File created: 16-Nov-87 17:15:41 **{ERINYES}<LISPUSERS>KOTO>FASTEDITBM.;3**

changes to: (FNS EXPANDBITMAP)
(VARS FASTEDITBMCOMS)

previous date: 4-Sep-87 15:58:23 {ERINYES}<LISPUSERS>KOTO>FASTEDITBM.;2

Read Table: OLD-INTERLISP-FILE

Package: INTERLISP

Format: XCCS


```
(RPAQQ FASTEDITBMCOMS
       ((DECLARE: DONTCOPY (MACROS UPDATE/BM/DISPLAY))
        (P (SETQ EDITBMMENU NIL))
        (FNS GRID)
        (FNS EDITBM EDITBMCLOSEFN TILEAREA EDITBMBUTTONFN EDITBMSCROLLFN \EDITBM/PUTUP/DISPLAY EDITBMRESHAPEFN
             EDITBMREPAINTFN.NEW EDITBMREPAINTFN RESETGRID.NEW)
        (FNS SCALEBM BLTPATTERN BLTPATTERN.REPLACEDISPLAY)
        (FNS EXPANDBITMAP EXPANDBM)))

(DECLARE: DONTCOPY

(DECLARE: EVAL@COMPILE

(PUTPROPS UPDATE/BM/DISPLAY MACRO ((BM W)
                                   (BITBLT BM (WINDOWPROP W (QUOTE DXOFFSET))
                                          (WINDOWPROP W (QUOTE DYOFFSET))
                                          W 0 (WINDOWPROP W (QUOTE BMDISPLAYBOTTOM))
                                          (WINDOWPROP W (QUOTE BMDISPLAYWIDTH))
                                          1000 NIL (QUOTE REPLACE))))
)
)

(SETQ EDITBMMENU NIL)

(DEFINEQ

(GRID
  [LAMBDA (GRIDSPEC WIDTH HEIGHT BORDER DS GRIDSHADE)          (* N.H.Briggs " 4-Sep-87 15:39")
                                                               ; draws a grid
    (PROG ((X0 (fetch (REGION LEFT) of GRIDSPEC))
           (Y0 (fetch (REGION BOTTOM) of GRIDSPEC))
           (SQWIDTH (fetch (REGION WIDTH) of GRIDSPEC))
           (SQHEIGHT (fetch (REGION HEIGHT) of GRIDSPEC))
           (GRIDSHADE (COND
                          ((TEXTUREP GRIDSHADE))
                          (T BLACKSHADE)))
           LINELENGTH TWICEBORDER MAXIMUMCOLOR TOTALHEIGHT GRIDBM TEMPBM)
          (SETQ TOTALHEIGHT (ITIMES HEIGHT SQHEIGHT))
          (COND
             ((OR (ZEROP BORDER)
                  (NULL BORDER))                               ; don't draw anything.
              (RETURN))
             [(NUMBERP BORDER)
              (SETQ TWICEBORDER (ITIMES BORDER 2))
              (PROGN
                     ;; draw vertical lines use BITBLT so that we don't have to correct for the width of the line since line drawing will put the
                     ;; coordinate in the middle.
                     (BLTSHADE GRIDSHADE DS X0 Y0 BORDER TOTALHEIGHT (QUOTE REPLACE))
                     (for X from (IDIFFERENCE (IPLUS X0 SQWIDTH)
                                              BORDER)
                        to (IDIFFERENCE (IPLUS X0 (ITIMES (SUB1 WIDTH)
                                                          SQWIDTH))
                                        BORDER)
                        by SQWIDTH do (BLTSHADE GRIDSHADE DS X Y0 TWICEBORDER TOTALHEIGHT (QUOTE REPLACE)))
                     (BLTSHADE GRIDSHADE DS (IDIFFERENCE (IPLUS X0 (ITIMES WIDTH SQWIDTH))
                                                         BORDER)
                            Y0 BORDER TOTALHEIGHT (QUOTE REPLACE)))
              (PROGN                                           ; draw horizontal lines
                     (BLTSHADE GRIDSHADE DS X0 Y0 (SETQ LINELENGTH (ITIMES WIDTH SQWIDTH))
                            BORDER
                            (QUOTE REPLACE))
                     (for Y from (IDIFFERENCE (IPLUS Y0 SQHEIGHT)
                                              BORDER)
                        to (IDIFFERENCE (IPLUS Y0 (ITIMES (SUB1 HEIGHT)
                                                          SQHEIGHT))
                                        BORDER)
                        by SQHEIGHT do (BLTSHADE GRIDSHADE DS X0 Y LINELENGTH TWICEBORDER (QUOTE REPLACE)))
                     (BLTSHADE GRIDSHADE DS X0 (IDIFFERENCE (IPLUS Y0 TOTALHEIGHT)
                                                            BORDER)
                            LINELENGTH BORDER (QUOTE REPLACE]
             [(EQ BORDER (QUOTE POINT))                        ; put a point in the lower left corner of each box
              (if (WINDOWP DS)
```

```
                then (SETQ TEMPBM (WINDOWPROP DS (QUOTE TEMPBM)))
                     (SETQ GRIDBM (WINDOWPROP DS (QUOTE GRIDBM)))
                     (if (NOT GRIDBM)
                         then (SETQ GRIDBM (BITMAPCREATE SQWIDTH SQHEIGHT))
                              (WINDOWPROP DS (QUOTE GRIDBM)
                                    GRIDBM))
                     (BLTSHADE WHITESHADE GRIDBM 0 0)          ; Clear temporary bitmap.
                     (BLTSHADE BLACKSHADE GRIDBM 0 0 1 1 (QUOTE REPLACE))
                                                               ; Put spot down.
                                                               ; Fill up temporary bitmap.
                     (BLTPATTERN GRIDBM 0 0 SQWIDTH SQHEIGHT DS X0 Y0 (ITIMES WIDTH SQWIDTH)
                            (ITIMES HEIGHT SQHEIGHT)
                            (QUOTE PAINT)
                            TEMPBM)
              else [SETQ MAXIMUMCOLOR (SUB1 (EXPT 2 (BITSPERPIXEL (DSPDESTINATION NIL DS]
                   ;; Crufty slow original code.
                   (for X from X0 to (IPLUS X0 (ITIMES WIDTH SQWIDTH)) by SQWIDTH
                      do (for Y from Y0 to (IPLUS Y0 TOTALHEIGHT) by SQHEIGHT
                            do (BITMAPBIT DS X Y MAXIMUMCOLOR]
          (T (\ILLEGAL.ARG BORDER])

)

(DEFINEQ

(EDITBM
  [LAMBDA (BMSPEC)                                             (* N.H.Briggs " 4-Sep-87 15:39")

;;; A simple bitmap editor.

    ;; The edit part of the display is from 0 to MAXGRIDWIDTH in width and from 0 to MAXGRIDHEIGHT in height.  The commands and display area
    ;; for the bitmap being edited are above the edit region.

    (DECLARE (GLOBALVARS SCREENWIDTH SCREENHEIGHT))
    (PROG (BMW BMWINTERIOR BMWWIDTH BMWHEIGHT WIDTH HEIGHT BM CR ORIGBM GRIDSQUARE BPP ORIGBPP ORIGWIDTH)
                                                               ; set ORIGBM to the input bitmap if any and BM to a copy of it for
                                                               ; editting.
          [COND
             ((OR (EQ BMSPEC CursorBitMap)
                  (AND (EQ BMSPEC (QUOTE CursorBitMap))
                       (SETQ BMSPEC CursorBitMap)))            ; editing cursor, save old value and make changes to the original.
              (SETQ ORIGBM (BITMAPCOPY CursorBitMap))
              (SETQ BM CursorBitMap))
             [(BITMAPP BMSPEC)
              (SETQ BM (BITMAPCOPY (SETQ ORIGBM BMSPEC]
             [(LITATOM BMSPEC)
              (COND
                 ([BITMAPP (SETQ ORIGBM (EVALV BMSPEC (QUOTE EDITBM]
                                                               ; use value.
                  (SETQ BM (BITMAPCOPY ORIGBM)))
                 (T (SETQ ORIGBM NIL)
                    (SETQ BM (\READBMDIMENSIONS]
             ((REGIONP BMSPEC)                                 ; if BMSPEC is a region, treat it as a region of the screen.
              [SETQ BM (BITMAPCREATE (fetch (REGION WIDTH) of BMSPEC)
                             (fetch (REGION HEIGHT) of BMSPEC)
                             (BITSPERPIXEL (SCREENBITMAP]      ; note that bm has initial bits in it.
              (SETQ ORIGBM BMSPEC)
              (BITBLT (SCREENBITMAP)
                      (fetch (REGION LEFT) of BMSPEC)
                      (fetch (REGION BOTTOM) of BMSPEC)
                      BM 0 0 NIL NIL (QUOTE INPUT)
                      (QUOTE REPLACE)))
             ((WINDOWP BMSPEC)
              (SETQ ORIGBM BMSPEC)
              ;; FS: Seems too big below, why not ClipRegion's Width & Height?  That's all that's used...
              (SETQ BM (BITMAPCREATE (WINDOWPROP BMSPEC (QUOTE WIDTH))
                             (WINDOWPROP BMSPEC (QUOTE HEIGHT))
                             (BITSPERPIXEL BMSPEC)))           ; open the window and bring it to the top.
              (TOTOPW BMSPEC)
              (SETQ CR (DSPCLIPPINGREGION NIL BMSPEC))
              (BITBLT BMSPEC (fetch (REGION LEFT) of CR)
                      (fetch (REGION BOTTOM) of CR)
                      BM 0 0 (fetch (REGION WIDTH) of CR)
                      (fetch (REGION HEIGHT) of CR)))
             (T                                                ; otherwise create a bitmap
                (SETQ BM (\READBMDIMENSIONS]
          (if (OR (EQ (BITMAPHEIGHT BM)
                      0)
                  (EQ (BITMAPWIDTH BM)
                      0))
              then (ERROR "Can't edit a bitmap with no bits in it." BMSPEC))
          (SETQ BPP (BITSPERPIXEL (SCREENBITMAP)))
          (SETQ ORIGBPP (fetch (BITMAP BITMAPBITSPERPIXEL) of BM))
          [COND
             ((NOT (EQ BPP ORIGBPP))
              ;; save the actual number of bits per pixel and set it to BPP in the bitmap being edited so that it can be BITBLT ed on the screen.
```

```
            (SETQ ORIGWIDTH (fetch (BITMAP BITMAPWIDTH) of BM))
            (replace (BITMAP BITMAPBITSPERPIXEL) of BM with BPP)
            (SETQ WIDTH (IQUOTIENT (ITIMES ORIGBPP ORIGWIDTH)
                                   BPP))
            (replace (BITMAP BITMAPWIDTH) of BM with WIDTH))
           (T (SETQ WIDTH (fetch (BITMAP BITMAPWIDTH) of BM]
      (SETQ HEIGHT (fetch (BITMAP BITMAPHEIGHT) of BM))
```

;; Calculate a default window size. Start by calculating the grid size from the bitmap size.

```
      (SETQ GRIDSQUARE (IMAX (IMIN (IQUOTIENT (IDIFFERENCE (IQUOTIENT (ITIMES SCREENWIDTH 2)
                                                                      3)
                                                           GRIDTHICKNESS)
                                              WIDTH)
                                   (IQUOTIENT (IDIFFERENCE (IQUOTIENT (ITIMES SCREENHEIGHT 2)
                                                                      3)
                                                           (ITIMES GRIDTHICKNESS 2))
                                              (ADD1 HEIGHT))
                                   NORMALGRIDSQUARE)
                             MINGRIDSQUARE))
      (SETQ BMWWIDTH (IMIN (IPLUS (ITIMES GRIDSQUARE WIDTH)
                                  GRIDTHICKNESS)
                           (IQUOTIENT (ITIMES SCREENWIDTH 2)
                                  3)))
      (SETQ BMWHEIGHT (IMIN (IPLUS (ITIMES HEIGHT (ADD1 GRIDSQUARE))
                                   (ITIMES GRIDTHICKNESS 2)
                                   1)
                            (IQUOTIENT (ITIMES SCREENHEIGHT 2)
                                   3)))
      (SETQ BMW (CREATEW (GETBOXREGION (WIDTHIFWINDOW BMWWIDTH)
                                       (HEIGHTIFWINDOW BMWHEIGHT T)
                                       NIL NIL NIL "Indicate the position for the Bitmap Edit window.")
                         "Bitmap Editor"))
      (WINDOWPROP BMW (QUOTE BM)
             BM)
      (WINDOWPROP BMW (QUOTE SCROLLFN)
             (FUNCTION EDITBMSCROLLFN))
      (WINDOWPROP BMW (QUOTE RESHAPEFN)
             (FUNCTION EDITBMRESHAPEFN))
      (WINDOWPROP BMW (QUOTE REPAINTFN)
             (FUNCTION EDITBMREPAINTFN))
      (WINDOWPROP BMW (QUOTE BUTTONEVENTFN)
             (FUNCTION EDITBMBUTTONFN))
      (WINDOWPROP BMW (QUOTE CLOSEFN)
             (FUNCTION EDITBMCLOSEFN))
      (WINDOWPROP BMW (QUOTE XOFFSET)
             0)
      (WINDOWPROP BMW (QUOTE YOFFSET)
             0)
      (WINDOWPROP BMW (QUOTE DXOFFSET)
             0)
      (WINDOWPROP BMW (QUOTE DYOFFSET)
             0)
      (WINDOWPROP BMW (QUOTE ORIGINALBITMAP)
             ORIGBM)
      (WINDOWPROP BMW (QUOTE FINISHEDFLG)
             NIL)
      (WINDOWPROP BMW (QUOTE COLOR)
             (SUB1 (EXPT 2 BPP)))
      (WINDOWPROP BMW (QUOTE GRIDON)
             T)                                          ; call reshapefn to initialize the display and values
      (EDITBMRESHAPEFN BMW NIL NIL NIL (NOT ORIGBM))     ; start a mouse process in case this process is the mouse
                                                         ; process.
      (SPAWN.MOUSE)
      (while (NOT (WINDOWPROP BMW (QUOTE FINISHEDFLG))) do (DISMISS 500))
                                                         ; remove the closefn before closing the window.
      (WINDOWPROP BMW (QUOTE CLOSEFN)
             NIL)
      (CLOSEW BMW)
      (COND
         ((NOT (EQ ORIGBPP BPP))
          (replace (BITMAP BITMAPBITSPERPIXEL) of BM with ORIGBPP)
          (replace (BITMAP BITMAPWIDTH) of BM with ORIGWIDTH)))
      (RETURN (COND
                 ((EQ T (WINDOWPROP BMW (QUOTE FINISHEDFLG)))
                                                         ; editor exited via ok, stuff contents into original bitmap.
                  (COND
                     ((EQ BMSPEC CursorBitMap)           ; editting happened in original, leave it alone.
                      CursorBitMap)
                     ((REGIONP ORIGBM)                   ; put it back into the screen.
                      (BITBLT BM 0 0 (SCREENBITMAP)
                             (fetch (REGION LEFT) of ORIGBM)
                             (fetch (REGION BOTTOM) of ORIGBM)
                             (fetch (REGION WIDTH) of ORIGBM)
                             (fetch (REGION HEIGHT) of ORIGBM)
                             (QUOTE INPUT)
                             (QUOTE REPLACE))
```

```
                    BM)
                   ((WINDOWP ORIGBM)                          ; put it back into the window
                    (BITBLT BM 0 0 ORIGBM (fetch (REGION LEFT) of CR)
                           (fetch (REGION BOTTOM) of CR)
                           (fetch (REGION WIDTH) of CR)
                           (fetch (REGION HEIGHT) of CR)
                           (QUOTE INPUT)
                           (QUOTE REPLACE))
                    BM)
                   (ORIGBM (BITBLT BM 0 0 ORIGBM 0 0 WIDTH HEIGHT)
                          [COND
                             ((AND BMSPEC (LITATOM BMSPEC))
                                                              ; if spec was an atom without a bm value, set it.  in the
                                                              ; environment above EDITBM.
                              (MARKASCHANGED BMSPEC (QUOTE VARS))
                              (STKEVAL (QUOTE EDITBM)
                                     (LIST (QUOTE SETQQ)
                                           BMSPEC BM]
                          ORIGBM)
                    (T BM)))
               (T                                             ; error exit, if cursor return it to original value.
                  (COND
                     ((EQ BMSPEC CursorBitMap)
                      (BITBLT ORIGBM NIL NIL CursorBitMap)))
                  (ERROR!])
```

(EDITBMCLOSEFN

```
[LAMBDA (BMW)                                                 ; Edited 27-Aug-87 21:26 by FS
```

;; the close function for a bitmap edit window. For now do what a STOP would have done.

;; FS: Assuming this window won't be reused, flush the temporary bm.

```
(WINDOWPROP BMW (QUOTE TEMPBM)
       NIL)
(WINDOWPROP BMW (QUOTE GRIDBM)
       NIL)
(WINDOWPROP BMW (QUOTE FINISHEDFLG)
       (QUOTE KILL])
```

(TILEAREA

```
[LAMBDA (LFT BTM WDTH HGHT SRCBM WIN)                         ; Edited 27-Aug-87 21:20 by FS
```

;; lays tiles out in an area of a window. This function only provided for backwards compatibility.

```
(BLTPATTERN.REPLACEDISPLAY SRCBM 0 0 (BITMAPWIDTH SRCBM)
       (BITMAPHEIGHT SRCBM)
       WIN LFT BTM WDTH HGHT])
```

(EDITBMBUTTONFN

```
[LAMBDA (W)                                                   (* N.H.Briggs " 4-Sep-87 15:30")
```

;; inner function of bitmap editor.

```
(DECLARE (GLOBALVARS \CURRENTCURSOR))
(PROG (GRIDX0 GRIDY0 BITMAPWIDTH BITMAPHEIGHT NEWGRIDSIZE PAINTW ORIGBM GRIDSPEC GRIDINTERIOR BM BITSWIDE
            BITSHIGH WREGION XOFFSET YOFFSET DXOFFSET DYOFFSET DISPLAYREGION EXTENT BITSPERPIXEL CURSORBM)
      (SETQ GRIDSPEC (WINDOWPROP W (QUOTE GRIDSPEC)))
      (SETQ GRIDINTERIOR (WINDOWPROP W (QUOTE GRIDINTERIOR)))
      (SETQ BM (WINDOWPROP W (QUOTE BM)))
      (SETQ BITSWIDE (WINDOWPROP W (QUOTE BITSWIDE)))
      (SETQ BITSHIGH (WINDOWPROP W (QUOTE BITSHIGH)))
      (SETQ WREGION (WINDOWPROP W (QUOTE REGION)))
      (SETQ XOFFSET (WINDOWPROP W (QUOTE XOFFSET)))
      (SETQ YOFFSET (WINDOWPROP W (QUOTE YOFFSET)))
      (SETQ DXOFFSET (WINDOWPROP W (QUOTE DXOFFSET)))
      (SETQ DYOFFSET (WINDOWPROP W (QUOTE DYOFFSET)))
      (SETQ DISPLAYREGION (WINDOWPROP W (QUOTE DISPLAYREGION)))
      (SETQ EXTENT (WINDOWPROP W (QUOTE EXTENT)))
      (SETQ GRIDX0 (fetch (REGION LEFT) of GRIDSPEC))
      (SETQ GRIDY0 (fetch (REGION BOTTOM) of GRIDSPEC))
      (SETQ BITMAPWIDTH (fetch (BITMAP BITMAPWIDTH) of BM))
      (SETQ BITMAPHEIGHT (fetch (BITMAP BITMAPHEIGHT) of BM))
      (SETQ BITSPERPIXEL (fetch (BITMAP BITMAPBITSPERPIXEL) of BM))
      (SETQ COLOR (WINDOWPROP W (QUOTE COLOR)))
```

;; mark the region of the bitmap that is being editted.

```
      (COND
         ((INSIDE? GRIDINTERIOR (LASTMOUSEX W)
                (LASTMOUSEY W))
```

;; if cursor is inside, shade it.

```
          (\SHADEBITS BM GRIDSPEC GRIDINTERIOR W BITSWIDE BITSHIGH COLOR))
         ((INSIDE? DISPLAYREGION (LASTMOUSEX W)
                (LASTMOUSEY W))
```

;; Run the menu foe re-windowing into the whole bitmap

```
          (SELECTQ [MENU (COND
```

```
                    ((type? MENU EDITBMWINDOWMENU)
                     EDITBMWINDOWMENU)
                    ((SETQ EDITBMWINDOWMENU (create MENU
                                                   ITEMS _ (QUOTE ((Move (QUOTE Move)
                                                                         "Selects a different part
                                                                         of the bitmap to edit.")))
                                                   CENTERFLG _ T]
(Move                                                ; move the editing window's location on the bitmap.
      (PROG (POS)
            [SETQ POS (GETBOXPOSITION BITSWIDE BITSHIGH [IPLUS 4 (fetch (REGION LEFT)
                                                                   of WREGION)
                                                              (DIFFERENCE
                                                               XOFFSET
                                                               (WINDOWPROP W (QUOTE DXOFFSET
                                                                                    ]
                                                        (IPLUS (WINDOWPROP W (QUOTE BMDISPLAYBOTTOM))
                                                               (DIFFERENCE YOFFSET (WINDOWPROP W (QUOTE DYOFFSET)))
                                                               4
                                                               (fetch (REGION BOTTOM) of WREGION]
            [WINDOWPROP W (QUOTE XOFFSET)
                  (SETQ XOFFSET (IMIN (IDIFFERENCE BITMAPWIDTH BITSWIDE)
                                      (IMAX [IPLUS (WINDOWPROP W (QUOTE DXOFFSET))
                                                   (DIFFERENCE (fetch (POSITION XCOORD)
                                                                  of POS)
                                                          (IPLUS 4 (fetch (REGION LEFT)
                                                                      of WREGION]
                                            0]
            [WINDOWPROP W (QUOTE YOFFSET)
                  (SETQ YOFFSET
                   (IMAX 0 (IMIN (DIFFERENCE BITMAPHEIGHT BITSHIGH)
                                 (DIFFERENCE (IPLUS (WINDOWPROP W (QUOTE DYOFFSET))
                                                    (DIFFERENCE (fetch (POSITION YCOORD)
                                                                   of POS)
                                                           (IPLUS (fetch (REGION BOTTOM)
                                                                     of WREGION)
                                                                  4)))
                                             (WINDOWPROP W (QUOTE BMDISPLAYBOTTOM]
            (replace (REGION LEFT) of EXTENT with (IMINUS (QUOTIENT (TIMES XOFFSET
                                                                           (fetch (REGION WIDTH)
                                                                              of EXTENT))
                                                                    BITMAPWIDTH)))
            (replace (REGION BOTTOM) of EXTENT with (IMINUS (QUOTIENT (TIMES YOFFSET
                                                                              (fetch (REGION
                                                                                        HEIGHT
                                                                                        )
                                                                                 of EXTENT))
                                                                      BITMAPHEIGHT)))
            [COND
               ([OR (ILESSP XOFFSET DXOFFSET)
                    (ILESSP YOFFSET DYOFFSET)
                    [IGREATERP (IPLUS XOFFSET BITSWIDE)
                           (IPLUS DXOFFSET (WINDOWPROP W (QUOTE BMDISPLAYWIDTH]
                    (IGREATERP (IPLUS YOFFSET BITSHIGH)
                           (IPLUS DYOFFSET (WINDOWPROP W (QUOTE BMDISPLAYHEIGHT]
                ;; Adjust the display region left lower corner so the selected region is near the center.
                [WINDOWPROP W (QUOTE DXOFFSET)
                       (SETQ DXOFFSET (IMAX 0 (IMIN (DIFFERENCE (fetch (BITMAP BITMAPWIDTH)
                                                                   of BM)
                                                              (WINDOWPROP W (QUOTE
                                                                               BMDISPLAYWIDTH
                                                                               )))
                                                    (DIFFERENCE (IPLUS XOFFSET
                                                                       (LRSH BITSWIDE 1))
                                                           (LRSH (WINDOWPROP W (QUOTE
                                                                                  BMDISPLAYWIDTH
                                                                                  ))
                                                                 1]
                (WINDOWPROP W (QUOTE DYOFFSET)
                       (SETQ DYOFFSET (IMAX 0 (IMIN (DIFFERENCE (fetch (BITMAP BITMAPHEIGHT)
                                                                   of BM)
                                                              (WINDOWPROP W (QUOTE
                                                                               BMDISPLAYHEIGHT
                                                                               )))
                                                    (DIFFERENCE (IPLUS YOFFSET
                                                                       (LRSH BITSHIGH 1))
                                                           (LRSH (WINDOWPROP W (QUOTE
                                                                                  BMDISPLAYHEIGHT
                                                                                  ))
                                                                 1]
                                                 (* DSPFILL GRIDINTERIOR WHITESHADE
                                                 (QUOTE REPLACE) W)
            (UPDATE/BM/DISPLAY BM W)
        ;; FS: More useless code: (COND ((WINDOWPROP W 'GRIDON) (GRID GRIDSPEC BITSWIDE BITSHIGH 'POINT
        ;; W)))
            (RESETGRID.NEW BM GRIDSPEC BITSWIDE BITSHIGH 0 0 W T)))
```

```
                NIL))
            ((LASTMOUSESTATE LEFT)
             (UPDATE/BM/DISPLAY/SELECTED/REGION W)
             (SETQ CURSORBM (BITMAPCREATE 16 16 (BITSPERPIXEL BM)))
             (BITBLT BM NIL NIL CURSORBM)
             [RESETFORM [CURSOR (CURSORCREATE CURSORBM (fetch (CURSOR CURSORHOTSPOTX) of (CURSOR))
                                      (fetch (CURSOR CURSORHOTSPOTY) of (CURSOR]
                    (until (MOUSESTATE (NOT LEFT]
             (UPDATE/BM/DISPLAY/SELECTED/REGION W))
            (T ;; the region being editted is inverted while the menu is active.  Each command must make sure that it is recomplemented.
               (UPDATE/BM/DISPLAY/SELECTED/REGION W)
               (SELECTQ [MENU (COND
                                ((type? MENU EDITBMMENU)
                                 EDITBMMENU)
                                (T (SETQ EDITBMMENU (create MENU
                                                            ITEMS _ [APPEND (COND
                                                                              [(COLORDISPLAYP)
                                                                               (QUOTE ((Color (QUOTE Color)
                                                                                              "Choose color to
                                                                                              set bits with"]
                                                                              (T NIL))
                                                                            (QUOTE ((Paint (QUOTE Paint)
                                                                                           "Calls the window
                                                                                           PAINT command on the
                                                                                           bitmap.")
                                                                                    (ShowAsTile (QUOTE
                                                                                                  ShowAsTile
                                                                                                  )
                                                                                           "tiles the upper part
                                                                                           of the edit window
                                                                                           with the bitmap.")
                                                                                    (Grid% On/Off (QUOTE
                                                                                                    GridOnOff
                                                                                                    )
                                                                                           "Grid On/Off Switch")
                                                                                    (GridSize_ (QUOTE GridSize_)
                                                                                           "Allows setting of
                                                                                           the size of a bit in
                                                                                           the edit area.")
                                                                                    (Reset (QUOTE Reset)
                                                                                           "Sets the bitmap back
                                                                                           to the state at the
                                                                                           start of this edit
                                                                                           session.")
                                                                                    (Clear (QUOTE Clear)
                                                                                           "Sets the entire
                                                                                           bitmap to 0")
                                                                                    (Cursor_ (QUOTE Cursor_)
                                                                                           "Puts the bitmap into
                                                                                           the cursor and exits
                                                                                           the editor.")
                                                                                    (OK (QUOTE OK)
                                                                                        "Leaves the edit
                                                                                        session.")
                                                                                    (Abort (QUOTE Abort)
                                                                                           "Restores the bitmap
                                                                                           to its original
                                                                                           values and leaves the
                                                                                           editor."]
                                                            CENTERFLG _ T]
                                (OK (WINDOWPROP W (QUOTE FINISHEDFLG)
                                          T))
                                (Abort (WINDOWPROP W (QUOTE FINISHEDFLG)
                                               (QUOTE KILL)))
                                (Reset ;; allow the user to choose between everything or just visible part.  This also give the user a chance to change their
                                       ;; mind.
                                       (COND
                                          ((SELECTQ (\EDITBMHOWMUCH BM BITSWIDE BITSHIGH "RESET how much?")
                                                (VISIBLE [COND
                                                            [(SETQ ORIGBM (WINDOWPROP W (QUOTE ORIGINALBITMAP)))
                                                             (COND
                                                                ((REGIONP ORIGBM)
                                                                 (BITBLT (SCREENBITMAP)
                                                                         (IPLUS XOFFSET (fetch (REGION LEFT) of ORIGBM))
                                                                         (IPLUS YOFFSET (fetch (REGION BOTTOM) of ORIGBM))
                                                                        BM XOFFSET YOFFSET BITSWIDE BITSHIGH (QUOTE INPUT)
                                                                         (QUOTE REPLACE)))
                                                                (T (BITBLT ORIGBM XOFFSET YOFFSET BM XOFFSET YOFFSET
                                                                           BITSWIDE BITSHIGH]
                                                            (T (BLTSHADE WHITESHADE BM XOFFSET YOFFSET BITSWIDE BITSHIGH
                                                                         (QUOTE REPLACE]
                                                      T)
                                                (WHOLE [COND
                                                          [(SETQ ORIGBM (WINDOWPROP W (QUOTE ORIGINALBITMAP)))
                                                           (COND
```

```
                                    ((REGIONP ORIGBM)
                                     (BITBLT (SCREENBITMAP)
                                            (fetch (REGION LEFT) of ORIGBM)
                                            (fetch (REGION BOTTOM) of ORIGBM)
                                            BM))
                                    (T (BITBLT ORIGBM NIL NIL BM]
                                  (T (BLTSHADE WHITESHADE BM NIL NIL NIL NIL (QUOTE REPLACE]
                                T)
                           (PROGN (UPDATE/BM/DISPLAY/SELECTED/REGION W)
                                  NIL))
                     (\EDITBM/PUTUP/DISPLAY W BM GRIDSPEC GRIDINTERIOR BITSWIDE BITSHIGH))))
(Clear ;; allow the user to choose between everything or just visible part.  This also give the user a chance to change their
       ;; mind.
       (COND
          ((SELECTQ (\EDITBMHOWMUCH BM BITSWIDE BITSHIGH "CLEAR how much?")
                (VISIBLE (BLTSHADE WHITESHADE BM XOFFSET YOFFSET BITSWIDE BITSHIGH
                                   (QUOTE REPLACE))
                         T)
                (WHOLE (\CLEARBM BM)
                       T)
                (PROGN (UPDATE/BM/DISPLAY/SELECTED/REGION W)
                       NIL))
           (DSPFILL GRIDINTERIOR WHITESHADE (QUOTE REPLACE)
              W)
           (COND
              ((WINDOWPROP W (QUOTE GRIDON))
               (GRID GRIDSPEC BITSWIDE BITSHIGH (QUOTE POINT)
                     W)))
           (UPDATE/BM/DISPLAY BM W))))
(GridOnOff (COND
              ((NOT (WINDOWPROP W (QUOTE GRIDON)))
                                                  ; Turn Grid On
               (WINDOWPROP W (QUOTE GRIDON)
                T)
               (GRID GRIDSPEC BITSWIDE BITSHIGH (QUOTE POINT)
                     W)
               ;; FS:  The update here was unnecessary.  (UPDATE/BM/DISPLAY BM W)
               NIL)
              (T                                  ; Turn off grid
                 (WINDOWPROP W (QUOTE GRIDON)
                        NIL)                      (* DSPFILL (create REGION LEFT _ 0 BOTTOM _ 0 WIDTH _
                                                  (ADD1 (fetch (REGION WIDTH) of GRIDINTERIOR)) HEIGHT _
                                                  (ADD1 (fetch (REGION HEIGHT) of GRIDINTERIOR)))
                                                  WHITESHADE (QUOTE REPLACE) W)
                 (RESETGRID.NEW BM GRIDSPEC BITSWIDE BITSHIGH 0 0 W T)
                 ;; FS:  The update here was unnecessary.  (UPDATE/BM/DISPLAY BM W)
                 NIL)))
(GridSize_                                        ; sets the grid square size and calls the reshapefn.
           (COND
              ([SETQ NEWGRIDSIZE
                (NUMBERP (MENU (COND
                                  ((TYPENAMEP GRIDSIZEMENU (QUOTE MENU))
                                   GRIDSIZEMENU)
                                  (T (SETQ GRIDSIZEMENU
                                       (create MENU
                                               ITEMS _
                                               (QUOTE (3 4 5 6 7 8 12 16 20 24 28 32))
                                               MENUROWS _ 4]
               (WINDOWPROP W (QUOTE GRIDSQUARE)
                      NEWGRIDSIZE)
               (EDITBMRESHAPEFN W))))
(ShowAsTile                                       ; tiles the upper part of the window with the bitmap so the user
                                                  ; can see what it would be as a shade.
            (UPDATE/SHADE/DISPLAY BM W))
(Paint                                            ; call the window paint command on the contents of the bitmap.
       [SETQ PAINTW (CREATEW (create REGION
                                     LEFT _ (IQUOTIENT (DIFFERENCE SCREENWIDTH BITMAPWIDTH)
                                                       2)
                                     BOTTOM _ (IQUOTIENT (DIFFERENCE SCREENHEIGHT
                                                                     BITMAPHEIGHT)
                                                         2)
                                     WIDTH _ (WIDTHIFWINDOW BITMAPWIDTH)
                                     HEIGHT _ (HEIGHTIFWINDOW BITMAPHEIGHT NIL]
       (OPENW PAINTW)
       (BITBLT BM 0 0 PAINTW)
       (PAINTW PAINTW)
       (COND
          ((MENU (create MENU
                         ITEMS _ (QUOTE ((YES T "Will put the newly painted bits back in the
                                                 bitmap being editted.")
                                         (NO NIL "Will discard the painted bits, not changing
                                             the bitmap being editted.")))
                         TITLE _ "Put change into bitmap?"
                         CENTERFLG _ T))
```

```
                              (BITBLT PAINTW 0 0 BM)
                              (CLOSEW PAINTW)                    ; set PAINTW so that space can be reclaimed
                              (SETQ PAINTW)
                              (\EDITBM/PUTUP/DISPLAY W BM GRIDSPEC GRIDINTERIOR BITSWIDE BITSHIGH))))
                       (Cursor_                                  ; Stuffs lower left part of image into the cursor and sets the
                                                                 ; hotspot.
                                (READHOTSPOT BM GRIDSPEC GRIDINTERIOR W)
                                (WINDOWPROP W (QUOTE FINISHEDFLG)
                                      T))
                       (Color (WINDOWPROP W (QUOTE COLOR)
                                    (OR (MENU (COLORMENU BITSPERPIXEL))
                                        COLOR)))
                       (UPDATE/BM/DISPLAY/SELECTED/REGION W])
```

(EDITBMSCROLLFN

```
  [LAMBDA (W DX DY)                                              ; Edited 31-Aug-87 13:29 by FS
                                                                 ; Do scrolling for the bitmap editor.
    (PROG (GRIDSPEC REG WHEIGHT WWIDTH (DXGRID 0)
                (DYGRID 0)
                EXTENT EXTENTWIDTH EXTENTHEIGHT GILEFT GIBOTTOM GIHEIGHT GWIDTH GHEIGHT GRIDINTERIOR EBMXLIMIT
                EBMYLIMIT EBMXOFFSET EBMYOFFSET BM BITMAPWIDTH BITMAPHEIGHT BITSWIDE BITSHIGH DXOFFSET DYOFFSET
                )
          (SETQ GRIDSPEC (WINDOWPROP W (QUOTE GRIDSPEC)))
          (SETQ REG (WINDOWPROP W (QUOTE REGION)))
          (SETQ WHEIGHT (WINDOWPROP W (QUOTE HEIGHT)))
          (SETQ WWIDTH (WINDOWPROP W (QUOTE WIDTH)))
          (SETQ GRIDINTERIOR (WINDOWPROP W (QUOTE GRIDINTERIOR)))
          (SETQ EBMXOFFSET (WINDOWPROP W (QUOTE XOFFSET)))
          (SETQ EBMYOFFSET (WINDOWPROP W (QUOTE YOFFSET)))
          (SETQ BM (WINDOWPROP W (QUOTE BM)))
          (SETQ BITMAPWIDTH (fetch BITMAPWIDTH of BM))
          (SETQ BITMAPHEIGHT (fetch BITMAPHEIGHT of BM))
          (SETQ BITSWIDE (WINDOWPROP W (QUOTE BITSWIDE)))
          (SETQ BITSHIGH (WINDOWPROP W (QUOTE BITSHIGH)))
          (SETQ DXOFFSET (WINDOWPROP W (QUOTE DXOFFSET)))
          (SETQ DYOFFSET (WINDOWPROP W (QUOTE DYOFFSET)))
          (SETQ EBMXLIMIT (IPLUS EBMXOFFSET BITSWIDE))
          (SETQ EBMYLIMIT (IPLUS EBMYOFFSET BITSHIGH))
          (COND
             (GRIDSPEC (SETQ GILEFT (fetch (REGION LEFT) of GRIDINTERIOR))
                    (SETQ GIBOTTOM (fetch (REGION BOTTOM) of GRIDINTERIOR))
                    (SETQ GIHEIGHT (fetch (REGION HEIGHT) of GRIDINTERIOR))
                    (SETQ GWIDTH (fetch (REGION WIDTH) of GRIDSPEC))
                    (SETQ GHEIGHT (fetch (REGION HEIGHT) of GRIDSPEC))
                    (SETQ EXTENT (WINDOWPROP W (QUOTE EXTENT)))
                    (SETQ EXTENTWIDTH (fetch (REGION WIDTH) of EXTENT))
                    (SETQ EXTENTHEIGHT (fetch (REGION HEIGHT) of EXTENT))
                                                                 ; Make a horizontal adjustment
                    (COND
                       ((FLOATP DX)                              ; Horizontal thumbing
                        [WINDOWPROP W (QUOTE XOFFSET)
                               (SETQ EBMXOFFSET (FIX (TIMES (IDIFFERENCE BITMAPWIDTH BITSWIDE)
                                                            DX]
                        (replace (REGION LEFT) of EXTENT with (IMINUS (QUOTIENT (TIMES EBMXOFFSET EXTENTWIDTH)
                                                                                BITMAPWIDTH)))
                                                                 (* BLTSHADE WHITESHADE W GILEFT GIBOTTOM
                                                                 SCREENWIDTH SCREENHEIGHT
                                                                 (QUOTE REPLACE) GRIDINTERIOR)
                        (RESETGRID.NEW BM GRIDSPEC BITSWIDE BITSHIGH 0 0 W T))
                       ((ILESSP DX 0)                            ; moving to the left.
                                                                 ; determine how many grid points to move.
                        (SETQ DXGRID (IMIN (GRIDXCOORD (IMINUS DX)
                                                       GRIDSPEC)
                                           (IDIFFERENCE BITMAPWIDTH EBMXLIMIT)))
                        (COND
                           ((NOT (IGREATERP DXGRID 0))           ; right edge is at the right margin
                            (RETURN)))
                        (WINDOWPROP W (QUOTE XOFFSET)
                               (SETQ EBMXOFFSET (IPLUS EBMXOFFSET DXGRID)))
                                                                 ; update EXTENT bar
                        (replace (REGION LEFT) of EXTENT with (IMAX (IMINUS (QUOTIENT (TIMES EBMXOFFSET
                                                                                      EXTENTWIDTH)
                                                                                      BITMAPWIDTH))
                                                                    (IMINUS EXTENTWIDTH)))
                                                                 ; move image to the left.
                        (BITBLT W (IPLUS GILEFT (TIMES DXGRID GWIDTH))
                               GIBOTTOM W GILEFT GIBOTTOM SCREENWIDTH SCREENHEIGHT (QUOTE INPUT)
                               (QUOTE REPLACE)
                               NIL GRIDINTERIOR)                 ; clear the newly exposed area.
                        (BLTSHADE WHITESHADE W (IPLUS GILEFT (TIMES (IDIFFERENCE BITSWIDE DXGRID)
                                                                    GWIDTH))
                               GIBOTTOM SCREENWIDTH SCREENHEIGHT (QUOTE REPLACE)
                               GRIDINTERIOR)
                        (RESETGRID.NEW BM GRIDSPEC DXGRID BITSHIGH (IDIFFERENCE BITSWIDE DXGRID)
                               0 W))
                       ((ILESSP 0 DX)                            ; determine how many grid point to the left to move.
```

```
        (SETQ DXGRID (IMIN EBMXOFFSET (GRIDXCOORD DX GRIDSPEC)))
        (COND
           ((NOT (IGREATERP DXGRID 0))          ; left edge is at the left margin
            (RETURN)))
        (WINDOWPROP W (QUOTE XOFFSET)
               (SETQ EBMXOFFSET (IDIFFERENCE EBMXOFFSET DXGRID)))
                                                ; update REGION bar
        (replace (REGION LEFT) of EXTENT with (IMIN (IMINUS (IQUOTIENT (TIMES EBMXOFFSET
                                                                          EXTENTWIDTH)
                                                                  BITMAPWIDTH))
                                                    0))
                                                ; move image to the right.
        (BITBLT W GILEFT GIBOTTOM W (IPLUS GILEFT (TIMES DXGRID GWIDTH))
               GIBOTTOM SCREENWIDTH SCREENHEIGHT (QUOTE INPUT)
               (QUOTE REPLACE)
               NIL GRIDINTERIOR)                ; clear the newly exposed area.
        (BLTSHADE WHITESHADE W GILEFT GIBOTTOM (TIMES DXGRID GWIDTH)
               GIHEIGHT
               (QUOTE REPLACE))
        (RESETGRID.NEW BM GRIDSPEC DXGRID BITSHIGH 0 0 W)))
                                                ; Make a vertical adjustment
(COND
   ((FLOATP DY)                                 ; Vertical Thumbing
    [WINDOWPROP W (QUOTE YOFFSET)
           (SETQ EBMYOFFSET (FIX (TIMES (IDIFFERENCE BITMAPHEIGHT BITSHIGH)
                                        (FDIFFERENCE 1.0 DY]
                                                ; set EXTENT bar
    (replace (REGION BOTTOM) of EXTENT with (IMINUS (QUOTIENT (TIMES EBMYOFFSET EXTENTHEIGHT)
                                                              BITMAPHEIGHT)))
                                                ; Clear Window
                                                (* BLTSHADE WHITESHADE W GILEFT GIBOTTOM
                                                SCREENWIDTH SCREENHEIGHT
                                                (QUOTE REPLACE) GRIDINTERIOR)
                                                ; Repaint the image using grid function
    (RESETGRID.NEW BM GRIDSPEC BITSWIDE BITSHIGH 0 0 W T))
   ((ILESSP DY 0)                               ; determine how many squares to move down.
    (SETQ DYGRID (IMIN (IDIFFERENCE (fetch (BITMAP BITMAPHEIGHT) of BM)
                                    EBMYLIMIT)
                       (GRIDYCOORD (IMIN GIHEIGHT (IMINUS DY))
                              GRIDSPEC)))
    (COND
       ((NOT (IGREATERP DYGRID 0))              ; top edge is at the top margin
        (RETURN)))
    (WINDOWPROP W (QUOTE YOFFSET)
           (SETQ EBMYOFFSET (IPLUS EBMYOFFSET DYGRID)))
    (replace (REGION BOTTOM) of EXTENT with (IMAX (IMINUS (QUOTIENT (TIMES EBMYOFFSET
                                                                          EXTENTHEIGHT)
                                                                  BITMAPHEIGHT))
                                                  (IMINUS EXTENTHEIGHT)))
    (BITBLT W GILEFT (IPLUS GIBOTTOM (ITIMES DYGRID GHEIGHT))
           W GILEFT GIBOTTOM SCREENWIDTH SCREENHEIGHT (QUOTE INPUT)
           (QUOTE REPLACE)
           NIL GRIDINTERIOR)                    (* BLTSHADE WHITESHADE W GILEFT
                                                (IPLUS GIBOTTOM (ITIMES (IDIFFERENCE BITSHIGH
                                                DYGRID) GHEIGHT)) SCREENWIDTH SCREENHEIGHT
                                                (QUOTE REPLACE) GRIDINTERIOR)
    (RESETGRID.NEW BM GRIDSPEC BITSWIDE DYGRID 0 (IDIFFERENCE BITSHIGH DYGRID)
           W T))
   ((ILESSP 0 DY)                               ; moving up;  determine how may grid squares to move.
    (SETQ DYGRID (IMIN EBMYOFFSET (GRIDYCOORD (IMIN GIHEIGHT DY)
                                              GRIDSPEC)))
    (COND
       ((NOT (IGREATERP DYGRID 0))              ; bottom edge is at the bottom margin
        (RETURN)))
    (WINDOWPROP W (QUOTE YOFFSET)
           (SETQ EBMYOFFSET (IDIFFERENCE EBMYOFFSET DYGRID)))
    (replace (REGION BOTTOM) of EXTENT with (IMIN (IMINUS (QUOTIENT (TIMES EBMYOFFSET
                                                                          EXTENTHEIGHT)
                                                                  BITMAPHEIGHT))
                                                  0))
    (BITBLT W GILEFT GIBOTTOM W GILEFT (IPLUS GIBOTTOM (ITIMES DYGRID GHEIGHT))
           SCREENWIDTH SCREENHEIGHT (QUOTE INPUT)
           (QUOTE REPLACE)
           NIL GRIDINTERIOR)                    (* BLTSHADE WHITESHADE W GILEFT GIBOTTOM
                                                (fetch (REGION WIDTH) of GRIDINTERIOR)
                                                (ITIMES DYGRID GHEIGHT) (QUOTE REPLACE))
    (RESETGRID.NEW BM GRIDSPEC BITSWIDE DYGRID 0 0 W T)))
;; This call to GRID is unnecessary as the grid dots get filled in earlier.
;; (COND ((WINDOWPROP W 'GRIDON) (GRID GRIDSPEC BITSWIDE BITSHIGH 'POINT W)))
[COND
   ([OR (ILESSP EBMXOFFSET DXOFFSET)
        (ILESSP EBMYOFFSET DYOFFSET)
        [IGREATERP (IPLUS EBMXOFFSET BITSWIDE)
               (IPLUS DXOFFSET (WINDOWPROP W (QUOTE BMDISPLAYWIDTH]
        (IGREATERP (IPLUS EBMYOFFSET BITSHIGH)
               (IPLUS DYOFFSET (WINDOWPROP W (QUOTE BMDISPLAYHEIGHT]
```

```
                                                                    ; Adjust the display region left lower corner so the selected
                                                                    ; region is near the center.
                        [WINDOWPROP W (QUOTE DXOFFSET)
                               (SETQ DXOFFSET (IMAX 0 (IMIN (IDIFFERENCE (fetch (BITMAP BITMAPWIDTH) of BM)
                                                                    (WINDOWPROP W (QUOTE BMDISPLAYWIDTH)))
                                                            (IDIFFERENCE (IPLUS EBMXOFFSET (LRSH BITSWIDE 1))
                                                                    (LRSH (WINDOWPROP W (QUOTE BMDISPLAYWIDTH))
                                                                          1]
                        (WINDOWPROP W (QUOTE DYOFFSET)
                               (SETQ DYOFFSET (IMAX 0 (IMIN (IDIFFERENCE (fetch (BITMAP BITMAPHEIGHT)
                                                                            of BM)
                                                                    (WINDOWPROP W (QUOTE BMDISPLAYHEIGHT)))
                                                            (IDIFFERENCE (IPLUS EBMYOFFSET (LRSH BITSHIGH 1))
                                                                    (LRSH (WINDOWPROP W (QUOTE BMDISPLAYHEIGHT))
                                                                          1]
                     (UPDATE/BM/DISPLAY BM W])
```

(\EDITBM/PUTUP/DISPLAY

```
  [LAMBDA (WINDOW BM GRIDSPEC GRIDINTERIOR BITSWIDE BITSHIGH)    ; Edited 31-Aug-87 13:05 by FS
                                                                 (* initializes the display for the bitmap editor.)
                                                                 (* DSPFILL GRIDINTERIOR WHITESHADE
                                                                 (QUOTE REPLACE) WINDOW)
                                                                 (* COND ((WINDOWPROP WINDOW
                                                                 (QUOTE GRIDON)) (GRID GRIDSPEC BITSWIDE BITSHIGH
                                                                 (QUOTE POINT) WINDOW)))
    (RESETGRID.NEW BM GRIDSPEC BITSWIDE BITSHIGH 0 0 WINDOW T)
    (UPDATE/BM/DISPLAY BM WINDOW])
```

(EDITBMRESHAPEFN

```
  [LAMBDA (BMEDITWINDOW OLDIMAGE OLDREGION OLDSCREENREGION ZEROBMFLG)
                                                                 ; Edited 31-Aug-87 12:41 by FS
    ;; allows the bitmap edit window to be reshaped to enlarge the editting area.  This is also called to set up the image during initialization.
    (PROG (BMWINTERIORWIDTH BMWINTERIORHEIGHT EDITAREABITWIDTH EDITAREABITHEIGHT GRIDSQUARE GRIDINTERIOR
                BITMAPWIDTH BMDISPLAYWIDTH BMDISPLAYBOTTOM BMDISPLAYHEIGHT BITMAPHEIGHT
                (BM (WINDOWPROP BMEDITWINDOW (QUOTE BM)))
                MINCOMMANDAREAWIDTH EXTENTWIDTH EXTENTHEIGHT)
          (SETQ MINCOMMANDAREAWIDTH 30)
          (SETQ BITMAPWIDTH (fetch (BITMAP BITMAPWIDTH) of BM))
          (SETQ BITMAPHEIGHT (fetch (BITMAP BITMAPHEIGHT) of BM))
          (SETQ BMWINTERIORWIDTH (WINDOWPROP BMEDITWINDOW (QUOTE WIDTH)))
    ;; leave room at the top for the full size display area.  But not more than half of the window.
          (SETQ BMWINTERIORHEIGHT (IMAX (IDIFFERENCE (WINDOWPROP BMEDITWINDOW (QUOTE HEIGHT))
                                                     (IPLUS BITMAPHEIGHT GRIDTHICKNESS))
                                        (IQUOTIENT (WINDOWPROP BMEDITWINDOW (QUOTE HEIGHT))
                                               2)))
    ;; if the user hasn't set it, determine the grid size as the largest size which fits the interior but not larger than NORMALGRIDSQUARE nor smaller
    ;; than MINGRIDSQUARE.  If GRIDSQUARE was specified, reset it to NIL so that if reshaped it will be recalculated.
          (SETQ GRIDSQUARE (OR (WINDOWPROP BMEDITWINDOW (QUOTE GRIDSQUARE)
                                       NIL)
                               (IMAX (IMIN (IQUOTIENT BMWINTERIORWIDTH BITMAPWIDTH)
                                           (IQUOTIENT BMWINTERIORHEIGHT BITMAPHEIGHT)
                                           NORMALGRIDSQUARE)
                                     MINGRIDSQUARE)))            ; calculate how many bits will be displayed at once.
          (SETQ EDITAREABITWIDTH (IMIN (IQUOTIENT BMWINTERIORWIDTH GRIDSQUARE)
                                       BITMAPWIDTH))
          (WINDOWPROP BMEDITWINDOW (QUOTE BITSWIDE)
                 EDITAREABITWIDTH)
          (SETQ EDITAREABITHEIGHT (IMIN (IQUOTIENT BMWINTERIORHEIGHT GRIDSQUARE)
                                        BITMAPHEIGHT))           ; calculate offset of display and command regions at the top of
                                                                 ; the window.
          (WINDOWPROP BMEDITWINDOW (QUOTE BITSHIGH)
                 EDITAREABITHEIGHT)
          (SETQ BMDISPLAYBOTTOM (IPLUS (ITIMES GRIDSQUARE EDITAREABITHEIGHT)
                                       GRIDTHICKNESS))
          (SETQ BMDISPLAYWIDTH (IMIN BITMAPWIDTH (IDIFFERENCE BMWINTERIORWIDTH MINCOMMANDAREAWIDTH)))
    ;; put the offset --- the lower left coordinate --- in the same place unless the new shape allows more to be shown past the upper right corner.
          (WINDOWPROP BMEDITWINDOW (QUOTE XOFFSET)
                 (IMIN (WINDOWPROP BMEDITWINDOW (QUOTE XOFFSET))
                       (IDIFFERENCE BITMAPWIDTH EDITAREABITWIDTH)))
          (WINDOWPROP BMEDITWINDOW (QUOTE YOFFSET)
                 (IMIN (WINDOWPROP BMEDITWINDOW (QUOTE YOFFSET))
                       (IDIFFERENCE BITMAPHEIGHT EDITAREABITHEIGHT)))
                                                                 ; Center edit square
          (SETQ GRIDINTERIOR (create REGION
                                     LEFT _ (IQUOTIENT (IDIFFERENCE BMWINTERIORWIDTH (ITIMES EDITAREABITWIDTH
                                                                                            GRIDSQUARE))
                                                       2)
                                     BOTTOM _ (IQUOTIENT (IDIFFERENCE BMDISPLAYBOTTOM (ITIMES EDITAREABITHEIGHT
                                                                                             GRIDSQUARE))
                                                         2)
                                     WIDTH _ (ITIMES EDITAREABITWIDTH GRIDSQUARE)
```

```
                                HEIGHT _ (ITIMES EDITAREABITHEIGHT GRIDSQUARE)))
        (WINDOWPROP BMEDITWINDOW (QUOTE GRIDINTERIOR)
               GRIDINTERIOR)
        (WINDOWPROP BMEDITWINDOW (QUOTE BMDISPLAYBOTTOM)
               BMDISPLAYBOTTOM)
        (WINDOWPROP BMEDITWINDOW (QUOTE BMDISPLAYWIDTH)
               BMDISPLAYWIDTH)
        (WINDOWPROP BMEDITWINDOW (QUOTE BMDISPLAYHEIGHT)
               (SETQ BMDISPLAYHEIGHT (IDIFFERENCE (WINDOWPROP BMEDITWINDOW (QUOTE HEIGHT))
                                                  BMDISPLAYBOTTOM)))
        (WINDOWPROP BMEDITWINDOW (QUOTE DISPLAYREGION)
               (create REGION
                      LEFT _ 0
                      BOTTOM _ BMDISPLAYBOTTOM
                      WIDTH _ BMDISPLAYWIDTH
                      HEIGHT _ BMDISPLAYHEIGHT))
        (WINDOWPROP BMEDITWINDOW (QUOTE GRIDSPEC)
               (create REGION
                      LEFT _ (fetch (REGION LEFT) of GRIDINTERIOR)
                      BOTTOM _ (fetch (REGION BOTTOM) of GRIDINTERIOR)
                      WIDTH _ GRIDSQUARE
                      HEIGHT _ GRIDSQUARE))
        (SETQ EXTENTHEIGHT (QUOTIENT (TIMES BITMAPHEIGHT (WINDOWPROP BMEDITWINDOW (QUOTE HEIGHT)))
                                     EDITAREABITHEIGHT))
        [SETQ EXTENTWIDTH (IDIFFERENCE (QUOTIENT (TIMES BITMAPWIDTH BMWINTERIORWIDTH)
                                                 EDITAREABITWIDTH)
                                       (WINDOWPROP BMEDITWINDOW (QUOTE BORDER]
        (WINDOWPROP BMEDITWINDOW (QUOTE EXTENT)
               (CREATEREGION (MINUS (QUOTIENT (TIMES (WINDOWPROP BMEDITWINDOW (QUOTE XOFFSET))
                                                     EXTENTWIDTH)
                                              BITMAPWIDTH))
                             (MINUS (QUOTIENT (TIMES (WINDOWPROP BMEDITWINDOW (QUOTE YOFFSET))
                                                     EXTENTHEIGHT)
                                              BITMAPHEIGHT))
                             EXTENTWIDTH EXTENTHEIGHT))
        (EDITBMREPAINTFN BMEDITWINDOW NIL ZEROBMFLG])
```

(EDITBMREPAINTFN.NEW

```
  [LAMBDA (WIN REGION ZEROBM)                                   ; Edited 27-Aug-87 22:02 by FS
```

;; Stub in case I missed a call to this guy. Take out later.

```
    (EDITBMREPAINTFN WIN REGION ZEROBM])
```

(EDITBMREPAINTFN

```
  [LAMBDA (WIN REGION ZEROBM)                                   (* N.H.Briggs " 4-Sep-87 15:07")
```

;; redisplays a bitmap editting window If ZEROBM is non-NIL, it doesn't bother to display the bits.

```
    (PROG [(GRIDSPEC (WINDOWPROP WIN (QUOTE GRIDSPEC)))
           (EDITAREABITWIDTH (WINDOWPROP WIN (QUOTE BITSWIDE)))
           (EDITAREABITHEIGHT (WINDOWPROP WIN (QUOTE BITSHIGH)))
           (BM (WINDOWPROP WIN (QUOTE BM]
          (CLEARW WIN)                                          ; gray the area above the edit grid that is not bitmap display area.
          (BLTSHADE NOTINUSEGRAY WIN (PLUS (WINDOWPROP WIN (QUOTE BMDISPLAYWIDTH))
                                           GRIDTHICKNESS)
                 (WINDOWPROP WIN (QUOTE BMDISPLAYBOTTOM)))
```

;; put in the display of the full sized bitmap.

```
          (UPDATE/BM/DISPLAY BM WIN)
```

;; FS: Now that RESETGRID displays the grid, don't need the call to GRID.

;; (COND ((WINDOWPROP WIN 'GRIDON) (GRID GRIDSPEC EDITAREABITWIDTH EDITAREABITHEIGHT 'POINT WIN)))

```
          (if ZEROBM
              then (if (WINDOWPROP WIN (QUOTE GRIDON))
                       then (GRID GRIDSPEC EDITAREABITWIDTH EDITAREABITHEIGHT (QUOTE POINT)
                                  WIN))
                   else (RESETGRID.NEW BM GRIDSPEC EDITAREABITWIDTH EDITAREABITHEIGHT 0 0 WIN])
```

(RESETGRID.NEW

```
  [LAMBDA (BM GRIDSPEC WIDTH HEIGHT ORIGX ORIGY WINDOW DOCLEARFLG)
                                                                (* N.H.Briggs " 4-Sep-87 15:08")
```

;; Copies the contents of a bitmap into the edit display grid of window. ORIGX & Y are used to offest into both bitmap and destination window.

```
    (LET (XOFFSET YOFFSET MAXX MAXY SHADE XSCALE YSCALE TEMPBM)
         (SETQ XSCALE (fetch (REGION WIDTH) of GRIDSPEC))
         (SETQ YSCALE (fetch (REGION HEIGHT) of GRIDSPEC))
         (if (NULL ORIGX)
             then (SETQ ORIGX 0))
         (if (NULL ORIGY)
             then (SETQ ORIGY 0))
         (SETQ XOFFSET (WINDOWPROP WINDOW (QUOTE XOFFSET)))
         (SETQ YOFFSET (WINDOWPROP WINDOW (QUOTE YOFFSET)))
         (SETQ MAXX (IPLUS ORIGX WIDTH -1))
         (SETQ MAXY (IPLUS ORIGY HEIGHT -1))
```

```
        ;; Build & cache a temporary bitmap.
        (SETQ TEMPBM (WINDOWPROP WINDOW (QUOTE TEMPBM)))
        (if (NOT TEMPBM)
            then (SETQ TEMPBM (BITMAPCREATE (BITMAPWIDTH WINDOW)
                                            (BITMAPHEIGHT BM)))
                 (WINDOWPROP WINDOW (QUOTE TEMPBM)
                        TEMPBM))
        ;; Use SCALEBM.  Bitmap destination must be empty (white).
        (if DOCLEARFLG
            then (BLTSHADE WHITESHADE WINDOW (LEFTOFGRIDCOORD ORIGX GRIDSPEC)
                         (BOTTOMOFGRIDCOORD ORIGY GRIDSPEC)
                         (TIMES WIDTH XSCALE)
                         (TIMES HEIGHT YSCALE)
                         (QUOTE REPLACE)))
        (SCALEBM BM (PLUS ORIGX XOFFSET)
               (PLUS ORIGY YOFFSET)
               WINDOW
               (LEFTOFGRIDCOORD ORIGX GRIDSPEC)
               (BOTTOMOFGRIDCOORD ORIGY GRIDSPEC)
               WIDTH HEIGHT XSCALE YSCALE TEMPBM)
        ;; Shade the pixels correctly.
        (BLTSHADE DARKBITSHADE WINDOW (LEFTOFGRIDCOORD ORIGX GRIDSPEC)
               (BOTTOMOFGRIDCOORD ORIGY GRIDSPEC)
               (TIMES WIDTH XSCALE)
               (TIMES HEIGHT YSCALE)
               (QUOTE ERASE))
        ;; Add grid
        (if (WINDOWPROP WINDOW (QUOTE GRIDON))
            then (if (OR (NEQ ORIGX (CAR GRIDSPEC))
                         (NEQ ORIGY (CADR GRIDSPEC)))
                     then (SETQ GRIDSPEC (COPYALL GRIDSPEC))
                          (replace (REGION LEFT) of GRIDSPEC with (LEFTOFGRIDCOORD ORIGX GRIDSPEC))
                          (replace (REGION BOTTOM) of GRIDSPEC with (BOTTOMOFGRIDCOORD ORIGY GRIDSPEC)))
                 (GRID GRIDSPEC WIDTH HEIGHT (QUOTE POINT)
                       WINDOW])

)

(DEFINEQ

(SCALEBM
  [LAMBDA (SRCEBM SRCEX SRCEY DESTBM DESTX DESTY SRCEWIDTH SRCEHEIGHT XSCALE YSCALE TEMPBM)
                                                                    (* N.H.Briggs " 4-Sep-87 15:48")
    ;; Magnify a bitmap as per EDITBM.  Use smearing algorithm.
    (LET ((DESTWIDTH (BITMAPWIDTH DESTBM))
          (DESTHEIGHT (if (WINDOWP DESTBM)
                          then (WINDOWPROP DESTBM (QUOTE HEIGHT))
                        else (BITMAPHEIGHT DESTBM)))
          XSTEPS YSTEPS POWER)
        ;; Check parameters, apply  defaults
        (if (NUMBERP SRCEWIDTH)
          else (SETQ SRCEWIDTH (BITMAPWIDTH SRCEBM)))
        (if (NUMBERP SRCEHEIGHT)
          else (SETQ SRCEHEIGHT (BITMAPHEIGHT SRCEBM)))
        ;; Save effort by considering min of srce and dest.
        (SETQ DESTWIDTH (MIN DESTWIDTH (TIMES SRCEWIDTH XSCALE)))
        (SETQ DESTHEIGHT (MIN DESTHEIGHT (TIMES SRCEHEIGHT YSCALE)))
        (SETQ SRCEWIDTH (MIN SRCEWIDTH (IQUOTIENT DESTWIDTH XSCALE)))
        (SETQ SRCEHEIGHT (MIN SRCEHEIGHT (IQUOTIENT DESTHEIGHT YSCALE)))
        (if TEMPBM
            then (BLTSHADE WHITESHADE TEMPBM)
          else (SETQ TEMPBM (BITMAPCREATE DESTWIDTH SRCEHEIGHT)))
        ;; CALL EXPANDBM twice, once for each direction, because we have a spare bitmap which makes it run faster than a single call to
        ;; EXPANDBM would (I think).
        ;;
        ;; Do X Direction Smearing.
        ;; ============
        (EXPANDBM SRCEBM SRCEX SRCEY SRCEWIDTH SRCEHEIGHT TEMPBM 0 0 DESTWIDTH SRCEHEIGHT XSCALE 1 XSCALE 1)
        ;;
        ;; Do Y Direction Smearing.
        ;; ============
        (EXPANDBM TEMPBM 0 0 DESTWIDTH SRCEHEIGHT DESTBM DESTX DESTY DESTWIDTH DESTHEIGHT 1 YSCALE 1 YSCALE)
        ;;
        ;; Return the temporary bitmap for recycling purposes.
        TEMPBM])
```

(BLTPATTERN

```
[LAMBDA (SRCE SX SY SW SH DEST DX DY DW DH OPER TEMPBM)        (* N.H.Briggs " 4-Sep-87 15:10")
```

;; Fills region of Destination with tiles of Source region, using operation. If Temporary bitmap is provided, use it for optimal performance.

```
(PROG (W H RX RW)
      (if (NULL SW)
          then (SETQ SW (BITMAPWIDTH SRCE)))
      (if (NULL SH)
          then (SETQ SH (BITMAPHEIGHT SRCE)))
```

;;

;; Fill columns

;;

```
      [if TEMPBM
          then
```

;; Temporary bitmap is only useful if larger than source.

```
               (if [AND (GREATERP (BITMAPWIDTH TEMPBM)
                                  (MIN SW (BITMAPWIDTH SRCE)))
                        (GREATERP (BITMAPHEIGHT TEMPBM)
                                  (MIN SH (BITMAPHEIGHT SRCE]
                   then (BLTPATTERN.REPLACEDISPLAY SRCE SX SY SW SH TEMPBM 0 0 (BITMAPWIDTH TEMPBM)
                               (BITMAPHEIGHT TEMPBM))
```

;; Allow code to fall through using TEMPBM as source area.

```
                   (SETQ SRCE TEMPBM)
                   (SETQ SX 0)
                   (SETQ SY 0)
                   (SETQ SW (ITIMES SW (IQUOTIENT (BITMAPWIDTH TEMPBM)
                                                  SW)))
                   (SETQ SH (ITIMES SH (IQUOTIENT (BITMAPHEIGHT TEMPBM)
                                                  SH]
      (if (AND (EQ OPER (QUOTE REPLACE))
               (OR (BITMAPP DEST)
                   (WINDOWP DEST)))
          then (BLTPATTERN.REPLACEDISPLAY SRCE SX SY SW SH DEST DX DY DW DH)
               (RETURN))
```

;; Even if operation is REPLACE, don't know if destination is inexpensively readable (e.g. Interpress stream. SO, this is the general case here.

```
      (while (GREATERP DH 0)
         do (SETQ H (MIN SH DH))
```

;;

```
            (SETQ RW DW)
            (SETQ RX DX)
```

;;

;; Fill rows

;;

```
            (while (GREATERP RW 0) do (SETQ W (MIN SW RW))
                                      (BITBLT SRCE SX SY DEST RX DY W H NIL OPER)
                                      (SETQ RW (DIFFERENCE RW W))
                                      (SETQ RX (PLUS RX W)))
```

;;

```
            (SETQ DH (DIFFERENCE DH H))
            (SETQ DY (PLUS DY H])
```

(BLTPATTERN.REPLACEDISPLAY

```
[LAMBDA (SRCE SX SY SW SH DEST DX DY DW DH)                    (* N.H.Briggs " 4-Sep-87 15:11")
```

;; This routine only replaces the destination with the source, and assumes the destination itself can be easily read from and blt'ed to.

;; Put initial bitmap into destination. Source should not be within destination area, otherwise it will be overwritten.

```
(LET (RX RY RW RH W H)                                         ; R's are remaining area.
     (SETQ W (MIN SW DW))
     (SETQ H (MIN SH DH))
     (BLTSHADE WHITESHADE DEST DX DY W H (QUOTE REPLACE))
     (BITBLT SRCE SX SY DEST DX DY W H NIL (QUOTE REPLACE))
     (SETQ RX (PLUS DX W))
     (SETQ RW (DIFFERENCE DW W))
```

;; Now power up until width is full.

```
     (while (GREATERP RW 0) do (SETQ W (MIN SW RW))
                               (BITBLT DEST DX DY DEST RX DY W H NIL (QUOTE REPLACE))
                               (SETQ RW (DIFFERENCE RW W))    ; Reduce remaining width
                               (SETQ RX (PLUS RX W))          ; Set next starting position
                               (SETQ SW (PLUS SW SW))         ; Can now use 2x area.
                               )
```

;;

```
     (SETQ RY (PLUS DY H))
     (SETQ RH (DIFFERENCE DH H))
     (SETQ SH H)
```

```
        (SETQ W DW)
        ;; Now power up until height is full.
        (while (GREATERP RH 0) do (SETQ H (MIN SH RH))
                                  (BITBLT DEST DX DY DEST DX RY W H NIL (QUOTE REPLACE))
                                  (SETQ RH (DIFFERENCE RH H))    ; Reduce remaining width
                                  (SETQ RY (PLUS RY H))          ; Set next starting position
                                  (SETQ SH (PLUS SH SH))         ; Can now use 2x area.
                                  ])

)

(DEFINEQ

(EXPANDBITMAP
  [LAMBDA (BITMAP WIDTHFACTOR HEIGHTFACTOR)                    (* N.H.Briggs "16-Nov-87 17:10")
    ;; Returns a new bitmap which is WidthFactor and HeightFactor bigger.
    ;; FS:  This slow piece of code has been replaced with a much faster, general one, EXPAND.l
    (LET (WIDTH HEIGHT BITSPERPIXEL NEWWIDTH NEWHEIGHT NEWX NEWY NEWBITMAP)
         (OR WIDTHFACTOR (SETQ WIDTHFACTOR 1))
         (OR HEIGHTFACTOR (SETQ HEIGHTFACTOR 1))
         (SETQ HEIGHT (fetch (BITMAP BITMAPHEIGHT) of BITMAP))
         (SETQ WIDTH (fetch (BITMAP BITMAPWIDTH) of BITMAP))
         (SETQ BITSPERPIXEL (fetch (BITMAP BITMAPBITSPERPIXEL) of BITMAP))
         (SETQ NEWWIDTH (ITIMES WIDTHFACTOR WIDTH))
         (SETQ NEWHEIGHT (ITIMES HEIGHTFACTOR HEIGHT))
         (SETQ NEWBITMAP (BITMAPCREATE NEWWIDTH NEWHEIGHT BITSPERPIXEL))
         ;; OLD code commented out here.
                                                               (* LET NIL (* Expand in x-direction.
                                                               *) (SETQ NEWX 0) (for X from 0 to
                                                               (SUB1 WIDTH) do (for I from 1 to WIDTHFACTOR do
                                                               (BITBLT BITMAP X 0 NEWBITMAP NEWX 0 1 HEIGHT
                                                               (QUOTE INPUT) (QUOTE REPLACE))
                                                               (add NEWX 1))) (* Expand in y-direction.
                                                               *) (SETQ NEWY (SUB1 NEWHEIGHT))
                                                               (for Y from (SUB1 HEIGHT) to 0 by -1 do
                                                               (for I from 1 to HEIGHTFACTOR do
                                                               (BITBLT NEWBITMAP 0 Y NEWBITMAP 0 NEWY NEWWIDTH
         1 (QUOTE INPUT) (QUOTE REPLACE)) (add NEWY -1))))
         (EXPANDBM BITMAP 0 0 WIDTH HEIGHT NEWBITMAP 0 0 NEWWIDTH NEWHEIGHT WIDTHFACTOR HEIGHTFACTOR
                WIDTHFACTOR HEIGHTFACTOR)
         NEWBITMAP])

(EXPANDBM
  [LAMBDA (SRCEBM SRCEX SRCEY SRCEW SRCEH DESTBM DESTX DESTY DESTW DESTH XSCALE YSCALE XSPACE YSPACE)
                                                               (* N.H.Briggs " 4-Sep-87 15:18")
    ;; Expands a region of SrceBM by X&Y scale into a region of DestBM, spaced Xspace by YSpace apart (space must be larger than scale).  SrceBM
    ;; cannot be the same bitmap as DestBM.  The entire region inside DestBM is cleared.
    (PROG (XSTEPS YSTEPS POWER)
     ;; Check parameters, apply  defaults
          (if (NUMBERP SRCEX)
            else (SETQ SRCEX 0))
          (if (NUMBERP SRCEY)
            else (SETQ SRCEY 0))
          (if (NUMBERP SRCEW)
            else (SETQ SRCEW (BITMAPWIDTH SRCEBM)))
          (if (NUMBERP SRCEH)
            else (SETQ SRCEH (BITMAPHEIGHT SRCEBM)))
          (if (NUMBERP DESTX)
            else (SETQ SRCEX 0))
          (if (NUMBERP DESTY)
            else (SETQ SRCEY 0))
     ;; Save effort by considering min of srce and dest.
          [SETQ DESTW (IMIN DESTW (TIMES SRCEW (IMAX XSCALE XSPACE]
          [SETQ DESTH (IMIN DESTH (TIMES SRCEH (IMAX YSCALE YSPACE]
          [SETQ SRCEW (IMIN SRCEW (PLUS 1 (IQUOTIENT DESTW (IMAX XSCALE XSPACE]
          [SETQ SRCEH (IMIN SRCEH (PLUS 1 (IQUOTIENT DESTH (IMAX YSCALE YSPACE]
          (BLTSHADE WHITESHADE DESTBM DESTX DESTY DESTW DESTH)
          (if (AND (EQP XSPACE 1)
                   (EQP YSPACE 1))
              then (BITBLT SRCEBM SRCEX SRCEY DESTBM DESTX DESTY SRCEW SRCEH)
                   (RETURN DESTBM))
     ;;
     ;; Do X Direction Smearing.
     ;; ============
     ;; Spread out bitmap by spacefactor.  Start from far side to avoid overwrite (if srce = dest)
          (if (EQP XSPACE 1)
              then ;; Don't fill destination, instead use srce in YSmear loop.
```

```
                    ;; (BITBLT SRCEBM SRCEX SRCEY DESTBM DESTX DESTY SRCEW SRCEH)

            else ;; Spread out bitmap by spacefactor.  Start from far side to avoid overwrite (if srce = dest)
                 (for I from (SUB1 SRCEW) to 0 by -1 do (BITBLT SRCEBM (PLUS SRCEX I)
                                                                SRCEY DESTBM (PLUS DESTX (TIMES I XSPACE))
                                                                DESTY 1 SRCEH)))
      ;; Now smear by scalefactor.  Each step smears out a power of two.  LSH is in ucode.
      [if (EQP XSCALE 1)
        else (SETQ POWER 1)
             (while (ILEQ POWER (LSH XSCALE -1)) do ;; In the X direction, only need to blt SRCEH bits high, and must shorten W to
                                                    ;; remain within DESTW
                                                    (BITBLT DESTBM DESTX DESTY DESTBM (PLUS DESTX POWER)
                                                           DESTY
                                                           (DIFFERENCE DESTW POWER)
                                                           SRCEH NIL (QUOTE PAINT))
                                                    (SETQ POWER (PLUS POWER POWER)))
             ;; Clean up for non power of two.
             (if (ZEROP (DIFFERENCE XSCALE POWER))
               else (BITBLT DESTBM DESTX DESTY DESTBM (PLUS DESTX (DIFFERENCE XSCALE POWER))
                          DESTY
                          (DIFFERENCE DESTW (DIFFERENCE XSCALE POWER))
                          SRCEH NIL (QUOTE PAINT]
      ;;
      ;; Do Y Direction Smearing.
      ;; ============
      ;; Spread out bitmap by spacefactor.  Start from far side to avoid overwrite (if srce = dest)
      [if (EQP YSPACE 1)
        else (if (EQP XSPACE 1)
                 then ;; Didn't need to paint in destination, so can avoid second loop by blting from SRCBM instead of DESTBM.
                      (for J from (SUB1 SRCEH) to 0 by -1 do (BITBLT SRCEBM SRCEX (PLUS SRCEY J)
                                                                    DESTBM DESTX (PLUS DESTY (TIMES J YSPACE))
                                                                    DESTW 1))
               else (for J from (SUB1 SRCEH) to 0 by -1 do (BITBLT DESTBM DESTX (PLUS DESTY J)
                                                                  DESTBM DESTX (PLUS DESTY (TIMES J YSPACE))
                                                                  DESTW 1))
                    ;; Since we reused DESTBM, parts of the dest have bits in them but shouldn't.  So, clear them.
                    (for J from 0 to SRCEH by YSPACE do (BLTSHADE WHITESHADE DESTBM DESTX (PLUS DESTY J 1)
                                                                 DESTW
                                                                 (SUB1 YSPACE]
      ;; Now smear correctly.  Each step smears out a power of two.  LSH is in ucode.
      [if (EQP YSCALE 1)
        else (SETQ POWER 1)
             (while (ILEQ POWER (LSH YSCALE -1)) do (BITBLT DESTBM DESTX DESTY DESTBM DESTX (PLUS DESTY POWER)
                                                           DESTW
                                                           (DIFFERENCE DESTH POWER)
                                                           NIL
                                                           (QUOTE PAINT))
                                                    (SETQ POWER (PLUS POWER POWER)))
             ;; Clean up for non power of two.
             (if (ZEROP (DIFFERENCE YSCALE POWER))
               else (BITBLT DESTBM DESTX DESTY DESTBM DESTX (PLUS DESTY (DIFFERENCE YSCALE POWER))
                          DESTW DESTH NIL (QUOTE PAINT]
      ;;
      ;; Return the temporary bitmap for recycling purposes.
       DESTBM])

)

(PUTPROPS FASTEDITBM COPYRIGHT ("Xerox Corporation" 1987))
```

FUNCTION INDEX

MACRO INDEX