C 12

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

```
_$2

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

```
**FILE**ID**SMGPRVINP                                    M 10
```

```
  SSSSSSSS  MM      MM    GGGGGGGG  PPPPPPPP    RRRRRRRR    VV      VV    IIIIII    NN      NN  PPPPPPPP
  SSSSSSSS  MM      MM    GGGGGGGG  PPPPPPPP    RRRRRRRR    VV      VV    IIIIII    NN      NN  PPPPPPPP
SS          MMMM  MMMM  GG          PP      PP  RR      RR  VV      VV      II      NN      NN  PP      PP
SS          MMMM  MMMM  GG          PP      PP  RR      RR  VV      VV      II      NN      NN  PP      PP
SS          MM  MM  MM  GG          PP      PP  RR      RR  VV      VV      II      NNNN    NN  PP      PP
SS          MM  MM  MM  GG          PP      PP  RR      RR  VV      VV      II      NNNN    NN  PP      PP
  SSSSSS    MM      MM  GG          PPPPPPPP    RRRRRRRR    VV      VV      II      NN  NN  NN  PPPPPPPP
  SSSSSS    MM      MM  GG          PPPPPPPP    RRRRRRRR    VV      VV      II      NN  NN  NN  PPPPPPPP
        SS  MM      MM  GG  GGGGGG  PP          RR  RR      VV      VV      II      NN    NNNN  PP
        SS  MM      MM  GG  GGGGGG  PP          RR  RR      VV      VV      II      NN    NNNN  PP
        SS  MM      MM  GG      GG  PP          RR    RR      VV  VV        II      NN      NN  PP              ....
        SS  MM      MM  GG      GG  PP          RR    RR      VV  VV        II      NN      NN  PP              ....
SSSSSSSS    MM      MM    GGGGGG    PP          RR      RR      VV        IIIIII    NN      NN  PP              ....
SSSSSSSS    MM      MM    GGGGGG    PP          RR      RR      VV        IIIIII    NN      NN  PP              ....

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

```
0001 0 %TITLE 'SMG$$PRVINP - Private Input support routines'
0002 0 MODULE SMG$$PRVINP (
0003 0                 IDENT = '1-001' ! File: SMGPRVINP.B32 Edit: STAN1001
0004 0                 ) =
0005 1 BEGIN
0006 1 !
0007 1 !****************************************************************************
0008 1 !*                                                                          *
0009 1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0010 1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0011 1 !*  ALL RIGHTS RESERVED.                                                    *
0012 1 !*                                                                          *
0013 1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0014 1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0015 1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0016 1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0017 1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0018 1 !*  TRANSFERRED.                                                            *
0019 1 !*                                                                          *
0020 1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0021 1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0022 1 !*  CORPORATION.                                                            *
0023 1 !*                                                                          *
0024 1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0025 1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0026 1 !*                                                                          *
0027 1 !*                                                                          *
0028 1 !****************************************************************************
0029 1 !
```

```
;    31          0030  1 !++
;    32          0031  1 ! FACILITY:     Screen Management
;    33          0032  1 !
;    34          0033  1 ! ABSTRACT:
;    35          0034  1 !       The procedures in this module act as interfaces between the
;    36          0035  1 ! virtual displays and pasteboards and associated data structures, and
;    37          0036  1 ! the keyboard input side of SMG.  These routines are called to pass
;    38          0037  1 ! information about changes to the physical screen that have been
;    39          0038  1 ! brought about by input-related activities.
;    40          0039  1 ! These are in a module by themselves so that they can be explicitly
;    41          0040  1 ! linked into programs that need them (like WHAT) since they
;    42          0041  1 ! will not be in the SMGSHR vector.
;    43          0042  1 !
;    44          0043  1 ! ENVIRONMENT:  User mode, Shared library routines.
;    45          0044  1 !
;    46          0045  1 ! AUTHOR: R. Reichert, CREATION DATE: 9-Mar-1983
;    47          0046  1 !
;    48          0047  1 ! MODIFIED BY:
;    49          0048  1 !
;    50          0049  1 ! 1-001 - Original.  Moved out of other modules. STAN 7-Mar-1984.
;    51          0050  1 !--
```

```
1 %SBTTL 'Declarations'
1 !
1 ! SWITCHES:
1 !
1
1 !
1 ! LINKAGES:
1 !
1 !      NONE
1 !
1 ! TABLE OF CONTENTS:
1 !
1
1 FORWARD ROUTINE
1
1 ! Private entry points:
1
1    SMG$$INVALIDATE_DISPLAY,          ! Mark contents of display as unknown
1
1    SMG$$REPORT_CHANGE_INSERT,        ! Report change to physical
1                                      ! screen in insert mode.
1
1    SMG$$REPORT_CHANGE_REPLACE;       ! Report change to physical
1                                      ! screen in replaec mode.
1
1 !
1 ! INCLUDE FILES
1 !
1
1 REQUIRE 'RTLIN:SMGPROLOG';           ! defines psects, macros, tcb,
1                                      !  wcb, & terminal symbols
1
1 !
1 ! EXTERNAL REFERENCES
1 !
1
1 EXTERNAL
1    PBD_L_COUNT,        ! No. of pasteboards we currently have
1
1    PBD_A_PBCB : VECTOR [PBD_K_MAX_PB, LONG],
1                        ! Table of addresses of PBCB's
1
1    PBD_V_PB_AVAIL : BITVECTOR [PBD_K_MAX_PB];
1                        ! Bit vector or pasteboard id numbers in use.
1
1 EXTERNAL ROUTINE
1    LIB$GET_VM,         ! Allocate heap storage
1
1    SMG$INSERT_CHARS,            ! Insert char in virtual display buffer
1                                 ! and cause output.
1
1    SMG$$FILL_WINDOW_BUFFER,     ! Map all virtual display buffers to
1                                 ! the window buffer for a given PBCB
1
1    SMG$$FLUSH_BUFFER,  ! Flush any pending output to terminal
1
1    SMG$$FORCE_SCROLL_REG,       ! Force scroll region to specified
```

```
1                                    ! lines.
1
1      SMG$$LOCATE_PP,         ! Locate pasting packet that joins a virtual
1                              ! display to a pasteboard.
1
1      SMG$$MOVE_TEXT_TO_SCREEN_BUF,
1
1      SMG$$MOVE_TEXT_TO_WINDOW_BUF,          ! Map single virtual display to
1                                             ! window buffer.
1
1      SMG$$OCCLUDE,           ! Determine overlap between two rectangular
1                              ! areas.
1      SMG$$MIN_UPD,           ! Minimum output routine
1
1      SMG$$PUT_TEXT_TO_BUFFER;     ! Text to virtual display buffer
1
1 EXTERNAL LITERAL
1
1      SMG$_FATERRLIB,         ! Fatal error in library procedure
1      SMG$_INVARG,            ! Invalid argument
1      SMG$_INVCOL,            ! Invalid column number
1      SMG$_INVDIS_ID,         ! Invalid virtual display id
1      SMG$_INVPAS_ID,         ! Invalid pasteboard id
1      SMG$_INVROW;            ! Invalid row number
```

```
;  135          0209  1 %SBTTL 'SMG$$INVALIDATE_DISPLAY - Mark display as being privately used'
;  136          0210  1 GLOBAL ROUTINE SMG$$INVALIDATE_DISPLAY ( DISPLAY_ID) =
;  137          0211  1
;  138          0212  1 !++
;  139          0213  1 ! FUNCTIONAL DESCRIPTION:
;  140          0214  1 !
;  141          0215  1 !       This routine is called when ever a change has been completed
;  142          0216  1 !       to a given virtual display and the user had previously
;  143          0217  1 !       written into that display on his own, without using
;  144          0218  1 !       SMG routines.
;  145          0219  1 !
;  146          0220  1 !       The virtual display must not be occluded.
;  147          0221  1 !
;  148          0222  1 !       Each pasteboard to which this display is pasted is isolated and
;  149          0223  1 !       its window image must be redrawn.
;  150          0224  1 !       The new physical pasteboard cursor position gets set.
;  151          0225  1 !
;  152          0226  1 ! CALLING SEQUENCE:
;  153          0227  1 !
;  154          0228  1 !       ret_status.wlc.v = SMG$$INVALIDATE_DISPLAY ( DISPLAY_ID.rl. )
;  155          0229  1 !
;  156          0230  1 ! FORMAL PARAMETERS:
;  157          0231  1 !
;  158          0232  1 !       DISPLAY_ID.rl.r         Display ID of virtual display.
;  159          0233  1 !
;  160          0234  1 ! IMPLICIT INPUTS:
;  161          0235  1 !
;  162          0236  1 !       NONE
;  163          0237  1 !
;  164          0238  1 ! IMPLICIT OUTPUTS:
;  165          0239  1 !
;  166          0240  1 !       NONE
;  167          0241  1 !
;  168          0242  1 ! COMPLETION STATUS:
;  169          0243  1 !
;  170          0244  1 !       SS$_NORMAL      Normal successful completion
;  171          0245  1 !
;  172          0246  1 ! SIDE EFFECTS:
;  173          0247  1 !
;  174          0248  1 !       NONE
;  175          0249  1 !--
;  177        0250  2 BEGIN
;  178        0251  2
;  179        0252  2 LOCAL
;  180        0253  2
;  181        0254  2         DCB     : REF BLOCK [,BYTE],    ! Addr of display control block
;  182        0255  2         CURR_PP : REF BLOCK [,BYTE],    ! Addr of pasting packet under
;  183        0256  2                                         ! inspection
;  184        0257  2
;  185        0258  2         STATUS; ! Status of subroutine calls
;  186        0259  2
;  187        0260  2 !+
;  188        0261  2 ! This routine is independent of buffering.
;  189        0262  2 !-
;  190        0263  2
;  191        0264  2 $SMG$GET_DCB (.DISPLAY_ID, DCB);        ! Get DCB address
;  192        0265  2
;  193        0266  2 CURR_PP = .DCB [DCB_A_PP_NEXT];         ! Start of chain of pasting
;  194        0267  2                                         ! packets to which this display
;  195        0268  2                                         ! is pasted.
;  196        0269  2
;  197        0270  2 !+
;  198        0271  2 ! Deal with each pasteboard that this virtual display is pasted to...
;  199        0272  2 !-
;  200        0273  2
;  201        0274  2 WHILE .CURR_PP NEQ DCB [DCB_A_PP_NEXT]
;  202        0275  2 DO
;  203        0276  3         BEGIN   ! Overall loop
;  204        0277  3
;  205        0278  3         LOCAL
;  206        0279  3             PBCB : REF BLOCK [,BYTE],   ! Address of pasteboard control
;  207        0280  3             WCB  : REF BLOCK [,BYTE],   ! Address of window control block
;  208        0281  3             TO_INDEX;                   ! Index into destination
;  209        0282  3
;  210        0283  3         PBCB = .CURR_PP [PP_A_PBCB_ADDR]; ! Select this pasteboard and WCB
;  211        0284  3         WCB  = .PBCB    [PBCB_A_WCB];     ! whose window needs rebuilding.
;  212        0285  3
;  213        0286  3         TO_INDEX   = .CURR_PP [PP_W_TO_INDEX];
;  214        0287  3
;  215        0288  3         INCR R FROM 1 TO .CURR_PP [PP_W_ROWS_TO_MOVE]
;  216        0289  3         DO
;  217        0290  4                 BEGIN   ! For all rows in this display
;  218        0291  4                 !+
;  219        0292  4                 ! Zero out the display buffer.
;  220        0293  4                 !-
;  221        0294  4                 CH$FILL (0,
;  222        0295  4                          .CURR_PP [PP_W_MOVE_LENGTH],
;  223        0296  4                          .WCB [WCB_A_SCR_TEXT_BUF] + .TO_INDEX);
;  224        0297  4
;  225        0298  4                 TO_INDEX   = .TO_INDEX   + .WCB [WCB_W_NO_COLS];
;  226        0299  3                 END;    ! For all rows to move
;  227        0300  3
;  228        0301  3         CURR_PP = .CURR_PP [PP_A_NEXT_DCB];     ! Walk the DCB side of
;  229        0302  3                                                 ! the chain from front
;  230        0303  3                                                 ! to back.
;  231        0304  2         END;    ! Overall loop
;  232        0305  2
;  233        0306  2 RETURN  SS$_NORMAL
;  234          0307  2
;  235          0308  1 END;                ! End of routine SMG$$INVALIDATE_DISPLAY


                                                                          .TITLE  SMG$$PRVINP SMG$$PRVINP - Private Input support
;                                                                                              routines
                                                                          .IDENT  \1-001\

                                                                          .EXTRN  PBD_L_COUNT, PBD_A_PBCB
                                                                          .EXTRN  PBD_V_PB_AVAIL, LIB$GET_VM
                                                                          .EXTRN  SMG$INSERT_CHARS
                                                                          .EXTRN  SMG$$FILL_WINDOW_BUFFER
                                                                          .EXTRN  SMG$$FLUSH_BUFFER
                                                                          .EXTRN  SMG$$FORCE_SCROLL_REG
                                                                          .EXTRN  SMG$$LOCATE_PP, SMG$$MOVE_TEXT_TO_SCREEN_BUF
                                                                          .EXTRN  SMG$$MOVE_TEXT_TO_WINDOW_BUF
                                                                          .EXTRN  SMG$$OCCLUDE, SMG$$MIN_UPD
                                                                          .EXTRN  SMG$$PUT_TEXT_TO_BUFFER
                                                                          .EXTRN  SMG$_FATERRLIB, SMG$_INVARG
                                                                          .EXTRN  SMG$_INVCOL, SMG$_INVDIS_ID
                                                                          .EXTRN  SMG$_INVPAS_ID, SMG$_INVROW

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

                                                       0FFC 00000         .ENTRY  SMG$$INVALIDATE_DISPLAY, Save R2,R3,R4,R5,- ; 0210
                                                                                  R6,R7,R8,R9,R10,R11                         ;
                                       50       04   BC  D0 00002         MOVL    @DISPLAY_ID, R0                             ; 0264
                                  04   BC       38   A0  D1 00006         CMPL    56(R0), @DISPLAY_ID                         ;
                                                     06  12 0000B         BNEQ    1$                                          ;
                                       11       44   A0  91 0000D         CMPB    68(R0), #17                                 ;
                                                     08  13 00011         BEQL    2$                                          ;
                                       50 00000000G  8F  D0 00013 1$:     MOVL    #SMG$_INVDIS_ID, R0                         ;
                                                         04 0001A         RET                                                 ;
                                       5A       04   BC  D0 0001B 2$:     MOVL    @DISPLAY_ID, DCB                            ;
                                       57       20   AA  D0 0001F         MOVL    32(DCB), CURR_PP                            ; 0266
                                       50       20   AA  9E 00023 3$:     MOVAB   32(DCB), R0                                 ; 0274
                                       50            57  D1 00027         CMPL    CURR_PP, R0                                 ;
                                                     2D  13 0002A         BEQL    6$                                          ;
                                       50       14   A7  D0 0002C         MOVL    20(CURR_PP), PBCB                           ; 0283
                                       56       08   A0  D0 00030         MOVL    8(PBCB), WCB                                ; 0284
                                       58       20   A7  3C 00034         MOVZWL  32(CURR_PP), TO_INDEX                       ; 0286
                                       5B       1C   A7  3C 00038         MOVZWL  28(CURR_PP), R11                            ; 0288
                                                     59  D4 0003C         CLRL    R                                           ; 0298
                                                     10  11 0003E         BRB     5$                                          ;
      22   A7           00             6E            00  2C 00040 4$:     MOVC5   #0, (SP), #0, 34(CURR_PP), @20(WCB)-        ; 0296
                                                14 B648     00046                 [TO_INDEX]
                                       50       06   A6  3C 00049         MOVZWL  6(WCB), R0                                  ; 0298
                                       58            50  C0 0004D         ADDL2   R0, TO_INDEX                                ;
                         EC            59            5B  F3 00050 5$:     AOBLEQ  R11, R, 4$                                  ; 0288
                                       57            67  D0 00054         MOVL    (CURR_PP), CURR_PP                          ; 0301
                                                     CA  11 00057         BRB     3$                                          ; 0274
                                       50            01  D0 00059 6$:     MOVL    #1, R0                                      ; 0306
                                                         04 0005C         RET                                                 ; 0308

; Routine Size:  93 bytes,    Routine Base:  _SMG$CODE + 0000
;  237          0309  1 %SBTTL 'SMG$$REPORT_CHANGE_INSERT - Report change to screen -- insert'
;  238          0310  1 GLOBAL ROUTINE SMG$$REPORT_CHANGE_INSERT (
;  239          0311  1                                   DISPLAY_ID,
;  240          0312  1                                   PASTEBOARD_ID,
;  241          0313  1                                   CHANGED_CHAR,
;  242          0314  1                                   CHANGED_ROW,
;  243          0315  1                                   CHANGED_COL,
;  244          0316  1                                   TERMINATING_CHAR
;  245          0317  1                                 ) =
;  246          0318  1 !++
;  247          0319  1 ! FUNCTIONAL DESCRIPTION:
;  248          0320  1 !
;  249          0321  1 !       This routine is called to report a change made to the physical
;  250          0322  1 !       screen via an input action involving insertion.  The character
;  251          0323  1 !       identified by CHANGED_CHAR has been positioned on the screen in
;  252          0324  1 !       a position corresponding to the virtual display coordinates
;  253          0325  1 !       CHANGED_ROW and CHANGED_COL.
;  254          0326  1 !
;  255          0327  1 !       TERMINATING_CHAR specifies the details of the change.
;  256          0328  1 !
;  257          0329  1 !       If TERMINATING_CHAR has not been specified, then CHANGED_CHAR
;  258          0330  1 !       has been echoed in the position indicated.  The original
;  259          0331  1 !       contents of the line from CHANGED_COL to the right-hand edge of
;  260          0332  1 !       the virtual display need to be shifted to the right by one
;  261          0333  1 !       character.  The new contents of CHANGED_COL+1 through the
;  262          0334  1 !       right-hand edge need to be redisplayed on the screen.
;  263          0335  1 !       Current cursor position is changed to reflect that one beyond
;  264          0336  1 !       that indicated by CHANGED_COL.  If this new position is now
;  265          0337  1 !       beyond the bounds of the virtual display's dimensions
;  266          0338  1 !       then ?????.
;  267          0339  1 !
;  268          0340  1 !       If TERMINATING_CHAR was specified, CHANGED_CHAR has been echoed
;  269          0341  1 !       in the position indicated.  Futhermore, this character was
;  270          0342  1 !       followed by a TERMINATING_CHAR.  The original contents of the
;  271          0343  1 !       line from CHANGED_COL to the right-hand edge of the virtual
;  272          0344  1 !       display need to be shifted to the right by one character.  The
;  273          0345  1 !       new contents of CHANGED_COL+1 through the right-hand edge need
;  274          0346  1 !       to be redisplayed on the screen.
;  275          0347  1 !       Current cursor position is changed to reflect the effects of
;  276          0348  1 !       the TERMINATING_CHAR.  If this new position is now beyond the
;  277          0349  1 !       bounds of the virtual display's dimensions then ????.
;  278          0350  1 !
;  279          0351  1 ! CALLING SEQUENCE:
;  280          0352  1 !
;  281          0353  1 !       ret_status.wlc.v = SMG$$REPORT_CHANGE_INSERT (
;  282          0354  1 !                                       DISPLAY_ID.rl.r,
;  283          0355  1 !                                       PASTEBOARD_ID.rl.r,
;  284          0356  1 !                                       CHANGED_CHAR.rb.r,
;  285          0357  1 !                                       CHANGED_ROW.rl.r,
;  286          0358  1 !                                       CHANGED_COL.rl.r
;  287          0359  1 !                                       [,TERMINATING_CHAR.rb.r])
;  288          0360  1 ! FORMAL PARAMETERS:
;  289          0361  1 !
;  290          0362  1 !       DISPLAY_ID.rl.r         Display id of virtual display.
;  291          0363  1 !
;  292          0364  1 !       PASTEBOARD_ID.rl.r      Pasteboard id.
;  293          0365  1 !
```

```
1 !         CHANGED_CHAR.rb.r       The character that modified the screen.
1 !
1 !         CHANGED_ROW.rl.r        Row number within the virtual display in
1 !                                 which CHANGED_CHAR was written.
1 !
1 !         CHANGED_COL.rl.r        Column number within the virtual display
1 !                                 where CHANGED_CHAR was written.
1 !
1 !         [,TERMINATING_CHAR.rb.r]        [Optional].
1 !                                 If supplied, the terminating character
1 !                                 that followed CHANGED_CHAR  (See
1 !                                 functional description for meaning).
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !         NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !         NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !         SS$_NORMAL      Normal successful completion
1 !         SMG$_INVDIS_ID  Invalid Display Id
1 !         SMG$_INVPAS_ID  Invalid Pasteboard Id
1 !         SMG$_INVROW     Invalid row specified
1 !         SMG$_INVCOL     Invalid column specified
1 !
1 ! SIDE EFFECTS:
1 !
1 !         NONE
1 !--
1
2     BEGIN
2     BUILTIN
2         NULLPARAMETER;
2
2     LOCAL
2         DESC : BLOCK [8,BYTE],          ! Local descriptor
2         STATUS,                         ! Status of subroutine calls
2         DCB  : REF $DCB_DECL,           ! Addr of display control block
2         PBCB : REF $PBCB_DECL,          ! Addr of pasteboard control
2                                         ! block.
2         PP   : REF $PP_DECL;            ! Addr of pasting packet
2
2     $SMG$VALIDATE_ARGCOUNT (5, 6);      ! Test for right no. of args
2
2 !+
2 ! Get addresses of control blocks we need
2 !-
2     $SMG$GET_DCB (.DISPLAY_ID, DCB);                        ! Get DCB addr.
2     $SMG$GET_PBCB ( .PASTEBOARD_ID, PBCB) ;                 ! Get PBCB addr.
3     IF NOT (STATUS = SMG$$LOCATE_PP (.DCB, .PBCB, PP)) ! Get PP addr.
2     THEN
2         RETURN (.STATUS);
2
```

```
2       $SMG$VALIDATE_ROW_COL (..CHANGED_ROW, ..CHANGED_COL); ! Valid Pos.?
2
2   !+
2   ! Initialize our local descriptor to point to the changed character.
2   !-
2       DESC [DSC$W_LENGTH] = 1;
2       DESC [DSC$B_CLASS]  = DSC$K_CLASS_S;
2       DESC [DSC$B_DTYPE]  = DSC$K_DTYPE_T;
2       DESC [DSC$A_POINTER]= .CHANGED_CHAR;
2
2   !+
2   ! Reflect this change in the virtual display text and attribute buffer,
2   ! including new virtual display cursor position.
2   ! Data from the affected column to the last-1 column of this line need
2   ! to be moved one character position to the right and the changed
2   ! character inserted at the indicated position.  The attributes for the
2   ! moved bytes must be moved as well.
2   !-
3       IF NOT (STATUS = SMG$INSERT_CHARS ( .DISPLAY_ID,
3                                           .CHANGED_ROW,
3                                           .CHANGED_COL,
3                                          DESC))
2       THEN
2           RETURN (.STATUS);
2
2       IF NOT NULLPARAMETER (6)
2       THEN
3           BEGIN   ! Terminator supplied
3           !+
3           ! Inspect supplied terminator to determine effect on cursor
3           ! position in virtual display.
3           !-
3           !+
3           ! ***** For now pretend the terminator is a <CR>. *****
3           ! ***** This needs to act like a <CR><LF> pair.
3           !-
4           IF NOT (STATUS = SMG$$PUT_TEXT_TO_BUFFER (
4                                   .DCB,
4                                   .DCB [DCB_B_DEF_VIDEO_ATTR],
4                                   1,
4                                   %REF (CR^8 + LF),       ! <CR><LF>
4                                   .DCB [DCB_B_DEF_CHAR_SET]))
3           THEN
3               RETURN (.STATUS);
2           END;    ! Terminator supplied
2
2   !+
2   ! Reflect this change in the appropriate positions of the window screen
2   ! text and attribute buffers, including new screen cursor position.
2   !-
3       IF NOT (STATUS = SMG$$MOVE_TEXT_TO_WINDOW_BUF ( .PP))
2       THEN
2           RETURN (.STATUS);
2
2   !+
2   ! Record what has happened to screen buffer as well.
2   !-
```

```
3       IF NOT (STATUS = SMG$$MOVE_TEXT_TO_SCREEN_BUF ( .PP))
2       THEN
2           RETURN (.STATUS);
2
2   !+
2   ! Must now force the changes to be output.
2   !-
2       IF .PP [PP_W_ROWS_TO_MOVE] NEQ 0
2       THEN
3           BEGIN
3           ! Assume damage confined to single row.
3           PBCB [PBCB_W_FIRST_CHANGED_ROW] = ..CHANGED_ROW;
3           PBCB [PBCB_W_LAST_CHANGED_ROW]  = ..CHANGED_ROW;
3
3           . Assume damage in row from given pos. to end of line
3           PBCB [PBCB_W_FIRST_CHANGED_COL] = ..CHANGED_COL;
3           PBCB [PBCB_W_LAST_CHANGED_COL]  = .PBCB [PBCB_W_WIDTH];
2           END;
2
2       STATUS = SMG$$MIN_UPD ( .PBCB);
2
2   !+
2   ! If this virtual display is pasted to pasteboards other than the one
2   ! identified in the call list, these additional pasteboard's window
2   ! buffers must be updated as well.  For these additional pasteboards,
2   ! the changed byte in addition to the shifted remainder of the line
2   ! must be output -- since they did not receive the originally echoed
2   ! character.
2   !-
2       PP = .DCB [DCB_A_PP_NEXT];            ! 1st in chain
2       WHILE .PP NEQ .DCB [DCB_A_PP_NEXT]    ! While any packets remain...
2       DO
3           BEGIN   ! Loop through all pasting packets for this DCB
3           LOCAL
3               NEW_PBCB : REF $PBCB_DECL;    ! PBCB being considered
3
3           NEW_PBCB = .PP [PP_A_PBCB_ADDR];          ! PBCB for this packet
3           IF .NEW_PBCB NEQ .PBCB  ! If this isn't the one we started with
3           THEN
4               BEGIN       ! Needs to be output
5               IF NOT (STATUS = SMG$$FILL_WINDOW_BUFFER (.NEW_PBCB))
4               THEN
4                   RETURN (.STATUS);
4
4               IF .PP [PP_W_ROWS_TO_MOVE] NEQ 0
4               THEN
5                   BEGIN
5                   ! Assume damage confined to single row.
5                   PBCB [PBCB_W_FIRST_CHANGED_ROW] = ..CHANGED_ROW;
5                   PBCB [PBCB_W_LAST_CHANGED_ROW]  = ..CHANGED_ROW;
5
5                   ! Assume damage in row from given pos. to end of line
5                   PBCB [PBCB_W_FIRST_CHANGED_COL] = ..CHANGED_COL;
5                   PBCB [PBCB_W_LAST_CHANGED_COL]  = .PBCB [PBCB_W_WIDTH];
4                   END;
4
5               IF NOT (STATUS = SMG$$MIN_UPD ( .NEW_PBCB))
```

```
;  465          0537  4              THEN
;  466          0538  4                  RETURN (.STATUS);
;  467          0539  3              END;        ! Needs to be output
;  468          0540  3
;  469          0541  3          PP = .PP [PP_A_NEXT_DCB];       ! Step to next packet in chain
;  470          0542  2          END;    ! Loop through all pasting packets for this DCB
;  471          0543  2
;  472          0544  2      RETURN (SS$_NORMAL);
;  473          0545  1      END;                 ! End of routine SMG$$REPORT_CHANGE_INSERT



                                                                           .EXTRN  SMG$_WRONUMARG

                                                        01FC 00000         .ENTRY  SMG$$REPORT_CHANGE_INSERT, Save R2,R3,R4,-   ; 0310
                                                                                   R5,R6,R7,R8                                  ;
                                       58 00000000G   00  9E 00002         MOVAB   SMG$$MIN_UPD, R8                             ;
                                       5E             10  C2 00009         SUBL2   #16, SP                                      ;
                         50            6C             05  83 0000C         SUBB3   #5, (AP), DIFF                               ; 0412
                                       01             50  91 00010         CMPB    DIFF, #1                                     ;
                                                      08  1B 00013         BLEQU   1$                                           ;
                                       50 00000000G   8F  D0 00015         MOVL    #SMG$_WRONUMARG, R0                          ;
                                                          04 0001C         RET                                                  ;
                                       50        04   BC  D0 0001D 1$:     MOVL    @DISPLAY_ID, R0                              ; 0417
                                  04   BC        38   A0  D1 00021         CMPL    56(R0), @DISPLAY_ID                          ;
                                                      06  12 00026         BNEQ    2$                                           ;
                                       11        44   A0  91 00028         CMPB    68(R0), #17                                  ;
                                                      08  13 0002C         BEQL    3$                                           ;
                                       50 00000000G   8F  D0 0002E 2$:     MOVL    #SMG$_INVDIS_ID, R0                          ;
                                                          04 00035         RET                                                  ;
                                       54        04   BC  D0 00036 3$:     MOVL    @DISPLAY_ID, DCB                             ;
                                       50        08   BC  D0 0003A         MOVL    @PASTEBOARD_ID, R0                           ; 0418
                                                      11  19 0003E         BLSS    4$                                           ;
                            00000000G  00             50  D1 00040         CMPL    R0, PBD_L_COUNT                              ;
                                                      08  14 00047         BGTR    4$                                           ;
                         08 00000000G  00             50  E0 00049         BBS     R0, PBD_V_PB_AVAIL, 5$                       ;
                                       50 00000000G   8F  D0 00051 4$:     MOVL    #SMG$_INVPAS_ID, R0                          ;
                                                          04 00058         RET                                                  ;
                                       52 00000000G0040   D0 00059 5$:     MOVL    PBD_A_PBCB[R0], PBCB                         ;
                                                 04   AE  9F 00061         PUSHAB  PP                                           ; 0419
                                                      52  DD 00064         PUSHL   PBCB                                         ;
                                                      54  DD 00066         PUSHL   DCB                                          ;
                            00000000G  00             03  FB 00068         CALLS   #3, SMG$$LOCATE_PP                           ;
                                       76             50  E9 0006F         BLBC    STATUS, 10$                                  ;
                                       56        10   AC  D0 00072         MOVL    CHANGED_ROW, R6                              ; 0423
                                                      66  D5 00076         TSTL    (R6)                                         ;
                                                      08  15 00078         BLEQ    6$                                           ;
           66       02   A4            10             00  ED 0007A         CMPZV   #0, #16, 2(DCB), (R6)                        ;
                                                      08  18 00080         BGEQ    7$                                           ;
                                       50 00000000G   8F  D0 00082 6$:     MOVL    #SMG$_INVROW, R0                             ;
                                                          04 00089         RET                                                  ;
                                       57        14   AC  D0 0008A 7$:     MOVL    CHANGED_COL, R7                              ;
                                                      67  D5 0008E         TSTL    (R7)                                         ;
                                                      08  15 00090         BLEQ    8$                                           ;
           67       06   A4            10             00  ED 00092         CMPZV   #0, #16, 6(DCB), (R7)                        ;
                                                      08  18 00098         BGEQ    9$                                           ;
                                       50 00000000G   8F  D0 0009A 8$:     MOVL    #SMG$_INVCOL, R0                             ;
```

```
                                   04 000A1        RET                                                  ;
          08   AE 010E0001    8F   D0 000A2 9$:    MOVL     #17694721, DESC                             ; 0428
          0C   AE       0C    AC   D0 000AA        MOVL     CHANGED_CHAR, DESC+4                        ; 0431
                        08    AE   9F 000AF        PUSHAB   DESC                                        ; 0441
               7E             56   7D 000B2        MOVQ     R6, -(SP)                                   ; 0441
                        04    AC   DD 000B5        PUSHL    DISPLAY_ID                                  ; 0441
     00000000G 00             04   FB 000B8        CALLS    #4, SMG$INSERT_CHARS                        ;
               42             50   E9 000BF        BLBC     STATUS, 12$                                 ;
               06             6C   91 000C2        CMPB     (AP), #6                                    ; 0448
                              24   1F 000C5        BLSSU    11$                                         ;
                        18    AC   D5 000C7        TSTL     24(AP)                                      ;
                              1F   13 000CA        BEQL     11$                                         ;
               7E       30    A4   9A 000CC        MOVZBL   48(DCB), -(SP)                              ; 0464
          04   AE     0D0A    8F   3C 000D0        MOVZWL   #3338, 4(SP)                                ; 0463
                        04    AE   9F 000D6        PUSHAB   4(SP)                                       ;
                              01   DD 000D9        PUSHL    #1                                          ; 0459
               7E       2E    A4   9A 000DB        MOVZBL   46(DCB), -(SP)                              ; 0461
                              54   DD 000DF        PUSHL    DCB                                         ; 0460
     00000000G 00             05   FB 000E1        CALLS    #5, SMG$$PUT_TEXT_TO_BUFFER                 ;
               5F             50   E9 000E8 10$:   BLBC     STATUS, 15$                                 ; 0459
               53       04    AE   D0 000EB 11$:   MOVL     PP, R3                                      ; 0473
                              53   DD 000EF        PUSHL    R3                                          ;
     00000000G 00             01   FB 000F1        CALLS    #1, SMG$$MOVE_TEXT_TO_WINDOW_BUF            ;
               7D             50   E9 000F8        BLBC     STATUS, 19$                                 ;
                              53   DD 000FB        PUSHL    R3                                          ; 0480
     00000000G 00             01   FB 000FD        CALLS    #1, SMG$$MOVE_TEXT_TO_SCREEN_BUF            ;
               71             50   E9 00104 12$:   BLBC     STATUS, 19$                                 ;
                        1C    A3   B5 00107        TSTW     28(R3)                                      ; 0487
                              15   13 0010A        BEQL     13$                                         ;
          00A8 C2             66   B0 0010C        MOVW     (R6), 168(PBCB)                             ; 0491
          00AA C2             66   B0 00111        MOVW     (R6), 170(PBCB)                             ; 0492
          00AC C2             67   B0 00116        MOVW     (R7), 172(PBCB)                             ; 0495
          00AE C2       5A    A2   B0 0011B        MOVW     90(PBCB), 174(PBCB)                         ; 0496
                              52   DD 00121 13$:   PUSHL    PBCB                                        ; 0499
               68             01   FB 00123        CALLS    #1, SMG$$MIN_UPD                            ;
          04   AE       20    A4   D0 00126        MOVL     32(DCB), PP                                 ; 0509
               53       04    AE   D0 0012B 14$:   MOVL     PP, R3                                      ; 0510
               51       20    A4   9E 0012F        MOVAB    32(DCB), R1                                 ;
               51             53   D1 00133        CMPL     R3, R1                                      ;
                              3D   13 00136        BEQL     18$                                         ;
               55       14    A3   D0 00138        MOVL     20(R3), NEW_PBCB                            ; 0516
               52             55   D1 0013C        CMPL     NEW_PBCB, PBCB                              ; 0517
                              2E   13 0013F        BEQL     17$                                         ;
                              55   DD 00141        PUSHL    NEW_PBCB                                    ; 0520
     00000000G 00             01   FB 00143        CALLS    #1, SMG$$FILL_WINDOW_BUFFER                 ;
               2B             50   E9 0014A 15$:   BLBC     STATUS, 19$                                 ;
                        1C    A3   B5 0014D        TSTW     28(R3)                                      ; 0524
                              15   13 00150        BEQL     16$                                         ;
          00A8 C2             66   B0 00152        MOVW     (R6), 168(PBCB)                             ; 0528
          00AA C2             66   B0 00157        MOVW     (R6), 170(PBCB)                             ; 0529
          00AC C2             67   B0 0015C        MOVW     (R7), 172(PBCB)                             ; 0532
          00AE C2       5A    A2   B0 00161        MOVW     90(PBCB), 174(PBCB)                         ; 0533
                              55   DD 00167 16$:   PUSHL    NEW_PBCB                                    ; 0536
               68             01   FB 00169        CALLS    #1, SMG$$MIN_UPD                            ;
               09             50   E9 0016C        BLBC     STATUS, 19$                                 ;
          04   AE             63   D0 0016F 17$:   MOVL     (R3), PP                                    ; 0541
                              B6   11 00173        BRB      14$                                         ; 0510
```

```
                         50          01  D0 00175 18$:     MOVL    #1, R0                                               ; 0544
                                         04 00178 19$:     RET                                                          ; 0545

; Routine Size:  377 bytes,    Routine Base:  _SMG$CODE + 005D

;  474          0546  1 !<BLF/PAGE>
```

```
1 %SBTTL 'SMG$$REPORT_CHANGE_REPLACE - Report change to screen -- replace'
1 GLOBAL ROUTINE SMG$$REPORT_CHANGE_REPLACE (
1                                   DISPLAY_ID,
1                                   PASTEBOARD_ID,
1                                   NUM_CHARS     : REF VECTOR [,WORD],
1                                   CHANGED_CHARS : REF VECTOR [,BYTE],
1                                   CHANGED_ROW,
1                                   CHANGED_COL,
1                                   TERMINATING_CHAR
1                                 ) =
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine is called to report a change made to the physical
1 !       screen via an input action involving replacement.  The
1 !       characters identified by CHANGED_CHARS and NUM_CHARS have been
1 !       positioned on the screen in a position corresponding to the
1 !       virtual display coordinates CHANGED_ROW and CHANGED_COL.
1 !
1 !       TERMINATING_CHAR specifies the details of the change.
1 !
1 !       If TERMINATING_CHAR has not been specified, then the changed
1 !       characters have been echoed in the position indicated.  No
1 !       further output is required.  Just need to update internal data
1 !       bases to reflect that change.  Current cursor position is
1 !       changed to reflect that beyond the changed text.  If this new
1 !       position is now beyond the bounds of the virtual display's
1 !       dimensions then ?????.
1 !
1 !       If TERMINATING_CHAR was specified, the changed characters have
1 !       been echoed in the positions indicated.  Futhermore, these
1 !       character were followed by a TERMINATING_CHAR.  Just need to
1 !       update internal data bases to reflect that change.  Current
1 !       cursor position is changed to reflect the effects of the
1 !       TERMINATING_CHAR.  If this new position is now beyond the
1 !       bounds of the virtual display's dimensions then ????.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !       ret_status.wlc.v = SMG$$REPORT_CHANGE_REPLACE (
1 !                                       DISPLAY_ID.rl.r,
1 !                                       PASTEBOARD_ID.rl.r,
1 !                                       NUM_CHARS.rwu.r,
1 !                                       CHANGED_CHARS.rab.r
1 !                                       [,CHANGED_ROW.rl.r]
1 !                                       [,CHANGED_COL.rl.r]
1 !                                       [,TERMINATING_CHAR.rb.r])
1 ! FORMAL PARAMETERS:
1 !
1 !       DISPLAY_ID.rl.r         Display id of virtual display.
1 !
1 !       PASTEBOARD_ID.rl.r      Pasteboard id.
1 !
1 !       NUM_CHARS.rwu.r         The number of characters that changed.
1 !
1 !       CHANGED_CHARS.rab.r     Address of the characters that modified
1 !                               the screen.
```

```
1 !
1 !         CHANGED_ROW.rl.r        Row number within the virtual display in
1 !                                 which CHANGED_CHARS were written.
1 !
1 !         CHANGED_COL.rl.r        Column number within the virtual display
1 !                                 where CHANGED_CHARS were written.
1 !
1 !         [,TERMINATING_CHAR.rb.r]        [Optional].
1 !                                 If supplied, the terminating character
1 !                                 that followed CHANGED_CHAR  (See
1 !                                 functional description for meaning).
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !         NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !         NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !         SS$_NORMAL      Normal successful completion
1 !         SMG$_INVDIS_ID  Invalid Display Id
1 !         SMG$_INVPAS_ID  Invalid Pasteboard Id
1 !         SMG$_INVROW     Invalid row specified
1 !         SMG$_INVCOL     Invalid co.umn specified
1 !
1 ! SIDE EFFECTS:
1 !
1 !         NONE
1 !--
1
2     BEGIN
2     BUILTIN
2         NULLPARAMETER;
2
2     LOCAL
2         STATUS,                         ! Status of subroutine calls
2         C_ROW,                          ! Working row
2         C_COL,                          ! Working col
2         DCB  : REF $DCB_DECL,           ! Addr of display control block
2         PBCB : REF BLOCK [,BYTE],       ! Addr of pasteboard control
2                                         ! block.
2         WCB  : REF BLOCK [,BYTE],       ! Address of window block
2         PP   : REF BLOCK [,BYTE];       ! Addr of pasting packet.
2
2     $SMG$VALIDATE_ARGCOUNT (4, 7);      ! Test for right no. of args
2
2 !+
2 ! Get addresses of control blocks needed.
2 !-
2     $SMG$GET_DCB (.DISPLAY_ID, DCB);                    ! Get DCB addr.
2     $SMG$GET_PBCB ( .PASTEBOARD_ID, PBCB) ;             ! Get PBCB addr.
3     IF NOT (STATUS = SMG$$LOCATE_PP (.DCB, .PBCB, PP)) ! Get PP addr.
2     THEN
2         RETURN (.STATUS);
```

```
2
2       IF NOT NULLPARAMETER (5)
2       THEN
3           BEGIN
3           C_ROW = ..CHANGED_ROW;
3           DCB [DCB_W_CURSOR_ROW] = .C_ROW;
3           END
2       ELSE
2           C_ROW = .DCB [DCB_W_CURSOR_ROW];
2
2       IF NOT NULLPARAMETER (6)
2       THEN
3           BEGIN
3           C_COL = ..CHANGED_COL;
3           DCB [DCB_W_CURSOR_COL] = .C_COL;
3           END
2       ELSE
2           C_COL = .DCB [DCB_W_CURSOR_COL];
2
2       $SMG$VALIDATE_ROW_COL (.C_ROW, .C_COL); ! Valid posit?
2
2   !+
2   ! Reflect this change in the virtual display text and attribute buffers.
2   !-
2
2       !+
2       ! Invalidate physical cursor position.
2       ! This will force output to begin with a direct cursor
2       ! movement to the proper place.
2       !-
2
2       WCB = .PBCB [PBCB_A_WCB];
2       WCB [WCB_W_OLD_CUR_ROW] = 0;
2       WCB [WCB_W_OLD_CUR_COL] = 0;
2
3       IF NOT (STATUS = SMG$$PUT_TEXT_TO_BUFFER (
3                                   .DCB,
3                                   .DCB [DCB_B_DEF_VIDEO_ATTR],
3                                   .NUM_CHARS[0],
3                                   .CHANGED_CHARS,
3                                   .DCB [DCB_B_DEF_CHAR_SET]))
2       THEN
2           RETURN (.STATUS);
2
2       IF NOT NULLPARAMETER (7)
2       THEN
3           BEGIN   ! Terminator supplied
3           !+
3           ! Inspect supplied terminator to determine effect on cursor
3           ! position in virtual display.
3           !-
4           IF NOT (STATUS = SMG$$PUT_TEXT_TO_BUFFER (
4                                   .DCB,
4                                   .DCB [DCB_B_DEF_VIDEO_ATTR],
4                                   1,
4                                   %REF (CR^8 + LF),       ! <CR><LF>
4                                   .DCB [DCB_B_DEF_CHAR_SET]))
```

```
3         THEN
3             RETURN (.STATUS);
2         END;    ! Terminator supplied
2
2 !+
2 ! Reflect this change in the appropriate positions of the window
2 ! text and attribute buffers, including new screen cursor position.
2 !-
3     IF NOT (STATUS = SMG$$MOVE_TEXT_TO_WINDOW_BUF ( .PP))
2     THEN
2         RETURN (.STATUS);
2
2 !+
2 ! Update screen buffers as well.
2 !-
3     IF NOT (STATUS = SMG$$MOVE_TEXT_TO_SCREEN_BUF ( .PP))
2     THEN
2         RETURN (.STATUS);
2
2 !+
2 ! If this virtual display is pasted to pasteboards other than the one
2 ! identified in the call list, these additional pasteboard's window
2 ! buffers must be updated as well.  For these additional pasteboards,
2 ! the changed bytes must be output -- since they did not receive the
2 ! originally echoed characters.
2 !-
2     PP = .DCB [DCB_A_PP_NEXT];          ! 1st in chain
2     WHILE .PP NEQ DCB [DCB_A_PP_NEXT]   ! While any packets remain...
2     DO
3         BEGIN   ! Loop through all pasting packets for this DCB
3         LOCAL
3             NEW_PBCB : REF $PBCB_DECL;    ! PBCB being considered
3
3         NEW_PBCB = .PP [PP_A_PBCB_ADDR];        ! PBCB for this packet
3         IF .NEW_PBCB NEQ .PBCB  ! If this isn't the one we started with
3         THEN
4             BEGIN       ! Needs to be output
5             IF NOT (STATUS = SMG$$FILL_WINDOW_BUFFER (.NEW_PBCB))
4             THEN
4                 RETURN (.STATUS);
4
4             IF .PP [PP_W_ROWS_TO_MOVE] NEQ 0
4             THEN
5                 BEGIN
5                 ! Assume damage confined to single row.
5                 NEW_PBCB [PBCB_W_FIRST_CHANGED_ROW] = .C_ROW;
5                 NEW_PBCB [PBCB_W_LAST_CHANGED_ROW]  = .C_ROW;
5
5                 ! Assume damage in row from given pos. to end of line
5                 NEW_PBCB [PBCB_W_FIRST_CHANGED_COL] = .C_COL;
5                 NEW_PBCB [PBCB_W_LAST_CHANGED_COL]  =
5                                         .NEW_PBCB [PBCB_W_WIDTH];
4                 END;
4
5             IF NOT (STATUS = SMG$$MIN_UPD (.NEW_PBCB))
4             THEN
4                 RETURN (.STATUS);
```

```
;  704          0775  4
;  705          0776  3               END;          ! Needs to be output
;  706          0777  3
;  707          0778  3            PP = .PP [PP_A_NEXT_DCB];        ! Step to next packet in chain
;  708          0779  2            END;    ! Loop through all pasting packets for this DCB
;  709          0780  2
;  710          0781  2        RETURN (SS$_NORMAL);
;  711          0782  1        END;                 ! End of routine SMG$$REPORT_CHANGE_REPLACE



                                          01FC 00000          .ENTRY  SMG$$REPORT_CHANGE_REPLACE, Save R2,R3,R4,-  ; 0548
                                                                      R5,R6,R7,R8                                   ;
                      58 00000000G   00   9E 00002            MOVAB   SMG$$PUT_TEXT_TO_BUFFER, R8                   ;
                      5E             08   C2 00009            SUBL2   #8, SP                                        ;
       50             6C             04   83 0000C            SUBB3   #4, (AP), DIFF                                ; 0651
                      03             50   91 00010            CMPB    DIFF, #3                                      ;
                                     08   1B 00013            BLEQU   1$                                            ;
                      50 00000000G   8F   D0 00015            MOVL    #SMG$_WRONUMARG, R0                           ;
                                     04   0001C               RET                                                   ;
                      50        04   BC   D0 0001D 1$:        MOVL    @DISPLAY_ID, R0                               ; 0656
              04      BC        38   A0   D1 00021            CMPL    56(R0), @DISPLAY_ID                           ;
                                     06   12 00026            BNEQ    2$                                            ;
                      11        44   A0   91 00028            CMPB    68(R0), #17                                   ;
                                     08   13 0002C            BEQL    3$                                            ;
                      50 00000000G   8F   D0 0002E 2$:        MOVL    #SMG$_INVDIS_ID, R0                           ;
                                     04   00035               RET                                                   ;
                      53        04   BC   D0 00036 3$:        MOVL    @DISPLAY_ID, DCB                              ;
                      50        08   BC   D0 0003A            MOVL    @PASTEBOARD_ID, R0                            ; 0657
                                     11   19 0003E            BLSS    4$                                            ;
         00000000G    00             50   D1 00040            CMPL    R0, PBD_L_COUNT                               ;
                                     08   14 00047            BGTR    4$                                            ;
       08 00000000G   00             50   E0 00049            BBS     R0, PBD_V_PB_AVAIL, 5$                        ;
                      50 00000000G   8F   D0 00051 4$:        MOVL    #SMG$_INVPAS_ID, R0                           ;
                                     04   00058               RET                                                   ;
                      55 00000000G0040    D0 00059 5$:        MOVL    PBD_A_PBCB[R0], PBCB                          ;
                                04   AE   9F 00061            PUSHAB  PP                                            ; 0658
                                     28   BB 00064            PUSHR   #^M<R3,R5>                                    ;
         00000000G    00             03   FB 00066            CALLS   #3, SMG$$LOCATE_PP                            ;
                      73             50   E9 0006D            BLBC    STATUS, 14$                                   ;
                      05             6C   91 00070            CMPB    (AP), #5                                      ; 0662
                                     0F   1F 00073            BLSSU   6$                                            ;
                                14   AC   D5 00075            TSTL    20(AP)                                        ;
                                     0A   13 00078            BEQL    6$                                            ;
                      56        14   BC   D0 0007A            MOVL    @CHANGED_ROW, C_ROW                           ; 0665
              28      A3             56   B0 0007E            MOVW    C_ROW, 40(DCB)                                ; 0666
                                     04   11 00082            BRB     7$                                            ; 0662
                      56        28   A3   3C 00084 6$:        MOVZWL  40(DCB), C_ROW                                ; 0669
                      06             6C   91 00088 7$:        CMPB    (AP), #6                                      ; 0671
                                     0F   1F 0008B            BLSSU   8$                                            ;
                                18   AC   D5 0008D            TSTL    24(AP)                                        ;
                                     0A   13 00090            BEQL    8$                                            ;
                      57        18   BC   D0 00092            MOVL    @CHANGED_COL, C_COL                           ; 0674
              2A      A3             57   B0 00096            MOVW    C_COL, 42(DCB)                                ; 0675
                                     04   11 0009A            BRB     9$                                            ; 0671
```

```
                                        57        2A   A3  3C 0009C 8$:      MOVZWL  42(DCB), C_COL                                ; 0678
                                                       56  D5 000A0 9$:      TSTL    C_ROW                                         ; 0680
                                                       08  15 000A2          BLEQ    10$                                           :
            56       02   A3            10             00  ED 000A4          CMPZV   #0, #16, 2(DCB), C_ROW                        :
                                                       08  18 000AA          BGEQ    11$                                           :
                                        50 00000000G   8F  D0 000AC 10$:     MOVL    #SMG$_INVROW, R0                              :
                                                           04 000B3          RET                                                   :
                                                       57  D5 000B4 11$:     TSTL    C_COL                                         :
                                                       08  15 000B6          BLEQ    12$                                           :
            57       06   A3            10             00  ED 000B8          CMPZV   #0, #16, 6'DCB), C_COL                        :
                                                       08  18 000BE          BGEQ    13$                                           :
                                        50 00000000G   8F  D0 000C0 12$:     MOVL    #SMG$_INVCOL, R0                              :
                                                           04 000C7          RET                                                   :
                                        51        08   A5  D0 000C8 13$:     MOVL    8(PBCB), WCB                                  ; 0692
                                                  24   A1  D4 000CC          CLRL    36(WCB)                                       ; 0693
                                        7E        30   A3  9A 000CF          MOVZBL  48(DCB), -(SP)                                ; 0701
                                                  10   AC  DD 000D3          PUSHL   CHANGED_CHARS                                 ; 0700
                                        7E        0C   BC  3C 000D6          MOVZWL  @NUM_CHARS, -(SP)                             ; 0699
                                        7E        2E   A3  9A 000DA          MOVZBL  46(DCB), -(SP)                                ; 0698
                                                       53  DD 000DE          PUSHL   DCB                                           ; 0697
                                        68             05  FB 000E0          CALLS   #5, SMG$$PUT_TEXT_TO_BUFFER                   :
                                        63             50  E9 000E3 14$:     BLBC    STATUS, 17$                                   ; 0696
                                        07             6C  91 000E6          CMPB    (AP), #7                                      ; 0705
                                                       2C  1F 000E9          BLSSU   15$                                           :
                                                  1C   AC  D5 000EB          TSTL    28(AP)                                        :
                                                       1B  13 000EE          BEQL    15$                                           :
                                        7E        30   A3  9A 000F0          MOVZBL  48(DCB), -(SP)                                ; 0717
                                   04   AE      0D0A   8F  3C 000F4          MOVZWL  #3338, 4(SP)                                  ; 0716
                                                  04   AE  9F 000FA          PUSHAB  4(SP)                                         :
                                                       01  DD 000FD          PUSHL   #1                                            ; 0712
                                        7E        2E   A3  9A 000FF          MOVZBL  46(DCB), -(SP)                                ; 0714
                                                       53  DD 00103          PUSHL   DCB                                           ; 0713
                                        68             05  FB 00105          CALLS   #5, SMG$$PUT_TEXT_TO_BUFFER                   :
                                        70             50  E9 00108          BLBC    STATUS, 21$                                   ; 0712
                                                  04   AE  DD 0010B 15$:     PUSHL   PP                                            ; 0726
                             00000000G  00             01  FB 0010E          CALLS   #1, SMG$$MOVE_TEXT_TO_WINDOW_BUF              :
                                        63             50  E9 00115          BLBC    STATUS, 21$                                   :
                                                  04   AE  DD 00118          PUSHL   PP                                            ; 0733
                             00000000G  00             01  FB 0011B          CALLS   #1, SMG$$MOVE_TEXT_TO_SCREEN_BUF              :
                                        56             50  E9 00122          BLBC    STATUS, 21$                                   :
                                   04   AE        20   A3  D0 00125          MOVL    32(DCB), PP                                   ; 0744
                                        54        04   AE  D0 0012A 16$:     MOVL    PP, R4                                        ; 0745
                                        51        20   A3  9E 0012E          MOVAB   32(DCB), R1                                   :
                                        51             54  D1 00132          CMPL    R4, R1                                        :
                                                       41  13 00135          BEQL    20$                                           :
                                        52        14   A4  D0 00137          MOVL    20(R4), NEW_PBCB                              ; 0751
                                        55             52  D1 0013B          CMPL    NEW_PBCB, PBCB                                ; 0752
                                                       32  13 0013E          BEQL    19$                                           :
                                                       52  DD 00140          PUSHL   NEW_PBCB                                      ; 0755
                             00000000G  00             01  FB 00142          CALLS   #1, SMG$$FILL_WINDOW_BUFFER                   :
                                        2F             50  E9 00149 17$:     BLBC    STATUS, 21$                                   :
                                                  1C   A4  B5 0014C          TSTW    28(R4)                                        ; 0759
                                                       15  13 0014F          BEQL    18$                                           :
                                  00A8  C2             56  B0 00151          MOVW    C_ROW, 168(NEW_PBCB)                          ; 0763
                                  00AA  C2             56  B0 00156          MOVW    C_ROW, 170(NEW_PBCB)                          ; 0764
                                  00AC  C2             57  B0 0015B          MOVW    C_COL, 172(NEW_PBCB)                          ; 0767
                                  00AE  C2        5A   A2  B0 00160          MOVW    90(NEW_PBCB), 174(NEW_PBCB)                   ; 0769
```

```
                                52  DD 00166 18$:    PUSHL   NEW_PBCB                                    ; 0772
              00000000G  00     01  FB 00168         CALLS   #1, SMG$$MIN_UPD                            ;
                         09     50  E9 0016F         BLBC    STATUS, 21$                                 ;
                   04    AE     64  D0 00172 19$:    MOVL    (R4), PP                                    ; 0778
                                B2  11 00176         BRB     16$                                         ; 0745
                         50     01  D0 00178 20$:    MOVL    #1, R0                                      ; 0781
                                    04 0017B 21$:    RET                                                 ; 0782

; Routine Size:  380 bytes,    Routine Base:  _SMG$CODE + 01D6

;  712          0783  1 !<BLF/PAGE>
;  714          0784  1 END                        ! End of module SMG$$PRVINP
;  715          0785  1
;  716          0786  0 ELUDOM



;                                   PSECT SUMMARY
;
;       Name                         Bytes                        Attributes
;
;  _SMG$CODE                            850  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)



;                              Library Statistics
;
;                                                 -------- Symbols --------        Pages       Processing
;       File                                      Total     Loaded   Percent       Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1               9776          9         0          581        00:01.0
;  _$255$DUA28:[SMGRTL.OBJ]RTLLIB.L32;1              36          0         0            8        00:00.1
;  _$255$DUA28:[SMGRTL.OBJ]SMGLIB.L32;1             469         38         8           38        00:00.4



;                                   COMMAND QUALIFIERS

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

; Size:          850 code + 0 data bytes
; Run Time:         00:18.4
; Elapsed Time:     01:10.5
; Lines/CPU Min:     2557
; Lexemes/CPU-Min: 16151
; Memory Used:  162 pages
; Compilation Complete
```

0360 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY