                      ; 3037
                                             63 DD 000DC      PUSHL   (R3)                                         ; 3036
                                             52 DD 000DE      PUSHL   R2                                           ;
                                    6B       03 FB 000E0      CALLS   #3, SMG$$OUTPUT                              ;
                                    56       50 D0 000E3      MOVL    R0, STATUS                                   ;
                                    59       56 E9 000E6 11$: BLBC    STATUS, 17$                                  ; 303P
                                    57       01 D0 000E9      MOVL    #1, RENDITION_CHANGED                        ; 3039
                                57  18   AC  02 E1 000EC 12$: BBC     #2, FLAGS, 18$                               ; 3043
                                04  64       1B E0 000F1      BBS     #27, (R4), 13$                               ; 3044
                                    4F   03  A4 E8 000F5      BLBS    3(R4), 18$                                   ;
                                  00FC   C2  D5 000F9 13$:    TSTL    252(R2)                                      ; 3046
                                             09 12 000FD      BNEQ    14$                                          ;
                                    53   0108  C2 9E 000FF     MOVAB   264(R2), R3                                  ;
                                             63 D4 00104      CLRL    (R3)                                         ;
                                             28 11 00106      BRB     16$                                          ;
                                    0C   AE  D4 00108 14$:    CLRL    INPUT_ARGS                                   ;
                                    0C   AE  9F 0010B         PUSHAB  INPUT_ARGS                                   ;
                                  0104   C2  DD 0010E         PUSHL   260(R2)                                      ;
```

```
                    53    0108  C2  9E 00112        MOVAB   264(R2), R3              :
                                53  DD 00117        PUSHL   R3                       :
                          0100  C2  9F 00119        PUSHAB  256(R2)                  :
              10    AE    01BA  8F  3C 0011D        MOVZWL  #442, 16(SP)             :
                            10  AE  9F 00123        PUSHAB  16(SP)                   :
                          00FC  C2  9F 00126        PUSHAB  252(R2)                  :
                    69          06  FB 0012A        CALLS   #6, SMG$GET_TERM_DATA    :
                    59          50  E9 0012D 15$:   BLBC    STATUS, 21$              :
                                63  D5 00130 16$:   TSTL    (R3)                     : 3048
                                14  13 00132        BEQL    18$                      :
                          0104  C2  DD 00134        PUSHL   260(R2)                  : 3051
                                63  DD 00138        PUSHL   (R3)                     : 3050
                                52  DD 0013A        PUSHL   R2                       :
                    6B          03  FB 0013C        CALLS   #3, SMG$$OUTPUT          :
                    56          50  D0 0013F        MOVL    R0, STATUS               :
                    59          56  E9 00142 17$:   BLBC    STATUS, 23$              : 3052
                    57          01  D0 00145        MOVL    #1, RENDITION_CHANGED    : 3053
    57        18    AC          01  E1 00148 18$:   BBC     #1, FLAGS, 24$           : 3057
    04              64          1B  E0 0014D        BBS     #27, (R4), 19$           : 3058
                    4F      03  A4  E8 00151        BLBS    3(R4), 24$               :
                          00FC  C2  D5 00155 19$:   TSTL    252(R2)                  : 3060
                                09  12 00159        BNEQ    20$                      :
                    53    0108  C2  9E 0015B        MOVAB   264(R2), R3              :
                                63  D4 00160        CLRL    (R3)                     :
                                28  11 00162        BRB     22$                      :
                            0C  AE  D4 00164 20$:   CLRL    INPUT_ARGS               :
                            0C  AE  9F 00167        PUSHAB  INPUT_ARGS               :
                          0104  C2  DD 0016A        PUSHL   260(R2)                  :
                    53    0108  C2  9E 0016E        MOVAB   264(R2), R3              :
                                53  DD 00173        PUSHL   R3                       :
                          0100  C2  9F 00175        PUSHAB  256(R2)                  :
              10    AE    01BF  8F  3C 00179        MOVZWL  #447, 16(SP)             :
                            10  AE  9F 0017F        PUSHAB  16(SP)                   :
                          00FC  C2  9F 00182        PUSHAB  252(R2)                  :
                    69          06  FB 00186        CALLS   #6, SMG$GET_TERM_DATA    :
                    59          50  E9 00189 21$:   BLBC    STATUS, 27$              :
                                63  D5 0018C 22$:   TSTL    (R3)                     : 3062
                                14  13 0018E        BEQL    24$                      :
                          0104  C2  DD 00190        PUSHL   260(R2)                  : 3065
                                63  DD 00194        PUSHL   (R3)                     : 3064
                                52  DD 00196        PUSHL   R2                       :
                    6B          03  FB 00198        CALLS   #3, SMG$$OUTPUT          :
                    56          50  D0 0019B        MOVL    R0, STATUS               :
                    59          56  E9 0019E 23$:   BLBC    STATUS, 29$              : 3066
                    57          01  D0 001A1        MOVL    #1, RENDITION_CHANGED    : 3067
    57        18    AC          03  E1 001A4 24$:   BBC     #3, FLAGS, 30$           : 3071
    04              64          1B  E0 001A9        BBS     #27, (R4), 25$           : 3072
                    4F      03  A4  E8 001AD        BLBS    3(R4), 30$               :
                          00FC  C2  D5 001B1 25$:   TSTL    252(R2)                  : 3074
                                09  12 001B5        BNEQ    26$                      :
                    53    0108  C2  9E 001B7        MOVAB   264(R2), R3              :
                                63  D4 001BC        CLRL    (R3)                     :
                                28  11 001BE        BRB     28$                      :
                            0C  AE  D4 001C0 26$:   CLRL    INPUT_ARGS               :
                            0C  AE  9F 001C3        PUSHAB  INPUT_ARGS               :
                          0104  C2  DD 001C6        PUSHL   260(R2)                  :
                    53    0108  C2  9E 001CA        MOVAB   264(R2), R3              :
```

```
                                                53  DD 001CF          PUSHL   R3
                                              0100  C2 9F 001D1          PUSHAB  256(R2)
                                        10    AE  01C0  8F 3C 001D5          MOVZWL  #448, 16(SP)
                                                10  AE 9F 001DB          PUSHAB  16(SP)
                                              00FC  C2 9F 001DE          PUSHAB  252(R2)
                                        69          06 FB 001E2          CALLS   #6, SMG$GET_TERM_DATA
                                        66          50 E9 001E5 27$:     BLBC    STATUS, 34$
                                                    63 D5 001E8 28$:     TSTL    (R3)                                    3076
                                                    14 13 001EA          BEQL    30$
                                              0104  C2 DD 001EC          PUSHL   260(R2)                                 3079
                                                    63 DD 001F0          PUSHL   (R3)                                    3078
                                                    52 DD 001F2          PUSHL   R2
                                        6B          03 FB 001F4          CALLS   #3, SMG$$OUTPUT
                                        56          50 D0 001F7          MOVL    R0, STATUS
                                        67          56 E9 001FA 29$:     BLBC    STATUS, 36$                             3080
                                        57          01 D0 001FD          MOVL    #1, RENDITION_CHANGED                   3081
                                                18  AC 95 00200 30$:     TSTB    FLAGS                                   3090
                                                    08 19 00203          BLSS    31$
                                     03     18  AC          06 E0 00205          BBS     #6, FLAGS, 31$
                                                0359  31 0020A          BRW     89$
                                        10    AE    08  AC 7D 0020D 31$:     MOVQ    TEXT_LEN, TEXT_DESC                     3106
                                     04         64    1B E0 00212          BBS     #27, (R4), 32$                          3113
                                        50          03  A4 E8 00216          BLBS    3(R4), 37$
                                              00FC  C2 D5 0021A 32$:     TSTL    252(R2)                                 3115
                                                    09 12 0021E          BNEQ    33$
                                        53    0108  C2 9E 00220          MOVAB   264(R2), R3
                                                    63 D4 00225          CLRL    (R3)
                                                    29 11 00227          BRB     35$
                                                04  AE D4 00229 33$:     CLRL    INPUT_ARGS                              3117
                                                04  AE 9F 0022C          PUSHAB  INPUT_ARGS
                                              0104  C2 DD 0022F          PUSHL   260(R2)                                 3120
                                        53    0108  C2 9E 00233          MOVAB   264(R2), R3                             3119
                                                    53 DD 00238          PUSHL   R3
                                              0100  C2 9F 0023A          PUSHAB  256(R2)
                                        10    AE  01BE  8F 3C 0023E          MOVZWL  #446, 16(SP)
                                                10  AE 9F 00244          PUSHAB  16(SP)
                                              00FC  C2 9F 00247          PUSHAB  252(R2)
                                        69          06 FB 0024B          CALLS   #6, SMG$GET_TERM_DATA
                                        01          50 E8 0024E 34$:     BLBS    STATUS, 35$
                                                    04 00251             RET
                                                    63 D5 00252 35$:     TSTL    (R3)                                    3121
                                                    14 13 00254          BEQL    37$
                                              0104  C2 DD 00256          PUSHL   260(R2)                                 3131
                                                    63 DD 0025A          PUSHL   (R3)                                    3134
                                                    52 DD 0025C          PUSHL   R2
                                        6B          03 FB 0025E          CALLS   #3, SMG$$OUTPUT                         3141
                                        56          50 D0 00261          MOVL    R0, STATUS                              3142
                                        03          56 E8 00264 36$:     BLBS    STATUS, 37$
                                                  03C1  31 00267          BRW     104$
                                     03     00D1  C2    01 E0 0026A 37$:     BBS     #1, 209(R2), 38$
                                                  014A  31 00270          BRW     60$
                                        55          6A 3C 00273 38$:     MOVZWL  TRANSLATED_TEXT_DESC, R5
                                        53          01 CE 00276          MNEGL   #1, I
                                                  0137  31 00279          BRW     58$
                                        50    04  BA43  9A 0027C 39$:     MOVZBL  @TRANSLATED_TEXT_DESC+4[I], CHAR
                                        10          50 D1 00281          CMPL    CHAR, #16
                                                    03 1E 00284          BGEQU   40$
```

```
                                  0113  31 00286          BRW     56$
                            50      10  C6 00289 40$:     DIVL2   #16, CHAR                                    ;3144
                            06      50  D1 0028C          CMPL    CHAR, #6                                      ;3156
                                    09  12 0028F          BNEQ    41$
                                00FC    C2  D5 00291      TSTL    252(R2)
                                    7B  13 00295          BEQL    45$
                                  00C5  31 00297          BRW     52$
                            09      50  D1 0029A 41$:     CMPL    CHAR, #9                                      ;3157
                                    20  12 0029D          BNEQ    42$
                                00FC    C2  D5 0029F      TSTL    252(R2)
                                    6D  13 002A3          BEQL    45$
                                  04    AE  D4 002A5      CLRL    INPUT_ARGS
                                  04    AE  9F 002A8      PUSHAB  INPUT_ARGS
                                0104    C2  DD 002AB      PUSHL   260(R2)
                                0108    C2  9F 002AF      PUSHAB  264(R2)
                                0100    C2  9F 002B3      PUSHAB  256(R2)
                        10  AE  024C    8F  3C 002B7      MOVZWL  #588, 16(SP)
                                    6D  11 002BD          BRB     46$
                            0A      50  D1 0C2BF 42$:     CMPL    CHAR, #10                                     ;3158
                                    20  12 002C2          BNEQ    43$
                                00FC    C2  D5 002C4      TSTL    252(R2)
                                    6D  13 002C8          BEQL    48$
                                  04    AE  D4 002CA      CLRL    INPUT_ARGS
                                  04    AE  9F 002CD      PUSHAB  INPUT_ARGS
                                0104    C2  DD 002D0      PUSHL   260(R2)
                                0108    C2  9F 002D4      PUSHAB  264(R2)
                                0100    C2  9F 002D8      PUSHAB  256(R2)
                        10  AE  024B    8F  3C 002DC      MOVZWL  #587, 16(SP)
                                    6D  11 002E2          BRB     49$
                            0B      50  D1 002E4 43$:     CMPL    CHAR, #11                                     ;3159
                                    20  12 002E7          BNEQ    44$
                                00FC    C2  D5 002E9      TSTL    252(R2)
                                    6A  13 002ED          BEQL    51$
                                  04    AE  D4 002EF      CLRL    INPUT_ARGS
                                  04    AE  9F 002F2      PUSHAB  INPUT_ARGS
                                0104    C2  DD 002F5      PUSHL   260(R2)
                                0108    C2  9F 002F9      PUSHAB  264(R2)
                                0100    C2  9F 002FD      PUSHAB  256(R2)
                        10  AE  024D    8F  3C 00301      MOVZWL  #589, 16(SP)
                                    6E  11 00307          BRB     53$
                            0C      50  D1 00309 44$:     CMPL    CHAR, #12                                     ;3160
                                    20  12 0030C          BNEQ    47$
                                00FC    C2  D5 0030E      TSTL    252(R2)
                                    15  13 00312 45$:     BEQL    51$
                                  04    AE  D4 00314      CLRL    INPUT_ARGS
                                  04    AE  9F 00317      PUSHAB  INPUT_ARGS
                                0104    C2  DD 0031A      PUSHL   260(R2)
                                0108    C2  9F 0031E      PUSHAB  264(R2)
                                0100    C2  9F 00322      PUSHAB  256(R2)
                        10  AE  024A    8F  3C 00326      MOVZWL  #586, 16(SP)
                                    49  11 0032C 46$:     BRB     53$
                            0D      50  D1 0032E 47$:     CMPL    CHAR, #13                                     ;3161
                                    20  12 00331          BNEQ    50$
                                00FC    C2  D5 00333      TSTL    252(R2)
                                    20  13 00337 48$:     BEQL    51$
                                  04    AE  D4 00339      CLRL    INPUT_ARGS
                                  04    AE  9F 0033C      PUSHAB  INPUT_ARGS
```

```
                                    0104   C2  DD 0033F          PUSHL   260(R2)                                    ;
                                    0108   C2  9F 00343          PUSHAB  264(R2)                                   ;
                                    0100   C2  9F 00347          PUSHAB  256(R2)                                   ;
                                10  AE  0249   8F  3C 0034B          MOVZWL  #585, 16(SP)                              ;
                                            24  11 00351 49$:       BRB     53$                                       ;
                                    00FC   C2  D5 00353 50$:       TSTL    252(R2)                                   ; 3162
                                            06  12 00357          BNEQ    52$                                       ;
                                    0108   C2  D4 00359 51$:       CLRL    264(R2)                                   ;
                                            26  11 0035D          BRB     54$                                       ;
                                      04   AE  D4 0035F 52$:       CLRL    INPUT_ARGS                                ;
                                      04   AE  9F 00362          PUSHAB  INPUT_ARGS                                ;
                                    0104   C2  DD 00365          PUSHL   260(R2)                                   ;
                                    0108   C2  9F 00369          PUSHAB  264(R2)                                   ;
                                    0100   C2  9F 0036D          PUSHAB  256(R2)                                   ;
                                10  AE  024E   8F  3C 00371          MOVZWL  #590, 16(SP)                              ;
                                      10   AE  9F 00377 53$:       PUSHAB  16(SP)                                    ;
                                    00FC   C2  9F 0037A          PUSHAB  252(R2)                                   ;
                                69          06  FB 0037E          CALLS   #6, SMG$GET_TERM_DATA                     ;
                                01          50  E8 00381          BLBS    STATUS, 54$                               ;
                                            04  00384          RET                                               ;
                                50  0108   C2  D0 00385 54$:       MOVL    264(R2), R0                               ; 3166
                                            08  13 0038A          BEQL    55$                                       ;
                                    0104   C2  DD 0038C          PUSHL   260(R2)                                   ; 3169
                                            50  DD 00390          PUSHL   R0                                        ; 3168
                                            14  11 00392          BRB     57$                                       ;
                                    FC67   CF  9F 00394 55$:       PUSHAB  P.AAE                                     ; 3173
                                            01  DD 00398          PUSHL   #1                                        ;
                                            0C  11 0039A          BRB     57$                                       ;
                                50  010C C240  D0 0039C 56$:       MOVL    268(R2)[CHAR], R0                         ; 3179
                                    01   A0  9F 003A2          PUSHAB  1(R0)                                     ; 3181
                                7E          60  9A 003A5          MOVZBL  (R0), -(SP)                               ;
                                            52  DD 003A8 57$:       PUSHL   R2                                        ;
                                6B          03  FB 003AA          CALLS   #3, SMG$$OUTPUT                           ;
                                56          5C  D0 003AD          MOVL    R0, STATUS                                ;
                                19          56  E9 003B0          BLBC    STATUS, 61$                               ; 3182
03                              53          55  F2 003B3 58$:       AOBLSS  R5, I, 59$                                ; 3142
                                        0161   31 003B7          BRW     84$                                       ; 3131
                                        FEBF   31 003BA 59$:       BRW     39$                                       ; 3142
                                            5A  DD 003BD 60$:       PUSHL   R10                                       ; 3194
                                      14   AE  9F 003BF          PUSHAB  TEXT_DESC                                 ;
                    00000000G   00          02  FB 003C2          CALLS   #2, [IB$SCOPY_DXDX                        ;
                                56          50  D0 003C9          MOVL    R0, STATUS                                ;
                                03          56  E8 003CC 61$:       BLBS    STATUS, 62$                               ; 3195
                                        0259   31 003CF          BRW     104$                                      ;
                                58          6A  3C 003D2 62$:       MOVZWL  TRANSLATED_TEXT_DESC, R8                  ; 320
                                55          01  CE 003D5          MNEGL   #1, I                                     ;
                                        0126   31 003D8          BRW     81$                                       ;
                                53      04  AA  D0 003DB 63$:       MOVL    TRANSLATED_TEXT_DESC+4, R3                ; 3204
                                50      6543   9A 003DF          MOVZBL  (I)[R3], CHAR                             ; 3217
                                10          50  D1 003E3          CMPL    CHAR, #16                                 ; 3218
                                            03  1E 003E6          BGEQU   64$                                       ;
                                        010F   31 003E8          BRW     80$                                       ;
                                50          10  C6 003EB 64$:       DIVL2   #16, CHAR                                 ; 3220
                                06          50  D1 003EE          CMPL    CHAR, #6                                  ; 3233
                                            09  12 003F1          BNEQ    65$                                       ;
                                    00FC   C2  D5 003F3          TSTL    252(R2)                                   ;
                                            7B  13 003F7          BEQL    69$                                       ;
```

```
                                00C5  31  003F9          BRW     76$
                09                    5C  D1  003FC 65$: CMPL    CHAR, #9                    3234
                                      20  12  003FF          BNEQ    66$
                          00FC        C2  D5  00401          TSTL    252(R2)
                                      6D  13  00405          BEQL    69$
                            04        AE  D4  00407          CLRL    INPUT_ARGS
                            04        AE  9F  0040A          PUSHAB  INPUT_ARGS
                          0104        C2  DD  0040D          PUSHL   260(R2)
                          0108        C2  9F  00411          PUSHAB  264(R2)
                          0100        C2  9F  00415          PUSHAB  256(R2)
            10    AE        024C      8F  3C  00419          MOVZWL  #588, 16(SP)
                                      6D  11  0041F          BRB     70$
                0A                    50  D1  00421 66$: CMPL    CHAR, #10                   3235
                                      20  12  00424          BNEQ    67$
                          00FC        C2  D5  00426          TSTL    252(R2)
                                      6D  13  0042A          BEQL    72$
                            04        AE  D4  0042C          CLRL    INPUT_ARGS
                            04        AE  9F  0042F          PUSHAB  INPUT_ARGS
                          0104        C2  DD  00432          PUSHL   260(R2)
                          0108        C2  9F  00436          PUSHAB  264(R2)
                          0100        C2  9F  0043A          PUSHAB  256(R2)
            10    AE        024B      8F  3C  0043E          MOVZWL  #587, 16(SP)
                                      6D  11  00444          BRB     73$
                0B                    50  D1  00446 67$: CMPL    CHAR, #11                   3236
                                      20  12  00449          BNEQ    68$
                          00FC        C2  D5  0044B          TSTL    252(R2)
                                      6A  13  0044F          BEQL    75$
                            04        AE  D4  00451          CLRL    INPUT_ARGS
                            04        AE  9F  00454          PUSHAB  INPUT_ARGS
                          0104        C2  DD  00457          PUSHL   260(R2)
                          0108        C2  9F  0045B          PUSHAB  264(R2)
                          0100        C2  9F  0045F          PUSHAB  256(R2)
            10    AE        024D      8F  3C  00463          MOVZWL  #589, 16(SP)
                                      6E  11  00469          BRB     77$
                0C                    50  D1  00468 68$: CMPL    CHAR, #12                   3237
                                      20  12  0046E          BNEQ    71$
                          00FC        C2  D5  00470          TSTL    252(R2)
                                      45  13  00474 69$: BEQL    75$
                            04        AE  D4  00476          CLRL    INPUT_ARGS
                            04        AE  9F  00479          PUSHAB  INPUT_ARGS
                          0104        C2  DD  0047C          PUSHL   260(R2)
                          0108        C2  9F  00480          PUSHAB  264(R2)
                          0100        C2  9F  00484          PUSHAB  256(R2)
            10    AE        024A      8F  3C  00489          MOVZWL  #586, 16(SP)
                                      49  11  0048E 70$: BRB     77$
                0D                    50  D1  00490 71$: CMPL    CHAR, #13                   3238
                                      20  12  00493          BNEQ    74$
                          00FC        C2  D5  00495          TSTL    252(R2)
                                      20  13  00499 72$: BEQL    75$
                            04        AE  D4  0049B          CLRL    INPUT_ARGS
                            04        AE  9F  0049E          PUSHAB  INPUT_ARGS
                          0104        C2  DD  004A1          PUSHL   260(R2)
                          0108        C2  9F  004A5          PUSHAB  264(R2)
                          0100        C2  9F  004A9          PUSHAB  256(R2)
            10    AE        0249      8F  3C  004AD          MOVZWL  #585, 16(SP)
                                      24  11  004B3 73$: BRB     77$
                          00FC        C2  D5  004B5 74$: TSTL    252(R2)                     3239
```

```
                                 06   12 004B9        BNEQ    76$
                         0108    C2   D4 004BB 75$:   CLRL    264(R2)                          :
                                 25   11 004BF        BRB     78$                              :
                           04    AE   D4 004C1 76$:   CLRL    INPUT_ARGS                       :
                           04    AE   9F 004C4        PUSHAB  INPUT_ARGS                       :
                         0104    C2   DD 004C7        PUSHL   260(R2)                          :
                         0108    C2   9F 004CB        PUSHAB  264(R2)                          :
                         0100    C2   9F 004CF        PUSHAB  256(R2)                          :
            10     AE    024E    8F   3C 004D3        MOVZWL  #590, 16(SP)                     :
                           10    AE   9F 004D9 77$:   PUSHAB  16(SP)                           :
                         00FC    C2   9F 004DC        PUSHAB  252(R2)                          :
                   69            06   FB 004E0        CALLS   #6, SMG$GET_TERM_DATA            :
                   71            50   E9 004E3        BLBC    STATUS, 87$                      :
                         0108    C2   D5 004E6 78$:   TSTL    264(R2)                          : 3243
                                 08   13 004EA        BEQL    79$                              :
                 6543    0104    D2   90 004EC        MOVB    @260(R2), (I)[R3]                : 3246
                                 0D   11 004F2        BRB     81$                              :
                 6543            2A   90 004F4 79$:   MOVB    #42, (I)[R3]                     : 3249
                                 07   11 004F8        BRB     81$                              : 3218
                 6543    010C C240    F6 004FA 80$:   CVTLB   268(R2)[CHAR], (I)[R3]           : 3253
       02          55            58   F2 00501 81$:   AOBLSS  R8, I, 82$                       : 3218
                                 03   11 00505        BRB     83$                              :
                               FED1   31 00507 82$:   BRW     63$                              :
                           04    AA   DD 0050A 83$:   PUSHL   TRANSLATED_TEXT_DESC+4           : 3265
                   7E            6A   3C 0050D        MOVZWL  TRANSLATED_TEXT_DESC, -(SP)      : 3264
                                 52   DD 00510        PUSHL   R2                               : 3263
                   6B            03   FB 00512        CALLS   #3, SMG$$OUTPUT                  :
                   56            50   D0 00515        MOVL    R0, STATUS                       :
                   57            56   E9 00518        BLBC    STATUS, 91$                      : 3266
       04          64            1B   E0 0051B 84$:   BBS     #27, (R4), 85$                   : 3274
                   52      03    A4   E8 0051F        BLBS    3(R4), 92$                       :
                         00FC    C2   D5 00523 85$:   TSTL    252(R2)                          : 3276
                                 09   12 00527        BNEQ    86$                              :
                   53    0108    C2   9E 00529        MOVAB   264(R2), R3                      :
                                 63   D4 0052E        CLRL    (R3)                             :
                                 28   11 00530        BRB     88$                              :
                           04    AE   D4 00532 86$:   CLRL    INPUT_ARGS                       :
                           04    AE   9F 00535        PUSHAB  INPUT_ARGS                       :
                         0104    C2   DD 00538        PUSHL   260(R2)                          :
                   53    0108    C2   9E 0053C        MOVAB   264(R2), R3                      :
                                 53   DD 00541        PUSHL   R3                               :
                         0100    C2   9F 00543        PUSHAB  256(R2)                          :
            10     AE    01D5    8F   3C 00547        MOVZWL  #469, 16(SP)                     :
                           10    AE   9F 0054D        PUSHAB  16(SP)                           :
                         00FC    C2   9F 00550        PUSHAB  252(R2)                          :
                   69            06   FB 00554        CALLS   #6, SMG$GET_TERM_DATA            :
                   5A            50   E9 00557 87$:   BLBC    STATUS, 95$                      :
                                 63   D5 0055A 88$:   TSTL    (R3)                             : 3278
                                 17   13 0055C        BEQL    92$                              :
                         0104    C2   DD 0055E        PUSHL   260(R2)                          : 3281
                                 63   DD 00562        PUSHL   (R3)                             : 3280
                                 04   11 00564        BRB     90$                              :
                   7E      08    AC   7D 00566 89$:   MOVQ    TEXT_LEN, -(SP)                  : 3288
                                 52   DD 0056A 90$:   PUSHL   R2                               :
                   6B            03   FB 0056C        CALLS   #3, SMG$$OUTPUT                  :
                   56            50   D0 0056F        MOVL    R0, STATUS                       :
                   54            56   E9 00572 91$:   BLBC    STATUS, 97$                      : 3289
```

```
                                54              57  E9  00575 92$:     BLBC    RENDITION_CHANGED, 98$                  ;3299
                             04 64              1B  E0  00578          BBS     #27, (R4), 93$
                                4C        03    A4  E8  0057C          BLBS    3(R4), 98$
                                        C0FC    C2  D5  00580 93$:     TSTL    252(R2)                                 ;3304
                                                09  12  00584          BNEQ    94$
                                53      0108    C2  9E  00586          MOVAB   264(R2), R3
                                                63  D4  0058B          CLRL    (R3)
                                                28  11  0058D          BRB     96$
                                          0C    AE  D4  0058F 94$:     CLRL    INPUT_ARGS
                                          0C    AE  9F  00592          PUSHAB  INPUT_ARGS
                                        0104    C2  DD  00595          PUSHL   260(R2)
                                52      0108    C2  9E  00599          MOVAB   264(R2), R3
                                                53  DD  0059E          PUSHL   R3
                                        0100    C2  9F  005A0          PUSHAB  256(R2)
                                10  AE  0253    8F  3C  005A4          MOVZWL  #595, 16(SP)
                                          10    AE  9F  005AA          PUSHAB  16(SP)
                                        00FC    C2  9F  005AD          PUSHAB  252(R2)
                                69              06  FB  005B1          CALLS   #6, SMG$GET_TERM_DATA
                                7B              50  E9  005B4 95$:     BLBC    STATUS, 106$
                                                63  D5  005B7 96$:     TSTL    (R3)                                    ;3309
                                                11  13  005B9          BEQL    98$
                                        0104    C2  DD  005BB          PUSHL   260(R2)                                 ;3312
                                                63  DD  005BF          PUSHL   (R3)                                    ;3311
                                                52  DD  005C1          PUSHL   R2
                                6B              03  FB  005C3          CALLS   #3, SMG$$OUTPUT
                                56              50  D0  005C6          MOVL    R0, STATUS
                                5F              56  E9  005C9 97$:     BLBC    STATUS, 104$                            ;3313
                             5E 18  AC              04  E1  005CC 98$:     BBC     #4, FLAGS, 105$                         ;3317
                             04 64                  1B  E0  005D1          BBS     #27, (R4), 99$                          ;3318
                                56        03    A4  E8  005D5          BLBS    3(R4), 105$
                                        00FC    C2  D5  005D9 99$:     TSTL    252(R2)                                 ;3320
                                                09  12  005DD          BNEQ    100$
                                53      0108    C2  9E  005DF          MOVAB   264(R2), R3
                                                63  D4  005E4          CLRL    (R3)
                                                28  11  005E6          BRB     101$
                                          0C    AE  D4  005E8 100$:    CLRL    INPUT_ARGS
                                          0C    AE  9F  005EB          PUSHAB  INPUT_ARGS
                                        0104    C2  DD  005EE          PUSHL   260(R2)
                                53      0108    C2  9E  005F2          MOVAB   264(R2), R3
                                                53  DD  005F7          PUSHL   R3
                                        0100    C2  9F  005F9          PUSHAB  256(R2)
                             10 AE      01D5    8F  3C  005FD          MOVZWL  #469, 16(SP)
                                          10    AE  9F  00603          PUSHAB  16(SP)
                                        00FC    C2  9F  00606          PUSHAB  252(R2)
                                69              06  FB  0060A          CALLS   #6, SMG$GET_TERM_DATA
                                22              50  E9  0060D          BLBC    STATUS, 106$
                                                63  D5  00610 101$:    TSTL    (R3)                                    ;3322
                                                1B  13  00612          BEQL    105$
                                        0104    C2  DD  00614          PUSHL   260(R2)                                 ;3325
                                                63  DD  00618          PUSHL   (R3)                                    ;3324
                                                04  11  0061A          BRB     103$
                                7E        08    AC  7D  0061C 102$:    MOVQ    TEXT_LEN, -(SP)                         ;3333
                                                52  DD  00620 103$:    PUSHL   R2
                                6B              03  FB  00622          CALLS   #3, SMG$$OUTPUT
                                56              50  D0  00625          MOVL    R0, STATUS
                                04              56  E8  00628          BLBS    STATUS, 105$                            ;3334
                                50              56  D0  0062B 104$:    MOVL    STATUS, R0
```

```
                                        04 0062E                RET
                                50      01 D0 0062F 105$:       MOVL    #1, R0                  ;
                                        04 00632 106$:          RET                             ; 3338
                                                                                                ; 3340

; Routine Size:  1587 bytes,    Routine Base:  _SMG$CODE + 170B
```

```
;  3106          3341  1 %SBTTL 'SMG$$AUTOB_OUTPUT - Autobended entry for output to screen'
;  3107          3342  1 GLOBAL ROUTINE SMG$$AUTOB_OUTPUT (PB_ID,TEXT_LEN,TEXT_ADR) =
;  3108          3343  1 !++
;  3109          3344  1 ! FUNCTIONAL DESCRIPTION:
;  3110          3345  1 !
;  3111          3346  1 !
;  3112          3347  1 ! CALLING SEQUENCE:
;  3113          3348  1 !
;  3114          3349  1 !       ret_status.wlc.v = SMG$$AUTOB_OUTPUT(
;  3115          3350  1 !                                            PB_ID.rl.v,
;  3116          3351  1 !                                            TEXT_LEN.rl.v,
;  3117          3352  1 !                                            TEXT_ADR.rt.r)
;  3118          3353  1 !
;  3119          3354  1 ! FORMAL PARAMETERS:
;  3120          3355  1 !
;  3121          3356  1 !       PB_ID.rl.v              Pasteboard id
;  3122          3357  1 !
;  3123          3358  1 !       TEXT_LEN.rl.v           Number of characters in text string
;  3124          3359  1 !
;  3125          3360  1 !       TEXT_ADR.rt.r           Address of start of text string
;  3126          3361  1 !                               The text may contain escape sequences.
;  3127          3362  1 !
;  3128          3363  1 ! IMPLICIT INPUTS:
;  3129          3364  1 !
;  3130          3365  1 !       None
;  3131          3366  1 !
;  3132          3367  1 ! IMPLICIT OUTPUTS:
;  3133          3368  1 !
;  3134          3369  1 !       None
;  3135          3370  1 !
;  3136          3371  1 ! COMPLETION STATUS:
;  3137          3372  1 !
;  3138          3373  1 !       SS$_NORMAL      Normal successful completion
;  3139          3374  1 !
;  3140          3375  1 ! SIDE EFFECTS:
;  3141          3376  1 !
;  3142          3377  1 !       The following may occur as a result of calling SMG$$OUTPUT:
;  3143          3378  1 !       Output may occur.
;  3144          3379  1 !       If buffering is enabled, buffers may fill and/or dump.
;  3145          3380  1 !--
```

```
;  3147          3381  2 BEGIN
;  3148          3382  2
;  3149          3383  2 LOCAL
;  3150          3384  2
;  3151          3385  2         PBCB;
;  3152          3386  2
;  3153          3387  2 $SMG$GET_PBCB(.PB_ID,PBCB);
;  3154          3388  2
;  3155          3389  2 RETURN SMG$$OUTPUT(.PBCB,.TEXT_LEN,.TEXT_ADR)
;  3156          3390  2
;  3157          3391  1 END;                              ! end of routine SMG$$AUTOB_OUTPUT



                                                0000 00000          .ENTRY  SMG$$AUTOB_OUTPUT, Save nothing                 ; 3342
                                50        04    BC   D0 00002       MOVL    @PB_ID, R0                                      ; 3387
                                                11   19 00006       BLSS    1$                                              ;
                     00000000G  00             50   D1 00008       CMPL    R0, PBD_L_COUNT                                 ;
                                                08   14 0000F       BGTR    1$                                              ;
                  08 00000000G  00             50   E0 00011       BBS     R0, PBD_V_PB_AVAIL, 2$                          ;
                                50 00000000G    8F   D0 00019 1$:    MOVL    #SMG$_INVPAS_ID, R0                             ;
                                                04   00020          RET                                                     ;
                                50 00000000G0040  D0 00021 2$:      MOVL    PBD_A_PBCB[R0], PBCB                            ;
                                7E        08    AC   7D 00029       MOVQ    TEXT_LEN, -(SP)                                 ; 3389
                                                50   DD 0002D       PUSHL   PBCB                                            ;
                         0000V  CF              03   FB 0002F       CALLS   #3, SMG$$OUTPUT                                 ;
                                                04   00034          RET                                                     ; 3391

; Routine Size:  53 bytes,    Routine Base:  _SMG$CODE + 1D3E
```

```
;  3159          3392  1 %SBTTL 'SMG$$OUTPUT - Output to screen'
;  3160          3393  1 GLOBAL ROUTINE SMG$$OUTPUT (P_PBCB,TEXT_LEN,TEXT_ADR) =
;  3161          3394  1 !++
;  3162          3395  1 ! FUNCTIONAL DESCRIPTION:
;  3163          3396  1 !
;  3164          3397  1 !
;  3165          3398  1 ! CALLING SEQUENCE:
;  3166          3399  1 !
;  3167          3400  1 !       ret_status.wlc.v = SMG$$OUTPUT(
;  3168          3401  1 !                                      P_PBCB.rab.r,
;  3169          3402  1 !                                      TEXT_LEN.rl.v,
;  3170          3403  1 !                                      TEXT_ADR.rt.r)
;  3171          3404  1 !
;  3172          3405  1 ! FORMAL PARAMETERS:
;  3173          3406  1 !
;  3174          3407  1 !       P_PBCB.rab.r            Address of pasteboard control block.
;  3175          3408  1 !
;  3176          3409  1 !       TEXT_LEN.rl.v           Number of characters in text string
;  3177          3410  1 !
;  3178          3411  1 !       TEXT_ADR.rt.r           Address of start of text string
;  3179          3412  1 !                               The text may contain escape sequences.
;  3180          3413  1 !
;  3181          3414  1 ! IMPLICIT INPUTS:
;  3182          3415  1 !
;  3183          3416  1 !       Contents of PBCB.
;  3184          3417  1 !
;  3185          3418  1 ! IMPLICIT OUTPUTS:
;  3186          3419  1 !
;  3187          3420  1 !       PBCB[PBCB_W_OUTPUT_BUFLEN]      may change if buffering is enabled.
;  3188          3421  1 !
;  3189          3422  1 ! COMPLETION STATUS:
;  3190          3423  1 !
;  3191          3424  1 !       SS$_NORMAL      Normal successful completion
;  3192          3425  1 !
;  3193          3426  1 ! SIDE EFFECTS:
;  3194          3427  1 !
;  3195          3428  1 !       Output may occur.
;  3196          3429  1 !       If buffering is enabled, buffers may fill and/or dump.
;  3197          3430  1 !--
```

```
;  3199          3431  2 BEGIN
;  3200          3432  2
;  3201          3433  2 BIND
;  3202          3434  2
;  3203          3435  2         PBCB    = .P_PBCB       : $PBCB_DECL,   ! pasteboard control block
;  3204          3436  2         TEXT    = .TEXT_ADR     : VECTOR[,BYTE],! string to be output
;  3205          3437  2         BUFLEN  =  PBCB[PBCB_W_OUTPUT_BUFLEN]   : WORD, ! number of chars in buffer
;  3206          3438  2         BUFSIZ  =  PBCB[PBCB_W_OUTPUT_BUFSIZ]   : WORD, ! size of buffer
;  3207          3439  2         BUFFER  = .PBCB[PBCB_A_OUTPUT_BUFFER]   : VECTOR[,BYTE];        ! Output buffer
;  3208          3440  2
;  3209          3441  2 LOCAL
;  3210          3442  2
;  3211          3443  2         STATUS;
```

```
;  3213          3444  2 !+
;  3214          3445  2 ! Do nothing if the output is being controlled by RMS.
;  3215          3446  2 !-
;  3216          3447  2
;  3217          3448  2 IF .PBCB[PBCB_V_RMS]
;  3218          3449  2   THEN  RETURN  SMG$_WILUSERMS;
;  3219          3450  2
;  3220          3451  2 !+
;  3221          3452  2 ! If buffering is enabled, and the new string won't fit in the buffer,
;  3222          3453  2 ! then we must output the buffer now.
;  3223          3454  2 ! If it will fit, then just put it in the buffer.
;  3224          3455  2 ! Do not break up the string since we do not want to output
;  3225          3456  2 ! a partial escape sequence.
;  3226          3457  2 !-
;  3227          3458  2
;  3228          3459  2 IF .PBCB[PBCB_V_BUF_ENABLED]
;  3229          3460  3   THEN  BEGIN   ! buffering is enabled
;  3230          3461  3
;  3231          3462  3         !+
;  3232          3463  3         ! See if the string will fit in the buffer.
;  3233          3464  3         !-
;  3234          3465  3
;  3235          3466  3         IF .TEXT_LEN+.BUFLEN GTRU .BUFSIZ
;  3236          3467  4           THEN  BEGIN    ! No - Dump buffer
;  3237          3468  4                 STATUS=OUTPUT(PBCB,.BUFLEN,BUFFER);
;  3238          3469  4                 IF NOT .STATUS THEN RETURN .STATUS;
;  3239          3470  4                 BUFLEN=0
;  3240          3471  4                 END     ! No - Dump buffer
;  3241          3472  4           ELSE  BEGIN   ! Yes - append to buffer
;  3242          3473  4
;  3243          3474  4                 !+
;  3244          3475  4                 ! Copy the text into the buffer,
;  3245          3476  4                 ! update BUFLEN which keeps track of
;  3246          3477  4                 ! how much data is in the buffer,
;  3247          3478  4                 ! and then return.
;  3248          3479  4                 !-
;  3249          3480  4
;  3250          3481  4                 CH$MOVE(.TEXT_LEN,TEXT,BUFFER[.BUFLEN]);
;  3251          3482  4                 BUFLEN=.BUFLEN+.TEXT_LEN;
;  3252          3483  4                 RETURN  SS$_NORMAL
;  3253          3484  4
;  3254          3485  3                 END;    ! Yes - append to buffer
;  3255          3486  3
;  3256          3487  3         !+
;  3257          3488  3         ! We reach here if the string would not fit in the buffer.
;  3258          3489  3         ! The buffer has been dumped.
;  3259          3490  3         ! Put the new string into the buffer.
;  3260          3491  3         ! If it will not fit, we output it in chunks.
;  3261          3492  3         ! We output as many full buffer chunks as we can.
;  3262          3493  3         ! When we are all done, we are left with a string
;  3263          3494  3         ! smaller than one buffer's worth, which we then
;  3264          3495  3         ! put into our output buffer.
;  3265          3496  3         !-
;  3266          3497  3
;  3267          3498  3         INCR I FROM 0 BY .BUFSIZ DO
;  3268          3499  3                 IF .I+.BUFSIZ LEQU .TEXT_LEN
;  3269          3500  4                   THEN  BEGIN   ! output next part of string
```

```
;  3270          3501  4                          STATUS=OUTPUT(PBCB,.BUFSIZ,TEXT[.I]);
;  3271          3502  4                          IF NOT .STATUS THEN RETURN .STATUS
;  3272          3503  4                          END      ! output next part of string
;  3273          3504  4                    ELSE  BEGIN    ! buffer final part of string
;  3274          3505  4                          BUFLEN=.TEXT_LEN-.I;                    ! could be 0
;  3275          3506  4                          CH$MOVE(.BUFLEN,TEXT[.I],BUFFER);
;  3276          3507  4                          EXITLOOP
;  3277          3508  4                          END      ! buffer final part of string
;  3278          3509  4
;  3279          3510  3              END      ! buffering is enabled
;  3280          3511  3          ELSE  BEGIN    ! no buffering
;  3281          3512  3
;  3282          3513  3              !+
;  3283          3514  3              ! Output the string directly.
;  3284          3515  3              !-
;  3285          3516  3
;  3286          3517  3              STATUS=OUTPUT(PBCB,.TEXT_LEN,TEXT);
;  3287          3518  3              IF NOT .STATUS THEN RETURN .STATUS
;  3288          3519  3
;  3289          3520  2              END;     ! no buffering
;  3290          3521  2
;  3291          3522  2      RETURN  SS$_NORMAL
;  3292          3523  2
;  3293          3524  1  END;


                                        0FFC 00000          .ENTRY  SMG$$OUTPUT, Save R2,R3,R4,R5,R6,R7,R8,R9,- ; 3393
                                                                    R10,R11
                                    5E          04  C2 00002        SUBL2   #4, SP
                                    57      04  AC  D0 00005        MOVL    P_PBCB, R7                            ; 3435
                                    58      72  A7  9E 00009        MOVAB   114(R7), R8                           ; 3437
                                    59      6C  A7  D0 0000D        MOVL    108(R7), R9                           ; 3439
                08      00D0        C7          03  E1 00011        BBC     #3, 208(R7), 1$                       ; 3448
                                    50 00000000G    8F  D0 00017    MOVL    #SMG$_WILUSERMS, R0                   ; 3449
                                                    04  0001E        RET
                                    5A      08  AC  D0 0001F 1$:    MOVL    TEXT_LEN, R10                         ; 3466
                                    69      0C  A7  E9 00023        BLBC    12(R7), 7$                            ; 3459
                                    52          68  3C 00027        MOVZWL  (R8), R2                              ; 3466
                50                  5A          52  C1 0002A        ADDL3   R2, R10, R0
                                    6E      70  A7  3C 0002E        MOVZWL  112(R7), (SP)
                                    6E          50  D1 00032        CMPL    R0, (SP)
                                                15  1B 00035        BLEQU   2$
                                            0204    8F  BB 00037    PUSHR   #^M<R2,R9>                            ; 3468
                                                57  DD 0003B        PUSHL   R7
                        0000V       CF              03  FB 0003D    CALLS   #3, OUTPUT
                                    5B              50  D0 00042    MOVL    R0, STATUS
                                    5A              5B  E9 00045    BLBC    STATUS, 8$                            ; 3469
                                                    68  B4 00048    CLRW    (R8)                                  ; 3470
                                                    0B  11 0004A    BRB     3$
        6249        0C          BC              5A  28 0004C 2$:    MOVC3   R10, @TEXT_ADR, (R2)[R9]              ; 3481
                                    68              5A  A0 00052    ADDW2   R10, (R8)                             ; 3482
                                                    4F  11 00055    BRB     9$                                    ; 3483
                                                    56  D4 00057 3$:    CLRL    I                                 ; 3498
                50                  56              6E  C1 00059 4$:    ADDL3   (SP), I, R0                       ; 3499
```

```
                                 5A              50  D1 0005D          CMPL    R0, R10                                 ;
                                                 16  1A 00060          BGTRU   5$                                      ;
                                          0C BC46    9F 00062          PUSHAB  @TEXT_ADR[I]                            ; 3501
                                          04    AE   DD 00066          PUSHL   4(SP)                                   ;
                                                57   DD 00069          PUSHL   R7                                      ;
                           0000V  CF                 03  FB 0006B          CALLS   #3, OUTPUT                              ;
                                  5B                 50  D0 00070          MOVL    R0, STATUS                              ;
                                  0E                 5B  E8 00073          BLBS    STATUS, 4$                              ;
                                                     2A  11 00076          BRB     8$                                      ; 3502
                     68              5A              56  A3 00078 5$:      SUBW3   I, R10, (R8)                            ; 3505
                     69        0C BC46               68  28 0007C          MOVC3   (R8), @TEXT_ADR[I], (R9)                ; 3506
                                                     22  11 00082          BRB     9$                                      ; 3504
            FFCB     56              6E 7FFFFFFF     8F  F1 00084 6$:      ACBL    #2147483647, (SP), I, 4$                ; 3499
                                                     16  11 0008E          BRB     9$                                      ; 3498
                                          0C    AC   DD 00090 7$:      PUSHL   TEXT_ADR                                ; 3517
                                        0480    8F   BB 00093          PUSHR   #^M<R7,R10>                             ;
                           0000V  CF                 03  FB 00097          CALLS   #3, OUTPUT                              ;
                                  5B                 50  D0 0009C          MOVL    R0, STATUS                              ;
                                  04                 5B  E8 0009F          BLBS    STATUS, 9$                              ; 3518
                                  50                 5B  D0 000A2 8$:      MOVL    STATUS, R0                              ;
                                                     04  000A5          RET                                             ;
                                  50                 01  D0 000A6 9$:      MOVL    #1, R0                                  ; 3522
                                                     04  000A9          RET                                             ; 3524

; Routine Size:  170 bytes,    Routine Base:  _SMG$CODE + 1D73
```

```
;  3295     3525  1 %SBTTL 'OUTPUT - Low level output'
;  3296     3526  1 ROUTINE OUTPUT(P_PBCB,TEXT_LEN,TEXT_ADR) =
;  3297     3527  1 !++
;  3298     3528  1 ! FUNCTIONAL DESCRIPTION:
;  3299     3529  1 !
;  3300     3530  1 !       Handles low level output by issuing a QIO or calling RMS.
;  3301     3531  1 !       No buffering occurs here.
;  3302     3532  1 !       If CTRL/O was encountered, then we invalidate our knowledge
;  3303     3533  1 !       of the screen.
;  3304     3534  1 !
;  3305     3535  1 ! CALLING SEQUENCE:
;  3306     3536  1 !
;  3307     3537  1 !       ret_status.wlc.v = OUTPUT(
;  3308     3538  1 !                                P_PBCB.rab.r,
;  3309     3539  1 !                                TEXT_LEN.rl.v,
;  3310     3540  1 !                                TEXT_ADR.rt.r)
;  3311     3541  1 !
;  3312     3542  1 ! FORMAL PARAMETERS:
;  3313     3543  1 !
;  3314     3544  1 !       P_PBCB.rab.r            Address of pasteboard control block.
;  3315     3545  1 !
;  3316     3546  1 !       TEXT_LEN.rl.v           Number of characters in text string
;  3317     3547  1 !
;  3318     3548  1 !       TEXT_ADR.rt.r           Address of start of text string
;  3319     3549  1 !                               The text may contain escape sequences.
;  3320     3550  1 !
;  3321     3551  1 ! IMPLICIT INPUTS:
;  3322     3552  1 !
;  3323     3553  1 !       Contents of PBCB.
;  3324     3554  1 !
;  3325     3555  1 ! IMPLICIT OUTPUTS:
;  3326     3556  1 !
;  3327     3557  1 !       NONE
;  3328     3558  1 !
;  3329     3559  1 ! COMPLETION STATUS:
;  3330     3560  1 !
;  3331     3561  1 !       SS$_NORMAL      Normal successful completion
;  3332     3562  1 !       SS$_xyz         errors from QIO
;  3333     3563  1 !       SS$_xyz         errors from $ASSIGN
;  3334     3564  1 !       RMS$_xyz        errors from RMS (STS value only)
;  3335     3565  1 !
;  3336     3566  1 ! SIDE EFFECTS:
;  3337     3567  1 !
;  3338     3568  1 !       NONE
;  3339     3569  1 !--
```

```
; 3341          3570  2 BEGIN
; 3342          3571  2
; 3343          3572  2 BIND
; 3344          3573  2
; 3345          3574  2         PBCB    = .P_PBCB       : $PBCB_DECL;   ! pasteboard control block
; 3346          3575  2
; 3347          3576  2 LOCAL
; 3348          3577  2
; 3349          3578  2         QIO_IOSB                : VECTOR[4],
; 3350          3579  2         STATUS;
; 3351          3580  2
; 3352          3581  2 !+
; 3353          3582  2 ! Do nothing if the output is being controlled by RMS.
; 3354          3583  2 !-
; 3355          3584  2
; 3356          3585  2 IF .PBCB[PBCB_V_RMS]
; 3357          3586  2   THEN  RETURN  SMG$_WILUSERMS;
; 3358          3587  2
; 3359          3588  2 !+
; 3360          3589  2 ! Null strings succeed no matter what.
; 3361          3590  2 !-
; 3362          3591  2
; 3363          3592  2 IF .TEXT_LEN EQL 0
; 3364          3593  2   THEN  RETURN  SS$_NORMAL;
; 3365          3594  2
; 3366          3595  2 !+
; 3367          3596  2 ! Normally, we use QIOs to talk to this terminal.
; 3368          3597  2 ! See if a channel has been assigned yet.
; 3369          3598  2 ! If not, assign a channel now.
; 3370          3599  2 !-
; 3371          3600  2
; 3372          3601  2 IF .PBCB[PBCB_W_CHAN] EQL 0
; 3373          3602  3   THEN  BEGIN    ! assigning channel
; 3374          3603  3
; 3375          3604  3         ! *** Perhaps this code should be moved to SMG$CREATE_PASTEBOARD.
; 3376          3605  3
; 3377          3606  3         LOCAL   NAME_DESC       : VECTOR[2];    ! Fixed length descriptor
; 3378          3607  3
; 3379          3608  3         !+
; 3380          3609  3         ! Create a fixed length descriptor for our device name string
; 3381          3610  3         ! for use by $ASSIGN.
; 3382          3611  3         !-
; 3383          3612  3
; 3384          3613  3         NAME_DESC[0]=.PBCB[PBCB_W_DEVNAM_LEN];
; 3385          3614  3         NAME_DESC[1]= PBCB[PBCB_T_DEVNAM];
; 3386          3615  3
; 3387          3616  3         !+
; 3388          3617  3         ! Assign the channel.
; 3389          3618  3         ! Put the resulting channel number in PBCB[PBCB_W_CHAN].
; 3390          3619  3         !-
; 3391          3620  3
; 3392        P 3621  3         STATUS=$ASSIGN( DEVNAM  = NAME_DESC,
; 3393          3622  3                         CHAN    = PBCB[PBCB_W_CHAN]);
; 3394          3623  3         IF NOT .STATUS THEN RETURN .STATUS
; 3395          3624  3
; 3396          3625  2         END;    ! assigning channel
; 3397          3626  2
```

```
;  3398          3627  2 !+
;  3399          3628  2 ! Issue a QIO to output the string.
;  3400          3629  2 !-
;  3401          3630  2
;  3402        P 3631  2 STATUS=$QIOW(    CHAN    = .PBCB[PBCB_W_CHAN],
;  3403        P 3632  2                  EFN     = .PBCB[PBCB_B_EFN],
;  3404        P 3633  2                  FUNC    =  IO$_WRITEVBLK OR IO$M_NOFORMAT OR
;  3405        P 3634  2                     (IF .PBCB[PBCB_V_UNSOLICIT] THEN IO$M_ENABLMBX ELSE 0),
;  3406        P 3635  2                  IOSB    = QIO_IOSB,
;  3407        P 3636  2                  P1      = .TEXT_ADR,
;  3408          3637  2                  P2      = .TEXT_LEN);
;  3409          3638  2 IF NOT .STATUS THEN RETURN .STATUS;
;  3410          3639  2 IF NOT .QIO_IOSB[0] THEN RETURN .QIO_IOSB[0];
;  3411          3640  2
;  3412          3641  2 !+
;  3413          3642  2 ! If the I/O was aborted by CTRL/O, then we have no idea how far
;  3414          3643  2 ! the I/O got and so our knowledge of the screen is invalid.
;  3415          3644  2 ! We therefore invalidate our screen buffer by setting the CONTROLO bit.
;  3416          3645  2 ! (We can't just invalidate the screen buffer here because our caller
;  3417          3646  2 ! may set that buffer.)
;  3418          3647  2 !-
;  3419          3648  2
;  3420          3649  2 IF .STATUS EQL SS$_CONTROLO
;  3421          3650  2 OR .QIO_IOSB[0] EQL SS$_CONTROLO
;  3422          3651  2   THEN PBCB[PBCB_V_CONTROLO]=1;
;  3423          3652  2
;  3424          3653  2 RETURN  SS$_NORMAL
;  3425          3654  2
;  3426          3655  1 END;


                                                                  .EXTRN  SYS$ASSIGN, SYS$QIOW

                                                        000C 00000 OUTPUT:  .WORD   Save R2,R3                          ;  3526
                                                5E        18   C2 00002          SUBL2  #24, SP
                                                52        04   AC  D0 00005      MOVL   P_PBCB, R2                    ;  3574
                                                53      00D0   C2  9E 00009      MOVAB  208(R2), R3                   ;  3585
                                        08      63             03  E1 0000E      BBC    #3, (R3), 1$                  ;
                                                50 00000000G   8F  D0 00012      MOVL   #SMG$_WILUSERMS, R0           ;  3586
                                                               04  00019          RET
                                                        08     AC  D5 0001A 1$:   TSTL   TEXT_LEN                     ;  3592
                                                               75  13 0001D      BEQL   7$
                                                        64     A2  B5 0001F      TSTW   100(R2)                       ;  3601
                                                               1B  12 00022      BNEQ   2$
                                                6E      12     A2  3C 00024      MOVZWL 18(R2), NAME_DESC             ;  3613
                                            04  AE      18     A2  9E 00028      MOVAB  24(R2), NAME_DESC+4           ;  3614
                                                               7E  7C 0002D      CLRQ   -(SP)                         ;  3622
                                                        64     A2  9F 0002F      PUSHAB 100(R2)
                                                        0C     AE  9F 00032      PUSHAB NAME_DESC
                                            00000000G   00             04  FB 00035      CALLS  #4, SYS$ASSIGN
                                                58                     50  E9 0003C      BLBC   STATUS, 8$            ;  3623
                                                               7E  7C 0003F 2$:   CLRQ   -(SP)                         ;  3637
                                                               7E  7C 00041      CLRQ   -(SP)
                                                        08     AC  DD 00043      PUSHL  TEXT_LEN
                                                        0C     AC  DD 00046      PUSHL  TEXT_ADR
                                                               7E  7C 00049      CLRQ   -(SP)
```

```
                                    28   AE  9F 0004B          PUSHAB  QIO_IOSB
                 06           63        01  E1 0004E          BBC     #1, (R3), 3$                    ;
                              51   80   8F  9A 00052          MOVZBL  #128, R1                        ;
                                        02  11 00056          BRB     4$                              ;
                                        51  D4 00058 3$:      CLRL    R1                              ;
                 7E           51 00000130  8F  C9 0005A 4$:   BISL3   #304, R1, -(SP)                 ;
                              7E   64      A2  3C 00062       MOVZWL  100(R2), -(SP)                  ;
                              7E   66      A2  9A 00066       MOVZBL  102(R2), -(SP)                  ;
                    00000000G 00           0C  FB 0006A       CALLS   #12, SYS$QIOW                   ;
                              23           50  E9 00071       BLBC    STATUS, 8$                      ; 3638
                              05   08      AE  E8 00074       BLBS    QIO_IOSB, 5$                    ; 3639
                              50   08      AE  D0 00078       MOVL    QIO_IOSB, R0                    ;
                                           04  0007C         RET                                     ;
                    00000609  8F           50  D1 0007D 5$:   CMPL    STATUS, #1545                   ; 3649
                                           0A  13 00084       BEQL    6$                              ;
                    00000609  8F   08      AE  D1 00086       CMPL    QIO_IOSB, #1545                 ; 3650
                                           04  12 0008E       BNEQ    7$                              ;
                              63   40      8F  88 00090 6$:   BISB2   #64, (R3)                       ; 3651
                              50           01  D0 00094 7$:   MOVL    #1, R0                          ; 3653
                                           04  00097 8$:      RET                                     ; 3655

; Routine Size:  152 bytes,    Routine Base:  _SMG$CODE + 1E1D
```

```
;  3428          3656  1 END
;  3429          3657  0 ELUDOM



                                                                .EXTRN  LIB$SIGNAL

;                                PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;   _SMG$DATA                      76  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,  PIC,ALIGN(2)
;   _SMG$CODE                    7861  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)



;                             Library Statistics
;                                                 -------- Symbols --------      Pages       Processing
;       File                                      Total    Loaded   Percent      Mapped      Time
;
;   _$255$DUA18:[SYSLIB]STARLET.L32;1             9776       135         1         581       00:01.0
;   _$255$DUA18:[SMGRTL.OBJ]RTLLIB.L32;1            36         0         0           8       00:00.1
;   _$255$DUA18.[SMGRTL.OBJ]SMGLIB.L32;1           469        78        16          38       00.00.4



;                                    COMMAND QUALIFIERS

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

; Size:          7751 code + 186 data bytes
; Run Time:         03:13.0
; Elapsed Time:     04:10.2
; Lines/CPU Min:     1136
; Lexemes/CPU-Min: 19480
; Memory Used:  700 pages
; Compilation Complete
```

0447 AH-EF71A-SE
VAX/VMS V4.1 SRC LST MCRF UPD

SYSLOA

SMGMINUPD
LIS

CSP
MAP

SMGPUTTEX
LIS

SYSLOAWS1
MAP