

```
SSSSSSSSSSSSS  MMM      MMM      GGGGGGGGGGG  RRRRRRRRRRR  TTTTTTTTTTTTT  LLL
SSSSSSSSSSSSS  MMM      MMM      GGGGGGGGGGG  RRRRRRRRRRR  TTTTTTTTTTTTT  LLL
SSSSSSSSSSSSS  MMM      MMM      GGGGGGGGGGG  RRRRRRRRRRR  TTTTTTTTTTTTT  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      GGG          RRR      RRR      TTT          LLL
SSS             MMM      MMM      GGG          RRR      RRR      TTT          LLL
SSS             MMM      MMM      GGG          RRR      RRR      TTT          LLL
SSS             MMM      MMM      GGG          RRR      RRR      TTT          LLL
SSSSSSSSSSS    MMM      MMM      GGG          RRRRRRRRRRR  TTT          LLL
SSSSSSSSSSS    MMM      MMM      GGG          RRRRRRRRRRR  TTT          LLL
SSSSSSSSSSS    MMM      MMM      GGG          RRRRRRRRRRR  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
SSSSSSSSSSSSS  MMM      MMM      GGGGGGGGG  RRR      RRR      TTT          LLLLLLLLLLLLLLLL
SSSSSSSSSSSSS  MMM      MMM      GGGGGGGGG  RRR      RRR      TTT          LLLLLLLLLLLLLLLL
SSSSSSSSSSSSS  MMM      MMM      GGGGGGGGG  RRR      RRR      TTT          LLLLLLLLLLLLLLLL
```

\_52  
Val  
---  
001  
001  
001  
001  
001  
001  
001  
001  
7FF  
7FF  
7FF  
7FF  
7FF  
7FF  
7FF

```

SSSSSSSS MM MM GGGGGGGG P P P P P P P P R R R R R R R R V V V V I I I I I I N N N N P P P P P P P P
SSSSSSSS MM MM GGGGGGGG P P P P P P P P R R R R R R R R V V V V I I I I I I N N N N P P P P P P P P
SS MM MM GG GG PP PP RR RR VV VV II II NN NN PP PP
SS MM MM GG GG PP PP RR RR VV VV II II NN NN PP PP
SS MM MM GG GG PP PP RR RR VV VV II II NN NN PP PP
SSSSSS SS MM MM GG GGGGGG PP P P P P P P R R R R R R V V V V II II NN NN P P P P P P P P
SSSSSS SS MM MM GG GGGGGG PP P P P P P P R R R R R R V V V V II II NN NN P P P P P P P P
SS MM MM GG GG GGGGGG PP PP RR RR VV VV II II NN NN P P P P P P P P
SS MM MM GG GG GGGGGG PP PP RR RR VV VV II II NN NN P P P P P P P P
SSSSSSSS MM MM GGGGGG GGGGGG PP PP RR RR VV VV II II NN NN P P P P P P P P
SSSSSSSS MM MM GGGGGG GGGGGG PP PP RR RR VV VV II II NN NN P P P P P P P P

```

```

LL LL I I I I I I S S S S S S S S
LL LL I I I I I I S S S S S S S S
LL LL I I I I I I S S S S S S S S
LL LL I I I I I I S S S S S S S S
LL LL I I I I I I S S S S S S S S
LL LL I I I I I I S S S S S S S S
LL LL I I I I I I S S S S S S S S
LL LL I I I I I I S S S S S S S S
LL LL I I I I I I S S S S S S S S
LLLLLLLLLLLL I I I I I I S S S S S S S S
LLLLLLLLLLLL I I I I I I S S S S S S S S

```

```
1 0001 0 %TITLE 'SMG$$PRVinp - Private Input support routines'  
2 0002 0 MODULE SMG$$PRVinp (  
3 0003 0 IDENT = '1-001' : File: SMGPRVinp.B32 Edit: STAN1001  
4 0004 0 ) =  
5 0005 1 BEGIN  
6 0006 1  
7 0007 1 *****  
8 0008 1 *  
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *  
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *  
11 0011 1 * ALL RIGHTS RESERVED. *  
12 0012 1 *  
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *  
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *  
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *  
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *  
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *  
18 0018 1 * TRANSFERRED. *  
19 0019 1 *  
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *  
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *  
22 0022 1 * CORPORATION. *  
23 0023 1 *  
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *  
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *  
26 0026 1 *  
27 0027 1 *  
28 0028 1 *****  
29 0029 1
```

```
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  --
```

```

53 0051 1 %SBTTL 'Declarations'
54 0052 1
55 0053 1 SWITCHES:
56 0054 1
57 0055 1
58 0056 1
59 0057 1 LINKAGES:
60 0058 1
61 0059 1 NONE
62 0060 1
63 0061 1 TABLE OF CONTENTS:
64 0062 1
65 0063 1
66 0064 1 FORWARD ROUTINE
67 0065 1
68 0066 1 ! Private entry points:
69 0067 1
70 0068 1 SMG$$INVALIDATE_DISPLAY, ! Mark contents of display as unknown
71 0069 1
72 0070 1 SMG$$REPORT_CHANGE_INSERT, ! Report change to physical
73 0071 1 ! screen in insert mode.
74 0072 1
75 0073 1 SMG$$REPORT_CHANGE_REPLACE; ! Report change to physical
76 0074 1 ! screen in replac mode.
77 0075 1
78 0076 1
79 0077 1 INCLUDE FILES
80 0078 1
81 0079 1
82 0080 1 REQUIRE 'RTLIN:SMGPROLOG'; ! defines psects, macros, tcb,
83 0158 1 ! wcb, & terminal symbols
84 0159 1
85 0160 1
86 0161 1 EXTERNAL REFERENCES
87 0162 1
88 0163 1
89 0164 1 EXTERNAL
90 0165 1 PBD_L_COUNT, ! No. of pasteboards we currently have
91 0166 1
92 0167 1 PBD_A_PBCB : VECTOR [PBD_K_MAX_PB, LONG],
93 0168 1 ! Table of addresses of PBCB's
94 0169 1
95 0170 1 PBD_V_PB_AVAIL : BITVECTOR [PBD_K_MAX_PB];
96 0171 1 ! Bit vector of pasteboard id numbers in use.
97 0172 1
98 0173 1 EXTERNAL ROUTINE
99 0174 1 LIB$GET_VM, ! Allocate heap storage
100 0175 1
101 0176 1 SMG$INSERT_CHARS, ! Insert char in virtual display buffer
102 0177 1 ! and cause output.
103 0178 1
104 0179 1 SMG$$FILL_WINDOW_BUFFER, ! Map all virtual display buffers to
105 0180 1 ! the window buffer for a given PBCB
106 0181 1
107 0182 1 SMG$$FLUSH_BUFFER, ! Flush any pending output to terminal
108 0183 1
109 0184 1 SMG$$FORCE_SCROLL_REG, ! Force scroll region to specified

```

```

: 110      0185 1          ! Lines.
: 111      0186 1
: 112      0187 1      SMG$LOCATE_PP,      ! Locate pasting packet that joins a virtual
: 113      0188 1          ! display to a pasteboard.
: 114      0189 1
: 115      0190 1      SMG$MOVE_TEXT_TO_SCREEN_BUF,
: 116      0191 1
: 117      0192 1      SMG$MOVE_TEXT_TO_WINDOW_BUF,      ! Map single virtual display to
: 118      0193 1          ! window buffer.
: 119      0194 1
: 120      0195 1      SMG$OCCLUDE,      ! Determine overlap between two rectangular
: 121      0196 1          ! areas.
: 122      0197 1      SMG$MIN_UPD,      ! Minimum output routine
: 123      0198 1
: 124      0199 1      SMG$PUT_TEXT_TO_BUFFER;      ! Text to virtual display buffer
: 125      0200 1
: 126      0201 1      EXTERNAL LITERAL
: 127      0202 1
: 128      0203 1      SMG$_FATERRLIB,      ! Fatal error in library procedure
: 129      0204 1      SMG$_INVARG,      ! Invalid argument
: 130      0205 1      SMG$_INVCOL,      ! Invalid column number
: 131      0206 1      SMG$_INVDIS_ID,      ! Invalid virtual display id
: 132      0207 1      SMG$_INV?AS_ID,      ! Invalid pasteboard id
: 133      0208 1      SMG$_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$$INVALIDATE_DISPLAY ( DISPLAY_ID.rl. )  
155 0229 1  
156 0230 1 FORMAL PARAMETERS:  
157 0231 1  
158 0232 1 DISPLAY_ID.rl.r Display ID of virtual display.  
159 0233 1  
160 0234 1 IMPLICIT INPUTS:  
161 0235 1  
162 0236 1 NONE  
163 0237 1  
164 0238 1 IMPLICIT OUTPUTS:  
165 0239 1  
166 0240 1 NONE  
167 0241 1  
168 0242 1 COMPLETION STATUS:  
169 0243 1  
170 0244 1 S$$_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 2     BEGIN ! Overall loop
204 0277 2
205 0278 2     LOCAL
206 0279 2         PBCB : REF BLOCK [ ,BYTE],      ! Address of pasteboard control
207 0280 2         WCB  : REF BLOCK [ ,BYTE],      ! Address of window control block
208 0281 2         TO_INDEX;      ! Index into destination
209 0282 2
210 0283 2         PBCB = .CURR_PP [PP_A_PBCB_ADDR]; ! Select this pasteboard and WCB
211 0284 2         WCB  = .PBCB [PBCB_A_WCB];      ! whose window needs rebuilding.
212 0285 2
213 0286 2         TO_INDEX = .CURR_PP [PP_W_TO_INDEX];
214 0287 2
215 0288 2         INCR R FROM 1 TO .CURR_PP [PP_W_ROWS_TO_MOVE]
216 0289 2         DO
217 0290 2             BEGIN ! For all rows in this display
218 0291 2                 !+
219 0292 2                 ! Zero out the display buffer.
220 0293 2                 !-
221 0294 2                 CHSFILL (0,
222 0295 2                     .CURR_PP [PP_W_MOVE_LENGTH],
223 0296 2                     .WCB [WCB_A_SCR_TEXT_BUF] + .TO_INDEX);
224 0297 2
225 0298 2                 TO_INDEX = .TO_INDEX + .WCB [WCB_W_NO_COLS];
226 0299 2                 END; ! For all rows to move
227 0300 2
228 0301 2         CURR_PP = .CURR_PP [PP_A_NEXT_DCB]; ! Walk the DCB side of
229 0302 2                                             ! the chain from front
230 0303 2                                             ! to back.
231 0304 2
232 0305 2     END; ! Overall loop
233 0306 2 RETURN SSS_NORMAL

```



: 234  
: 235  
0307 2  
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, [IB\$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

OFFC 00000

.ENTRY SMG\$\$INVALIDATE\_DISPLAY, Save R2,R3,R4,R5,- ; 0210  
R6,R7,R8,R9,R10,R11  
MOVL @DISPLAY\_ID, R0 ; 0264  
CML 56(R0), @DISPLAY\_ID  
BNEQ 1\$  
CMPB 68(R0), #17  
BEQL 2\$  
MOVL #SMG\$INVDIS\_ID, R0  
RET  
MOVL @DISPLAY\_ID, DCB ; 0266  
MOVL 32(DCB), CURR\_PP ; 0274  
MOVAB 32(DCB), R0  
CML CURR\_PP, R0  
BEQL 6\$  
MOVL 20(CURR\_PP), PBCB ; 0283  
MOVL 8(PBCB), WCB ; 0284  
MOVZWL 32(CURR\_PP), TO\_INDEX ; 0286  
MOVZWL 28(CURR\_PP), R1T ; 0288  
CLRL R ; 0298  
BRB 5\$  
MOVCS #0, (SP), #0, 34(CURR\_PP), @20(WCB)- ; 0296  
[TO\_INDEX]  
MOVZWL 6(WCB), R0 ; 0298  
ADDL2 R0, TO\_INDEX  
AOBLEQ R1T, R, 4\$ ; 0288  
MOVL (CURR\_PP), CURR\_PP ; 0301  
BRB 3\$ ; 0274  
MOVL #1, R0 ; 0306  
RET ; 0308

04 50 04 BC D0 00002  
BC 38 A0 D1 00006  
06 12 0000B  
11 44 A0 91 0000D  
08 13 00011  
50 00000000G 8F D0 00013 1\$:  
04 0001A  
5A 04 BC D0 0001B 2\$:  
57 20 AA D0 0001F  
50 20 AA 9E 00023 3\$:  
50 57 D1 00027  
2D 13 0002A  
50 14 A7 D0 0002C  
56 08 A0 D0 00030  
58 20 A7 3C 00034  
5B 1C A7 3C 00038  
59 D4 0003C  
10 11 0003E  
22 A7 00 2C 00040 4\$:  
06 B648 00046  
50 06 A6 3C 00049  
58 50 C0 0004D  
EC 59 5B F3 00050 5\$:  
57 67 D0 00054  
CA 11 00057  
50 01 D0 00059 6\$:  
04 0005C

; 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 ++
248 0320 1 FUNCTIONAL DESCRIPTION:
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. Furthermore, 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
    
```

```

294 0366 1 | CHANGED_CHAR.rb.r | The character that modified the screen.
295 0367 1 |
296 0368 1 | CHANGED_ROW.rl.r | Row number within the virtual display in
297 0369 1 | | which CHANGED_CHAR was written.
298 0370 1 |
299 0371 1 | CHANGED_COL.rl.r | Column number within the virtual display
300 0372 1 | | where CHANGED_CHAR was written.
301 0373 1 |
302 0374 1 | [,TERMINATING_CHAR.rb.r] | [Optional].
303 0375 1 | | If supplied, the terminating character
304 0376 1 | | that followed CHANGED_CHAR (See
305 0377 1 | | functional description for meaning).
306 0378 1 |
307 0379 1 | IMPLICIT INPUTS:
308 0380 1 |
309 0381 1 | NONE
310 0382 1 |
311 0383 1 | IMPLICIT OUTPUTS:
312 0384 1 |
313 0385 1 | NONE
314 0386 1 |
315 0387 1 | COMPLETION STATUS:
316 0388 1 |
317 0389 1 | S$$ NORMAL | Normal successful completion
318 0390 1 | SMG$_INVDIS_ID | Invalid Display Id
319 0391 1 | SMG$_INVPAS_ID | Invalid Pasteboard Id
320 0392 1 | SMG$_INVROW | Invalid row specified
321 0393 1 | SMG$_INVCOL | Invalid column specified
322 0394 1 |
323 0395 1 | SIDE EFFECTS:
324 0396 1 |
325 0397 1 | NONE
326 0398 1 | --
327 0399 1 |
328 0400 2 | BEGIN
329 0401 2 | BUILTIN
330 0402 2 | NULLPARAMETER;
331 0403 2 |
332 0404 2 | LOCAL
333 0405 2 | DESC : BLOCK [8,BYTE], | Local descriptor
334 0406 2 | STATUS, | Status of subroutine calls
335 0407 2 | DCB : REF $DCB_DECL, | Addr of display control block
336 0408 2 | PBCB : REF $PBCB_DECL, | Addr of pasteboard control
337 0409 2 | | block.
338 0410 2 | PP : REF $PP_DECL; | Addr of pasting packet
339 0411 2 |
340 0412 2 | $SMG$VALIDATE_ARGCOUNT (5, 6); | Test for right no. of args
341 0413 2 |
342 0414 2 | +
343 0415 2 | | Get addresses of control blocks we need
344 0416 2 | -
345 0417 2 | $SMG$GET_DCB (.DISPLAY ID, DCB); | Get DCB addr.
346 0418 2 | $SMG$GET_PBCB (.PASTEBOARD ID, PBCB); | Get PBCB addr.
347 0419 2 | IF NOT (STATUS = SMG$LOCATE_PP (.DCB, .PBCB, PP)) | Get PP addr.
348 0420 2 | THEN
349 0421 2 | RETURN (.STATUS);
350 0422 2 |

```

```

351 0423 2      SSMG$VALIDATE_ROW_COL (..CHANGED_ROW, ..CHANGED_COL); ! Valid Pos.?
352 0424 2
353 0425 2
354 0426 2      + Initialize our local descriptor to point to the changed character.
355 0427 2      -
356 0428 2      DESC [DSC$W_LENGTH] = 1;
357 0429 2      DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
358 0430 2      DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
359 0431 2      DESC [DSC$A_POINTER] = .CHANGED_CHAR;
360 0432 2
361 0433 2      +
362 0434 2      Reflect this change in the virtual display text and attribute buffer,
363 0435 2      including new virtual display cursor position.
364 0436 2      Data from the affected column to the last-1 column of this line need
365 0437 2      to be moved one character position to the right and the changed
366 0438 2      character inserted at the indicated position. The attributes for the
367 0439 2      moved bytes must be moved as well.
368 0440 2      -
369 0441 2      IF NOT (STATUS = SMG$INSERT_CHARS ( .DISPLAY_ID,
370 0442 2      .CHANGED_ROW,
371 0443 2      .CHANGED_COL,
372 0444 2      DESC))
373 0445 2      THEN
374 0446 2      RETURN (.STATUS);
375 0447 2
376 0448 2      IF NOT NULLPARAMETER (6)
377 0449 2      THEN
378 0450 2      BEGIN ! Terminator supplied
379 0451 2      +
380 0452 2      Inspect supplied terminator to determine effect on cursor
381 0453 2      position in virtual display.
382 0454 2      -
383 0455 2      +
384 0456 2      ***** For now pretend the terminator is a <CR>. *****
385 0457 2      ***** This needs to act like a <CR><LF> pair.
386 0458 2      -
387 0459 2      IF NOT (STATUS = SMG$SPUT_TEXT_TO_BUFFER (
388 0460 2      .DCB,
389 0461 2      .DCB [DCB_B_DEF_VIDEO_ATTR],
390 0462 2      1,
391 0463 2      %REF (CR*8 + LF), ! <CR><LF>
392 0464 2      .DCB [DCB_B_DEF_CHAR_SET]))
393 0465 2      THEN
394 0466 2      RETURN (.STATUS);
395 0467 2      END; ! Terminator supplied
396 0468 2
397 0469 2      +
398 0470 2      Reflect this change in the appropriate positions of the window screen
399 0471 2      text and attribute buffers, including new screen cursor position.
400 0472 2      -
401 0473 2      IF NOT (STATUS = SMG$MOVE_TEXT_TO_WINDOW_BUF ( .PP))
402 0474 2      THEN
403 0475 2      RETURN (.STATUS);
404 0476 2
405 0477 2      +
406 0478 2      Record what has happened to screen buffer as well.
407 0479 2      -

```

```

408 0480 3      IF NOT (STATUS = SMG$$MOVE_TEXT_TO_SCREEN_BUF ( .PP))
409 0481 2      THEN
410 0482 2      RETURN (.STATUS);
411 0483 2
412 0484 2      !+
413 0485 2      Must now force the changes to be output.
414 0486 2      -
415 0487 2      IF .PP [PP_W_ROWS_TO_MOVE] NEQ 0
416 0488 2      THEN
417 0489 2      BEGIN
418 0490 2      ! Assume damage confined to single row.
419 0491 3      PBCB [PBCB_W_FIRST_CHANGED_ROW] = ..CHANGED_ROW;
420 0492 3      PBCB [PBCB_W_LAST_CHANGED_ROW] = ..CHANGED_ROW;
421 0493 3
422 0494 3      ! Assume damage in row from given pos. to end of line
423 0495 3      PBCB [PBCB_W_FIRST_CHANGED_COL] = ..CHANGED_COL;
424 0496 3      PBCB [PBCB_W_LAST_CHANGED_COL] = .PBCB [PBCB_W_WIDTH];
425 0497 2      END;
426 0498 2
427 0499 2      STATUS = SMG$$MIN_UPD ( .PBCB);
428 0500 2
429 0501 2      !+
430 0502 2      If this virtual display is pasted to pasteboards other than the one
431 0503 2      identified in the call list, these additional pasteboard's window
432 0504 2      buffers must be updated as well. For these additional pasteboards,
433 0505 2      the changed byte in addition to the shifted remainder of the line
434 0506 2      must be output -- since they did not receive the originally echoed
435 0507 2      character.
436 0508 2      -
437 0509 2      PP = .DCB [DCB_A_PP_NEXT];          ! 1st in chain
438 0510 2      WHILE .PP NEQ ^CB [DCB_A_PP_NEXT] ! While any packets remain...
439 0511 2      DO
440 0512 3      BEGIN ! Loop through all pasting packets for this DCB
441 0513 3      LOCAL
442 0514 3      NEW_PBCB : REF $PBCB_DECL;      ! PBCB being considered
443 0515 3
444 0516 3      NEW_PBCB = .PP [PP_A_PBCB_ADDR];    ! PBCB for this packet
445 0517 3      IF .NEW_PBCB NEQ .PBCB ! If this isn't the one we started with
446 0518 3      THEN
447 0519 4      BEGIN ! Needs to be output
448 0520 5      IF NOT (STATUS = SMG$$FILL_WINDOW_BUFFER (.NEW_PBCB))
449 0521 4      THEN
450 0522 4      RETURN (.STATUS);
451 0523 4
452 0524 4      IF .PP [PP_W_ROWS_TO_MOVE] NEQ 0
453 0525 4      THEN
454 0526 5      BEGIN
455 0527 5      ! Assume damage confined to single row.
456 0528 5      PBCB [PBCB_W_FIRST_CHANGED_ROW] = ..CHANGED_ROW;
457 0529 5      PBCB [PBCB_W_LAST_CHANGED_ROW] = ..CHANGED_ROW;
458 0530 5
459 0531 5      ! Assume damage in row from given pos. to end of line
460 0532 5      PBCB [PBCB_W_FIRST_CHANGED_COL] = ..CHANGED_COL;
461 0533 5      PBCB [PBCB_W_LAST_CHANGED_COL] = .PBCB [PBCB_W_WIDTH];
462 0534 4      END;
463 0535 4
464 0536 5      IF NOT (STATUS = SMG$$MIN_UPD ( .NEW_PBCB))

```

```

: 465      0537 4      THEN
: 466      0538 4      RETURN (.STATUS);
: 467      0539 4      END;          ! Needs to be output
: 468      0540 4
: 469      0541 4      PP = .PP [PP_A_NEXT_DCB];      ! Step to next packet in chain
: 470      0542 4      END;          ! Loop through all pasting packets for this DCB
: 471      0543 4
: 472      0544 4      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	
					MOVAB	SMG\$\$MIN_UPD, R8	
					SUBL2	#16, SP	
	50				SUBB3	#5, (AP), DIFF	0412
					CMPB	DIFF, #1	
					BLEQU	1\$	
					MOVL	#SMG\$_WRONUMARG, R0	
					RET		
					MOVL	@DISPLAY_ID, R0	0417
	04				CMPB	56(R0), @DISPLAY_ID	
					BNEQ	2\$	
					CMPB	68(R0), #17	
					BEQL	3\$	
					MOVL	#SMG\$_INVDIS_ID, R0	
					RET		
					MOVL	@DISPLAY_ID, DCB	
					MOVL	@PASTEBOARD_ID, R0	0418
					BLSS	4\$	
					CMPB	R0, PBD_L_COUNT	
					BGTR	4\$	
	08				BBS	R0, PBD V PB_AVAIL, 5\$	
					MOVL	#SMG\$_INVPAS_ID, R0	
					RET		
					MOVL	PBD_A_PBCB[R0], PBCB	
					PUSHAB	PP	0419
					PUSHL	PBCB	
					PUSHL	DCB	
					CALLS	#3, SMG\$\$LOCATE_PP	
					BLBC	STATUS, 10\$	
					MOVL	CHANGED_ROW, R6	0423
					TSTL	(R6)	
					BLEQ	6\$	
	66	02	A4		CMPZV	#0, #16, 2(DCB), (R6)	
					BGEQ	7\$	
					MOVL	#SMG\$_INVROW, R0	
					RET		
					MOVL	CHANGED_COL, R7	
					TSTL	(R7)	
					BLEQ	8\$	
	67	06	A4		CMPZV	#0, #16, 6(DCB), (R7)	
					BGEQ	9\$	
					MOVL	#SMG\$_INVCOL, R0	

08	AE	010E0001	8F	D0	000A1	9\$:	RET		0428
0C	AE	0C	AC	D0	000A2		MOVL	#17694721, DESC	0431
		08	AE	9F	000AF		MOVL	CHANGED_CHAR, DESC+4	0441
	7E		56	7D	000B2		PUSHAB	DESC	0444
00000000G	00	04	AC	DD	000B5		MOVQ	R6, -(SP)	0441
	42		04	FB	000B8		PUSHL	DISPLAY_ID	
	06		50	E9	000BF		CALLS	#4, SMG\$INSERT_CHARS	
			6C	91	000C2		BLBC	STATUS, 12\$	0448
			24	1F	000C5		CMPB	(AP), #6	
		18	AC	D5	000C7		BLSSU	11\$	
	7E	30	1F	13	000CA		TSTL	24(AP)	
04	AE	0DOA	A4	9A	000CC		BEQL	11\$	0464
		04	8F	3C	000D0		MOVZBL	48(DCB), -(SP)	0463
			AE	9F	000D6		MOVZWL	#3338, 4(SP)	
	7E	2E	01	DD	000D9		PUSHAB	4(SP)	0459
00000000G	00		A4	9A	000DB		PUSHL	#1	0461
	5F		54	DD	000DF		MOVZBL	46(DCB), -(SP)	0460
	53	04	05	FB	000E1	10\$:	PUSHL	DCB	0459
			50	E9	000E8	11\$:	CALLS	#5, SMG\$PUT_TEXT_TO_BUFFER	0473
00000000G	00		AE	D0	000EB		BLBC	STATUS, 15\$	
	7D		53	DD	000EF		MOVL	PP, R3	0480
00000000G	00		01	FB	000F1		PUSHL	R3	
			50	E9	000F8		CALLS	#1, SMG\$MOVE_TEXT_TO_WINDOW_BUF	
			53	DD	000FB		BLBC	STATUS, 19\$	0480
00000000G	00		01	FB	000FD		PUSHL	R3	
			50	E9	00104	12\$:	CALLS	#1, SMG\$MOVE_TEXT_TO_SCREEN_BUF	
		1C	A3	B5	00107		BLBC	STATUS, 19\$	0487
			15	13	0010A		TSTW	28(R3)	
00A8	C2		66	B0	0010C		BEQL	13\$	0491
00AA	C2		66	B0	00111		MOVW	(R6), 168(PBCB)	0492
00AC	C2		67	B0	00116		MOVW	(R6), 170(PBCB)	0495
00AE	C2	5A	A2	B0	0011B		MOVW	(R7), 172(PBCB)	0496
			52	DD	00121	13\$:	MOVW	90(PBCB), 174(PBCB)	0499
	68	20	01	FB	00123		PUSHL	PBCB	
04	AE	04	A4	D0	00126	14\$:	CALLS	#1, SMG\$MIN_UPD	0509
	53	20	AE	D0	0012B		MOVL	32(DCB), PP	0510
	51		A4	9E	0012F		MOVL	PP, R3	
	51		53	D1	00133		MOVAB	32(DCB), R1	
			3D	13	00136		CPL	R3, R1	
	55	14	A3	D0	00138		BEQL	18\$	0516
	52		55	D1	0013C		MOVL	20(R3), NEW_PBCB	0517
			2E	13	0013F		CPL	NEW_PBCB, PBCB	
			55	DD	00141		BEQL	17\$	0520
00000000G	00		01	FB	00143		PUSHL	NEW_PBCB	
	2B		50	E9	0014A	15\$:	CALLS	#1, SMG\$FILL_WINDOW_BUFFER	
		1C	A3	B5	0014D		BLBC	STATUS, 19\$	0524
			15	13	00150		TSTW	28(R3)	
00A8	C2		66	B0	00152		BEQL	16\$	0528
00AA	C2		66	B0	00157		MOVW	(R6), 168(PBCB)	0529
00AC	C2		67	B0	0015C		MOVW	(R6), 170(PBCB)	0532
00AE	C2	5A	A2	B0	00161		MOVW	(R7), 172(PBCB)	0533
			55	DD	00167	16\$:	MOVW	90(PBCB), 174(PBCB)	0536
	68		01	FB	00169		PUSHL	NEW_PBCB	
	09		50	E9	0016C		CALLS	#1, SMG\$MIN_UPD	
04	AE		63	D0	0016F	17\$:	BLBC	STATUS, 19\$	0541
			B6	11	00173		MOVL	(R3), PP	0510
							BRB	14\$	

SMG\$\$PRVinp  
1-001

N 11  
SMG\$\$PRVinp - Private Input support routines 16-Sep-1984 01:10:09 VAX-11 Bliss-32 V4.0-742  
SMG\$\$REPORT\_CHANGE\_INSERT - Report change to sc 14-Sep-1984 13:09:59 [SMGRTL.SRC]SMGPRVinp.B32;1

50 01 00 00175 18\$: MOVL #1, R0  
04 00178 19\$: RET

: 0544  
: 0545

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

: 474 0546 1 !<BLF/PAGE>

S  
1

.....



```

476 0547 1 %SBTTL 'SMG$$REPORT_CHANGE_REPLACE - Report change to screen -- replace'
477 0548 1 GLOBAL ROUTINE SMG$$REPORT_CHANGE_REPLACE (
478 0549 1     DISPLAY_ID,
479 0550 1     PASTEBOARD_ID,
480 0551 1     NUM_CHARS      : REF VECTOR [,WORD],
481 0552 1     CHANGED_CHARS : REF VECTOR [,BYTE],
482 0553 1     CHANGED_ROW,
483 0554 1     CHANGED_COL,
484 0555 1     TERMINATING_CHAR
485 0556 1 ) =
486 0557 1
487 0558 1 ++
488 0559 1 FUNCTIONAL DESCRIPTION:
489 0560 1     This routine is called to report a change made to the physical
490 0561 1     screen via an input action involving replacement. The
491 0562 1     characters identified by CHANGED_CHARS and NUM_CHARS have been
492 0563 1     positioned on the screen in a position corresponding to the
493 0564 1     virtual display coordinates CHANGED_ROW and CHANGED_COL.
494 0565 1
495 0566 1     TERMINATING_CHAR specifies the details of the change.
496 0567 1
497 0568 1     If TERMINATING_CHAR has not been specified, then the changed
498 0569 1     characters have been echoed in the position indicated. No
499 0570 1     further output is required. Just need to update internal data
500 0571 1     bases to reflect that change. Current cursor position is
501 0572 1     changed to reflect that beyond the changed text. If this new
502 0573 1     position is now beyond the bounds of the virtual display's
503 0574 1     dimensions then ??????.
504 0575 1
505 0576 1     If TERMINATING_CHAR was specified, the changed characters have
506 0577 1     been echoed in the positions indicated. Futhermore, these
507 0578 1     character were followed by a TERMINATING_CHAR. Just need to
508 0579 1     update internal data bases to reflect that change. Current
509 0580 1     cursor position is changed to reflect the effects of the
510 0581 1     TERMINATING CHAR. If this new position is now beyond the
511 0582 1     bounds of the virtual display's dimensions then ??????.
512 0583 1
513 0584 1 CALLING SEQUENCE:
514 0585 1
515 0586 1     ret_status.wlc.v = SMG$$REPORT_CHANGE_REPLACE (
516 0587 1         DISPLAY_ID.rl.r,
517 0588 1         PASTEBOARD_ID.r(.r,
518 0589 1         NUM_CHARS.rwu.r,
519 0590 1         CHANGED_CHARS.rab.r
520 0591 1         [,CHANGED_ROW.rl.r]
521 0592 1         [,CHANGED_COL.rl.r]
522 0593 1         [,TERMINATING_CHAR.rb.r])
523 0594 1
524 0595 1 FORMAL PARAMETERS:
525 0596 1     DISPLAY_ID.rl.r     Display id of virtual display.
526 0597 1     PASTEBOARD_ID.rl.r  Pasteboard id.
527 0598 1     NUM_CHARS.rwu.r     The number of characters that changed.
528 0599 1
529 0600 1     CHANGED_CHARS.rab.r  Address of the characters that modified
530 0601 1
531 0602 1     the screen.
532 0603 1

```

```

533 0604 1 |
534 0605 1 |   CHANGED_ROW.rl.r   Row number within the virtual display in
535 0606 1 |                       which CHANGED_CHARS were written.
536 0607 1 |
537 0608 1 |   CHANGED_COL.rl.r   Column number within the virtual display
538 0609 1 |                       where CHANGED_CHARS were written.
539 0610 1 |
540 0611 1 |   [,TERMINATING_CHAR.rb.r] [Optional].
541 0612 1 |                       If supplied, the terminating character
542 0613 1 |                       that followed CHANGED_CHAR (See
543 0614 1 |                       functional description for meaning).
544 0615 1 |
545 0616 1 | IMPLICIT INPUTS:
546 0617 1 |
547 0618 1 |   NONE
548 0619 1 |
549 0620 1 | IMPLICIT OUTPUTS:
550 0621 1 |
551 0622 1 |   NONE
552 0623 1 |
553 0624 1 | COMPLETION STATUS:
554 0625 1 |
555 0626 1 |   S$$ NORMAL          Normal successful completion
556 0627 1 |   SMG$_INVDIS_ID      Invalid Display Id
557 0628 1 |   SMG$_INVPAS_ID      Invalid Pasteboard Id
558 0629 1 |   SMG$_INVROW         Invalid row specified
559 0630 1 |   SMG$_INVCOL         Invalid column specified
560 0631 1 |
561 0632 1 | SIDE EFFECTS:
562 0633 1 |
563 0634 1 |   NONE
564 0635 1 | --
565 0636 1 |
566 0637 2 | BEGIN
567 0638 2 | BUILTIN
568 0639 2 |   NULLPARAMETER;
569 0640 2 |
570 0641 2 | LOCAL
571 0642 2 |   STATUS,              | Status of subroutine calls
572 0643 2 |   C_ROW,               | Working row
573 0644 2 |   C_COL,               | Working col
574 0645 2 |   DCB : REF $DCB DECL, | Addr of display control block
575 0646 2 |   PBCB : REF BLOCK [,.BYTE], | Addr of pasteboard control
576 0647 2 |                               | block.
577 0648 2 |   WCB : REF BLOCK [,.BYTE], | Address of window block
578 0649 2 |   PP  : REF BLOCK [,.BYTE]; | Addr of pasting packet.
579 0650 2 |
580 0651 2 |   $SMG$VALIDATE_ARGCOUNT (4, 7); | Test for right no. of args
581 0652 2 |
582 0653 2 | |
583 0654 2 | | Get addresses of control blocks needed.
584 0655 2 | |
585 0656 2 | | $SMG$GET_DCB (.DISPLAY ID, DCB); | Get DCB addr.
586 0657 2 | | $SMG$GET_PBCB (.PASTEBOARD ID, PBCB); | Get PBCB addr.
587 0658 2 | | IF NOT (STATUS = SMG$$LOCATE_PP (.DCB, .PBCB, PP)) | Get PP addr.
588 0659 2 | | THEN
589 0660 2 | | RETURN (.STATUS);

```

```

590 0661 2
591 0662 2 IF NOT NULLPARAMETER (5)
592 0663 2 THEN
593 0664 2 BEGIN
594 0665 2 C_ROW = ..CHANGED_ROW;
595 0666 2 DCB [DCB_W_CURSOR_ROW] = .C_ROW;
596 0667 2 END
597 0668 2 ELSE
598 0669 2 C_ROW = .DCB [DCB_W_CURSOR_ROW];
599 0670 2
600 0671 2 IF NOT NULLPARAMETER (6)
601 0672 2 THEN
602 0673 2 BEGIN
603 0674 2 C_COL = ..CHANGED_COL;
604 0675 2 DCB [DCB_W_CURSOR_COL] = .C_COL;
605 0676 2 END
606 0677 2 ELSE
607 0678 2 C_COL = .DCB [DCB_W_CURSOR_COL];
608 0679 2
609 0680 2 $SMG$VALIDATE_ROW_COL (.C_ROW, .C_COL); ! Valid posit?
610 0681 2
611 0682 2 !+
612 0683 2 Reflect this change in the virtual display text and attribute buffers.
613 0684 2 !-
614 0685 2
615 0686 2 !+
616 0687 2 Invalidate physical cursor position.
617 0688 2 This will force output to begin with a direct cursor
618 0689 2 movement to the proper place.
619 0690 2 !-
620 0691 2
621 0692 2 WCB = .PBCB [PBCB_A_WCB];
622 0693 2 WCB [WCB_W_OLD_CUR_ROW] = 0;
623 0694 2 WCB [WCB_W_OLD_CUR_COL] = 0;
624 0695 2
625 0696 2 IF NOT (STATUS = SMG$$PUT_TEXT_TO_BUFFER (
626 0697 2 .DCB,
627 0698 2 .DCB [DCB_B_DEF_VIDEO_ATTR],
628 0699 2 .NUM_CHARS[0],
629 0700 2 .CHANGED_CHARS,
630 0701 2 .DCB [DCB_B_DEF_CHAR_SET]))
631 0702 2 THEN
632 0703 2 RETURN (.STATUS);
633 0704 2
634 0705 2 IF NOT NULLPARAMETER (7)
635 0706 2 THEN
636 0707 2 BEGIN ! Terminator supplied
637 0708 2 !+
638 0709 2 Inspect supplied terminator to determine effect on cursor
639 0710 2 position in virtual display.
640 0711 2 !-
641 0712 2 IF NOT (STATUS = SMG$$PUT_TEXT_TO_BUFFER (
642 0713 2 .DCB,
643 0714 2 .DCB [DCB_B_DEF_VIDEO_ATTR],
644 0715 2 !
645 0716 2 %REF (CR^8 + LF), ! <CR><LF>
646 0717 2 .DCB [DCB_B_DEF_CHAR_SET]))

```

```

647 0718 3      THEN
648 0719 3      RETURN (.STATUS);
649 0720 3      END;      ! Terminator supplied
650 0721 3
651 0722 3      +
652 0723 3      | Reflect this change in the appropriate positions of the window
653 0724 3      | text and attribute buffers, including new screen cursor position.
654 0725 3      |
655 0726 3      | IF NOT (STATUS = SMG$$MOVE_TEXT_TO_WINDOW_BUF ( .PP))
656 0727 3      | THEN
657 0728 3      |     RETURN (.STATUS);
658 0729 3      |
659 0730 3      | +
660 0731 3      | | Update screen buffers as well.
661 0732 3      | |
662 0733 3      | | IF NOT (STATUS = SMG$$MOVE_TEXT_TO_SCREEN_BUF ( .PP))
663 0734 3      | | THEN
664 0735 3      | |     RETURN (.STATUS);
665 0736 3      | |
666 0737 3      | | +
667 0738 3      | | | If this virtual display is pasted to pasteboards other than the one
668 0739 3      | | | identified in the call list, these additional pasteboard's window
669 0740 3      | | | buffers must be updated as well. For these additional pasteboards,
670 0741 3      | | | the changed bytes must be output -- since they did not receive the
671 0742 3      | | | originally echoed characters.
672 0743 3      | | |
673 0744 3      | | | PP = .DCB [DCB_A_PP_NEXT];      ! 1st in chain
674 0745 3      | | | WHILE .PP NEQ DCB [DCB_A_PP_NEXT] ! While any packets remain...
675 0746 3      | | | DO
676 0747 3      | | |     BEGIN ! Loop through all pasting packets for this DCB
677 0748 3      | | |     LOCAL
678 0749 3      | | |     NEW_PBCB : REF $PBCB_DLL;      ! PBCB being considered
679 0750 3      | | |
680 0751 3      | | |     NEW_PBCB = .PP [PP_A_PBCB_ADDR];      ! PBCB for this packet
681 0752 3      | | |     IF .NEW_PBCB NEQ .PBCB ! If this isn't the one we started with
682 0753 3      | | |     THEN
683 0754 4      | | |     BEGIN ! Needs to be output
684 0755 5      | | |     IF NOT (STATUS = SMG$$FILL_WINDOW_BUFFER (.NEW_PBCB))
685 0756 4      | | |     THEN
686 0757 4      | | |     RETURN (.STATUS);
687 0758 4      | | |
688 0759 4      | | |     IF .PP [PP_W_ROWS_TO_MOVE] NEQ 0
689 0760 4      | | |     THEN
690 0761 5      | | |     BEGIN
691 0762 5      | | |     ! Assume damage confined to single row.
692 0763 5      | | |     NEW_PBCB [PBCB_W_FIRST_CHANGED_ROW] = .C_ROW;
693 0764 5      | | |     NEW_PBCB [PBCB_W_LAST_CHANGED_ROW] = .C_ROW;
694 0765 5      | | |
695 0766 5      | | |     ! Assume damage in row from given pos. to end of line
696 0767 5      | | |     NEW_PBCB [PBCB_W_FIRST_CHANGED_COL] = .C_COL;
697 0768 5      | | |     NEW_PBCB [PBCB_W_LAST_CHANGED_COL] =
698 0769 5      | | |     .NEW_PBCB [PBCB_W_WIDTH];
699 0770 4      | | |     END;
700 0771 4      | | |
701 0772 5      | | | IF NOT (STATUS = SMG$$MIN_UPD (.NEW_PBCB))
702 0773 4      | | | THEN
703 0774 4      | | |     RETURN (.STATUS);

```

704  
705  
706  
707  
708  
709  
710  
711  
0775  
0776  
0777  
0778  
0779  
0780  
0781  
0782

END; ! Needs to be output  
PP = .PP [PP\_A\_NEXT\_DCB]; ! Step to next packet in chain  
END; ! Loop through all pasting packets for this DCB  
RETURN (SS\$ \_NORMAL);  
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	CMPB	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	CMPB	R0, PBD_L_COUNT	
			08 14 00047	BGTR	4\$	
08	00000000G	00	50 E0 00049	BBS	R0, PBD V PB_AVAIL, 5\$	
			50 0000000G 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	#SMGS_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	#SMGS_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	0DOA	AE	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
			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
			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		CPL	R4, R1	
					41	13	00135		BEQL	20\$	
			52	14	A4	D0	00137		MOVL	20(R4), NEW_PBCB	0751
			55		52	D1	0013B		CPL	NEW_PBCB, PBCB	0752
					32	13	0013E		BEQL	19\$	
					52	DD	00140		PUSHL	NEW_PBCB	0755
			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), T74(NEW_PBCB)	0769

SMG\$\$PRVINP  
1-001

SMG\$\$PRVINP - Private Input support routines  
SMG\$\$REPORT\_CHANGE\_REPLACE - Report change to s

H 12  
16-Sep-1984 01:10:09  
14-Sep-1984 13:09:59

VAX-11 Bliss-32 V4.0-742  
[SMGRTL.SRC]SMGPRVINP.B32;1

Page 21  
(7)

00000000G	00	52	DD	00166	18\$:	PUSHL	NEW_PCB	:	0772
	09	01	FB	00168		CALLS	#1, _SMG\$\$MIN_UPD	:	
		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

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing 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

This image displays a grid of numerous small, illegible terminal windows or document pages, each containing text and possibly graphical elements. The windows are arranged in a dense, repeating pattern across the page. Several windows contain the following text:

- SMGNUMTAB LIS
- SMGMSGPTR LIS
- SMGSCROLL LIS
- SMGMISC LIS
- SMGMSGTXT LIS
- SMGPLTENC LIS
- SMGPLTTEX LIS
- SMGSIMTRM LIS
- SMGNUMPAR LIS
- SMGPRUNP LIS