File created: 31-Jul-2023 13:39:50 **{WMEDLEY}<library>BIGBITMAPS.;13**

edit by: rmk

changes to: (VARS BIGBITMAPSCOMS)
(FNS BIGBITMAPEQUAL)

previous date: 9-Jul-2022 09:41:26 {WMEDLEY}<library>BIGBITMAPS.;12

Read Table: XCL

Package: INTERLISP

Format: XCCS


```
(RPAQQ BIGBITMAPSCOMS
       ((DECLARE\: EVAL@COMPILE DONTCOPY (RECORDS BIGBM)
              (CONSTANTS (|\\MaxBitMapHeight| 65535)
                     (|\\MaxBitMapWidth| 65535)
                     (|\\MaxBitMapWords| 131066))
              (MACROS |GetNewFragment|)
              (MACROS |\\SFInvert|))
        (INITRECORDS BIGBM)
        (FNS BIGBITMAPP BITBLT.BIGBM BITMAPCREATE.BIGBM BITMAPCREATE BITMAPCOPY BIGBITMAPEQUAL BLTSHADE.BIGBM
             BITBLT \\ORG.BITBLT \\BLTSHADE.DISPLAY \\RESHOWBORDER1)
        (FNS \\DRAWCIRCLE.BIGBM \\FILLCIRCLE.BIGBM \\DRAWELLIPSE.BIGBM \\DRAWCURVE.BIGBM \\DRAWLINE.BIGBM.DASH
             \\DRAWLINE.BIGBM.NODASH)
        (FNS \\GENERIC.DSPCREATE.DESTINATION.BITMAP?.BIGBM)
        (DECLARE\: DONTEVAL@LOAD DOCOPY (P (MOVD '\\GENERIC.DSPCREATE.DESTINATION.BITMAP?.BIGBM
                                                 '\\GENERIC.DSPCREATE.DESTINATION.BITMAP?)))
        (FNS DSPDESTINATION |\\SFFixY| |\\SFFixDestination| |\\SFFixClippingRegion|)
        (FNS \\SW2BM BITMAPHEIGHT BITMAPWIDTH |\\SFFixFont| BITSPERPIXEL)
        (FNS COLORIZEBITMAP \\BWTOCOLORBLT UNCOLORIZEBITMAP)
        (DECLARE\: DONTEVAL@LOAD DOCOPY (P (MOVD '\\ORG.BITBLT 'ORG.BITBLT)
                                           (MOVD? 'BLTSHADE 'ORG.BLTSHADE)
                                           (MOVD 'BLTSHADE.BIGBM 'BLTSHADE)
                                           (MOVD 'BITBLT 'BKBITBLT)))))

(DECLARE\: EVAL@COMPILE DONTCOPY

(DECLARE\: EVAL@COMPILE

(DATATYPE BIGBM (BIGBMWIDTH BIGBMHEIGHT BIGBMLIST))
)

(/DECLAREDATATYPE 'BIGBM '(POINTER POINTER POINTER)
        ;; ---field descriptor list elided by lister---
        '6)

(DECLARE\: EVAL@COMPILE

(RPAQQ |\\MaxBitMapHeight| 65535)

(RPAQQ |\\MaxBitMapWidth| 65535)

(RPAQQ |\\MaxBitMapWords| 131066)

(CONSTANTS (|\\MaxBitMapHeight| 65535)
       (|\\MaxBitMapWidth| 65535)
       (|\\MaxBitMapWords| 131066))
)

(DECLARE\: EVAL@COMPILE

(PUTPROPS |GetNewFragment| MACRO ((LIST)
                                  (PROG1 (CAR LIST)
                                      (SETQ LIST (CDR LIST)))))
)

(DECLARE\: EVAL@COMPILE

(PUTPROPS |\\SFInvert| MACRO ((|BitMap| \y)

            (* |corrects| |for| |the| |fact| |that| |alto| |bitmaps| |are| |stored| |with| 0\,0 |as| |upper| |left| |while| |lisp| |bitmaps| |have| 0\,0 |as|
            |lower| |left.| |The| |correction| |is| |actually| |off| |by| |one| (|greater|) |because| \a |majority| |of| |the| |places| |that| |it| |is|
            |called| |actually| |need| |one| |more| |than| |corrected| Y |value.|)

                              (IDIFFERENCE (|fetch| (BITMAP BITMAPHEIGHT) |of| |BitMap|)
                                     \y)))
)
)

(/DECLAREDATATYPE 'BIGBM '(POINTER POINTER POINTER)
        ;; ---field descriptor list elided by lister---
```

```
       '6)

(DEFINEQ
```

(BIGBITMAPP

```
  (LAMBDA (X)                                                    ; Edited 13-Jun-2021 13:27 by rmk:
    (TYPE? BIGBM X)))
```

(BITBLT.BIGBM

```
  (LAMBDA (SRCE SRCELEFT SRCEBOTTOM DEST DESTLEFT DESTBOTTOM WIDTH HEIGHT SRCETYPE OPERATION TEXTURE
                CLIPPINGREGION)                                  ; Edited 24-Jan-91 11:19 by matsuda
    (PROG (SRCEBMLIST DESTBMLIST SRCEBIGBMHEIGHT DESTBIGBMHEIGHT SRCETOP DESTTOP SRCEFRAG DESTFRAG SRCEFRAGTOP
                  DESTFRAGTOP SRCEFRAGBOTTOM DESTFRAGBOTTOM SRCE-H DEST-H H NEXT-S-TOP NEXT-D-TOP SBOTTOM DBOTTOM
                  )
          (SETQ SRCETOP (IPLUS (OR SRCEBOTTOM (SETQ SRCEBOTTOM 0))
                               (OR HEIGHT (SETQ HEIGHT (BITMAPHEIGHT SRCE)))))
          (SETQ DESTTOP (IPLUS (OR DESTBOTTOM (SETQ DESTBOTTOM 0))
                               HEIGHT))
          (COND
             ((< DESTBOTTOM 0)
              (SETQ HEIGHT (+ HEIGHT DESTBOTTOM))
              (SETQ SRCEBOTTOM (- SRCEBOTTOM DESTBOTTOM))
              (SETQ DESTBOTTOM 0)
              (SETQ DESTTOP HEIGHT)))
          (COND
             ((|type?| BIGBM SRCE)
              (SETQ SRCEBMLIST (|fetch| (BIGBM BIGBMLIST) |of| SRCE))
              (SETQ SRCEBIGBMHEIGHT (|fetch| (BIGBM BIGBMHEIGHT) |of| SRCE))
              (SETQ SRCEFRAG (|GetNewFragment| SRCEBMLIST))
              (SETQ SRCEFRAGTOP SRCEBIGBMHEIGHT)
              (SETQ SRCEFRAGBOTTOM (- SRCEFRAGTOP (BITMAPHEIGHT SRCEFRAG)))
              (|until| (< SRCEFRAGBOTTOM SRCETOP) |do|   ;; Search the first fragment of SRCE bitmaps
                                                         (SETQ SRCEFRAG (|GetNewFragment| SRCEBMLIST))
                                                         (SETQ SRCEFRAGTOP SRCEFRAGBOTTOM)
                                                         (SETQ SRCEFRAGBOTTOM (- SRCEFRAGTOP (BITMAPHEIGHT SRCEFRAG))
                                                          ))
              (COND
                 ((|type?| BIGBM DEST)
                  (PROG NIL
                   ;; BIGBM to BIGBM case
                        (SETQ DESTBMLIST (|fetch| (BIGBM BIGBMLIST) |of| DEST))
                        (SETQ DESTBIGBMHEIGHT (|fetch| (BIGBM BIGBMHEIGHT) |of| DEST))
                        (SETQ DESTFRAG (|GetNewFragment| DESTBMLIST))
                        (SETQ DESTFRAGTOP DESTBIGBMHEIGHT)
                        (SETQ DESTFRAGBOTTOM (- DESTFRAGTOP (BITMAPHEIGHT DESTFRAG)))
                    LOOP
                        (|until| (<= DESTFRAGBOTTOM DESTTOP) |do|   ;; Serch the first fragment of DEST bitmaps
                                                                    (SETQ DESTFRAG (|GetNewFragment| DESTBMLIST))
                                                                    (SETQ DESTFRAGTOP DESTFRAGBOTTOM)
                                                                    (SETQ DESTFRAGBOTTOM (- DESTFRAGTOP (BITMAPHEIGHT
                                                                                                         DESTFRAG))))
                        (COND
                           ((<= SRCEFRAGBOTTOM SRCEBOTTOM)
                            (SETQ SRCE-H (- SRCETOP SRCEBOTTOM)))
                           (T (SETQ SRCE-H (- SRCETOP SRCEFRAGBOTTOM))))
                        (COND
                           ((<= DESTFRAGBOTTOM DESTBOTTOM)
                            (SETQ DEST-H (- DESTTOP DESTBOTTOM)))
                           (T (SETQ DEST-H (- DESTTOP DESTFRAGBOTTOM))))
                        (SETQ H (MIN DEST-H SRCE-H))             ; Decriments Height
                        (SETQ NEXT-S-TOP (- SRCETOP H))
                        (SETQ NEXT-D-TOP (- DESTTOP H))
                        (SETQ SBOTTOM (- NEXT-S-TOP SRCEFRAGBOTTOM))
                        (SETQ DBOTTOM (- NEXT-D-TOP DESTFRAGBOTTOM))
                        (ORG.BITBLT SRCEFRAG SRCELEFT SBOTTOM DESTFRAG DESTLEFT DBOTTOM WIDTH H SRCETYPE
                               OPERATION TEXTURE CLIPPINGREGION)
                        (COND
                           ((> (SETQ HEIGHT (- HEIGHT H))
                               0)
                            (SETQ SRCETOP NEXT-S-TOP)
                            (SETQ DESTTOP NEXT-D-TOP)
                            (COND
                               ((<= NEXT-S-TOP SRCEFRAGBOTTOM)
                                (SETQ SRCEFRAG (|GetNewFragment| SRCEBMLIST))
                                (SETQ SRCEFRAGTOP SRCEFRAGBOTTOM)
                                (SETQ SRCEFRAGBOTTOM (- SRCEFRAGTOP (BITMAPHEIGHT SRCEFRAG)))))
                            (COND
                               ((<= NEXT-D-TOP DESTFRAGBOTTOM)
                                (SETQ DESTFRAG (|GetNewFragment| DESTBMLIST))
                                (COND
                                   ((NOT DESTFRAG)
                                    (RETURN)))
                                (SETQ DESTFRAGTOP DESTFRAGBOTTOM)
```

```
                            (SETQ DESTFRAGBOTTOM (- DESTFRAGTOP (BITMAPHEIGHT DESTFRAG)))))
                        (GO LOOP)
                        ;; I hate goto, but this is temporary one

                        ))))
                (T (PROG NIL
                     LOOP2
                          (COND
                             ((<= SRCEFRAGBOTTOM SRCEBOTTOM)
                              ;; bottom edge
                              (SETQ SRCE-H (- SRCETOP SRCEBOTTOM))
                              ;; BIGBM to BITMAP case

                              )
                             (T (SETQ SRCE-H (- SRCETOP SRCEFRAGBOTTOM))))
                          (SETQ H (MIN HEIGHT SRCE-H))
                          (SETQ NEXT-S-TOP (- SRCETOP H))
                          (SETQ NEXT-D-TOP (- DESTTOP H))
                          (SETQ SBOTTOM (- NEXT-S-TOP SRCEFRAGBOTTOM))
                          (ORG.BITBLT SRCEFRAG SRCELEFT SBOTTOM DEST DESTLEFT NEXT-D-TOP WIDTH H SRCETYPE
                                 OPERATION TEXTURE CLIPPINGREGION)
                          (COND
                             ((> (SETQ HEIGHT (- HEIGHT H))
                                 0)
                              (SETQ SRCETOP NEXT-S-TOP)
                              (SETQ DESTTOP NEXT-D-TOP)
                              (COND
                                 ((<= NEXT-S-TOP SRCEFRAGBOTTOM)
                                  (SETQ SRCEFRAG (|GetNewFragment| SRCEBMLIST))
                                  ;; Get next SRCE fragment
                                  (SETQ SRCEFRAGTOP SRCEFRAGBOTTOM)
                                  (SETQ SRCEFRAGBOTTOM (- SRCEFRAGTOP (BITMAPHEIGHT SRCEFRAG)))))
                              (GO LOOP2)
                              ;; I hate goto, but this is temporary one

                              ))))))
          ((OR (|type?| BIGBM DEST))
           (PROG NIL
                 (SETQ DESTBMLIST (|fetch| (BIGBM BIGBMLIST) |of| DEST))
                 (SETQ DESTBIGBMHEIGHT (|fetch| (BIGBM BIGBMHEIGHT) |of| DEST))
                 (SETQ DESTFRAG (|GetNewFragment| DESTBMLIST))
                 (SETQ DESTFRAGTOP DESTBIGBMHEIGHT)
                 (SETQ DESTFRAGBOTTOM (- DESTFRAGTOP (BITMAPHEIGHT DESTFRAG)))
                 (|until| (< DESTFRAGBOTTOM DESTTOP) |do| ;; Serch the first fragment of DEST bitmaps
                                                          (SETQ DESTFRAG (|GetNewFragment| DESTBMLIST))
                                                          (SETQ DESTFRAGTOP DESTFRAGBOTTOM)
                                                          (SETQ DESTFRAGBOTTOM (- DESTFRAGTOP (BITMAPHEIGHT
                                                                                               DESTFRAG))))
                 (COND
                    ((<= DESTFRAGBOTTOM DESTBOTTOM)
                     ;; bottom edge
                     (SETQ DEST-H (- DESTTOP DESTBOTTOM)))
                    (T (SETQ DEST-H (- DESTTOP DESTFRAGBOTTOM))))
            LOOP3
                 (COND
                    ((<= DESTFRAGBOTTOM DESTBOTTOM)
                     (SETQ DEST-H (- DESTTOP DESTBOTTOM)))
                    (T (SETQ DEST-H (- DESTTOP DESTFRAGBOTTOM))))
                 (SETQ H (MIN DEST-H HEIGHT))
                 (SETQ NEXT-S-TOP (- SRCETOP H))
                 (SETQ NEXT-D-TOP (- DESTTOP H))
                 (SETQ DBOTTOM (- NEXT-D-TOP DESTFRAGBOTTOM))
                 (ORG.BITBLT SRCE SRCELEFT NEXT-S-TOP DESTFRAG DESTLEFT DBOTTOM WIDTH H SRCETYPE OPERATION
                        TEXTURE CLIPPINGREGION)
                 (COND
                    ((> (SETQ HEIGHT (- HEIGHT H))
                        0)
                     (SETQ DESTTOP NEXT-D-TOP)
                     (SETQ SRCETOP NEXT-S-TOP)
                     (COND
                        ((<= NEXT-D-TOP DESTFRAGBOTTOM)
                         (SETQ DESTFRAG (|GetNewFragment| DESTBMLIST))
                         (SETQ DESTFRAGTOP DESTFRAGBOTTOM)
                         (SETQ DESTFRAGBOTTOM (- DESTFRAGTOP (BITMAPHEIGHT DESTFRAG)))))
                     (GO LOOP3)
                     ;; I hate goto, but this is temporary one

                     ))))
          (T ;; Normal case, use BITBLT
             (ORG.BITBLT SRCE SRCELEFT SRCEBOTTOM DEST DESTLEFT DESTBOTTOM WIDTH HEIGHT SRCETYPE OPERATION
```

```
                    TEXTURE CLIPPINGREGION))))))
```

(BITMAPCREATE.BIGBM

```
(LAMBDA (WIDTH HEIGHT BITSPERPIXEL)                          ; Edited  7-Sep-89 18:14 by takeshi
  (LET (H HLEFT BM BIGBM)
       (SETQ H (FLOOR (IQUOTIENT |\\MaxBitMapWords| WIDTH)
                      BITSPERWORD))                          ; slice should be a multiple of 16 so that textures tesselate nicely.
       (SETQ HLEFT HEIGHT)
       (SETQ BIGBM (|create| BIGBM))
       (|freplace| (BIGBM BIGBMWIDTH) OF BIGBM WITH WIDTH)
       (|freplace| (BIGBM BIGBMHEIGHT) OF BIGBM WITH HEIGHT)
       (|freplace| (BIGBM BIGBMLIST) OF BIGBM WITH (|while| (IGREATERP HLEFT 0)
                                                           |collect| (SETQ BM (BITMAPCREATE WIDTH (MIN H HLEFT)
                                                                                            BITSPERPIXEL))
                                                                     (SETQ HLEFT (- HLEFT H))
                                                                     BM))
       BIGBM)))
```

(BITMAPCREATE

```
(LAMBDA (WIDTH HEIGHT BITSPERPIXEL)                          ; Edited  1-Nov-91 15:47 by jds
                                                             ; creates a bitmap  & bigbm data structure.
  (PROG (RW)
        (OR (AND (IGEQ WIDTH 0)
                 (ILEQ WIDTH |\\MaxBitMapWidth|))
            (\\ILLEGAL.ARG WIDTH))
        (OR (AND (IGEQ HEIGHT 0)
                 (ILEQ HEIGHT |\\MaxBitMapHeight|))
            (\\ILLEGAL.ARG HEIGHT))
```

;; WIDTH & HEIGHT are now known to be OK.

```
        (SETQ BITSPERPIXEL (\\INSUREBITSPERPIXEL BITSPERPIXEL))
        (SETQ RW (FOLDHI (ITIMES WIDTH BITSPERPIXEL)
                         BITSPERWORD))
        (RETURN (COND
                   ((NOT (IGREATERP (ITIMES RW HEIGHT)
                                    |\\MaxBitMapWords|))
                    (|create| BITMAP
                           BITMAPRASTERWIDTH _ RW
                           BITMAPWIDTH _ WIDTH
                           BITMAPHEIGHT _ HEIGHT
                           BITMAPBITSPERPIXEL _ BITSPERPIXEL
                           BITMAPBASE _ (\\ALLOCBLOCK (FOLDHI (ITIMES RW HEIGHT)
                                                              WORDSPERCELL)
                                                      NIL
                                                      (AND (NULL WINDFLG)
                                                           0))))
                   (T (BITMAPCREATE.BIGBM WIDTH HEIGHT BITSPERPIXEL)))))))
```

(BITMAPCOPY

```
(LAMBDA (BITMAP)                                             ; Edited  1-Nov-91 15:49 by jds
```

;; makes a copy of an existing BitMap

```
  (PROG (NEWBITMAP)
        (BITBLT BITMAP 0 0 (SETQ NEWBITMAP (BITMAPCREATE (BITMAPWIDTH BITMAP)
                                                         (BITMAPHEIGHT BITMAP)
                                                         (BITSPERPIXEL BITMAP)))
               0 0 NIL NIL 'INPUT 'REPLACE 0)
        (RETURN NEWBITMAP))))
```

(BIGBITMAPEQUAL

```
(LAMBDA (BM1 BM2)                                            ; Edited 31-Jul-2023 13:08 by rmk
```

;; Fields may not be SMALLP

```
  (AND (|type?| BIGBM |of| BM1)
       (|type?| BIGBM |of| BM2)
       (IEQP (|ffetch| (BIGBM BIGBMWIDTH) |of| BM1)
             (|ffetch| (BIGBM BIGBMWIDTH) |of| BM2))
       (IEQP (|ffetch| (BIGBM BIGBMHEIGHT) |of| BM1)
             (|ffetch| (BIGBM BIGBMHEIGHT) |of| BM2))
       (|for| B1 |in| (|ffetch| (BIGBM BIGBMLIST) |of| BM1) |as| B2 |in| (|ffetch| (BIGBM BIGBMLIST) |of| BM2)
          |always| (EQUALBITMAPP B1 B2)))))
```

(BLTSHADE.BIGBM

```
(LAMBDA (TEXTURE DESTINATION DESTLEFT DESTBOTTOM WIDTH HEIGHT OPERATION CLIPPINGREGION)
                                                             ; Edited 17-Oct-89 19:01 by takeshi
  (LET (H SLITHEIGHT)
```

;; Clippingregion is handled incorrectly (at least in the Y direction).

```
       (COND
          ((NOT (|type?| BIGBM DESTINATION))
           (ORG.BLTSHADE TEXTURE DESTINATION DESTLEFT DESTBOTTOM WIDTH HEIGHT OPERATION CLIPPINGREGION))
          (T (PROG (DESTTOP DESTBMLIST DESTBIGBMHEIGHT DESTFRAG DESTFRAGTOP DESTFRAGBOTTOM DEST-H NEXT-D-TOP
```

```
                        DBOTTOM)
                    (SETQ DESTTOP (IPLUS DESTBOTTOM HEIGHT))
                    (SETQ DESTBMLIST (|fetch| (BIGBM BIGBMLIST) |of| DESTINATION))
                    (SETQ DESTBIGBMHEIGHT (|fetch| (BIGBM BIGBMHEIGHT) |of| DESTINATION))
                    (SETQ DESTFRAG (|GetNewFragment| DESTBMLIST))
                    (SETQ DESTFRAGTOP DESTBIGBMHEIGHT)
                    (SETQ DESTFRAGBOTTOM (- DESTFRAGTOP (BITMAPHEIGHT DESTFRAG)))
                LOOP
                    (|until| (<= DESTFRAGBOTTOM DESTTOP) |do|  ;; Serch the first fragment of DEST bitmaps
                                                               (SETQ DESTFRAG (|GetNewFragment| DESTBMLIST))
                                                               (SETQ DESTFRAGTOP DESTFRAGBOTTOM)
                                                               (SETQ DESTFRAGBOTTOM (- DESTFRAGTOP (BITMAPHEIGHT
                                                                                                    DESTFRAG))))
                    (COND
                       ((<= DESTFRAGBOTTOM DESTBOTTOM)
                        (SETQ DEST-H (- DESTTOP DESTBOTTOM)))
                       (T (SETQ DEST-H (- DESTTOP DESTFRAGBOTTOM))))
                    (SETQ NEXT-D-TOP (- DESTTOP DEST-H))
                    (SETQ DBOTTOM (- NEXT-D-TOP DESTFRAGBOTTOM))
                    (ORG.BLTSHADE TEXTURE DESTFRAG DESTLEFT DBOTTOM WIDTH DEST-H OPERATION CLIPPINGREGION)
                    (COND
                       ((> (SETQ HEIGHT (- HEIGHT DEST-H))
                           0)
                        (SETQ DESTTOP NEXT-D-TOP)
                        (COND
                           ((<= NEXT-D-TOP DESTFRAGBOTTOM)
                            (SETQ DESTFRAG (|GetNewFragment| DESTBMLIST))
                            (SETQ DESTFRAGTOP DESTFRAGBOTTOM)
                            (SETQ DESTFRAGBOTTOM (- DESTFRAGTOP (BITMAPHEIGHT DESTFRAG)))))
                        (GO LOOP)
                       ;; I hate goto, but this is temporary one

                       ))))))))
```

(**BITBLT**

```
(LAMBDA (SOURCE SOURCELEFT SOURCEBOTTOM DESTINATION DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT SOURCETYPE
           OPERATION TEXTURE CLIPPINGREGION)                    ; Edited 29-Jun-90 10:43 by matsuda
   (DECLARE (LOCALVARS . T))
   ;; IRM defined defaults
   (OR DESTINATIONLEFT (SETQ DESTINATIONLEFT 0))
   (OR DESTINATIONBOTTOM (SETQ DESTINATIONBOTTOM 0))
   (COND
      ((EQ SOURCETYPE 'TEXTURE)
       (COND
          ((|type?| BITMAP DESTINATION)
           (\\BLTSHADE.BITMAP TEXTURE DESTINATION DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT OPERATION
                  CLIPPINGREGION))
          ((|type?| BIGBM DESTINATION)
           (BLTSHADE.BIGBM TEXTURE DESTINATION DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT OPERATION
                  CLIPPINGREGION))
          (T (PROG ((STREAM (\\OUTSTREAMARG DESTINATION)))
                   (RETURN (IMAGEOP 'IMBLTSHADE STREAM TEXTURE STREAM DESTINATIONLEFT DESTINATIONBOTTOM WIDTH
                                    HEIGHT OPERATION CLIPPINGREGION))))))
      (T (COND
            ((OR (|type?| BIGBM SOURCE)
                 (|type?| BIGBM DESTINATION))
             (BITBLT.BIGBM SOURCE SOURCELEFT SOURCEBOTTOM DESTINATION DESTINATIONLEFT DESTINATIONBOTTOM WIDTH
                    HEIGHT SOURCETYPE OPERATION TEXTURE CLIPPINGREGION))
            (T (ORG.BITBLT SOURCE SOURCELEFT SOURCEBOTTOM DESTINATION DESTINATIONLEFT DESTINATIONBOTTOM WIDTH
                      HEIGHT SOURCETYPE OPERATION TEXTURE CLIPPINGREGION)))))))
```

(**ORG.BITBLT**

```
(LAMBDA (SOURCE SOURCELEFT SOURCEBOTTOM DESTINATION DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT SOURCETYPE
           OPERATION TEXTURE CLIPPINGREGION)
   (DECLARE (LOCALVARS . T))                                    ; Edited 24-Jul-90 16:34 by matsuda
   ;; IRM defined defaults
   (OR DESTINATIONLEFT (SETQ DESTINATIONLEFT 0))
   (OR DESTINATIONBOTTOM (SETQ DESTINATIONBOTTOM 0))
   (COND
      ((EQ SOURCETYPE 'TEXTURE)
       (COND
          ((|type?| BITMAP DESTINATION)
           (\\BLTSHADE.BITMAP TEXTURE DESTINATION DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT OPERATION
                  CLIPPINGREGION))
          ((|type?| BIGBM DESTINATION)
           (BLTSHADE.BIGBM TEXTURE DESTINATION DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT OPERATION
                  CLIPPINGREGION))
          (T (PROG ((STREAM (\\OUTSTREAMARG DESTINATION)))
                   (RETURN (IMAGEOP 'IMBLTSHADE STREAM TEXTURE STREAM DESTINATIONLEFT DESTINATIONBOTTOM WIDTH
                                    HEIGHT OPERATION CLIPPINGREGION))))))
      (T (PROG (SOURCEDD SOURCEBM CLIPPEDSOURCELEFT CLIPPEDSOURCEBOTTOM)
               (COND
```

```
                ((|type?| BITMAP SOURCE)
                 (OR SOURCELEFT (SETQ SOURCELEFT 0))
                 (OR SOURCEBOTTOM (SETQ SOURCEBOTTOM 0))
                 (SETQ SOURCEBM SOURCE)
                 (SETQ CLIPPEDSOURCELEFT SOURCELEFT)
                 (SETQ CLIPPEDSOURCEBOTTOM SOURCEBOTTOM)      ; limit the WIDTH and HEIGHT to the source size.
                 (SETQ WIDTH (COND
                                (WIDTH (IMIN WIDTH (IDIFFERENCE (|fetch| (BITMAP BITMAPWIDTH) |of| SOURCE)
                                                                SOURCELEFT)))
                                (T (|fetch| (BITMAP BITMAPWIDTH) |of| SOURCE))))
                 (SETQ HEIGHT (COND
                                 (HEIGHT (IMIN HEIGHT (IDIFFERENCE (|fetch| (BITMAP BITMAPHEIGHT) |of| SOURCE)
                                                                   SOURCEBOTTOM)))
                                 (T (|fetch| (BITMAP BITMAPHEIGHT) |of| SOURCE)))))
                ((SETQ SOURCEDD (\\GETDISPLAYDATA SOURCE))
                 (OR SOURCELEFT (SETQ SOURCELEFT (|fetch| (REGION LEFT) |of| (|ffetch| (\\DISPLAYDATA
                                                                                        |DDClippingRegion|)
                                                                                    |of| SOURCEDD))))
                 (OR SOURCEBOTTOM (SETQ SOURCEBOTTOM (|fetch| (REGION BOTTOM) |of| (|ffetch| (\\DISPLAYDATA
                                                                                               |DDClippingRegion|
                                                                                               )
                                                                                           |of| SOURCEDD))))
                                                              ; do transformations coming out of source
                 (SETQ SOURCEBM (|fetch| (\\DISPLAYDATA |DDDestination|) |of| SOURCEDD))
                 (SETQ CLIPPEDSOURCELEFT (IMAX (SETQ SOURCELEFT (\\DSPTRANSFORMX SOURCELEFT SOURCEDD))
                                               (|fetch| (\\DISPLAYDATA |DDClippingLeft|) |of| SOURCEDD)))
                 (SETQ CLIPPEDSOURCEBOTTOM (IMAX (SETQ SOURCEBOTTOM (\\DSPTRANSFORMY SOURCEBOTTOM SOURCEDD))
                                                 (|fetch| (\\DISPLAYDATA |DDClippingBottom|) |of| SOURCEDD)))
                                                              ; limit the WIDTH and HEIGHT by the source dimensions.
                 (SETQ WIDTH (COND
                                (WIDTH (IMIN WIDTH (IDIFFERENCE (|fetch| (\\DISPLAYDATA |DDClippingRight|)
                                                                         |of| SOURCEDD)
                                                                CLIPPEDSOURCELEFT)))
                                (T (IDIFFERENCE (|fetch| (\\DISPLAYDATA |DDClippingRight|) |of| SOURCEDD)
                                                CLIPPEDSOURCELEFT))))
                 (SETQ HEIGHT (COND
                                 (HEIGHT (IMIN HEIGHT (IDIFFERENCE (|fetch| (\\DISPLAYDATA |DDClippingTop|)
                                                                           |of| SOURCEDD)
                                                                   CLIPPEDSOURCEBOTTOM)))
                                 (T (IDIFFERENCE (|fetch| (\\DISPLAYDATA |DDClippingTop|) |of| SOURCEDD)
                                                 CLIPPEDSOURCEBOTTOM))))
                                                              ; if texture is not given, use the display stream's.
                 (OR TEXTURE (SETQ TEXTURE (|ffetch| (\\DISPLAYDATA |DDTexture|) |of| SOURCEDD)))))
            (COND
               ((OR (IGEQ 0 WIDTH)
                    (IGEQ 0 HEIGHT))                          ; if either width or height is 0, don't do anything.
                (RETURN)))
            (RETURN (COND
                       ((|type?| BITMAP DESTINATION)
                        (COND
                           ((WINDOWP SOURCE)
                            ;; bring source window to the top.  Note: this doesn't work if the user passes in a display stream onto the
                            ;; screen instead of a window.
                            (.WHILE.TOP.DS. (\\OUTSTREAMARG SOURCE)
                                   (\\BITBLT.BITMAP SOURCEBM SOURCELEFT SOURCEBOTTOM DESTINATION
                                          DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT SOURCETYPE
                                          OPERATION TEXTURE CLIPPINGREGION CLIPPEDSOURCELEFT
                                          CLIPPEDSOURCEBOTTOM)))
                            (T (PROG ((DESTNBITS (BITSPERPIXEL DESTINATION))
                                      (SRCNBITS (BITSPERPIXEL SOURCEBM)))
                                     (COND
                                        ((NOT (EQ SRCNBITS DESTNBITS))
                                         (COND
                                            ((EQ DESTNBITS 1)
                                             (SETQ SOURCEBM (UNCOLORIZEBITMAP SOURCEBM (COLORMAP SRCNBITS)))
                                             )))))
                                  (\\BITBLT.BITMAP SOURCEBM SOURCELEFT SOURCEBOTTOM DESTINATION DESTINATIONLEFT
                                         DESTINATIONBOTTOM WIDTH HEIGHT SOURCETYPE OPERATION TEXTURE
                                         CLIPPINGREGION CLIPPEDSOURCELEFT CLIPPEDSOURCEBOTTOM))))
                       (T (PROG (STREAM)
                                (SETQ STREAM (\\OUTSTREAMARG DESTINATION))
                                (COND
                                   ((AND (NEQ SOURCE DESTINATION)
                                         (WINDOWP SOURCE))
                                    ;; both source and destination are windows, see if they overlap and use an intermediate bitmap.
                                    ;; Note: this doesn't work if the user passes in a display stream onto the screen instead of a window.
                                    (COND
                                       ((WINDOWP DESTINATION)
                                        (COND
                                           ((WOVERLAPP SOURCE DESTINATION)
                                            (RETURN (PROG (SCRATCHBM)
                                                          (.WHILE.TOP.DS. (\\OUTSTREAMARG SOURCE)
                                                                 (BITBLT SOURCEBM SOURCELEFT SOURCEBOTTOM
                                                                        (SETQ SCRATCHBM (BITMAPCREATE
                                                                                         WIDTH HEIGHT
```

```
                                                                                    (BITSPERPIXEL
                                                                                       SOURCEBM)))
                                                                          0 0 WIDTH HEIGHT 'INPUT 'REPLACE))
                                                                (RETURN (BITBLT SCRATCHBM 0 0 STREAM
                                                                          DESTINATIONLEFT DESTINATIONBOTTOM
                                                                          WIDTH HEIGHT SOURCETYPE OPERATION
                                                                          TEXTURE CLIPPINGREGION))))))))
                                                                ; bring the source to the top.  this should be done uninterruptably
                                                                ; but is better than nothing.
                                                    (TOTOPW SOURCE)))
                                               (IMAGEOP 'IMBITBLT STREAM SOURCEBM SOURCELEFT SOURCEBOTTOM STREAM
                                                      DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT SOURCETYPE OPERATION
                                                      TEXTURE CLIPPINGREGION CLIPPEDSOURCELEFT CLIPPEDSOURCEBOTTOM))))))))
)
```

(**BLTSHADE.DISPLAY**

```
(LAMBDA (TEXTURE STREAM DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT OPERATION CLIPPINGREGION)
                                                                ; Edited 21-Dec-90 10:41 by matsuda
                                                                ; BLTSHADE to a display stream
    (DECLARE (LOCALVARS . T))
    (PROG (|left| |top| |bottom| |right| DESTINATIONBITMAP DESTDD DESTINATIONNBITS)
          (SETQ DESTDD (|fetch| (STREAM IMAGEDATA) |of| STREAM))
          (SETQ DESTINATIONBITMAP (|fetch| (\\DISPLAYDATA |DDDestination|) |of| DESTDD))
     ;; bring it to top so that its TOTOPFNs will get called before the destination information is cached in case one of them moves, reshapes, etc. the
     ;; window
     ;; We'd rather handle the slow case when we are interruptable, so we do it here as a heuristic.  But we might get interrupted before we go
     ;; interruptable, so we do it there too.
          (\\INSURETOPWDS STREAM)
          (SETQ DESTINATIONLEFT (\\DSPTRANSFORMX DESTINATIONLEFT DESTDD))
          (SETQ DESTINATIONBOTTOM (\\DSPTRANSFORMY DESTINATIONBOTTOM DESTDD))
          (PROGN                                                ; compute limits based on clipping regions.
                 (SETQ |left| (|fetch| (\\DISPLAYDATA |DDClippingLeft|) |of| DESTDD))
                 (SETQ |bottom| (|fetch| (\\DISPLAYDATA |DDClippingBottom|) |of| DESTDD))
                 (SETQ |right| (|fetch| (\\DISPLAYDATA |DDClippingRight|) |of| DESTDD))
                 (SETQ |top| (|fetch| (\\DISPLAYDATA |DDClippingTop|) |of| DESTDD))
                 (COND
                    (CLIPPINGREGION                             ; hard case, two destination clipping regions: do calculations to
                                                                ; merge them.
                           (PROG (CRLEFT CRBOTTOM)
                                 (SETQ |left| (IMAX |left| (SETQ CRLEFT (\\DSPTRANSFORMX (|fetch| (REGION LEFT)
                                                                                             |of| CLIPPINGREGION)
                                                                                        DESTDD))))
                                 (SETQ |bottom| (IMAX |bottom| (SETQ CRBOTTOM (\\DSPTRANSFORMY
                                                                                 (|fetch| (REGION BOTTOM)
                                                                                    |of| CLIPPINGREGION)
                                                                                 DESTDD))))
                                 (SETQ |right| (IMIN |right| (IPLUS CRLEFT (|fetch| (REGION WIDTH) |of|
                                                                                                  CLIPPINGREGION
                                                                                    ))))
                                 (SETQ |top| (IMIN |top| (IPLUS CRBOTTOM (|fetch| (REGION HEIGHT) |of|
                                                                                                  CLIPPINGREGION
                                                                                 ))))))))
          (SETQ DESTINATIONNBITS (BITSPERPIXEL DESTINATIONBITMAP))
     ;; left, right top and bottom are the limits in destination taking into account Clipping Regions.  Clip to region in the arguments of this call.
          (PROGN (SETQ |left| (IMAX DESTINATIONLEFT |left|))
                 (SETQ |bottom| (IMAX DESTINATIONBOTTOM |bottom|))
                 (COND
                    (WIDTH                                      ; WIDTH is optional
                           (SETQ |right| (IMIN (IPLUS DESTINATIONLEFT WIDTH)
                                               |right|))))
                 (COND
                    (HEIGHT                                     ; HEIGHT is optional
                           (SETQ |top| (IMIN (IPLUS DESTINATIONBOTTOM HEIGHT)
                                             |top|)))))
          (COND
             ((OR (ILEQ |right| |left|)
                  (ILEQ |top| |bottom|))                        ; there is nothing to move.
              (RETURN)))
          (SETQ TEXTURE (SELECTQ (TYPENAME TEXTURE)
                                 (LITATOM (COND
                                             ((NULL TEXTURE)    ; NIL case.  default texture to background texture.
                                              (|ffetch| (\\DISPLAYDATA |DDTexture|) |of| DESTDD))
                                             ((NOT (EQ DESTINATIONNBITS 1))
                                                                ; should be a color name
                                              (OR (COLORNUMBERP TEXTURE DESTINATIONNBITS T)
                                                  (\\ILLEGAL.ARG TEXTURE)))
                                             (T (\\ILLEGAL.ARG TEXTURE))))
                                 ((SMALLP FIXP)
                                     (LOGAND TEXTURE (MAXIMUMSHADE DESTINATIONNBITS)))
                                 (BITMAP TEXTURE)
                                 (LISTP                         ; should be a list of levels rgb or hls.
                                        (OR (AND (NOT (EQ DESTINATIONNBITS 1))
                                                 (COLORNUMBERP TEXTURE DESTINATIONNBITS))
                                            (\\ILLEGAL.ARG TEXTURE)))
```

```
                        (\\ILLEGAL.ARG TEXTURE)))
          (COND
             ((NOT (EQ DESTINATIONNBITS 1))
              (COND
                 ((NOT (|type?| BIGBM DESTINATIONBITMAP))
                  (SETQ |left| (ITIMES DESTINATIONNBITS |left|))
                  (SETQ |right| (ITIMES DESTINATIONNBITS |right|))))
              (SETQ TEXTURE (COLORTEXTUREFROMCOLOR# TEXTURE DESTINATIONNBITS))))
          (.WHILE.TOP.DS. STREAM (COND
                                    ((NOT (|type?| BIGBM DESTINATIONBITMAP))
                                     (PROG (HEIGHT)
                                           (SETQ HEIGHT (IDIFFERENCE |top| |bottom|))
                                           (|replace| (PILOTBBT PBTWIDTH) |of| \\SYSPILOTBBT |with| (IDIFFERENCE
                                                                                                    |right| |left|))
                                           (|replace| (PILOTBBT PBTHEIGHT) |of| \\SYSPILOTBBT |with| HEIGHT)
                                           (\\BITBLTSUB \\SYSPILOTBBT NIL |left| NIL DESTINATIONBITMAP |left|
                                                  (|\\SFInvert| DESTINATIONBITMAP |top|)
                                                  HEIGHT
                                                  'TEXTURE
                                                  (OR OPERATION (|ffetch| (\\DISPLAYDATA DDOPERATION) |of| DESTDD))
                                                  TEXTURE
                                                  (ITIMES DESTINATIONNBITS (|fetch| (\\DISPLAYDATA DDXOFFSET)
                                                                                   |of| DESTDD))
                                                  (|fetch| (\\DISPLAYDATA DDYOFFSET) |of| DESTDD))))
                                    (T (PROG (HEIGHT)
                                             (SETQ HEIGHT (IDIFFERENCE |top| |bottom|))
                                             (**BLTSHADE.BIGBM** TEXTURE DESTINATIONBITMAP |left| |bottom|
                                                    (IDIFFERENCE |right| |left|)
                                                    (IDIFFERENCE |top| |bottom|)
                                                    (OR OPERATION (|ffetch| (\\DISPLAYDATA DDOPERATION)
                                                                            |of| DESTDD))
                                                    CLIPPINGREGION)))))
          (RETURN T))))
```

(**RESHOWBORDER1**

```
(LAMBDA (NEWBORDER OLDBORDER WINDOW)                    ; Edited 26-Jul-90 12:52 by matsuda

    ;; redisplays the border of a window.  Is called by RESHOWBORDER and RESHOWTITLE.  It doesn't check for equality between the new and old
    ;; borders because it is also used when a title is added or deleted.

    (PROG ((REGION (|fetch| (WINDOW REG) |of| WINDOW))
           (OLDSAVE (|fetch| (WINDOW SAVE) |of| WINDOW))
           NUSAV DELTA NUWIDTH NUHEIGHT)
          (SETQ DELTA (IDIFFERENCE NEWBORDER OLDBORDER))
          (SETQ NUWIDTH (IPLUS (|fetch| (REGION WIDTH) |of| REGION)
                               (ITIMES DELTA 2)))
          (SETQ NUHEIGHT (IDIFFERENCE (IPLUS (|fetch| (REGION HEIGHT) |of| (DSPCLIPPINGREGION NIL
                                                                                (|fetch| (WINDOW DSP) |of| WINDOW)))
                                             (ITIMES NEWBORDER 2))
                              (COND
                                 ((|fetch| (WINDOW WTITLE) |of| WINDOW)
                                  (DSPLINEFEED NIL (|fetch| (SCREEN SCTITLEDS) |of| (|fetch| (WINDOW SCREEN)
                                                                                            |of| WINDOW))))
                                 (T 0))))
          (SETQ NUSAV (**BITMAPCREATE** NUWIDTH NUHEIGHT (**BITSPERPIXEL** OLDSAVE)))
          (.WHILE.TOP.DS. WINDOW                        ; Save window image
                 (\\**SW2BM** (|fetch| (SCREEN SCDESTINATION) |of| (|fetch| (WINDOW SCREEN) |of| WINDOW))
                        REGION
                        (|fetch| (WINDOW SAVE) |of| WINDOW)
                        NIL)                            ; put new save image into window
                 (|replace| (WINDOW SAVE) |of| WINDOW |with| NUSAV)
                 (|replace| (WINDOW WBORDER) |of| WINDOW |with| NEWBORDER)
                                                        ; create a region that coresponds to the old region with the new
                                                        ; border.
                 (|replace| (WINDOW REG) |of| WINDOW |with| (|create| REGION
                                                                LEFT _ (IDIFFERENCE (|fetch| (REGION LEFT)
                                                                                            |of| REGION)
                                                                                    DELTA)
                                                                BOTTOM _ (IDIFFERENCE (|fetch| (REGION BOTTOM)
                                                                                             |of| REGION)
                                                                                      DELTA)
                                                                WIDTH _ NUWIDTH
                                                                HEIGHT _ NUHEIGHT))
                 (UPDATE/SCROLL/REG WINDOW)             ; draw border in the new image.
                 (SHOWWFRAME WINDOW)                    ; copy the visible part from the old image into the new one.
                 (**BITBLT** OLDSAVE OLDBORDER OLDBORDER NUSAV NEWBORDER NEWBORDER (IDIFFERENCE (**BITMAPWIDTH**
                                                                                                   OLDSAVE)
                                                                                                (ITIMES 2 OLDBORDER))
                        (|fetch| (REGION HEIGHT) |of| (DSPCLIPPINGREGION NIL (|fetch| (WINDOW DSP) |of| WINDOW)))
                        'INPUT
                        'REPLACE)                       ; put the new image up on the screen.
                 (\\**SW2BM** (|fetch| (SCREEN SCDESTINATION) |of| (|fetch| (WINDOW SCREEN) |of| WINDOW))
                        (|fetch| (WINDOW REG) |of| WINDOW)
                        (|fetch| (WINDOW SAVE) |of| WINDOW)
                        NIL)))))

)
```

```
(DEFINEQ
```

(\\DRAWCIRCLE.BIGBM

```
  (LAMBDA (DISPLAYSTREAM CENTERX CENTERY RADIUS BRUSH DASHING)   ; Edited 29-Jan-91 16:25 by matsuda
    (DECLARE (LOCALVARS . T))
    (PROG ((DD (|fetch| IMAGEDATA |of| DISPLAYSTREAM))
           BITMAP)
          (SETQ BITMAP (|fetch| (\\DISPLAYDATA |DDDestination|) |of| DD))
          (COND
             ((|type?| BIGBM BITMAP)
              (PROG (BIGBMLIST HEIGHT BOTTOM BM |ClippingTop| |ClippingBottom| |CTop| |CBottom|)
                    (SETQ BIGBMLIST (|fetch| (BIGBM BIGBMLIST) |of| BITMAP))
                    (SETQ HEIGHT (BITMAPHEIGHT BITMAP))
                    (SETQ |ClippingTop| (|ffetch| (\\DISPLAYDATA |DDClippingTop|) |of| DD))
                    (SETQ |ClippingBottom| (|ffetch| (\\DISPLAYDATA |DDClippingBottom|) |of| DD))
                    (SETQ BM (|GetNewFragment| BIGBMLIST))
                    (|while| (AND BM (IGREATERP HEIGHT |ClippingBottom|))
                       |do| (SETQ BOTTOM (IDIFFERENCE HEIGHT (BITMAPHEIGHT BM)))
                            (SETQ |CTop| (COND
                                           ((IGREATERP |ClippingTop| HEIGHT)
                                            (IDIFFERENCE HEIGHT BOTTOM))
                                           (T (IDIFFERENCE |ClippingTop| BOTTOM))))
                            (COND
                               ((IGEQ |CTop| 0)
                                (SETQ |CBottom| (COND
                                                  ((ILESSP |ClippingBottom| BOTTOM)
                                                   0)
                                                  (T (IDIFFERENCE |ClippingBottom| BOTTOM))))
                                (|replace| (\\DISPLAYDATA |DDDestination|) |of| DD |with| BM)
                                (|replace| (\\DISPLAYDATA |DDClippingTop|) |of| DD |with| |CTop|)
                                (|replace| (\\DISPLAYDATA |DDClippingBottom|) |of| DD |with| |CBottom|)
                                (\\DRAWCIRCLE.DISPLAY DISPLAYSTREAM CENTERX (IDIFFERENCE CENTERY BOTTOM)
                                       RADIUS BRUSH DASHING)
                                (SETQ BM (|GetNewFragment| BIGBMLIST))
                                (SETQ HEIGHT BOTTOM))))
                    (|replace| (\\DISPLAYDATA |DDDestination|) |of| DD |with| BITMAP)
                    (|replace| (\\DISPLAYDATA |DDClippingTop|) |of| DD |with| |ClippingTop|)
                    (|replace| (\\DISPLAYDATA |DDClippingBottom|) |of| DD |with| |ClippingBottom|)
                    (MOVETO CENTERX CENTERY DISPLAYSTREAM)
                    (RETURN NIL)))
             (T (\\DRAWCIRCLE.DISPLAY DISPLAYSTREAM CENTERX CENTERY RADIUS BRUSH DASHING))))))
```

(\\FILLCIRCLE.BIGBM

```
  (LAMBDA (DISPLAYSTREAM CENTERX CENTERY RADIUS TEXTURE)
    (DECLARE (LOCALVARS . T))                                   ; Edited 29-Jan-91 16:21 by matsuda
    (COND
       ((OR (NOT (NUMBERP RADIUS))
            (ILESSP (SETQ RADIUS (FIXR RADIUS))
                    0))
        (\\ILLEGAL.ARG RADIUS))
       (T (PROG ((DD (|fetch| IMAGEDATA |of| DISPLAYSTREAM))
                 BITMAP)
                (SETQ BITMAP (|fetch| (\\DISPLAYDATA |DDDestination|) |of| DD))
                (COND
                   ((|type?| BIGBM BITMAP)
                    (PROG (BIGBMLIST HEIGHT BOTTOM BM |ClippingTop| |ClippingBottom| |CTop| |CBottom|)
                          (SETQ BIGBMLIST (|fetch| (BIGBM BIGBMLIST) |of| BITMAP))
                          (SETQ HEIGHT (BITMAPHEIGHT BITMAP))
                          (SETQ |ClippingTop| (|ffetch| (\\DISPLAYDATA |DDClippingTop|) |of| DD))
                          (SETQ |ClippingBottom| (|ffetch| (\\DISPLAYDATA |DDClippingBottom|) |of| DD))
                          (SETQ BM (|GetNewFragment| BIGBMLIST))
                          (|while| (AND BM (IGREATERP HEIGHT |ClippingBottom|))
                             |do| (SETQ BOTTOM (IDIFFERENCE HEIGHT (BITMAPHEIGHT BM)))
                                  (SETQ |CTop| (COND
                                                 ((IGREATERP |ClippingTop| HEIGHT)
                                                  (IDIFFERENCE HEIGHT BOTTOM))
                                                 (T (IDIFFERENCE |ClippingTop| BOTTOM))))
                                  (COND
                                     ((IGEQ |CTop| 0)
                                      (SETQ |CBottom| (COND
                                                        ((ILESSP |ClippingBottom| BOTTOM)
                                                         0)
                                                        (T (IDIFFERENCE |ClippingBottom| BOTTOM))))
                                      (|replace| (\\DISPLAYDATA |DDDestination|) |of| DD |with| BM)
                                      (|replace| (\\DISPLAYDATA |DDClippingTop|) |of| DD |with| |CTop|)
                                      (|replace| (\\DISPLAYDATA |DDClippingBottom|) |of| DD |with| |CBottom|)
                                      (\\FILLCIRCLE.DISPLAY DISPLAYSTREAM CENTERX (IDIFFERENCE CENTERY BOTTOM)
                                             RADIUS TEXTURE)
                                      (SETQ BM (|GetNewFragment| BIGBMLIST))
                                      (SETQ HEIGHT BOTTOM))))
                          (|replace| (\\DISPLAYDATA |DDDestination|) |of| DD |with| BITMAP)
                          (|replace| (\\DISPLAYDATA |DDClippingTop|) |of| DD |with| |ClippingTop|)
                          (|replace| (\\DISPLAYDATA |DDClippingBottom|) |of| DD |with| |ClippingBottom|)
                          (MOVETO CENTERX CENTERY DISPLAYSTREAM)
                          (RETURN NIL)))
```

```
                    (T (\\FILLCIRCLE.DISPLAY DISPLAYSTREAM CENTERX CENTERY RADIUS TEXTURE))))))))
```

(\\DRAWELLIPSE.BIGBM

```
(LAMBDA (DISPLAYSTREAM CENTERX CENTERY SEMIMINORRADIUS SEMIMAJORRADIUS ORIENTATION BRUSH DASHING)
  (DECLARE (LOCALVARS . T))                                    ; Edited 29-Jan-91 12:52 by matsuda
  (PROG ((DD (|fetch| IMAGEDATA |of| DISPLAYSTREAM)
             BITMAP))
        (SETQ BITMAP (|fetch| (\\DISPLAYDATA |DDDestination|) |of| DD))
        (COND
           ((|type?| BIGBM BITMAP)
            (PROG ((CENTERX (FIXR CENTERX))
                   (CENTERY (FIXR CENTERY))
                   (SEMIMINORRADIUS (FIXR SEMIMINORRADIUS))
                   (SEMIMAJORRADIUS (FIXR SEMIMAJORRADIUS)))
                  (COND
                     ((OR (EQ 0 SEMIMINORRADIUS)
                          (EQ 0 SEMIMAJORRADIUS))
                      (MOVETO CENTERX CENTERY DISPLAYSTREAM)
                      (RETURN)))
                  (PROG (BIGBMLIST HEIGHT BOTTOM BM YY1 YY2 |ClippingTop| |ClippingBottom| |CTop| |CBottom|)
                        (SETQ BIGBMLIST (|fetch| (BIGBM BIGBMLIST) |of| BITMAP))
                        (SETQ HEIGHT (BITMAPHEIGHT BITMAP))
                        (SETQ |ClippingTop| (|ffetch| (\\DISPLAYDATA |DDClippingTop|) |of| DD))
                        (SETQ |ClippingBottom| (|ffetch| (\\DISPLAYDATA |DDClippingBottom|) |of| DD))
                        (SETQ BM (|GetNewFragment| BIGBMLIST))
                        (|while| (AND BM (IGREATERP HEIGHT |ClippingBottom|))
                           |do| (SETQ BOTTOM (IDIFFERENCE HEIGHT (BITMAPHEIGHT BM)))
                                (SETQ |CTop| (COND
                                                ((IGREATERP |ClippingTop| HEIGHT)
                                                 (IDIFFERENCE HEIGHT BOTTOM))
                                                (T (IDIFFERENCE |ClippingTop| BOTTOM))))
                                (COND
                                   ((IGEQ |CTop| 0)
                                    (SETQ |CBottom| (COND
                                                       ((ILESSP |ClippingBottom| BOTTOM)
                                                        0)
                                                       (T (IDIFFERENCE |ClippingBottom| BOTTOM))))
                                    (|replace| (\\DISPLAYDATA |DDDestination|) |of| DD |with| BM)
                                    (|replace| (\\DISPLAYDATA |DDClippingTop|) |of| DD |with| |CTop|)
                                    (|replace| (\\DISPLAYDATA |DDClippingBottom|) |of| DD |with| |CBottom|)
                                    (\\DRAWELLIPSE.DISPLAY DISPLAYSTREAM CENTERX (IDIFFERENCE CENTERY BOTTOM)
                                           SEMIMINORRADIUS SEMIMAJORRADIUS ORIENTATION BRUSH DASHING)
                                    (SETQ BM (|GetNewFragment| BIGBMLIST))
                                    (SETQ HEIGHT BOTTOM))))
                        (|replace| (\\DISPLAYDATA |DDDestination|) |of| DD |with| BITMAP)
                        (|replace| (\\DISPLAYDATA |DDClippingTop|) |of| DD |with| |ClippingTop|)
                        (|replace| (\\DISPLAYDATA |DDClippingBottom|) |of| DD |with| |ClippingBottom|)
                        (MOVETO CENTERX CENTERY DISPLAYSTREAM)
                        (RETURN NIL))))
           (T (\\DRAWELLIPSE.DISPLAY DISPLAYSTREAM CENTERX CENTERY SEMIMINORRADIUS SEMIMAJORRADIUS ORIENTATION
                     BRUSH DASHING))))))
```

(\\DRAWCURVE.BIGBM

```
(LAMBDA (DISPLAYSTREAM KNOTS CLOSED BRUSH DASHING)
  (DECLARE (LOCALVARS . T))                                    ; Edited 29-Jan-91 17:48 by matsuda
  (PROG ((DD (|fetch| (STREAM IMAGEDATA) |of| DISPLAYSTREAM))
         BITMAP)
        (SETQ BITMAP (|fetch| (\\DISPLAYDATA |DDDestination|) |of| DD))
        (COND
           ((|type?| BIGBM BITMAP)
            (PROG (BIGBMLIST HEIGHT BOTTOM BM |ClippingTop| |ClippingBottom| |CTop| |CBottom| POINTS)
                  (|for| KNOT |in| KNOTS |do| (OR (|type?| POSITION KNOT)
                                                  (ERROR "bad knot" KNOT)))
                  (SETQ BIGBMLIST (|fetch| (BIGBM BIGBMLIST) |of| BITMAP))
                  (SETQ HEIGHT (BITMAPHEIGHT BITMAP))
                  (SETQ |ClippingTop| (|ffetch| (\\DISPLAYDATA |DDClippingTop|) |of| DD))
                  (SETQ |ClippingBottom| (|ffetch| (\\DISPLAYDATA |DDClippingBottom|) |of| DD))
                  (SETQ BM (|GetNewFragment| BIGBMLIST))
                  (|while| (AND BM (IGREATERP HEIGHT |ClippingBottom|))
                     |do| (SETQ BOTTOM (IDIFFERENCE HEIGHT (BITMAPHEIGHT BM)))
                          (SETQ |CTop| (COND
                                          ((IGREATERP |ClippingTop| HEIGHT)
                                           (IDIFFERENCE HEIGHT BOTTOM))
                                          (T (IDIFFERENCE |ClippingTop| BOTTOM))))
                          (COND
                             ((IGEQ |CTop| 0)
                              (SETQ |CBottom| (COND
                                                 ((ILESSP |ClippingBottom| BOTTOM)
                                                  0)
                                                 (T (IDIFFERENCE |ClippingBottom| BOTTOM))))
                              (|replace| (\\DISPLAYDATA |DDDestination|) |of| DD |with| BM)
                              (|replace| (\\DISPLAYDATA |DDClippingTop|) |of| DD |with| |CTop|)
                              (|replace| (\\DISPLAYDATA |DDClippingBottom|) |of| DD |with| |CBottom|)
                              (SETQ POINTS (|for| KNOT |in| KNOTS |collect| (|create| POSITION
                                                                                  XCOORD _ (CAR KNOT)
```

```
                                                                YCOORD _ (DIFFERENCE (CDR KNOT)
                                                                                     BOTTOM))))
                                    (\\DRAWCURVE.DISPLAY DISPLAYSTREAM POINTS CLOSED BRUSH DASHING)
                                    (SETQ BM (|GetNewFragment| BIGBMLIST))
                                    (SETQ HEIGHT BOTTOM))))
                            (|replace| (\\DISPLAYDATA |DDDestination|) |of| DD |with| BITMAP)
                            (|replace| (\\DISPLAYDATA |DDClippingTop|) |of| DD |with| |ClippingTop|)
                            (|replace| (\\DISPLAYDATA |DDClippingBottom|) |of| DD |with| |ClippingBottom|)
                            (RETURN DISPLAYSTREAM)))
                    (T (\\DRAWCURVE.DISPLAY DISPLAYSTREAM KNOTS CLOSED BRUSH DASHING))))))
```

(**DRAWLINE.BIGBM.DASH**

```
(LAMBDA (DISPLAYSTREAM X1 Y1 X2 Y2 BRUSH DASHING OPERATION)    ; Edited 13-Jun-2021 14:02 by rmk:
  (GLOBALRESOURCES \\BRUSHBBT (LET ((DD (|fetch| IMAGEDATA |of| DISPLAYSTREAM))
                                    BITMAP BIGBMLIST HEIGHT BOTTOM BM YY1 YY2 |ClippingTop| |ClippingBottom|
                                    |CTop| |CBottom|)
                                   (SETQ BITMAP (|ffetch| |DDDestination| |of| DD))
                                   (SETQ BIGBMLIST (|fetch| (BIGBM BIGBMLIST) |of| BITMAP))
                                   (SETQ HEIGHT (BITMAPHEIGHT BITMAP))
                                   (SETQ |ClippingTop| (|ffetch| |DDClippingTop| |of| DD))
                                   (SETQ |ClippingBottom| (|ffetch| |DDClippingBottom| |of| DD))
                                   (SETQ BM (|GetNewFragment| BIGBMLIST))
                                   (|while| (AND BM (IGREATERP HEIGHT |ClippingBottom|))
                                      |do| (SETQ BOTTOM (IDIFFERENCE HEIGHT (BITMAPHEIGHT BM)))
                                           (SETQ |CTop| (COND
                                                          ((IGREATERP |ClippingTop| HEIGHT)
                                                           (IDIFFERENCE HEIGHT BOTTOM))
                                                          (T (IDIFFERENCE |ClippingTop| BOTTOM))))
                                           (|if| (IGEQ |CTop| 0)
                                               |then| (SETQ |CBottom| (COND
                                                                        ((ILESSP |ClippingBottom| BOTTOM)
                                                                         0)
                                                                        (T (IDIFFERENCE |ClippingBottom| BOTTOM)
                                                                           )))
                                                      (|replace| |DDDestination| |of| DD |with| BM)
                                                      (|replace| |DDClippingTop| |of| DD |with| |CTop|)
                                                      (|replace| |DDClippingBottom| |of| DD |with| |CBottom|)
                                                      (\\LINEWITHBRUSH X1 (IDIFFERENCE Y1 BOTTOM)
                                                             X2
                                                             (IDIFFERENCE Y2 BOTTOM)
                                                             BRUSH
                                                             (\\GOOD.DASHLST DASHING BRUSH)
                                                             DISPLAYSTREAM \\BRUSHBBT OPERATION)
                                                      (SETQ BM (|GetNewFragment| BIGBMLIST))
                                                      (SETQ HEIGHT BOTTOM)))
                                   (|freplace| |DDDestination| |of| DD |with| BITMAP)
                                   (|freplace| |DDClippingTop| |of| DD |with| |ClippingTop|)
                                   (|freplace| |DDClippingBottom| |of| DD |with| |ClippingBottom|))))))
```

(**DRAWLINE.BIGBM.NODASH**

```
(LAMBDA (DISPLAYSTREAM X1 Y1 X2 Y2 WIDTH OPERATION COLOR)      ; Edited 13-Jun-2021 13:59 by rmk:
  (LET ((DD (|fetch| IMAGEDATA |of| DISPLAYSTREAM))
        BITMAP BIGBMLIST HEIGHT BOTTOM BM |CTop| |CBottom| |ClippingTop| |ClippingBottom| YY1 YY2)
       (SETQ BITMAP (|ffetch| |DDDestination| |of| DD))
       (SETQ BIGBMLIST (|fetch| (BIGBM BIGBMLIST) |of| BITMAP))
       (SETQ HEIGHT (BITMAPHEIGHT BITMAP))
       (SETQ BM (|GetNewFragment| BIGBMLIST))
       (SETQ |ClippingTop| (|ffetch| |DDClippingTop| |of| DD))
       (SETQ |ClippingBottom| (|ffetch| |DDClippingBottom| |of| DD))
       (SETQ YY1 (\\DSPTRANSFORMY (OR (FIXP Y1)
                                      (FIXR Y1))
                                  DD))
       (SETQ YY2 (\\DSPTRANSFORMY (OR (FIXP Y2)
                                      (FIXR Y2))
                                  DD))
       (|while| (AND BM (IGREATERP HEIGHT |ClippingBottom|))
          |do| (SETQ BOTTOM (IDIFFERENCE HEIGHT (BITMAPHEIGHT BM)))
               (SETQ |CTop| (COND
                              ((IGREATERP |ClippingTop| HEIGHT)
                               (IDIFFERENCE HEIGHT BOTTOM))
                              (T (IDIFFERENCE |ClippingTop| BOTTOM))))
               (COND
                  ((IGEQ |CTop| 0)
                   (SETQ |CBottom| (COND
                                     ((ILESSP |ClippingBottom| BOTTOM)
                                      0)
                                     (T (IDIFFERENCE |ClippingBottom| BOTTOM))))
                   (\\CLIPANDDRAWLINE (\\DSPTRANSFORMX (OR (FIXP X1)
                                                           (FIXR X1))
                                                       DD)
                                      (IDIFFERENCE YY1 BOTTOM)
                                      (\\DSPTRANSFORMX (OR (FIXP X2)
                                                           (FIXR X2))
                                                       DD)
                                      (IDIFFERENCE YY2 BOTTOM)
```

```
                           (COND
                              ((NULL WIDTH)
                               1)
                              ((OR (FIXP WIDTH)
                                   (FIXR WIDTH))))
                           (SELECTQ OPERATION
                                (NIL (|ffetch| DDOPERATION |of| DD))
                                ((REPLACE PAINT INVERT ERASE)
                                    OPERATION)
                                (\\ILLEGAL.ARG OPERATION))
                           BM
                           (|ffetch| |DDClippingLeft| |of| DD)
                           (SUB1 (|ffetch| |DDClippingRight| |of| DD))
                           |CBottom|
                           (SUB1 |CTop|)
                           DISPLAYSTREAM COLOR)))
                  (SETQ BM (|GetNewFragment| BIGBMLIST))
                  (SETQ HEIGHT BOTTOM)))))

)

(DEFINEQ
```

(**GENERIC.DSPCREATE.DESTINATION.BITMAP?.BIGBM**

```
  (LAMBDA (DESTINATION)                                     ; Edited  9-Jul-2022 09:24 by rmk
    (\\DTEST (COND
                ((|type?| BIGBM DESTINATION)
                 (CAR (|fetch| (BIGBM BIGBMLIST) OF DESTINATION)))
                (T DESTINATION))
             'BITMAP)))

)

(DECLARE\: DONTEVAL@LOAD DOCOPY

(MOVD '\\GENERIC.DSPCREATE.DESTINATION.BITMAP?.BIGBM '\\GENERIC.DSPCREATE.DESTINATION.BITMAP?)
)

(DEFINEQ
```

(**DSPDESTINATION**

```
  (LAMBDA (DESTINATION DISPLAYSTREAM)                       ; Edited 22-Sep-89 13:53 by takeshi
    (DECLARE (GLOBALVARS \\DISPLAYIMAGEOPS \\4DISPLAYIMAGEOPS \\8DISPLAYIMAGEOPS \\24DISPLAYIMAGEOPS))
    (PROG (DD)
          (SETQ DD (\\GETDISPLAYDATA DISPLAYSTREAM DISPLAYSTREAM))
          (RETURN (PROG1 (|ffetch| (\\DISPLAYDATA |DDDestination|) |of| DD)
                         (COND
                            (DESTINATION                    ; (SETQ DESTINATION (OR (\\DTEST DESTINATION 'BITMAP)
                                                            ; (\\DTEST DESTINATION 'BIGBM)))
                                   (COND
                                      ((|type?| BITMAP DESTINATION)
                                       (UNINTERRUPTABLY
                                            (|replace| (STREAM DEVICE) |of| DISPLAYSTREAM
                                               |with| (SELECTQ (|fetch| (BITMAP BITMAPBITSPERPIXEL) |of| DESTINATION)
                                                               (1 |DisplayFDEV|)
                                                               (4 \\4DISPLAYFDEV)
                                                               (8 \\8DISPLAYFDEV)
                                                               (24 \\24DISPLAYFDEV)
                                                               (SHOULDNT)))
                                            (|replace| (STREAM IMAGEOPS) |of| DISPLAYSTREAM
                                               |with| (SELECTQ (|fetch| (BITMAP BITMAPBITSPERPIXEL) |of| DESTINATION)
                                                               (1 \\DISPLAYIMAGEOPS)
                                                               (4 \\4DISPLAYIMAGEOPS)
                                                               (8 \\8DISPLAYIMAGEOPS)
                                                               (24 \\24DISPLAYIMAGEOPS)
                                                               (SHOULDNT)))
                                            (|freplace| (\\DISPLAYDATA |DDDestination|) |of| DD |with| DESTINATION)
                                            (|\\SFFixDestination| DD DISPLAYSTREAM)))
                                      ((|type?| BIGBM DESTINATION)
                                       (UNINTERRUPTABLY
                                            (|replace| (STREAM DEVICE) |of| DISPLAYSTREAM |with| \\8DISPLAYFDEV)
                                            ;; I'll add the bpp slot in BIGBM
                                            (|replace| (STREAM IMAGEOPS) |of| DISPLAYSTREAM |with| \\8DISPLAYIMAGEOPS)
                                            (|freplace| (\\DISPLAYDATA |DDDestination|) |of| DD |with| DESTINATION)
                                            (|\\SFFixDestination| DD DISPLAYSTREAM)))))))))))
```

(|**SFFixY**|

```
  (LAMBDA (DISPLAYDATA CSINFO)                              ; Edited  6-Jul-90 10:13 by matsuda
    ;; makes that part of the bitblt table of a display stream which deals with the Y information consistent.  This is called from \BLTCHAR whenever a
    ;; character is being printed and the charset/y-position caches are invalid        ; assumes DISPLAYDATA has already been type checked.
    (PROG ((PBT (|ffetch| DDPILOTBBT |of| DISPLAYDATA))
           (Y (\\DSPTRANSFORMY (|ffetch| DDYPOSITION |of| DISPLAYDATA)
                     DISPLAYDATA))
           TOP CHARTOP BM)
          (SETQ CHARTOP (IPLUS Y (|freplace| DDCHARSETASCENT |of| DISPLAYDATA |with| (|ffetch| CHARSETASCENT
```

```
                                                                                                |of| CSINFO))))
        (SETQ BM (|ffetch| |DDDestination| |of| DISPLAYDATA))
        (COND
           ((|type?| BIGBM BM)
            (SETQ TOP (IMAX (IMIN (|ffetch| |DDClippingTop| |of| DISPLAYDATA)
                                  CHARTOP)
                            0))
            (|freplace| PBTDEST |of| PBT |with| NIL)
            (|freplace| PBTSOURCE |of| PBT |with| (\\ADDBASE (|ffetch| BITMAPBASE |of| (SETQ BM (|ffetch| (CHARSETINFO
                                                                                                          CHARSETBITMAP)
                                                                                                  |of| CSINFO)))
                                                          (ITIMES (|ffetch| BITMAPRASTERWIDTH |of| BM)
                                                                  (|freplace| DDCHARHEIGHTDELTA |of| DISPLAYDATA
                                                                     |with| (IMIN (IMAX (IDIFFERENCE CHARTOP TOP)
                                                                                       0)
                                                                                  MAX.SMALL.INTEGER)))))
            (|freplace| PBTHEIGHT |of| PBT |with| (IMAX (IDIFFERENCE TOP (IMAX (IDIFFERENCE Y
                                                                                  (|freplace| DDCHARSETDESCENT
                                                                                     |of| DISPLAYDATA
                                                                                     |with| (|ffetch| CHARSETDESCENT
                                                                                                |of| CSINFO)))
                                                                           (|ffetch| |DDClippingBottom|
                                                                              |of| DISPLAYDATA)))
                                                     0)))
           (T (|freplace| PBTDEST |of| PBT |with| (\\ADDBASE (|fetch| BITMAPBASE |of| BM)
                                                             (ITIMES (|ffetch| BITMAPRASTERWIDTH |of| BM)
                                                                     (|\\SFInvert| BM
                                                                                   (SETQ TOP (IMAX (IMIN (|ffetch| |DDClippingTop|
                                                                                                            |of| DISPLAYDATA)
                                                                                                         CHARTOP)
                                                                                                   0))))))
              (|freplace| PBTSOURCE |of| PBT |with| (\\ADDBASE (|ffetch| BITMAPBASE |of| (SETQ BM (|ffetch| (CHARSETINFO
                                                                                                          CHARSETBITMAP
                                                                                                          )
                                                                                                  |of| CSINFO)))
                                                          (ITIMES (|ffetch| BITMAPRASTERWIDTH |of| BM)
                                                                  (|freplace| DDCHARHEIGHTDELTA |of| DISPLAYDATA
                                                                     |with| (IMIN (IMAX (IDIFFERENCE CHARTOP TOP)
                                                                                       0)
                                                                                  MAX.SMALL.INTEGER)))))
              (|freplace| PBTHEIGHT |of| PBT |with| (IMAX (IDIFFERENCE TOP
                                                                (IMAX (IDIFFERENCE Y (|freplace| DDCHARSETDESCENT
                                                                                        |of| DISPLAYDATA
                                                                                        |with| (|ffetch| CHARSETDESCENT
                                                                                                   |of| CSINFO)))
                                                                      (|ffetch| |DDClippingBottom| |of| DISPLAYDATA)))
                                                           0)))))))
```

(|\\SFFixDestination|

```
(LAMBDA (DISPLAYDATA DISPLAYSTREAM)                    ; Edited 6-Jul-90 13:55 by matsuda
```

;; fixes up those parts of the bitblt array which are dependent upon the destination

```
(PROG ((PBT (|ffetch| (\\DISPLAYDATA DDPILOTBBT) |of| DISPLAYDATA))
       (BM (|ffetch| (\\DISPLAYDATA |DDDestination|) |of| DISPLAYDATA)))
      (|replace| (PILOTBBT PBTDESTBPL) |of| PBT |with| (UNFOLD (COND
                                                                ((|type?| BITMAP BM)
                                                                 (|ffetch| (BITMAP BITMAPRASTERWIDTH) |of| BM))
                                                                (T (|ffetch| (BITMAP BITMAPRASTERWIDTH)
                                                                      |of| (CAR (|fetch| (BIGBM BIGBMLIST)
                                                                                   OF BM)))))
                                                           BITSPERWORD))
                                                    ; line width information will be updated by \SFFixFont
      (|\\SFFixClippingRegion| DISPLAYDATA)
      (\\INVALIDATEDISPLAYCACHE DISPLAYDATA)
      (|\\SFFixFont| DISPLAYSTREAM DISPLAYDATA)
      (RETURN))))
```

(|\\SFFixClippingRegion|

```
(LAMBDA (DISPLAYDATA)                                  ; Edited 6-Jul-90 13:55 by matsuda
```

;; compute the top, bottom, left and right edges of the clipping region in destination coordinates to save computation every BltChar and coordinate
;; transformation taking into account the size of the bit map as well as the clipping region.

```
(PROG ((CLIPREG (|ffetch| (\\DISPLAYDATA |DDClippingRegion|) |of| DISPLAYDATA))
       (BM (|ffetch| (\\DISPLAYDATA |DDDestination|) |of| DISPLAYDATA)))
      (|freplace| (\\DISPLAYDATA |DDClippingRight|) |of| DISPLAYDATA
         |with| (IMAX 0 (IMIN (\\DSPTRANSFORMX (IPLUS (|ffetch| (REGION LEFT) |of| CLIPREG)
                                                      (|ffetch| (REGION WIDTH) |of| CLIPREG))
                                                 DISPLAYDATA)
                              (BITMAPWIDTH BM))))
      (|freplace| (\\DISPLAYDATA |DDClippingLeft|) |of| DISPLAYDATA
         |with| (IMIN (IMAX (\\DSPTRANSFORMX (|ffetch| (REGION LEFT) |of| CLIPREG)
                                             DISPLAYDATA)
                            0)
                      MAX.SMALL.INTEGER))
      (|freplace| (\\DISPLAYDATA |DDClippingTop|) |of| DISPLAYDATA
```

```
            |with| (IMAX 0 (IMIN (\\DSPTRANSFORMY (IPLUS (|ffetch| (REGION BOTTOM) |of| CLIPREG)
                                                        (|ffetch| (REGION HEIGHT) |of| CLIPREG))
                                                DISPLAYDATA)
                                        (BITMAPHEIGHT BM))))
         (|freplace| (\\DISPLAYDATA |DDClippingBottom|) |of| DISPLAYDATA
            |with| (IMIN (IMAX (\\DSPTRANSFORMY (|ffetch| (REGION BOTTOM) |of| CLIPREG)
                                          DISPLAYDATA)
                               0)
                         MAX.SMALL.INTEGER)))))

)

(DEFINEQ

(\\SW2BM
  (LAMBDA (P PR Q QR)                                    ; Edited 8-Sep-89 16:14 by takeshi

          (* |Switches| |the| |areas| |of| P |and| Q |defined| |by| |the| |regions| PR |and| QR |respectively|)

    (PROG (PL PH PW PB QL QH QW QB)
          (COND
             (PR (SETQ PL (|fetch| (REGION LEFT) |of| PR))
                 (SETQ PB (|fetch| (REGION BOTTOM) |of| PR))
                 (SETQ PH (|fetch| (REGION HEIGHT) |of| PR))
                 (SETQ PW (|fetch| (REGION WIDTH) |of| PR)))
             (T (SETQ PL (SETQ PB 0))
                (COND
                   ((|type?| BITMAP P)
                    (SETQ PW (|ffetch| (BITMAP BITMAPWIDTH) |of| P))
                    (SETQ PH (|ffetch| (BITMAP BITMAPHEIGHT) |of| P)))
                   (T (SETQ PW (|fetch| (BIGBM BIGBMWIDTH) |of| P))
                      (SETQ PH (|fetch| (BIGBM BIGBMHEIGHT) |of| P))))))
          (COND
             (QR (SETQ QL (|fetch| (REGION LEFT) |of| QR))
                 (SETQ QB (|fetch| (REGION BOTTOM) |of| QR))
                 (SETQ QW (|fetch| (REGION WIDTH) |of| QR))
                 (SETQ QH (|fetch| (REGION HEIGHT) |of| QR)))
             (T (SETQ QL (SETQ QB 0))
                (COND
                   ((|type?| BITMAP Q)
                    (SETQ QW (|ffetch| (BITMAP BITMAPWIDTH) |of| Q))
                    (SETQ QH (|ffetch| (BITMAP BITMAPHEIGHT) |of| Q)))
                   (T (SETQ QW (|fetch| (BIGBM BIGBMWIDTH) |of| Q))
                      (SETQ QH (|fetch| (BIGBM BIGBMHEIGHT) |of| Q))))))
          (PROG ((CL (IMAX (IMINUS PL)
                           (IMINUS QL)
                           0))
                 (CB (IMAX (IMINUS PB)
                           (IMINUS QB)
                           0)))
                (PROG ((XP (IPLUS CL PL))
                       (YP (IPLUS CB PB))
                       (XQ (IPLUS CL QL))
                       (YQ (IPLUS CB QB))
                       CW CH)
                      (SETQ CW (IMIN (COND
                                        ((|type?| BITMAP P)
                                         (IDIFFERENCE (IMIN (|ffetch| (BITMAP BITMAPWIDTH) |of| P)
                                                            (IPLUS PL PW))
                                                      XP))
                                        (T (IDIFFERENCE (IMIN (|fetch| (BIGBM BIGBMWIDTH) |of| P)
                                                              (IPLUS PL PW))
                                                        XP)))
                                     (COND
                                        ((|type?| BITMAP Q)
                                         (IDIFFERENCE (IMIN (|ffetch| (BITMAP BITMAPWIDTH) |of| Q)
                                                            (IPLUS QL QW))
                                                      XQ))
                                        (T (IDIFFERENCE (IMIN (|fetch| (BIGBM BIGBMWIDTH) |of| Q)
                                                              (IPLUS QL QW))
                                                        XQ)))))
                      (SETQ CH (IMIN (IDIFFERENCE (IMIN (COND
                                                           ((|type?| BITMAP P)
                                                            (|fetch| (BITMAP BITMAPHEIGHT) |of| P))
                                                           (T (|fetch| (BIGBM BIGBMHEIGHT) |of| P)))
                                                        (IPLUS PB PH))
                                                  YP)
                                     (IDIFFERENCE (IMIN (COND
                                                           ((|type?| BITMAP Q)
                                                            (|fetch| (BITMAP BITMAPHEIGHT) |of| Q))
                                                           (T (|fetch| (BIGBM BIGBMHEIGHT) |of| Q)))
                                                        (IPLUS QB QH))
                                                  YQ)))
                      (UNINTERRUPTABLY
                          (BITBLT P XP YP Q XQ YQ CW CH 'INPUT 'INVERT)
                          (BITBLT Q XQ YQ P XP YP CW CH 'INPUT 'INVERT)
                          (BITBLT P XP YP Q XQ YQ CW CH 'INPUT 'INVERT)))))))
```

(BITMAPHEIGHT

```
(LAMBDA (BITMAP)                                        ; Edited 22-Sep-89 14:05 by takeshi
    ;; returns the height in pixels of a bitmap.
    (COND
       ((|type?| BITMAP BITMAP)
        (|ffetch| (BITMAP BITMAPHEIGHT) |of| BITMAP))
       ((|type?| WINDOW BITMAP)
        (WINDOWPROP BITMAP 'HEIGHT))
       ((|type?| BIGBM BITMAP)
        (|ffetch| (BIGBM BIGBMHEIGHT) |of| BITMAP))
       (T (\\ILLEGAL.ARG BITMAP)))))
```

(BITMAPWIDTH

```
(LAMBDA (BITMAP)                                        ; Edited 22-Sep-89 14:07 by takeshi
    ;; returns the width of a bitmap in pixels
    (COND
       ((|type?| BITMAP BITMAP)
        (|ffetch| (BITMAP BITMAPWIDTH) |of| BITMAP))
       ((|type?| WINDOW BITMAP)
        (WINDOWPROP BITMAP 'WIDTH))
       ((|type?| BIGBM BITMAP)
        (|ffetch| (BIGBM BIGBMWIDTH) |of| BITMAP))
       (T (\\ILLEGAL.ARG BITMAP)))))
```

(|\\SFFixFont|

```
(LAMBDA (DISPLAYSTREAM DISPLAYDATA)                     ; Edited  6-Jul-90 10:11 by matsuda
    ;; used to fix up those parts of the bitblt table which depend upon the FONT.  DISPLAYDATA is the IMAGEDATA for DISPLAYSTREAM, for
    ;; convenience.
    (PROG ((PILOTBBT (|ffetch| (\\DISPLAYDATA DDPILOTBBT) |of| DISPLAYDATA))
           (FONT (|ffetch| (\\DISPLAYDATA DDFONT) |of| DISPLAYDATA))
           (BITSPERPIXEL (BITSPERPIXEL (|ffetch| (\\DISPLAYDATA |DDDestination|) |of| DISPLAYDATA))))
          (|freplace| (\\DISPLAYDATA |DDSlowPrintingCase|) |of| DISPLAYDATA
             |with| (OR (NOT (EQ BITSPERPIXEL 1))
                        (NOT (EQ (|ffetch| (FONTDESCRIPTOR ROTATION) |of| FONT)
                                 0)))))
    (\\INVALIDATEDISPLAYCACHE DISPLAYDATA)
    (\\SFFIXLINELENGTH DISPLAYSTREAM)))
```

(BITSPERPIXEL

```
(LAMBDA (BITMAP)                                        ; Edited 29-Jun-90 10:15 by matsuda
    ;; returns the height in pixels of a bitmap.
    (COND
       ((|type?| BITMAP BITMAP)
        (|fetch| (BITMAP BITMAPBITSPERPIXEL) |of| BITMAP))
       ((|type?| BIGBM BITMAP)
        (|fetch| (BITMAP BITMAPBITSPERPIXEL) |of| (CAR (|fetch| (BIGBM BIGBMLIST) |of| BITMAP))))
       ((|type?| SCREEN BITMAP)
        (BITSPERPIXEL (|fetch| (SCREEN SCDESTINATION) |of| BITMAP)))
       ((|type?| WINDOW BITMAP)
        (BITSPERPIXEL (|fetch| (WINDOW SCREEN) |of| BITMAP)))
       ((ARRAYP BITMAP)                                 ; Consider array to be a colormap.
        (SELECTQ (ARRAYSIZE BITMAP)
             (256 8)
             (16 4)
             (LISPERROR "ILLEGAL ARG" BITMAP)))
       (T (LISPERROR "ILLEGAL ARG" BITMAP)))))

)

(DEFINEQ
```

(COLORIZEBITMAP

```
(LAMBDA (BITMAP 0COLOR 1COLOR BITSPERPIXEL)             ; Edited 26-Oct-2021 14:23 by larry
                                                        ; Edited 13-Jul-90 14:42 by matsuda
    ;; creates a copy of BITMAP that is in color form allowing BITSPERPIXEL per pixel.  0COLOR and 1COLOR are the color numbers that get
    ;; translated from 0 and 1 respectively.
    (PROG (COLORBITMAP)
          (SETQ COLORBITMAP (BITMAPCREATE (|fetch| (BITMAP BITMAPWIDTH) |of| BITMAP)
                                          (|fetch| (BITMAP BITMAPHEIGHT) |of| BITMAP)
                                          BITSPERPIXEL))
          (COND
             ((NOT (|type?| BIGBM COLORBITMAP))
              (\\BWTOCOLORBLT BITMAP 0 0 COLORBITMAP 0 0 (|fetch| (BITMAP BITMAPWIDTH) |of| BITMAP)
                     (|fetch| (BITMAP BITMAPHEIGHT) |of| BITMAP)
                     (COLORNUMBERP 0COLOR BITSPERPIXEL)
                     (COLORNUMBERP 1COLOR BITSPERPIXEL)
                     BITSPERPIXEL))
```

```
            (T (PROG (DESTBMLIST DESTBITMAP SOURCEBOOTTOM)
                     (SETQ DESTBMLIST (|fetch| (BIGBM BIGBMLIST) |of| COLORBITMAP))
                     (SETQ DESTBM (|GetNewFragment| DESTBMLIST))
                     (SETQ SOURCEBOOTTOM (|fetch| (BITMAP BITMAPHEIGHT) |of| BITMAP))
                     (|while| DESTBM |do| (SETQ DESTBMHEIGHT (|fetch| (BITMAP BITMAPHEIGHT) |of| DESTBM))
                                          (SETQ SOURCEBOOTTOM (- SOURCEBOOTTOM DESTBMHEIGHT))
                                          (\\BWTOCOLORBLT BITMAP 0 SOURCEBOOTTOM DESTBM 0 0 (|fetch| (BITMAP
                                                                                                    BITMAPWIDTH
                                                                                                          )
                                                                                                |of| BITMAP)
                                                DESTBMHEIGHT
                                                (COLORNUMBERP 0COLOR BITSPERPIXEL)
                                                (COLORNUMBERP 1COLOR BITSPERPIXEL)
                                                BITSPERPIXEL)
                                          (SETQ DESTBM (|GetNewFragment| DESTBMLIST))))))
         (RETURN COLORBITMAP))))
```

(\\BWTOCOLORBLT

```
(LAMBDA (SOURCEBWBM SLEFT SBOTTOM DESTCOLORBM DLEFT DBOTTOM WIDTH HEIGHT 0COLOR 1COLOR DESTNBITS)
                                                                        ; Edited 26-Oct-2021 14:36 by larry
                                                                        ; Edited 26-Oct-2021 14:32 by larry
                                                                        ; Edited 26-Oct-2021 14:26 by larry
                                                                        ; Edited  8-May-2021 22:31 by rmk:
```

;; blits from a black and white bitmap into a color bitmap which has DESTNBITS bits per pixel. DESTCOLORBM is a pointer to the color bitmap.

;; assumes all datatypes and bounds have been checked

```
(SELECTQ DESTNBITS
     (4 (PROG (MAP SRCBASE SRCHEIGHT SRCRW SRCWRD SRCOFFSET DESBASE DESHEIGHT DESRW DESWRD DESOFF NBITS
                   DESALIGNLEFT SCR)
              (SETQ MAP (|fetch| (ARRAYP BASE) |of| (\\MAP4 0COLOR 1COLOR)))
              (SETQ SRCBASE (|fetch| (BITMAP BITMAPBASE) |of| SOURCEBWBM))
              (SETQ SRCHEIGHT (|fetch| (BITMAP BITMAPHEIGHT) |of| SOURCEBWBM))
              (SETQ SRCRW (|fetch| (BITMAP BITMAPRASTERWIDTH) |of| SOURCEBWBM))
              (SETQ SRCWRD (FOLDLO SLEFT BITSPERWORD))
              (SETQ SRCOFFSET (MOD SLEFT BITSPERWORD))
              (SETQ DESBASE (|fetch| (BITMAP BITMAPBASE) |of| DESTCOLORBM))
              (SETQ DESHEIGHT (|fetch| (BITMAP BITMAPHEIGHT) |of| DESTCOLORBM))
              (SETQ DESRW (|fetch| (BITMAP BITMAPRASTERWIDTH) |of| DESTCOLORBM))
              (SETQ DESWRD (FOLDLO DLEFT 4))
              (SETQ DESOFF (MOD DLEFT 4))
              (SETQ NBITS 4)
```

;; DESTCOLORBM is used to allow one bit per pixel bitblt operations on the bitmap.

```
              (COND
                 ((NOT (EQ 0 DESOFF))
```

;; save the left bits of the destination bitmap so it can be word aligned.

```
                  (SETQ SCR (BITMAPCREATE 4 HEIGHT 4))
                  (BITBLT DESTCOLORBM (SETQ DESALIGNLEFT (LLSH DESWRD 2))
                          DBOTTOM SCR 0 0 DESOFF HEIGHT 'INPUT 'REPLACE)))
              (|for| LINECOUNTER |from| 1 |to| HEIGHT
                 |do| ;; linecounter goes from 1 to height because bitmaps are stored internally with top first so subtracting height is
                      ;; necessary to get offset of line and the 1 corrects for height difference.
                      (\\4BITLINEBLT (\\ADDBASE SRCBASE (IPLUS (ITIMES (IDIFFERENCE SRCHEIGHT (IPLUS
                                                                                                LINECOUNTER
                                                                                                  SBOTTOM))
                                                                                       SRCRW)
                                                               SRCWRD))
                                     SRCOFFSET
                                     (\\ADDBASE DESBASE (IPLUS (ITIMES (IDIFFERENCE DESHEIGHT (IPLUS LINECOUNTER
                                                                                                    DBOTTOM))
                                                                       DESRW)
                                                               DESWRD))
                                     WIDTH MAP 0COLOR 1COLOR))
              (COND
                 (DESALIGNLEFT
```

;; move the color bits to the right and restore the saved color bits.

```
                        (BITBLT DESTCOLORBM DESALIGNLEFT DBOTTOM DESTCOLORBM (IPLUS DESALIGNLEFT DESOFF)
                                DBOTTOM WIDTH HEIGHT 'INPUT 'REPLACE)
                        (BITBLT SCR 0 0 DESTCOLORBM DESALIGNLEFT DBOTTOM DESOFF HEIGHT 'INPUT 'REPLACE)))))
     (8 (SUBRCALL COLORIZE-BITMAP SOURCEBWBM SLEFT SBOTTOM DESTCOLORBM DLEFT DBOTTOM WIDTH HEIGHT 0COLOR
                  1COLOR DESTNBITS))
     (24 (PROG (SRCBASE SRCHEIGHT SRCRW DESBASE DESHEIGHT DESRW)
               (SETQ SRCBASE (|fetch| (BITMAP BITMAPBASE) |of| SOURCEBWBM))
               (SETQ SRCHEIGHT (|fetch| (BITMAP BITMAPHEIGHT) |of| SOURCEBWBM))
               (SETQ SRCRW (|fetch| (BITMAP BITMAPRASTERWIDTH) |of| SOURCEBWBM))
               (SETQ DESBASE (|fetch| (BITMAP BITMAPBASE) |of| DESTCOLORBM))
               (SETQ DESHEIGHT (|fetch| (BITMAP BITMAPHEIGHT) |of| DESTCOLORBM))
               (SETQ DESRW (|fetch| (BITMAP BITMAPRASTERWIDTH) |of| DESTCOLORBM))
               (|for| LINECOUNTER |from| 1 |to| HEIGHT |do|
```

;; linecounter goes from 1 to height because bitmaps are stored internally with top first so subtracting
;; height is necessary to get offset of line and the 1 corrects for height difference.

```
                                    (\\24BITLINEBLT (\\ADDBASE SRCBASE
```

```
                                                                    (ITIMES (IDIFFERENCE
                                                                             SRCHEIGHT
                                                                             (IPLUS LINECOUNTER
                                                                                    SBOTTOM))
                                                                            SRCRW))
                                                            SLEFT
                                                            (\\ADDBASE DESBASE
                                                                   (ITIMES (IDIFFERENCE DESHEIGHT
                                                                                        (IPLUS LINECOUNTER
                                                                                               DBOTTOM))
                                                                           DESRW))
                                                            DLEFT WIDTH 0COLOR 1COLOR))))
        (SHOULDNT))))
```

(**UNCOLORIZEBITMAP**

```
  (LAMBDA (BITMAP COLORMAP)                                 ; Edited 26-Oct-2021 14:51 by larry
                                                            ; Edited 26-Oct-2021 14:44 by larry
                                                            ; Edited 26-Oct-2021 14:44 by larry
                                                            ; Edited 13-Jul-90 16:54 by matsuda
    (PROG (BITSPERPIXEL MAXCOLOR MAXX MAXY BWBITMAP TABLE RGB R G B BIT BASE BWBASE RASTERWIDTH BWRASTERWIDTH
                 WORD)
          (SETQ MAXX (SUB1 (BITMAPWIDTH BITMAP)))
          (SETQ MAXY (SUB1 (BITMAPHEIGHT BITMAP)))
          (SETQ BITSPERPIXEL (BITSPERPIXEL BITMAP))
          (SETQ COLORMAP (OR COLORMAP (COLORMAP BITSPERPIXEL)))
          (SETQ MAXCOLOR (MAXIMUMCOLOR BITSPERPIXEL))
          (SETQ BWBITMAP (BITMAPCREATE (ADD1 MAXX)
                                       (ADD1 MAXY)
                                       1))
          (SETQ TABLE (\\ALLOCBLOCK (FOLDHI (ADD1 MAXCOLOR)
                                            2)))
          (|for| I |from| 0 |to| MAXCOLOR |do| (SETQ RGB (ELT COLORMAP I))
                                               (SETQ R (|fetch| (RGB RED) |of| RGB))
                                               (SETQ G (|fetch| (RGB GREEN) |of| RGB))
                                               (SETQ B (|fetch| (RGB BLUE) |of| RGB))
                                               (SETQ BIT (IDIFFERENCE 1 (IQUOTIENT (IPLUS R G B)
                                                                                   384)))
                                               (\\PUTBASE TABLE I BIT))
          (COND
             ((|type?| BITMAP BITMAP)
              (SETQ BASE (|fetch| (BITMAP BITMAPBASE) |of| BITMAP))
              (SETQ BWBASE (|fetch| (BITMAP BITMAPBASE) |of| BWBITMAP))
              (SETQ RASTERWIDTH (|fetch| (BITMAP BITMAPRASTERWIDTH) |of| BITMAP))
              (SETQ BWRASTERWIDTH (|fetch| (BITMAP BITMAPRASTERWIDTH) |of| BWBITMAP))))
          (SELECTQ BITSPERPIXEL
               (4 (|for| Y |from| 0 |to| MAXY |do| (SETQ WORD 0)
                                                   (|for| X |from| 0 |to| MAXX
                                                      |do| (SETQ WORD (LOGOR (LLSH WORD 1)
                                                                            (\\GETBASE TABLE (\\GETBASENYBBLE BASE X))
                                                                            ))
                                                           (COND
                                                              ((EQ (LOGAND X 15)
                                                                   15)
                                                               (\\PUTBASE BWBASE (FOLDLO X 16)
                                                                      WORD)
                                                               (SETQ WORD 0))))
                                                   (COND
                                                      ((NOT (EQ (LOGAND MAXX 15)
                                                                15))
                                                       (SETQ WORD (LLSH WORD (IDIFFERENCE 15 (LOGAND MAXX 15))))
                                                       (\\PUTBASE BWBASE (FOLDLO MAXX 16)
                                                              WORD)))
                                                   (COND
                                                      ((NOT (EQ Y MAXY))
                                                       (SETQ BASE (\\ADDBASE BASE RASTERWIDTH))
                                                       (SETQ BWBASE (\\ADDBASE BWBASE BWRASTERWIDTH))))))
               (8 (COND
                     ((NOT (|type?| BIGBM BITMAP))
                      (SUBRCALL UNCOLORIZE-BITMAP BITMAP BWBITMAP TABLE))
                     (T (PROG ((SRCBIGBMLIST (|fetch| (BIGBM BIGBMLIST) |of| BITMAP))
                               SRCBITMAP
                               (WIDTH (ADD1 MAXX))
                               HEIGHT
                               (DESTBOTTOM (ADD1 MAXY))
                               (TEMPBM (BITMAPCREATE (ADD1 MAXX)
                                                     (ADD1 MAXY)
                                                     1)))
                              (SETQ SRCBITMAP (|GetNewFragment| SRCBIGBMLIST))
                              (|while| SRCBITMAP |do| (SETQ DESTBOTTOM (IDIFFERENCE DESTBOTTOM
                                                                                    (SETQ HEIGHT (|fetch| (BITMAP
                                                                                                           BITMAPHEIGHT
                                                                                                           )
                                                                                                  |of| SRCBITMAP))))
                                                      (SUBRCALL UNCOLORIZE-BITMAP SRCBITMAP TEMPBM TABLE)
                                                      (BITBLT TEMPBM 0 (IDIFFERENCE (ADD1 MAXY)
                                                                                    HEIGHT)
```

```
                                                        BWBITMAP 0 DESTBOTTOM WIDTH HEIGHT 'INPUT 'REPLACE)
                                                (SETQ SRCBITMAP (|GetNewFragment| SRCBIGBMLIST)))))))
              NIL)
          (RETURN BWBITMAP))))

)

(DECLARE\: DONTEVAL@LOAD DOCOPY

(MOVD '\\ORG.BITBLT 'ORG.BITBLT)

(MOVD? 'BLTSHADE 'ORG.BLTSHADE)

(MOVD 'BLTSHADE.BIGBM 'BLTSHADE)

(MOVD 'BITBLT 'BKBITBLT)
)

(PUTPROPS BIGBITMAPS COPYRIGHT ("Venue" 1991 1993 1994))
```

FUNCTION INDEX

CONSTANT INDEX

MACRO INDEX

RECORD INDEX