```
(RPAQQ NOTEPADCOMS
       [(FNS ACQUIRE.STYLE ACQUIRE.SYMMETRY ADD.NOTEPAD.TO.BACKGROUND.MENU BITMAP.INTERIOR DELDEF.NOTEPAD.STYLE
             DISTANCE DUMP.NOTEPAD.STYLE ERASE.REGION EXTEND.AREA EXTEND.AREA.X EXTEND.AREA.Y GET.BITMAP
             GET.NOTEPAD.BITMAP GET.NOTEPAD.REGION GET.SYMMETRIC.BITMAPS GETDEF.NOTEPAD.STYLE
             GETPOSITION.RUBBERBAND GETPOSITION.RUBBERBAND1 MARK.SPOT MASKED.BRUSH MOVE.BITMAP MOVE.BITMAP1
             NOTEPAD NOTEPAD.ADD.TO.BACKGROUND.MENU NOTEPAD.BUTTONEVENTFN NOTEPAD.BUTTONFN NOTEPAD.CIRCLE
             NOTEPAD.COMMAND.MENU NOTEPAD.COMMAND.MENU.CREATE NOTEPAD.COOKIE.CUT NOTEPAD.COPY.FROM.SCREEN
             NOTEPAD.CURVE NOTEPAD.DEFINE.BRUSH NOTEPAD.DEFINE.FONT NOTEPAD.HELP NOTEPAD.READ.FONT
             NOTEPAD.FONTS.IN.CORE READ.NOTEPAD.STYLE NOTEPAD.DEFINE.GRID NOTEPAD.DEFINE.GRID1
             NOTEPAD.DEFINE.MASK NOTEPAD.DELETE.STYLE NOTEPAD.EDIT.BRUSH NOTEPAD.EDIT.MASK
             NOTEPAD.EDIT.RECTANGLE NOTEPAD.EDIT.SHADE NOTEPAD.ELLIPSE NOTEPAD.LINE NOTEPAD.MASK=BRUSH.OUTLINE
             NOTEPAD.NAMED.OBJECT NOTEPAD.POINT.OF.SYMMETRY NOTEPAD.RESTORE.STYLE NOTEPAD.SETUP.TO.PAINT
             NOTEPAD.SHADE.RECTANGLE NOTEPAD.SKETCH NOTEPAD.SKETCH1 NOTEPAD.TEXT NOTEPAD.USE.GRID
             NOTEPAD.USE.MASK NOTEPAD.USE.SYMMETRIC.BRUSH/MASK NOTEPAD.USE.SYMMETRY NOTEPAD.FILL NOTEPAD.CONFIRM
             NOTEPAD.CREATE NOTEPAD.DEFAULT.CHARACTERISTICS NOTEPAD.GETPOSITION NOTEPAD.MASK NOTEPAD.ON.GRID
             NOTEPAD.ON.GRID.X NOTEPAD.OPERATION NOTEPAD.SOLID.AREA BITMAP.EXTERIOR NOTEPAD.TITLEBUTTONFN
             PAINT.A.BITMAP PAINT.ALL.BITMAPS PAINT.AT.POSSIBLE.POINT PAINT.WITH.BITMAP PICKUP.BITMAP
             PICKUP.SCREEN.BITMAP READ.FROM.PROMPT.WINDOW PUTBACK.BITMAP PUTDEF.NOTEPAD.STYLE SAVE.STYLE SBIT
             TEST.AND.SET)
        (FNS GET.WINDOW.REGION GETCOLORPOSITION COLORBITMAPP)
        (VARS NOTEPAD.DEFAULT.FONT NOTEPAD.STYLE.REPRESENTATION.NUMBER (NOTEPAD.COMMAND.MENU)
              (NOTEPAD.SHOW.FILL)
              (NOTEPAD.USE.GRID.MENU))
        (VARS (.NOTEPAD.BRUSH.1)
              (.NOTEPAD.BRUSH.2)
              (.NOTEPAD.BRUSH.3)
              (.NOTEPAD.BRUSH.4)
              (.NOTEPAD.BRUSH.5)
              (.NOTEPAD.BRUSH.6)
              (.NOTEPAD.BRUSH.7)
              (.NOTEPAD.BRUSH.8)
              (.NOTEPAD.MASK.1)
              (.NOTEPAD.MASK.2)
              (.NOTEPAD.MASK.3)
              (.NOTEPAD.MASK.4)
              (.NOTEPAD.MASK.5)
              (.NOTEPAD.MASK.6)
              (.NOTEPAD.MASK.7)
              (.NOTEPAD.MASK.8))
        (GLOBALVARS NOTEPAD.COMMAND.MENU NOTEPAD.USE.GRID.MENU .NOTEPAD.WINDOW .NOTEPAD.OPERATION
             .NOTEPAD.BRUSH.1 .NOTEPAD.BRUSH.2 .NOTEPAD.BRUSH.3 .NOTEPAD.BRUSH.4 .NOTEPAD.BRUSH.5
             .NOTEPAD.BRUSH.6 .NOTEPAD.BRUSH.7 .NOTEPAD.BRUSH.8 .NOTEPAD.MASK.1 .NOTEPAD.MASK.2
             .NOTEPAD.MASK.3 .NOTEPAD.MASK.4 .NOTEPAD.MASK.5 .NOTEPAD.MASK.6 .NOTEPAD.MASK.7 .NOTEPAD.MASK.8
             .NOTEPAD.USE.GRID .NOTEPAD.GRID.X0 .NOTEPAD.GRID.Y0 .NOTEPAD.GRID.DX .NOTEPAD.GRID.DY
             .NOTEPAD.USE.MASK .NOTEPAD.INVERSE.OPERATION .NOTEPAD.USE.SYMMETRIC.BRUSH/MASK
             .NOTEPAD.USE.SYMMETRY .NOTEPAD.POSX .NOTEPAD.POSY .NOTEPAD.BRUSH.WIDTH .NOTEPAD.BRUSH.HEIGHT
             .NOTEPAD.BRUSH.HALF.WIDTH .NOTEPAD.BRUSH.HALF.HEIGHT .NOTEPAD.MASK.WIDTH .NOTEPAD.MASK.HEIGHT
             .NOTEPAD.MASK.HALF.WIDTH .NOTEPAD.MASK.HALF.HEIGHT .NOTEPAD.PREVIOUS.MIDX .NOTEPAD.PREVIOUS.MIDY)
        (VARS (COLORSPOTMARKER))
        (INITVARS (NOTEPAD.STYLES))
        (GLOBALVARS NOTEPAD.STYLES)
        (BITMAPS SPOTMARKER NOTEPAD.DEFAULT.BRUSH NOTEPAD.DEFAULT.MASK)
        (GLOBALVARS SPOTMARKER NOTEPAD.DEFAULT.BRUSH NOTEPAD.DEFAULT.MASK)
        (CURSORS CIRCLE.CENTER CIRCLE.EDGE ELLIPSE.CENTER ELLIPSE.SEMI.MAJOR ELLIPSE.SEMI.MINOR CURVE.KNOT)
        (GLOBALVARS CIRCLE.CENTER CIRCLE.EDGE ELLIPSE.CENTER ELLIPSE.SEMI.MAJOR ELLIPSE.SEMI.MINOR CURVE.KNOT)
        (FILES (FROM VALUEOF LISPUSERSDIRECTORIES)
             EDITBITMAP READNUMBER)
        (FILEPKGCOMS NOTEPADSTYLE)
        [P (ADD.NOTEPAD.TO.BACKGROUND.MENU)
           (FONTCREATE NOTEPAD.DEFAULT.FONT)
           (COND ((NULL NOTEPAD.STYLES)
                  (FILESLOAD NOTEPAD-CORESTYLES]
        (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)
                                                                              (NLAML READ.NOTEPAD.STYLE)
                                                                              (LAMA])

(DEFINEQ
```

(ACQUIRE.STYLE
```
  [LAMBDA (BITS.PER.PIXEL NAME.PLEASE)                          (* DAHJr "26-JAN-83 14:29")
    (PROG (NAME.MENU NAME)
          (SETQ NAME.MENU (create MENU
                                  TITLE _ "Styles"
                                  ITEMS _ (for ELEMENT in NOTEPAD.STYLES when (EQ (LISTGET (CDR ELEMENT)
                                                                                           'BITS.PER.PIXEL)
                                                                                   BITS.PER.PIXEL)
                                             collect (CAR ELEMENT))
                                  CENTERFLG _ T
                                  CHANGEOFFSETFLG _ T))
          (SETQ NAME (MENU NAME.MENU))
          (RETURN (COND
                    [NAME (COND
                            (NAME.PLEASE NAME)
                            (T (CDR (FASSOC NAME NOTEPAD.STYLES]
                    (T NIL])
```

(ACQUIRE.SYMMETRY
```
  [LAMBDA (MESSAGE)                                             (* edited%: "17-DEC-82 10:36")
    (PROG (MENU)
          (SETQ MENU (create MENU
                             TITLE _ MESSAGE
                             ITEMS _ '((none NIL)
                                       LEFT/RIGHT UP/DOWN 4-FOLD 8-FOLD)
                             CENTERFLG _ T
                             CHANGEOFFSETFLG _ T))
          (RETURN (MENU MENU])
```

(ADD.NOTEPAD.TO.BACKGROUND.MENU
```
  [LAMBDA NIL                                                   (* DAHJr " 7-APR-83 20:09")
    (NOTEPAD.ADD.TO.BACKGROUND.MENU (LIST 'Notepad '(NOTEPAD)
                                          "Opens a NOTEPAD window"))

        (* don't add the color notepad unless color display is on the machine.
        But since we can't test for the machine having a color board, only do it if it is actually on.)

    (AND (COLORDISPLAYP)
         (NOTEPAD.ADD.TO.BACKGROUND.MENU (LIST 'Color% Notepad '(NOTEPAD NIL T)
                                               "Opens a NOTEPAD window for color"])
```

(BITMAP.INTERIOR
```
  [LAMBDA (BITMAP PT)                                           (* rrb "19-JAN-83 18:53")

        (* returns a bitmap which has all 1's at all points that are the same value as PT and touch it or another point of the same
        value that touches it.)

    (PROG ((WIDTH (BITMAPWIDTH BITMAP))
           (HEIGHT (BITMAPHEIGHT BITMAP))
           INTERIOR
           (X (fetch (POSITION XCOORD) of PT))
           (Y (fetch (POSITION YCOORD) of PT))
           (NBITS (BITSPERPIXEL BITMAP))
           FROMVALUE MAXVAL)
          (SETQ MAXVAL (SUB1 (EXPT 2 NBITS)))
          (SETQ FROMVALUE (BITMAPBIT BITMAP X Y))
          (SETQ INTERIOR (BITMAPCREATE WIDTH HEIGHT NBITS))
          (EXTEND.AREA BITMAP INTERIOR X Y (SUB1 WIDTH)
                 (SUB1 HEIGHT)
                 FROMVALUE MAXVAL)
          (RETURN INTERIOR])
```

(DELDEF.NOTEPAD.STYLE
```
  [LAMBDA (NAME TYPE)                                           (* DAHJr "26-JAN-83 09:46")
    (SETQ NOTEPAD.STYLES (DREMOVE (FASSOC NAME NOTEPAD.STYLES)
                                  NOTEPAD.STYLES])
```

(DISTANCE
```
  [LAMBDA (P1 P2)                                               (* edited%: "18-OCT-82 18:04")
    (PROG (DX DY)
          (SETQ DX (IDIFFERENCE (fetch (POSITION XCOORD) of P1)
                                (fetch (POSITION XCOORD) of P2)))
          (SETQ DY (IDIFFERENCE (fetch (POSITION YCOORD) of P1)
                                (fetch (POSITION YCOORD) of P2)))
          (RETURN (SQRT (IPLUS (ITIMES DX DX)
                               (ITIMES DY DY])
```

(DUMP.NOTEPAD.STYLE
```
  [LAMBDA (NAME)                                                (* DAHJr "26-JAN-83 10:14")
    (PROG (NOTEPAD.STYLE)
          (SETQ NOTEPAD.STYLE (GETDEF.NOTEPAD.STYLE NAME))
```

```
            (PRINT (LIST 'READ.NOTEPAD.STYLE NAME NOTEPAD.STYLE.REPRESENTATION.NUMBER))
            (HPRINT NOTEPAD.STYLE NIL T)
            (RETURN])
```

(**ERASE.REGION**
```
  [LAMBDA (WINDOW REGION)                                    (* edited%: "13-DEC-82 15:51")
    (BITBLT NIL 0 0 WINDOW (fetch (REGION LEFT) of REGION)
            (fetch (REGION BOTTOM) of REGION)
            (fetch (REGION WIDTH) of REGION)
            (fetch (REGION HEIGHT) of REGION)
            'TEXTURE
            'REPLACE WHITESHADE])
```

(**EXTEND.AREA**
```
  [LAMBDA (DEFINING.BITMAP AREA.BITMAP X Y MAXX MAXY FROMVALUE TOVALUE)
                                                             (* rrb "19-JAN-83 19:21")
    (DECLARE (LOCALVARS . T))
    (COND
      ((TEST.AND.SET DEFINING.BITMAP AREA.BITMAP X Y FROMVALUE TOVALUE)
       (EXTEND.AREA.X DEFINING.BITMAP AREA.BITMAP X Y MAXX MAXY FROMVALUE TOVALUE])
```

(**EXTEND.AREA.X**
```
  [LAMBDA (DEFINING.BITMAP AREA.BITMAP X Y MAXX MAXY FROMVAL TOVAL)
                                                             (* rrb "19-JAN-83 19:21")
    (DECLARE (LOCALVARS . T))
    (PROG (LEFT RIGHT)
          (SETQ LEFT X)
          (for I from (SUB1 X) to 0 by -1 while (TEST.AND.SET DEFINING.BITMAP AREA.BITMAP I Y FROMVAL TOVAL)
             do (SETQ LEFT I))
          (SETQ RIGHT X)
          (for I from (ADD1 X) to MAXX while (TEST.AND.SET DEFINING.BITMAP AREA.BITMAP I Y FROMVAL TOVAL)
             do (SETQ RIGHT I))
          (for I from LEFT to RIGHT unless (EQ I X) do (EXTEND.AREA.Y DEFINING.BITMAP AREA.BITMAP I Y MAXX MAXY
                                                        FROMVAL TOVAL))
          (RETURN])
```

(**EXTEND.AREA.Y**
```
  [LAMBDA (DEFINING.BITMAP AREA.BITMAP X Y MAXX MAXY FROMVAL TOVAL)
                                                             (* rrb "19-JAN-83 19:21")
    (DECLARE (LOCALVARS . T))
    (PROG (BOTTOM TOP)
          (SETQ BOTTOM Y)
          (for I from (SUB1 Y) to 0 by -1 while (TEST.AND.SET DEFINING.BITMAP AREA.BITMAP X I FROMVAL TOVAL)
             do (SETQ BOTTOM I))
          (SETQ TOP Y)
          (for I from (ADD1 Y) to MAXY while (TEST.AND.SET DEFINING.BITMAP AREA.BITMAP X I FROMVAL TOVAL)
             do (SETQ TOP I))
          (for I from BOTTOM to TOP unless (EQ I Y) do (EXTEND.AREA.X DEFINING.BITMAP AREA.BITMAP X I MAXX MAXY
                                                       FROMVAL TOVAL))
          (RETURN])
```

(**GET.BITMAP**
```
  [LAMBDA (CHARACTERISTICS PROP)                             (* edited%: "17-DEC-82 12:43")
    (PROG (BITMAPS)
          (SETQ BITMAPS (LISTGET CHARACTERISTICS PROP))
          (RETURN (COND
                    ((LISTP BITMAPS)
                     (CAR BITMAPS))
                    (T BITMAPS])
```

(**GET.NOTEPAD.BITMAP**
```
  [LAMBDA (WINDOW)                                           (* edited%: "18-OCT-82 10:10")
    (DSPDESTINATION NIL (WINDOWPROP WINDOW 'DSP))
```

(**GET.NOTEPAD.REGION**
```
  [LAMBDA (WINDOW)                                           (* rrb "29-JAN-83 14:22")
    (PROG (REGION LEFT BOTTOM RIGHT TOP CHARACTERISTICS GRID)
          (SETQ REGION (GET.WINDOW.REGION WINDOW))
          (SETQ LEFT (fetch (REGION LEFT) of REGION))
          (SETQ BOTTOM (fetch (REGION BOTTOM) of REGION))
          (SETQ RIGHT (IPLUS LEFT (fetch (REGION WIDTH) of REGION)))
          (SETQ TOP (IPLUS BOTTOM (fetch (REGION HEIGHT) of REGION)))
          (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
          (COND
            [(LISTGET CHARACTERISTICS 'USE.GRID.TO.DRAW)
             (SETQ GRID (LISTGET CHARACTERISTICS 'GRID.DEFINITION))
             (SETQ LEFT (NOTEPAD.ON.GRID.X LEFT (fetch (REGION LEFT) of GRID)
                                           (fetch (REGION WIDTH) of GRID)))
             (SETQ RIGHT (NOTEPAD.ON.GRID.X RIGHT (fetch (REGION LEFT) of GRID)
                                            (fetch (REGION WIDTH) of GRID)))
```

```
                    (SETQ BOTTOM (NOTEPAD.ON.GRID.X BOTTOM (fetch (REGION BOTTOM) of GRID)
                                        (fetch (REGION HEIGHT) of GRID)))
                    (SETQ TOP (NOTEPAD.ON.GRID.X TOP (fetch (REGION BOTTOM) of GRID)
                                        (fetch (REGION HEIGHT) of GRID)))
                    (RETURN (create REGION
                                    LEFT _ LEFT
                                    BOTTOM _ BOTTOM
                                    WIDTH _ (IDIFFERENCE RIGHT LEFT)
                                    HEIGHT _ (IDIFFERENCE TOP BOTTOM]
                (T (RETURN REGION])
```

(**GET.SYMMETRIC.BITMAPS**
```
  [LAMBDA (CHARACTERISTICS PROP)                              (* edited%: "17-DEC-82 15:33")
    (PROG (BITMAPS BM1 BM2 BM3 BM4 BM5 BM6 BM7 BM8 NEW.BITMAPS)
          (SETQ BITMAPS (LISTGET CHARACTERISTICS PROP))
          (RETURN (COND
                      ((LISTP BITMAPS)
                       BITMAPS)
                      (T (SETQ BM1 BITMAPS)
                         (SETQ BM2 (INVERT.BITMAP.HORIZONTALLY BM1))
                         (SETQ BM3 (INVERT.BITMAP.VERTICALLY BM1))
                         (SETQ BM4 (INVERT.BITMAP.HORIZONTALLY BM3))
                         (SETQ BM5 (INVERT.BITMAP.DIAGONALLY BM1))
                         (SETQ BM6 (INVERT.BITMAP.HORIZONTALLY BM5))
                         (SETQ BM7 (INVERT.BITMAP.VERTICALLY BM5))
                         (SETQ BM8 (INVERT.BITMAP.HORIZONTALLY BM7))
                         (SETQ NEW.BITMAPS (LIST BM1 BM2 BM3 BM4 BM5 BM6 BM7 BM8))
                         (LISTPUT CHARACTERISTICS PROP NEW.BITMAPS)
                         NEW.BITMAPS])
```

(**GETDEF.NOTEPAD.STYLE**
```
  [LAMBDA (NAME TYPE)                                         (* DAHJr "26-JAN-83 09:44")
    (CDR (FASSOC NAME NOTEPAD.STYLES])
```

(**GETPOSITION.RUBBERBAND**
```
  [LAMBDA (STARTPOSITION WINDOW)                              (* rrb "27-DEC-82 16:42")
                                                             (* gets the other end of a line via a rubberband prompting)
    (PROG [CHARACTERISTICS GRID X0 GX0 GY0 GDX GDY Y0 DS (COLORDS (WINDOWPROP WINDOW 'INCOLOR]
          (SETQ X0 (fetch (POSITION XCOORD) of STARTPOSITION))
          (SETQ Y0 (fetch (POSITION YCOORD) of STARTPOSITION))
          [COND
              ((LISTGET (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
                        'USE.GRID)
               (SETQ GRID (LISTGET CHARACTERISTICS 'GRID.DEFINITION))
               (SETQ GX0 (fetch (REGION LEFT) of GRID))
               (SETQ GY0 (fetch (REGION BOTTOM) of GRID))
               (SETQ GDX (fetch (REGION WIDTH) of GRID))
               (SETQ GDY (fetch (REGION HEIGHT) of GRID]
          [SETQ DS (COND
                      (COLORDS)
                      (T (WINDOWPROP WINDOW 'DSP]
          (RETURN (COND
                      (COLORDS (RESETFORM (CHANGECURSORSCREEN (COLORSCREENBITMAP))
                                      (GETPOSITION.RUBBERBAND1 DS X0 Y0 GX0 GY0 GDX GDY)))
                      (T (GETPOSITION.RUBBERBAND1 DS X0 Y0 GX0 GY0 GDX GDY])
```

(**GETPOSITION.RUBBERBAND1**
```
  [LAMBDA (WHERE X0 Y0 GRIDX0 GRIDY0 GRIDDX GRIDDY)           (* rrb "27-DEC-82 17:45")
    (PROG (X1 Y1 NEW.X1 NEW.Y1 DONE DOWN)
          [until DONE do (GETMOUSESTATE)
                          (COND
                              ((LASTMOUSESTATE (NOT UP))
                               (SETQ DOWN T))
                              ((AND X1 DOWN (LASTMOUSESTATE UP))
                               (SETQ DONE T))
                              (T (SETQ NEW.X1 (LASTMOUSEX WHERE))
                                 (SETQ NEW.Y1 (LASTMOUSEY WHERE))
                                 [COND
                                     (GRIDX0 (SETQ NEW.X1 (NOTEPAD.ON.GRID.X NEW.X1 GRIDX0 GRIDDX))
                                             (SETQ NEW.Y1 (NOTEPAD.ON.GRID.X NEW.Y1 GRIDY0 GRIDDY]
                                 (COND
                                     ((OR (NEQ X1 NEW.X1)
                                          (NEQ Y1 NEW.Y1))
                                      (COND
                                          (X1 (DRAWLINE X0 Y0 X1 Y1 1 'INVERT WHERE)))
                                      (SETQ X1 NEW.X1)
                                      (SETQ Y1 NEW.Y1)
                                      (DRAWLINE X0 Y0 X1 Y1 1 'INVERT WHERE]
          (DRAWLINE X0 Y0 X1 Y1 1 'INVERT WHERE)
          (RETURN (create POSITION
                          XCOORD _ X1
                          YCOORD _ Y1])
```

(MARK.SPOT
```
  [LAMBDA (X/POSITION Y WINDOW)                              (* rrb "14-JAN-83 15:40")
    (PROG [X WIDTH HEIGHT (COLORDS (WINDOWPROP WINDOW 'INCOLOR]
          (COND
            ((POSITIONP X/POSITION)
             (SETQ X (fetch (POSITION XCOORD) of X/POSITION))
             (SETQ Y (fetch (POSITION YCOORD) of X/POSITION)))
            (T (SETQ X X/POSITION)))
          (SETQ WIDTH (BITMAPWIDTH SPOTMARKER))
          (SETQ HEIGHT (BITMAPHEIGHT SPOTMARKER))
          (BITBLT (COND
                    [COLORDS (COND
                              ((AND (BITMAPP COLORSPOTMARKER)
                                    (EQ (BITSPERPIXEL COLORSPOTMARKER)
                                        (COLORNUMBERBITSPERPIXEL)))
                               COLORSPOTMARKER)
                              (T (SETQ COLORSPOTMARKER (COLORIZEBITMAP SPOTMARKER 0 (MAXIMUMCOLOR)
                                                                         (COLORNUMBERBITSPERPIXEL]
                    (T SPOTMARKER))
                  0 0 (OR COLORDS WINDOW)
                  (IDIFFERENCE X (IQUOTIENT WIDTH 2))
                  (IDIFFERENCE Y (IQUOTIENT HEIGHT 2))
                  WIDTH HEIGHT 'INPUT 'INVERT])
```

(MASKED.BRUSH
```
  [LAMBDA (WINDOW MASK)                                      (* edited%: " 5-DEC-82 15:00")
    (PROG (BEEN.DOWN DONE POSITION BUFFER.BITMAP LEFT BOTTOM WIDTH HEIGHT NEW.LEFT NEW.BOTTOM BRUSH)
          (SETQ WIDTH (fetch (BITMAP BITMAPWIDTH) of MASK))
          (SETQ HEIGHT (fetch (BITMAP BITMAPHEIGHT) of MASK))
          (SETQ BUFFER.BITMAP (BITMAPCREATE WIDTH HEIGHT))
          (printout PROMPTWINDOW T "Place mask over desired brush area")
          [until DONE do (GETMOUSESTATE)
                (COND
                  ((AND BEEN.DOWN (MOUSESTATE UP))        (* restore bitmap)
                   [COND
                     (LEFT (BITBLT BUFFER.BITMAP 0 0 WINDOW LEFT BOTTOM WIDTH HEIGHT 'INPUT
                                   'REPLACE]
                   (SETQ DONE T))
                  (T (COND
                       ((MOUSESTATE (NOT UP))
                        (SETQ BEEN.DOWN T)))
                     (SETQ NEW.LEFT (LASTMOUSEX WINDOW))
                     (SETQ NEW.BOTTOM (LASTMOUSEY WINDOW))
                     (COND
                       ((OR (NEQ NEW.LEFT LEFT)
                            (NEQ NEW.BOTTOM BOTTOM))       (* restore bitmap)
                        [COND
                          (LEFT (BITBLT BUFFER.BITMAP 0 0 WINDOW LEFT BOTTOM WIDTH HEIGHT
                                        'INPUT
                                        'REPLACE]
                        (SETQ LEFT NEW.LEFT)
                        (SETQ BOTTOM NEW.BOTTOM)
                        (BITBLT WINDOW LEFT BOTTOM BUFFER.BITMAP 0 0 WIDTH HEIGHT 'INPUT 'REPLACE)
                        (BITBLT MASK 0 0 WINDOW LEFT BOTTOM WIDTH HEIGHT 'INPUT 'PAINT]
          (SETQ BRUSH (BITMAPCOPY BUFFER.BITMAP))
          (BITBLT MASK 0 0 BRUSH 0 0 NIL NIL 'INVERT 'ERASE)
          (RETURN BRUSH])
```

(MOVE.BITMAP
```
  [LAMBDA (WINDOW BITMAP OPERATION)                         (* rrb "22-DEC-82 11:28")
    (PROG [BUFFER.BITMAP WIDTH HEIGHT (COLORDS (WINDOWPROP WINDOW 'INCOLOR]
          (SETQ WIDTH (BITMAPWIDTH BITMAP))
          (SETQ HEIGHT (BITMAPHEIGHT BITMAP))
          [SETQ BUFFER.BITMAP (BITMAPCREATE WIDTH HEIGHT (COND
                                                           (COLORDS (COLORNUMBERBITSPERPIXEL))
                                                           (T 1]
          (printout PROMPTWINDOW T "Indicate where to copy area to")
          (COND
            (COLORDS (RESETFORM (CHANGECURSORSCREEN (COLORSCREENBITMAP))
                      (MOVE.BITMAP1 COLORDS BUFFER.BITMAP WIDTH HEIGHT)))
            (T (MOVE.BITMAP1 (WINDOWPROP WINDOW 'DSP)
                     BUFFER.BITMAP WIDTH HEIGHT)))
          (RETURN])
```

(MOVE.BITMAP1
```
  [LAMBDA (DS BUFFER.BITMAP WIDTH HEIGHT)                   (* DAHJr "31-MAY-83 12:16")
    (PROG (DONE DOWN LEFT BOTTOM NEW.LEFT NEW.BOTTOM)
          (until DONE do (GETMOUSESTATE)
                (COND
                  ((LASTMOUSESTATE (NOT UP))
                   (SETQ DOWN T)))
                (COND
                  ((AND DOWN (LASTMOUSESTATE UP))
```

```
                                     (SETQ DONE T))
                                    (T (SETQ NEW.LEFT (LASTMOUSEX DS))
                                       (SETQ NEW.BOTTOM (LASTMOUSEY DS))
                                       (COND
                                          ((OR (NEQ NEW.LEFT LEFT)
                                               (NEQ NEW.BOTTOM BOTTOM))
                                           [COND
                                              (LEFT (BITBLT BUFFER.BITMAP 0 0 DS LEFT BOTTOM WIDTH HEIGHT 'INPUT
                                                            'REPLACE]
                                           (SETQ LEFT NEW.LEFT)
                                           (SETQ BOTTOM NEW.BOTTOM)
                                           (BITBLT WINDOW LEFT BOTTOM BUFFER.BITMAP 0 0 WIDTH HEIGHT 'INPUT 'REPLACE)
                                           (BITBLT BITMAP 0 0 DS LEFT BOTTOM WIDTH HEIGHT 'INPUT OPERATION]
```

(**NOTEPAD**
```
  [LAMBDA (BITMAP INCOLORFLG)                                        (* rrb "29-JAN-83 14:30")
    (NOTEPAD.CREATE BITMAP (COND
                            (INCOLORFLG (PROMPTPRINT "Making sure the color functions are loaded...")
                                        (FILESLOAD LLCOLOR COLOR HLCOLOR)
                                        (CLRPROMPT)
                                        (SELECTQ (MACHINETYPE)
                                            (DOLPHIN                   (* make sure color is on in 4 bit mode)
                                                    (COLORDISPLAY T 4)
                                                  T)
                                            (DORADO                    (* make sure color is on)
                                                    (COND
                                                       ((COLORDISPLAYP))
                                                       (T
```

            (* used to turn the display on but this hangs the system if the machine doesn't have a color board.
            (COLORDISPLAY T (OR (SMALLP INCOLORFLG) 8)))

```
                                                          (PROMPTPRINT " You will have to turn on the color
                                                                    display before you will see anything. ")))
                                              T)
                              NIL])
```

(**NOTEPAD.ADD.TO.BACKGROUND.MENU**
```
  [LAMBDA (ITEM)                                                     (* rrb "22-FEB-83 14:40")
                                                                     (* adds an item to the background menu.)
    (DECLARE (GLOBALVARS BackgroundMenu BackgroundMenuCommands))
    (COND
       ((FASSOC (CAR ITEM)
                BackgroundMenuCommands))
       (T (SETQ BackgroundMenuCommands (NCONC1 BackgroundMenuCommands ITEM))
          (SETQ BackgroundMenu])
```

(**NOTEPAD.BUTTONEVENTFN**
```
  [LAMBDA (WINDOW)                                                   (* edited%: " 4-DEC-82 17:10")
    (COND
       ((INSIDEP (DSPCLIPPINGREGION NIL (WINDOWPROP WINDOW 'DSP))
                 (LASTMOUSEX WINDOW)
                 (LASTMOUSEY WINDOW))
        (NOTEPAD.BUTTONFN WINDOW))
       (T (NOTEPAD.TITLEBUTTONFN WINDOW])
```

(**NOTEPAD.BUTTONFN**
```
  [LAMBDA (WINDOW)                                                   (* DAHJr " 4-APR-83 16:06")
    (PROG (OPERATION CHARACTERISTICS DSP COMMAND.MENU COMMAND REGION BRUSH NEW.BRUSH FONT MASK NEW.MASK ORIGIN
                DX DY USE.GRID USE.MASK SHADE NEW.SHADE P1 P2 P3 POSITION OLD.BITMAP NEW.BITMAP TEXT NAME
                OBJECT TEXTWIDTH)
          (SETQ OPERATION (NOTEPAD.OPERATION))
          (SETQ COMMAND.MENU (NOTEPAD.COMMAND.MENU))            (* USES NOTEPAD.COMMAND.MENU.CREATE)
          (SETQ COMMAND (MENU COMMAND.MENU))
          (SELECTQ COMMAND
              ((NIL -- -trajectories- -objects/editing- -styles-)
                 NIL)
              (* -trajectories-)
              (SKETCH (NOTEPAD.SKETCH WINDOW OPERATION))
              (LINE (NOTEPAD.LINE WINDOW OPERATION))
              (CIRCLE (NOTEPAD.CIRCLE WINDOW OPERATION))
              (ELLIPSE (NOTEPAD.ELLIPSE WINDOW OPERATION))
              (OPEN.CURVE (NOTEPAD.CURVE WINDOW OPERATION))
              (CLOSED.CURVE (NOTEPAD.CURVE WINDOW OPERATION T))
              (* -objects/editing-)
              (TEXT (NOTEPAD.TEXT WINDOW OPERATION))
              (EDIT.RECTANGLE
                   (NOTEPAD.EDIT.RECTANGLE WINDOW))
              (SHADE.RECTANGLE
                   (NOTEPAD.SHADE.RECTANGLE WINDOW OPERATION))
              (FILL (NOTEPAD.FILL WINDOW OPERATION))
              (COPY.FROM.SCREEN
                   (NOTEPAD.COPY.FROM.SCREEN WINDOW OPERATION))
```

```
                    (NAMED.OBJECT (NOTEPAD.NAMED.OBJECT WINDOW OPERATION))
                    (* -styles-)
                    (DEFINE.BRUSH (NOTEPAD.DEFINE.BRUSH WINDOW))
                    (EDIT.BRUSH (EVAL.AS.PROCESS (LIST (FUNCTION NOTEPAD.EDIT.BRUSH)
                                                       (KWOTE WINDOW))))
                    (BRUSH=COOKIE.CUT.WITH.MASK
                        (NOTEPAD.COOKIE.CUT WINDOW))
                    (USE.MASK (NOTEPAD.USE.MASK WINDOW))
                    (DEFINE.MASK (NOTEPAD.DEFINE.MASK WINDOW))
                    (EDIT.MASK (EVAL.AS.PROCESS (LIST (FUNCTION NOTEPAD.EDIT.MASK)
                                                      (KWOTE WINDOW)
                                                      (KWOTE WINDOW))))
                    (MASK=OUTLINE.OF.BRUSH
                        (NOTEPAD.MASK=BRUSH.OUTLINE WINDOW))
                    (DEFINE.FONT (NOTEPAD.DEFINE.FONT WINDOW))
                    (DEFINE.GRID (NOTEPAD.DEFINE.GRID WINDOW))
                    (USE.GRID (NOTEPAD.USE.GRID WINDOW))
                    (USE.SYMMETRY (NOTEPAD.USE.SYMMETRY WINDOW))
                    (USE.SYMMETRIC.BRUSH/MASK
                        (NOTEPAD.USE.SYMMETRIC.BRUSH/MASK WINDOW))
                    (DEFINE.POINT.OF.SYMMETRY
                        (NOTEPAD.POINT.OF.SYMMETRY WINDOW))
                    (EDIT.SHADE (NOTEPAD.EDIT.SHADE WINDOW))
                    (SAVE.STYLE (SAVE.STYLE WINDOW))
                    (RESTORE.STYLE
                        (NOTEPAD.RESTORE.STYLE WINDOW))
                    (DELETE.STYLE (NOTEPAD.DELETE.STYLE WINDOW))
                    (SHOULDNT (CONCAT "Unrecognized COMMAND in NOTEPAD.BUTTONFN: " COMMAND])
```

(NOTEPAD.CIRCLE

```
  [LAMBDA (WINDOW OPERATION)                                (* rrb "14-JAN-83 09:56")
    (PROG (CHARACTERISTICS BRUSH P1 P2)
          (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
          (SETQ BRUSH (GET.BITMAP CHARACTERISTICS 'BRUSH.DEFINITION))
          (printout PROMPTWINDOW T "Indicate center of circle")
          (SETQ P1 (NOTEPAD.GETPOSITION WINDOW CIRCLE.CENTER))
          (MARK.SPOT P1 NIL WINDOW)
          (printout PROMPTWINDOW T "Indicate a point of the circumference of the circle")
          (SETQ P2 (NOTEPAD.GETPOSITION WINDOW CIRCLE.EDGE))    (* erase the center pt.)
          (MARK.SPOT P1 NIL WINDOW)
          (NOTEPAD.SETUP.TO.PAINT WINDOW OPERATION)
          (DRAWCIRCLE (fetch (POSITION XCOORD) of P1)
                 (fetch (POSITION YCOORD) of P1)
                 (DISTANCE P1 P2)
                 'PAINT.AT.POSSIBLE.POINT NIL WINDOW])
```

(NOTEPAD.COMMAND.MENU

```
  [LAMBDA NIL                                               (* edited%: " 6-DEC-82 21:07")
    (COND
        (NOTEPAD.COMMAND.MENU)
        (T (SETQ NOTEPAD.COMMAND.MENU (NOTEPAD.COMMAND.MENU.CREATE])
```

(NOTEPAD.COMMAND.MENU.CREATE

```
  [LAMBDA NIL                                               (* edited%: "18-DEC-82 08:50")
    (PROG (OPERATIONS STYLES)                               (* AFTER EDITING RESET WITH%:
                                                            (SETQ NOTEPAD.COMMAND.MENU))

          (SETQ OPERATIONS
           '(-trajectories- SKETCH LINE CIRCLE ELLIPSE OPEN.CURVE CLOSED.CURVE -objects/editing- TEXT
                 NAMED.OBJECT COPY.FROM.SCREEN SHADE.RECTANGLE FILL EDIT.RECTANGLE -- -- -- -- --))
          (SETQ STYLES
           '(-styles- DEFINE.BRUSH EDIT.BRUSH BRUSH=COOKIE.CUT.WITH.MASK USE.MASK DEFINE.MASK EDIT.MASK
                 MASK=OUTLINE.OF.BRUSH USE.GRID DEFINE.GRID USE.SYMMETRY DEFINE.POINT.OF.SYMMETRY
                 USE.SYMMETRIC.BRUSH/MASK EDIT.SHADE DEFINE.FONT -- SAVE.STYLE RESTORE.STYLE DELETE.STYLE))
          (RETURN (create MENU
                     TITLE _ "Adding and editing"
                     MENUCOLUMNS _ 2
                     ITEMS _ (for OPERATION in OPERATIONS as STYLE in STYLES join (LIST OPERATION STYLE))
                     CHANGEOFFSETFLG _ T])
```

(NOTEPAD.COOKIE.CUT

```
  [LAMBDA (WINDOW)                                          (* edited%: "17-DEC-82 16:15")
    (PROG (CHARACTERISTICS NEW.BRUSH)
          (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
          (SETQ MASK (GET.BITMAP CHARACTERISTICS 'MASK.DEFINITION))
          (SETQ NEW.BRUSH (MASKED.BRUSH WINDOW MASK))
          (LISTPUT CHARACTERISTICS 'BRUSH.DEFINITION NEW.BRUSH)
          (printout PROMPTWINDOW T "Brush defined"])
```

(NOTEPAD.COPY.FROM.SCREEN

```
  [LAMBDA (WINDOW OPERATION)                                (* rrb " 4-JAN-83 17:00")
    (PROG (REGION NEW.BITMAP)
          (printout PROMPTWINDOW T "Indicate a region of the screen to be copied")
```

```
              [SETQ NEW.BITMAP (COND
                                  ((WINDOWPROP WINDOW 'INCOLOR)
                                   (PICKUP.SCREEN.BITMAP (GETCOLORREGION)
                                          (COLORSCREENBITMAP)))
                                  (T (PICKUP.SCREEN.BITMAP (GETREGION)
                                            (SCREENBITMAP]
              (RETURN (MOVE.BITMAP WINDOW NEW.BITMAP OPERATION])
```

(**NOTEPAD.CURVE**
```
  [LAMBDA (WINDOW OPERATION CLOSED)                          (* DAHJr " 4-APR-83 16:53")
    (PROG (CHARACTERISTICS BRUSH REGION P1 P2 PTS DONE)
          (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
          (SETQ BRUSH (GET.BITMAP CHARACTERISTICS 'BRUSH.DEFINITION))
          [SETQ REGION (DSPCLIPPINGREGION NIL (COND
                                              ((WINDOWPROP WINDOW 'INCOLOR))
                                              (T (WINDOWPROP WINDOW 'DSP]
          (printout PROMPTWINDOW T "Indicate first point of curve")
          (SETQ P1 (NOTEPAD.GETPOSITION WINDOW CURVE.KNOT))
          (MARK.SPOT P1 NIL WINDOW)
          (printout PROMPTWINDOW " (indicate last point by holding down left shift key)")
          (printout PROMPTWINDOW ", point 2")
          (SETQ P2 (NOTEPAD.GETPOSITION WINDOW CURVE.KNOT))
          (SETQ DONE (KEYDOWNP 'LSHIFT))
          (MARK.SPOT P2 NIL WINDOW)
          (SETQ PTS (LIST P1 P2))
          (until DONE as I from 3 do (printout PROMPTWINDOW ", " I)
                          (SETQ P2 (NOTEPAD.GETPOSITION WINDOW CURVE.KNOT))
                          (SETQ DONE (KEYDOWNP 'LSHIFT))
                          (NCONC1 PTS P2)
                          (MARK.SPOT P2 NIL WINDOW))
          (for PT in PTS do (MARK.SPOT PT NIL WINDOW))
          (NOTEPAD.SETUP.TO.PAINT WINDOW OPERATION)
          (DRAWCURVE PTS CLOSED 'PAINT.AT.POSSIBLE.POINT NIL WINDOW])
```

(**NOTEPAD.DEFINE.BRUSH**
```
  [LAMBDA (WINDOW)                                           (* rrb "20-DEC-82 15:30")
    (PROG (CHARACTERISTICS BRUSH)
          (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
          (printout PROMPTWINDOW T "Indicate a region to be used as the brush")
          [SETQ BRUSH (COND
                         ((WINDOWPROP WINDOW 'INCOLOR)
                          (PICKUP.SCREEN.BITMAP (GETCOLORREGION)
                                 (COLORSCREENBITMAP)))
                         (T (PICKUP.SCREEN.BITMAP (GETREGION)
                                   (SCREENBITMAP]
          (LISTPUT CHARACTERISTICS 'BRUSH.DEFINITION BRUSH)
          (printout PROMPTWINDOW T "Brush defined"])
```

(**NOTEPAD.DEFINE.FONT**
```
  [LAMBDA (WINDOW)                                           (* DAHJr " 7-APR-83 20:13")
    (PROG (CHARACTERISTICS FONT)
          (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
          (SETQ FONT (NOTEPAD.READ.FONT))
          (COND
             (FONT (LISTPUT CHARACTERISTICS 'PRINTING.FONT.DEFINITION FONT)))
          (printout PROMPTWINDOW T "Font defined to be " FONT])
```

(**NOTEPAD.HELP**
```
  [LAMBDA NIL                                                (* DAHJr " 4-APR-83 16:01")
    (CLEARW PROMPTWINDOW)
    (printout PROMPTWINDOW "NOTEPAD is menu-driven")
    (printout PROMPTWINDOW T "Selection: ")
    (printout PROMPTWINDOW T 3 "In the title bar: commands affecting the whole")
    (printout PROMPTWINDOW T 3 "Within the window:")
    (printout PROMPTWINDOW T 6 "Using Left button:" " => paint")
    (printout PROMPTWINDOW T 6 "Using Middle button:" " => erase")
    (printout PROMPTWINDOW T 6 "Using Right button:" " window commands"])
```

(**NOTEPAD.READ.FONT**
```
  [LAMBDA NIL                                                (* DAHJr "31-MAY-83 12:12")
    (PROG (LOADED.FONTS COMMAND.MENU COMMAND REGION BRUSH NEW.FONT)
          (SETQ LOADED.FONTS (NOTEPAD.FONTS.IN.CORE))
          [SETQ COMMAND.MENU (create MENU
                                    TITLE _ "Style"
                                    ITEMS _ (NCONC1 (for FONT in LOADED.FONTS collect (LIST FONT (KWOTE FONT)))
                                                'NEW.FONT]
          (SETQ COMMAND (MENU COMMAND.MENU))
          (RETURN (SELECTQ COMMAND
                      (NIL NIL)
                      (NEW.FONT (SETQ NEW.FONT (READ.FROM.PROMPT.WINDOW "New font (FAMILY SIZE FACE):"))
                            (printout PROMPTWINDOW T "Reading font " NEW.FONT " ... ")
                            (FONTCREATE NEW.FONT)
```

```
                                       (printout PROMPTWINDOW "done")
                                    NEW.FONT)
                      COMMAND])
```

(**NOTEPAD.FONTS.IN.CORE**
```
    [LAMBDA NIL                                                 (* edited%: " 4-DEC-82 15:28")
      (for FAMILY in \FONTSINCORE join (for SIZE in (CDR FAMILY)
                                    join (for FACE in (CDR SIZE) when (EQ (CAR (CADR (CADR FACE)))
                                                                             'DISPLAY)
                                        collect (LIST (CAR FAMILY)
                                                      (CAR SIZE)
                                                      (CAR FACE])
```

(**READ.NOTEPAD.STYLE**
```
    [NLAMBDA (NAME VERSION)                                      (* DAHJr "26-JAN-83 10:17")
      (PROG (NOTEPAD.STYLE)
            (SELECTQ VERSION
                (1                                              (* 1%: HPRINTED PROPERTY LIST)
                    (SETQ NOTEPAD.STYLE (HREAD)))
                (SHOULDNT "Unrecognized version number in READ.NOTEPAD.STYLE"))
            (PUTDEF.NOTEPAD.STYLE NAME 'NOTEPAD.STYLES NOTEPAD.STYLE)
            (RETURN])
```

(**NOTEPAD.DEFINE.GRID**
```
    [LAMBDA (WINDOW)                                             (* rrb "27-DEC-82 16:21")
      (PROG [(COLORDS (WINDOWPROP WINDOW 'INCOLOR]
             [LISTPUT (WINDOWPROP WINDOW 'CHARACTERISTICS)
                     'GRID.DEFINITION
                     (COND
                        (COLORDS (RESETFORM (CHANGECURSORSCREEN (COLORSCREENBITMAP))
                                             (NOTEPAD.DEFINE.GRID1 COLORDS)))
                        (T (NOTEPAD.DEFINE.GRID1 WINDOW]
            (printout PROMPTWINDOW T "Grid defined"])
```

(**NOTEPAD.DEFINE.GRID1**
```
    [LAMBDA (WHERE)                                              (* rrb "22-FEB-83 14:47")
                                                                (* prompts for two points and returns a REGION that defines a
                                                                grid.)
      (PROG (ORIGIN P1)
            (printout PROMPTWINDOW T "Indicate an origin for the grid")
            (SETQ ORIGIN (NOTEPAD.GETPOSITION WHERE NIL T))
            (MARK.SPOT ORIGIN NIL WHERE)
            (printout PROMPTWINDOW T "Indicate (1, 1) on the grid")
            (SETQ P1 (NOTEPAD.GETPOSITION WHERE NIL T))
            (MARK.SPOT ORIGIN NIL WHERE)
            (RETURN (create REGION
                            LEFT _ (fetch (POSITION XCOORD) of ORIGIN)
                            BOTTOM _ (fetch (POSITION YCOORD) of ORIGIN)
                            WIDTH _ [MAX 1 (ABS (IDIFFERENCE (fetch (POSITION XCOORD) of P1)
                                                            (fetch (POSITION XCOORD) of ORIGIN]
                            HEIGHT _ (MAX 1 (ABS (IDIFFERENCE (fetch (POSITION YCOORD) of P1)
                                                             (fetch (POSITION YCOORD) of ORIGIN])
```

(**NOTEPAD.DEFINE.MASK**
```
    [LAMBDA (WINDOW)                                             (* rrb "21-DEC-82 15:26")
      (PROG (CHARACTERISTICS MASK)
            (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
            (printout PROMPTWINDOW T "Indicate a region to be used as the brush")
            [SETQ MASK (COND
                        ((WINDOWPROP WINDOW 'INCOLOR)
                         (PICKUP.SCREEN.BITMAP (GETCOLORREGION)
                                (COLORSCREENBITMAP)))
                        (T (PICKUP.SCREEN.BITMAP (GETREGION)
                                (SCREENBITMAP]
            (LISTPUT CHARACTERISTICS 'MASK.DEFINITION MASK)
            (printout PROMPTWINDOW T "Brush defined"])
```

(**NOTEPAD.DELETE.STYLE**
```
    [LAMBDA (WINDOW)                                             (* DAHJr "26-JAN-83 14:27")
      (PROG (BITS.PER.PIXEL NAME NEW.STYLES)
            (SETQ BITS.PER.PIXEL (LISTGET (WINDOWPROP WINDOW 'CHARACTERISTICS)
                                    'BITS.PER.PIXEL))
            (COND
               [NOTEPAD.STYLES (SETQ NAME (ACQUIRE.STYLE BITS.PER.PIXEL T))
                       (COND
                           ((AND NAME (NOTEPAD.CONFIRM (CONCAT "Delete style " NAME)))
                            (DELDEF.NOTEPAD.STYLE NAME)
                            (printout PROMPTWINDOW T "Style " NAME " deleted"]
               (T (printout PROMPTWINDOW T "No styles to delete"]))
```

⟨**NOTEPAD.EDIT.BRUSH**
```
  [LAMBDA (WINDOW)                                          (* DAHJr " 4-APR-83 16:44")
    (PROG (CHARACTERISTICS BRUSH NEW.BRUSH)
          (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
          (SETQ BRUSH (GET.BITMAP CHARACTERISTICS 'BRUSH.DEFINITION))
          (SETQ NEW.BRUSH (EDIT.BITMAP BRUSH))
          (LISTPUT CHARACTERISTICS 'BRUSH.DEFINITION NEW.BRUSH)
          (printout PROMPTWINDOW T "Brush defined"])
```

⟨**NOTEPAD.EDIT.MASK**
```
  [LAMBDA (WINDOW WINDOW)                                   (* edited%: "17-DEC-82 12:43")
    (PROG (CHARACTERISTICS MASK NEW.MASK)
          (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
          (SETQ MASK (GET.BITMAP CHARACTERISTICS 'MASK.DEFINITION))
          (SETQ NEW.MASK (EDIT.BITMAP MASK))
          (LISTPUT CHARACTERISTICS 'MASK.DEFINITION NEW.MASK)
          (printout PROMPTWINDOW T "Mask defined"])
```

⟨**NOTEPAD.EDIT.RECTANGLE**
```
  [LAMBDA (WINDOW)                                          (* rrb "11-AUG-83 12:24")
    (PROG (REGION OLD.BITMAP NEW.BITMAP ALLINFLG)
          (printout PROMPTWINDOW T "Indicate a rectangle to be edited")
          (SETQ REGION (GET.NOTEPAD.REGION WINDOW))
          [COND
            ((SUBREGIONP (DSPCLIPPINGREGION NIL WINDOW)
                     REGION)                                (* region is entirely within window)
             (SETQ ALLINFLG T)
             (SETQ OLD.BITMAP (PICKUP.BITMAP WINDOW REGION)))
            (T                                              (* region is at least partially outside of window Translate the
                                                            region into screen coordinates)
              (SETQ OLD.BITMAP (PICKUP.BITMAP NIL (CREATEREGION (IPLUS (fetch (REGION LEFT) of REGION)
                                                                       (DSPXOFFSET NIL WINDOW))
                                                          (IPLUS (fetch (REGION BOTTOM) of REGION)
                                                                 (DSPYOFFSET NIL WINDOW))
                                                          (fetch (REGION WIDTH) of REGION)
                                                          (fetch (REGION HEIGHT) of REGION]
          (SETQ NEW.BITMAP (EDIT.BITMAP OLD.BITMAP))
          (COND
            ((AND ALLINFLG (EQP (fetch (BITMAP BITMAPWIDTH) of OLD.BITMAP)
                                (fetch (BITMAP BITMAPWIDTH) of NEW.BITMAP))
                  (EQP (fetch (BITMAP BITMAPHEIGHT) of OLD.BITMAP)
                       (fetch (BITMAP BITMAPHEIGHT) of NEW.BITMAP)))
             (ERASE.REGION WINDOW REGION)
             (PUTBACK.BITMAP WINDOW REGION NEW.BITMAP))
            (T (MOVE.BITMAP WINDOW NEW.BITMAP OPERATION])
```

⟨**NOTEPAD.EDIT.SHADE**
```
  [LAMBDA (WINDOW)                                          (* DAHJr " 4-APR-83 16:44")
    (PROG (CHARACTERISTICS SHADE NEW.SHADE COLORDS)
          (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
          (SETQ SHADE (LISTGET CHARACTERISTICS 'SHADE.DEFINITION))
          [COND
            ((SETQ COLORDS (WINDOWPROP WINDOW 'INCOLOR))
             (SETQ NEW.SHADE (RNUMBER))
```

(* set the color of the display stream although it doesn't appear to make any difference it seems like it should.)

```
             (DSPCOLOR NEW.SHADE COLORDS))
            (T (SETQ NEW.SHADE (EDITSHADE SHADE]
          (LISTPUT CHARACTERISTICS 'SHADE.DEFINITION NEW.SHADE)
          (printout PROMPTWINDOW T "Shade redefined"])
```

⟨**NOTEPAD.ELLIPSE**
```
  [LAMBDA (WINDOW OPERATION)                                (* rrb "14-JAN-83 09:57")
    (PROG (CHARACTERISTICS BRUSH P1 P2 P3 ANGLE)
          (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
          (SETQ BRUSH (GET.BITMAP CHARACTERISTICS 'BRUSH.DEFINITION))
          (printout PROMPTWINDOW T "Indicate center of ellipse")
          (SETQ P1 (NOTEPAD.GETPOSITION WINDOW ELLIPSE.CENTER))
          (MARK.SPOT P1 NIL WINDOW)
          (printout PROMPTWINDOW T "Indicate semi-major axis")
          (SETQ P2 (NOTEPAD.GETPOSITION WINDOW ELLIPSE.SEMI.MAJOR))
          (MARK.SPOT P2 NIL WINDOW)
          (printout PROMPTWINDOW T "Indicate semi-minor axis")
          (SETQ P3 (NOTEPAD.GETPOSITION WINDOW ELLIPSE.SEMI.MINOR))
          (MARK.SPOT P2 NIL WINDOW)
          (MARK.SPOT P1 NIL WINDOW)
          (SETQ ANGLE (DIFFERENCE (ATAN (IDIFFERENCE (fetch (POSITION XCOORD) of P2)
                                                     (fetch (POSITION XCOORD) of P1))
                                        (IDIFFERENCE (fetch (POSITION YCOORD) of P2)
                                                     (fetch (POSITION YCOORD) of P1)))
                          90))
          (NOTEPAD.SETUP.TO.PAINT WINDOW OPERATION)
```

```
                    (DRAWELLIPSE (fetch (POSITION XCOORD) of P1)
                            (fetch (POSITION YCOORD) of P1)
                            (DISTANCE P1 P2)
                            (DISTANCE P1 P3)
                            ANGLE
                            'PAINT.AT.POSSIBLE.POINT NIL WINDOW])
```

(**NOTEPAD.LINE**
```
  [LAMBDA (WINDOW OPERATION)                                    (* rrb "14-JAN-83 09:57")
    (PROG (CHARACTERISTICS BRUSH P1 P2)
          (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
          (SETQ BRUSH (GET.BITMAP CHARACTERISTICS 'BRUSH.DEFINITION))
          (printout PROMPTWINDOW T "Indicate start point for the line")
          (SETQ P1 (NOTEPAD.GETPOSITION WINDOW))
          (printout PROMPTWINDOW T "Indicate end point for the line")
          (SETQ P2 (GETPOSITION.RUBBERBAND P1 WINDOW))
          (NOTEPAD.SETUP.TO.PAINT WINDOW OPERATION)
          (DRAWCURVE (LIST P1 P2)
                  NIL
                  'PAINT.AT.POSSIBLE.POINT NIL WINDOW])
```

(**NOTEPAD.MASK=BRUSH.OUTLINE**
```
  [LAMBDA (WINDOW)                                              (* edited%: "17-DEC-82 16:15")
    (PROG (CHARACTERISTICS NEW.MASK)
          (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
          (SETQ BRUSH (GET.BITMAP CHARACTERISTICS 'BRUSH.DEFINITION))
          (SETQ NEW.MASK (BITMAP.EXTERIOR BRUSH))
          (BITBLT NIL 0 0 NEW.MASK 0 0 NIL NIL 'TEXTURE 'INVERT BLACKSHADE)
          (LISTPUT CHARACTERISTICS 'MASK.DEFINITION NEW.MASK)
          (printout PROMPTWINDOW T "Mask defined to an outline of the brush"])
```

(**NOTEPAD.NAMED.OBJECT**
```
  [LAMBDA (WINDOW OPERATION)                                    ; Edited 27-Sep-91 16:42 by jds
    (PROG (NAME OBJECT NEW.BITMAP)
          (SETQ NAME (READ.FROM.PROMPT.WINDOW "Lisp expression to EVAL to get object to add to window: "))
          (SETQ OBJECT (EVALV NAME))
          (COND
             ((BITMAPP OBJECT)
              (SETQ NEW.BITMAP OBJECT)
              (MOVE.BITMAP WINDOW NEW.BITMAP OPERATION))
             ((CURSORP OBJECT)
              (SETQ NEW.BITMAP (fetch (CURSOR CUIMAGE) of OBJECT))
              (MOVE.BITMAP WINDOW NEW.BITMAP OPERATION))
             (T (printout PROMPTWINDOW T "Not a bitmap" OBJECT]))
```

(**NOTEPAD.POINT.OF.SYMMETRY**
```
  [LAMBDA (WINDOW)                                              (* rrb "22-FEB-83 14:47")
    (PROG (CHARACTERISTICS ORIGIN)
          (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
          (printout PROMPTWINDOW T "Indicate the point of symmetry")
          (SETQ ORIGIN (NOTEPAD.GETPOSITION WINDOW))
          (LISTPUT CHARACTERISTICS 'POINT.OF.SYMMETRY.DEFINITION ORIGIN)
          (printout PROMPTWINDOW T "Point of symmetry defined"])
```

(**NOTEPAD.RESTORE.STYLE**
```
  [LAMBDA (WINDOW)                                              (* DAHJr "26-JAN-83 14:05")
    (PROG (BITS.PER.PIXEL NEW.CHARACTERISTICS)
          (SETQ BITS.PER.PIXEL (LISTGET (WINDOWPROP WINDOW 'CHARACTERISTICS)
                                        'BITS.PER.PIXEL))
          (COND
             [NOTEPAD.STYLES (SETQ NEW.CHARACTERISTICS (ACQUIRE.STYLE BITS.PER.PIXEL))
                     (COND
                        (NEW.CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS (COPYALL NEW.CHARACTERISTICS))
                                (printout PROMPTWINDOW T "Style restored"]
             (T (printout PROMPTWINDOW T "No styles to restore"])
```

(**NOTEPAD.SETUP.TO.PAINT**
```
  [LAMBDA (WINDOW OPERATION)                                    (* rrb "27-DEC-82 16:56")
    (PROG (CHARACTERISTICS GRID POINT.OF.SYMMETRY BITMAPS BRUSH MASK)
          (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))

          (* if WINDOW is onto the color screen, its INCOLOR property is a color display stream)

          (SETQ .NOTEPAD.WINDOW (COND
                                   ((WINDOWPROP WINDOW 'INCOLOR))
                                   (T WINDOW)))
          (SETQ .NOTEPAD.OPERATION OPERATION)
          (SETQ .NOTEPAD.USE.SYMMETRIC.BRUSH/MASK (LISTGET CHARACTERISTICS 'USE.SYMMETRIC.BRUSH/MASK))
          (COND
             [.NOTEPAD.USE.SYMMETRIC.BRUSH/MASK (SETQ BITMAPS (GET.SYMMETRIC.BITMAPS CHARACTERISTICS
                                                                       'BRUSH.DEFINITION))
```

```
                            (SETQ .NOTEPAD.BRUSH.1 (CAR BITMAPS))
                            (SETQ .NOTEPAD.BRUSH.2 (CADR BITMAPS))
                            (SETQ .NOTEPAD.BRUSH.3 (CADDR BITMAPS))
                            (SETQ .NOTEPAD.BRUSH.4 (CADDDR BITMAPS))
                            (SETQ .NOTEPAD.BRUSH.5 (CAR (CDDDDR BITMAPS)))
                            (SETQ .NOTEPAD.BRUSH.6 (CADR (CDDDDR BITMAPS)))
                            (SETQ .NOTEPAD.BRUSH.7 (CADDR (CDDDDR BITMAPS)))
                            (SETQ .NOTEPAD.BRUSH.8 (CADDDR (CDDDDR BITMAPS]
                  (T (SETQ BRUSH (GET.BITMAP CHARACTERISTICS 'BRUSH.DEFINITION))
                      (SETQ .NOTEPAD.BRUSH.1 BRUSH)
                      (SETQ .NOTEPAD.BRUSH.2 BRUSH)
                      (SETQ .NOTEPAD.BRUSH.3 BRUSH)
                      (SETQ .NOTEPAD.BRUSH.4 BRUSH)
                      (SETQ .NOTEPAD.BRUSH.5 BRUSH)
                      (SETQ .NOTEPAD.BRUSH.6 BRUSH)
                      (SETQ .NOTEPAD.BRUSH.7 BRUSH)
                      (SETQ .NOTEPAD.BRUSH.8 BRUSH)))
              (SETQ .NOTEPAD.BRUSH.WIDTH (BITMAPWIDTH .NOTEPAD.BRUSH.1))
              (SETQ .NOTEPAD.BRUSH.HEIGHT (BITMAPHEIGHT .NOTEPAD.BRUSH.1))
              (SETQ .NOTEPAD.BRUSH.HALF.WIDTH (RSH .NOTEPAD.BRUSH.WIDTH 1))
              (SETQ .NOTEPAD.BRUSH.HALF.HEIGHT (RSH .NOTEPAD.BRUSH.HEIGHT 1))
              (SETQ .NOTEPAD.USE.GRID (LISTGET CHARACTERISTICS 'USE.GRID.TO.DRAW))
              [COND
                  (.NOTEPAD.USE.GRID (SETQ GRID (LISTGET CHARACTERISTICS 'GRID.DEFINITION))
                          (SETQ .NOTEPAD.GRID.X0 (fetch (REGION LEFT) of GRID))
                          (SETQ .NOTEPAD.GRID.Y0 (fetch (REGION BOTTOM) of GRID))
                          (SETQ .NOTEPAD.GRID.DX (fetch (REGION WIDTH) of GRID))
                          (SETQ .NOTEPAD.GRID.DY (fetch (REGION HEIGHT) of GRID]
              (SETQ .NOTEPAD.USE.MASK (LISTGET CHARACTERISTICS 'USE.MASK))
              [COND
                  (.NOTEPAD.USE.MASK (COND
                                        [.NOTEPAD.USE.SYMMETRIC.BRUSH/MASK (SETQ BITMAPS (GET.SYMMETRIC.BITMAPS
                                                                                            CHARACTERISTICS
                                                                                            'MASK.DEFINITION))
                                            (SETQ .NOTEPAD.MASK.1 (CAR BITMAPS))
                                            (SETQ .NOTEPAD.MASK.2 (CADR BITMAPS))
                                            (SETQ .NOTEPAD.MASK.3 (CADDR BITMAPS))
                                            (SETQ .NOTEPAD.MASK.4 (CADDDR BITMAPS))
                                            (SETQ .NOTEPAD.MASK.5 (CAR (CDDDDR BITMAPS)))
                                            (SETQ .NOTEPAD.MASK.6 (CADR (CDDDDR BITMAPS)))
                                            (SETQ .NOTEPAD.MASK.7 (CADDR (CDDDDR BITMAPS)))
                                            (SETQ .NOTEPAD.MASK.8 (CADDDR (CDDDDR BITMAPS]
                                        (T (SETQ MASK (GET.BITMAP CHARACTERISTICS 'MASK.DEFINITION))
                                            (SETQ .NOTEPAD.MASK.1 MASK)
                                            (SETQ .NOTEPAD.MASK.2 MASK)
                                            (SETQ .NOTEPAD.MASK.3 MASK)
                                            (SETQ .NOTEPAD.MASK.4 MASK)
                                            (SETQ .NOTEPAD.MASK.5 MASK)
                                            (SETQ .NOTEPAD.MASK.6 MASK)
                                            (SETQ .NOTEPAD.MASK.7 MASK)
                                            (SETQ .NOTEPAD.MASK.8 MASK)))
                              (SETQ .NOTEPAD.MASK.WIDTH (BITMAPWIDTH .NOTEPAD.MASK.1))
                              (SETQ .NOTEPAD.MASK.HEIGHT (BITMAPHEIGHT .NOTEPAD.MASK.1))
                              (SETQ .NOTEPAD.MASK.HALF.WIDTH (RSH .NOTEPAD.MASK.WIDTH 1))
                              (SETQ .NOTEPAD.MASK.HALF.HEIGHT (RSH .NOTEPAD.MASK.HEIGHT 1))
                              (SETQ .NOTEPAD.INVERSE.OPERATION (COND
                                                                ((EQ OPERATION 'PAINT)
                                                                 'ERASE)
                                                                (T 'PAINT]
              (SETQ .NOTEPAD.USE.SYMMETRY (LISTGET CHARACTERISTICS 'USE.SYMMETRY))
              [COND
                  (.NOTEPAD.USE.SYMMETRY (SETQ POINT.OF.SYMMETRY (LISTGET CHARACTERISTICS
                                                                     'POINT.OF.SYMMETRY.DEFINITION))
                          (SETQ .NOTEPAD.POSX (fetch (POSITION XCOORD) of POINT.OF.SYMMETRY))
                          (SETQ .NOTEPAD.POSY (fetch (POSITION YCOORD) of POINT.OF.SYMMETRY]
              (SETQ .NOTEPAD.PREVIOUS.MIDX)
              (SETQ .NOTEPAD.PREVIOUS.MIDY])
```

(**NOTEPAD.SHADE.RECTANGLE**
```
  [LAMBDA (WINDOW OPERATION)                                   (* rrb "20-DEC-82 15:03")
     (PROG (CHARACTERISTICS REGION)
           (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
           (printout PROMPTWINDOW T "Indicate rectangle to be PAINTED/ERASED with SHADE")
           (SETQ REGION (GET.NOTEPAD.REGION WINDOW))
           (BITBLT NIL 0 0 (OR (WINDOWPROP WINDOW 'INCOLOR)
                               WINDOW)
                   (fetch (REGION LEFT) of REGION)
                   (fetch (REGION BOTTOM) of REGION)
                   (fetch (REGION WIDTH) of REGION)
                   (fetch (REGION HEIGHT) of REGION)
                   'TEXTURE
                   (COND
                       ((WINDOWPROP WINDOW 'INCOLOR)
                        (SELECTQ OPERATION
                            (PAINT 'REPLACE)
                            OPERATION))
```

```
                              (T OPERATION))
                       (LISTGET CHARACTERISTICS 'SHADE.DEFINITION])
```

(NOTEPAD.SKETCH

```
   [LAMBDA (WINDOW OPERATION)                                    (* rrb "27-DEC-82 15:43")
      (PROG [BRUSH (COLORDS (WINDOWPROP WINDOW 'INCOLOR]
            (SETQ BRUSH (GET.BITMAP (WINDOWPROP WINDOW 'CHARACTERISTICS)
                                    'BRUSH.DEFINITION))
            (NOTEPAD.SETUP.TO.PAINT WINDOW OPERATION)
            (COND
               [COLORDS (RESETFORM (CHANGECURSORSCREEN (COLORSCREENBITMAP))
                               (RESETFORM (SETDISPLAYHEIGHT 8)
                                      (NOTEPAD.SKETCH1 COLORDS (BITMAPCREATE .NOTEPAD.BRUSH.WIDTH
                                                                      .NOTEPAD.BRUSH.HEIGHT (
                                                                            COLORNUMBERBITSPERPIXEL
                                                                                  ))
                                             BRUSH]
               (T (NOTEPAD.SKETCH1 (WINDOWPROP WINDOW 'DSP)
                         (BITMAPCREATE .NOTEPAD.BRUSH.WIDTH .NOTEPAD.BRUSH.HEIGHT)
                         BRUSH)))
            (RETURN])
```

(NOTEPAD.SKETCH1

```
   [LAMBDA (DS BUFFER.BITMAP BRUSH)                              (* rrb "22-DEC-82 11:36")
      (PROG (DOWN DONE MIDX MIDY NEW.MIDX NEW.MIDY)
            (until DONE do (GETMOUSESTATE)
                       (COND
                          ((LASTMOUSESTATE (NOT UP))
                           (SETQ DOWN T)))
                       (COND
                          ((AND DOWN (LASTMOUSESTATE UP))         (* RESTORE BITMAP)
                           [COND
                              (MIDX (BITBLT BUFFER.BITMAP 0 0 DS (IDIFFERENCE MIDX .NOTEPAD.BRUSH.HALF.WIDTH)
                                           (IDIFFERENCE MIDY .NOTEPAD.BRUSH.HALF.HEIGHT)
                                           .NOTEPAD.BRUSH.WIDTH .NOTEPAD.BRUSH.HEIGHT 'INPUT 'REPLACE]
                           (SETQ DONE T))
                          (T (SETQ NEW.MIDX (LASTMOUSEX DS))
                             (SETQ NEW.MIDY (LASTMOUSEY DS))
                             (COND
                                ((OR (NEQ NEW.MIDX MIDX)
                                     (NEQ NEW.MIDY MIDY)
                                     (LASTMOUSESTATE (NOT UP))) (* RESTORE BITMAP)
                                 [COND
                                    (MIDX (BITBLT BUFFER.BITMAP 0 0 DS (IDIFFERENCE MIDX
                                                                            .NOTEPAD.BRUSH.HALF.WIDTH)
                                                 (IDIFFERENCE MIDY .NOTEPAD.BRUSH.HALF.HEIGHT)
                                                 .NOTEPAD.BRUSH.WIDTH .NOTEPAD.BRUSH.HEIGHT 'INPUT 'REPLACE]
                                 (SETQ MIDX NEW.MIDX)
                                 (SETQ MIDY NEW.MIDY)
                                 (COND
                                    ((LASTMOUSESTATE (NOT UP))
                                     (PAINT.AT.POSSIBLE.POINT MIDX MIDY)))
                                 (BITBLT DS (IDIFFERENCE MIDX .NOTEPAD.BRUSH.HALF.WIDTH)
                                        (IDIFFERENCE MIDY .NOTEPAD.BRUSH.HALF.HEIGHT)
                                        BUFFER.BITMAP 0 0 .NOTEPAD.BRUSH.WIDTH .NOTEPAD.BRUSH.HEIGHT
                                        'INPUT
                                        'REPLACE)
                                 (BITBLT BRUSH 0 0 DS (IDIFFERENCE MIDX .NOTEPAD.BRUSH.HALF.WIDTH)
                                        (IDIFFERENCE MIDY .NOTEPAD.BRUSH.HALF.HEIGHT)
                                        .NOTEPAD.BRUSH.WIDTH .NOTEPAD.BRUSH.HEIGHT 'INPUT 'PAINT])
```

(NOTEPAD.TEXT

```
   [LAMBDA (WINDOW OPERATION)                                    (* rrb " 4-JAN-83 16:45")
      (PROG [CHARACTERISTICS TEXT FONT TEXTWIDTH NEW.BITMAP DSP (COLORDS (WINDOWPROP WINDOW 'INCOLOR]
            (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
            (SETQ TEXT (READ.FROM.PROMPT.WINDOW "Text to be printed: "))
            [SETQ FONT (FONTCREATE (LISTGET CHARACTERISTICS 'PRINTING.FONT.DEFINITION]
            (SETQ TEXTWIDTH (STRINGWIDTH TEXT FONT))
            [SETQ NEW.BITMAP (BITMAPCREATE TEXTWIDTH (FONTHEIGHT FONT)
                                    (COND
                                       (COLORDS (COLORNUMBERBITSPERPIXEL))
                                       (T 1]
            (SETQ DSP (DSPCREATE NEW.BITMAP))
            (DSPFONT FONT DSP)
            (AND COLORDS (DSPCOLOR (LISTGET CHARACTERISTICS 'SHADE.DEFINITION)
                               DSP))
            (MOVETO 0 (FONTDESCENT FONT)
                   DSP)
            (PRIN3 TEXT DSP)
            (MOVE.BITMAP WINDOW NEW.BITMAP OPERATION])
```

(NOTEPAD.USE.GRID

```
   [LAMBDA (WINDOW)                                              (* rrb "27-DEC-82 16:53")
```

```
        (PROG (CHARACTERISTICS USE.GRID USE.GRID.TO.DRAW)
              (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
              (SELECTQ [MENU (COND
                                ((type? MENU NOTEPAD.USE.GRID.MENU)
                                 NOTEPAD.USE.GRID.MENU)
                                (T (SETQ NOTEPAD.USE.GRID.MENU (create MENU
                                                        ITEMS _ '((No% Grid 'NOGRID "no grid will be
                                                                        used")
                                                          (When% specifying 'SPECIFY "the grid
                                                                        will be used when specifying
                                                                        points on the notepad.")
                                                          (When% drawing 'DRAW "the grid will
                                                                        be used as points are layed
                                                                        down in the trajectories.")
                                                          (Both 'BOTH "The grid will be used
                                                                        to specify points and when
                                                                        drawing trajectories."]
                (NOGRID (SETQ USE.GRID NIL)
                        (SETQ USE.GRID.TO.DRAW NIL))
                (SPECIFY (SETQ USE.GRID T)
                        (SETQ USE.GRID.TO.DRAW NIL))
                (DRAW (SETQ USE.GRID NIL)
                        (SETQ USE.GRID.TO.DRAW T))
                (BOTH (SETQ USE.GRID T)
                        (SETQ USE.GRID.TO.DRAW T))
                (RETURN))
              (LISTPUT CHARACTERISTICS 'USE.GRID USE.GRID)
              (LISTPUT CHARACTERISTICS 'USE.GRID.TO.DRAW USE.GRID.TO.DRAW)
              (printout PROMPTWINDOW T "Grid is now " (COND
                                                (USE.GRID (COND
                                                        (USE.GRID.TO.DRAW "in use when drawing and
                                                                specifying.")
                                                        (T "in use when specifying points.")))
                                                (T (COND
                                                        (USE.GRID.TO.DRAW "in use when drawing.")
                                                        (T "not in use"])
```

(NOTEPAD.USE.MASK
```
  [LAMBDA (WINDOW)                                      (* edited%: "17-DEC-82 10:05")
    (PROG (CHARACTERISTICS USE.MASK)
          (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
          [SETQ USE.MASK (NOT (LISTGET CHARACTERISTICS 'USE.MASK]
          (LISTPUT CHARACTERISTICS 'USE.MASK USE.MASK)
          (printout PROMPTWINDOW T "Mask is now " (COND
                                                (USE.MASK "in use")
                                                (T "not in use"])
```

(NOTEPAD.USE.SYMMETRIC.BRUSH/MASK
```
  [LAMBDA (WINDOW)                                      (* edited%: "17-DEC-82 12:04")
    (PROG (CHARACTERISTICS USE.SYMMETRIC.BRUSH/MASK)
          (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
          [SETQ USE.SYMMETRIC.BRUSH/MASK (NOT (LISTGET CHARACTERISTICS 'USE.SYMMETRIC.BRUSH/MASK]
          (LISTPUT CHARACTERISTICS 'USE.SYMMETRIC.BRUSH/MASK USE.SYMMETRIC.BRUSH/MASK)
          (printout PROMPTWINDOW T "Symmetric brush and mask are now " (COND
                                                        (USE.SYMMETRIC.BRUSH/MASK "in use")
                                                        (T "not in use"])
```

(NOTEPAD.USE.SYMMETRY
```
  [LAMBDA (WINDOW)                                      (* edited%: "17-DEC-82 10:36")
    (PROG (CHARACTERISTICS USE.SYMMETRY)
          (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
          (SETQ USE.SYMMETRY (ACQUIRE.SYMMETRY))
          (LISTPUT CHARACTERISTICS 'USE.SYMMETRY USE.SYMMETRY)
          (printout PROMPTWINDOW T "Symmetry is now " (SELECTQ USE.SYMMETRY
                                                        (NIL "not in use")
                                                        USE.SYMMETRY])
```

(NOTEPAD.FILL
```
  [LAMBDA (WINDOW OPERATION)                            (* DAHJr " 7-APR-83 18:23")
    (PROG [CHARACTERISTICS BITMAP REGION PT INTERIOR LEFT BOTTOM (COLORDS (WINDOWPROP WINDOW 'INCOLOR]
          (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
      LP  (printout PROMPTWINDOW T "Indicate an exterior box for the filling.")
          [SETQ BITMAP (COND
                          [COLORDS (PICKUP.BITMAP COLORDS (SETQ REGION (GETCOLORREGION]
                          (T [SETQ REGION (INTERSECTREGIONS (GET.WINDOW.REGION WINDOW)
                                                        (DSPCLIPPINGREGION NIL (WINDOWPROP WINDOW 'DSP]
                                (COND
                                  (REGION (PICKUP.BITMAP WINDOW REGION))
                                  (T (printout PROMPTWINDOW T "An empty region: try again")
                                        (GO LP]
          (printout PROMPTWINDOW T "Indicate position at which to start filling")
          (SETQ PT (NOTEPAD.GETPOSITION WINDOW))
          (COND
```

```
            [(INSIDE? REGION (fetch XCOORD of PT)
                           (fetch YCOORD of PT))
              (SETQ PT (create POSITION
                              XCOORD _ (IDIFFERENCE (fetch XCOORD of PT)
                                             (SETQ LEFT (fetch LEFT of REGION)))
                              YCOORD _ (IDIFFERENCE (fetch YCOORD of PT)
                                             (SETQ BOTTOM (fetch BOTTOM of REGION]
              (T (PROMPTPRINT "The point should be inside the bounding region.")
                 (GO LP)))
            (SETQ INTERIOR (BITMAP.INTERIOR BITMAP PT))
```

(* simulate merge since this will work for both color and b&w where as BITBLT merge only works for b&w NIL)

```
            (BITBLT INTERIOR 0 0 (OR COLORDS WINDOW)
                    LEFT BOTTOM NIL NIL 'INPUT 'ERASE)
            [BITBLT NIL NIL NIL INTERIOR 0 0 NIL NIL 'TEXTURE 'ERASE (LOGXOR (LISTGET CHARACTERISTICS
                                                                                   'SHADE.DEFINITION)
                                                         (COND
                                                            (COLORDS (MAXIMUMCOLOR))
                                                            (T BLACKSHADE]
            (BITBLT INTERIOR 0 0 (OR COLORDS WINDOW)
                    LEFT BOTTOM NIL NIL 'INPUT 'PAINT])
```

(**NOTEPAD.CONFIRM**
```
  [LAMBDA (MESSAGE)                                              (* edited%: " 2-APR-82 18:04")
    (PROG ((MENU (create MENU
                        TITLE _ MESSAGE
                        ITEMS _ '((YES T)
                                  (NO NIL))
                        CENTERFLG _ T
                        CHANGEOFFSETFLG _ T)))
          (RETURN (MENU MENU)])
```

(**NOTEPAD.CREATE**
```
  [LAMBDA (BITMAP INCOLORFLG)                                    (* DAHJr "23-OCT-83 12:06")
    (PROG ((TITLE (COND
                    (INCOLORFLG "Color Notepad control window")
                    (T "Black and white Notepad control window")))
           NPWINDOW WIDTH HEIGHT WINDOW.WIDTH WINDOW.HEIGHT)
          (COND
            ((AND (NULL INCOLORFLG)
                  BITMAP)
             (GETMOUSESTATE)
             (SETQ WIDTH (BITMAPWIDTH BITMAP))
             (SETQ HEIGHT (BITMAPHEIGHT BITMAP))
             (SETQ WINDOW.WIDTH (IPLUS 8 WIDTH))
             (SETQ WINDOW.HEIGHT (IPLUS 20 HEIGHT))
             (SETQ NPWINDOW (CREATEW (create REGION
                                            LEFT _ (MAX 0 (MIN (IDIFFERENCE SCREENWIDTH WINDOW.WIDTH)
                                                               LASTMOUSEX))
                                            BOTTOM _ (MAX 0 (MIN (IDIFFERENCE SCREENHEIGHT WINDOW.HEIGHT)
                                                                 LASTMOUSEY))
                                            WIDTH _ WINDOW.WIDTH
                                            HEIGHT _ WINDOW.HEIGHT)
                                     TITLE 4))
             (BITBLT BITMAP 0 0 NPWINDOW 0 0 WIDTH HEIGHT 'INPUT 'REPLACE)
             (printout PROMPTWINDOW T "Move this new notepad to desired position")
             (MOVEW NPWINDOW)
             (CLRPROMPT))
            (T (printout PROMPTWINDOW T "Indicate a region for a NOTEPAD window")
               (SETQ NPWINDOW (CREATEW (GETREGION)
                                       TITLE))
               (CLRPROMPT)))
          (COND
            (INCOLORFLG (COLORDISPLAY T (AND (FIXP INCOLORFLG)
                                             INCOLORFLG))
                 [WINDOWPROP NPWINDOW 'INCOLOR (SETQ INCOLORFLG (DSPCREATE (COLORSCREENBITMAP]
                 (DSPCLIPPINGREGION WHOLECOLORDISPLAY INCOLORFLG)))
          (WINDOWPROP NPWINDOW 'BUTTONEVENTFN 'NOTEPAD.BUTTONEVENTFN)
          (WINDOWPROP NPWINDOW 'CHARACTERISTICS (NOTEPAD.DEFAULT.CHARACTERISTICS INCOLORFLG))
          (NOTEPAD.HELP)
          (RETURN NPWINDOW])
```

(**NOTEPAD.DEFAULT.CHARACTERISTICS**
```
  [LAMBDA (INCOLORFLG)                                           (* DAHJr "31-MAY-83 12:05")
                                                                 (* returns the default characteristics of a notepad window's
                                                                 brush style etc.)
    (COND
      (INCOLORFLG (LIST 'BITS.PER.PIXEL (COLORNUMBERBITSPERPIXEL)
                        'BRUSH.DEFINITION
                        (COLORIZEBITMAP NOTEPAD.DEFAULT.BRUSH 0 (MAXIMUMCOLOR)
                                (COLORNUMBERBITSPERPIXEL))
                        'MASK.DEFINITION
                        (COLORIZEBITMAP NOTEPAD.DEFAULT.MASK 0 (MAXIMUMCOLOR)
```

```
                                   (COLORNUMBERBITSPERPIXEL))
                          'USE.MASK NIL 'PRINTING.FONT.DEFINITION NOTEPAD.DEFAULT.FONT 'GRID.DEFINITION
                          '(0 0 16 16)
                          'USE.GRID NIL 'USE.GRID.TO.DRAW NIL 'USE.SYMMETRY NIL 'POINT.OF.SYMMETRY.DEFINITION
                          (CONS (IQUOTIENT COLORSCREENWIDTH 2)
                                (IQUOTIENT COLORSCREENHEIGHT 2))
                          'USE.SYMMETRIC.BRUSH/MASK NIL 'SHADE.DEFINITION (MAXIMUMCOLOR)))
          (T (LIST 'BITS.PER.PIXEL 1 'BRUSH.DEFINITION NOTEPAD.DEFAULT.BRUSH 'MASK.DEFINITION NOTEPAD.DEFAULT.MASK
                   'USE.MASK NIL 'PRINTING.FONT.DEFINITION NOTEPAD.DEFAULT.FONT 'GRID.DEFINITION
                   '(0 0 16 16)
                   'USE.GRID NIL 'USE.GRID.TO.DRAW NIL 'USE.SYMMETRY NIL 'POINT.OF.SYMMETRY.DEFINITION
                   (CONS 100 100)
                   'USE.SYMMETRIC.BRUSH/MASK NIL 'SHADE.DEFINITION BLACKSHADE])
```

(**NOTEPAD.GETPOSITION**

```
  [LAMBDA (WINDOW CURSOR NOGRID)                              (* rrb "22-FEB-83 14:46")
                                                             (* IF USE.GRID IS NOT GIVEN, SEE IF YOU NEED TO
                                                             GRIDIFY; OTHERWISE ALWAYS GRIDIFY)

      (PROG (PT CHARACTERISTICS USE.GRID DS)
            [SETQ PT (COND
                         ((SETQ DS (WINDOWPROP WINDOW 'INCOLOR))
                          (GETCOLORPOSITION DS CURSOR))
                         (T (GETPOSITION WINDOW CURSOR]
            (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
            (SETQ USE.GRID (LISTGET CHARACTERISTICS 'USE.GRID))
            (RETURN (COND
                        [USE.GRID (NOTEPAD.ON.GRID PT (LISTGET CHARACTERISTICS 'GRID.DEFINITION]
                        (T PT])
```

(**NOTEPAD.MASK**

```
  [LAMBDA (BITMAP)                                            (* edited%: " 2-DEC-82 10:21")
      (PROG ((WIDTH (fetch (BITMAP BITMAPWIDTH) of BITMAP))
              (HEIGHT (fetch (BITMAP BITMAPHEIGHT) of BITMAP))
              LEFT RIGHT BOTTOM TOP NEW.BITMAP)
            (SETQ NEW.BITMAP (BITMAPCREATE WIDTH HEIGHT))
            (DSPFILL NIL −1 PAINT NEW.BITMAP)
            [for X from 0 to (SUB1 WIDTH) DO (for Y from 0 to (SUB1 HEIGHT)
                                     THEREIS (COND
                                                 ((EQ 1 (BITMAPBIT BITMAP X Y))
                                                  T)
                                                 ((BITMAPBIT BITMAP X Y 0]
            (SETQ RIGHT (for X from (SUB1 WIDTH) to 0 by −1 thereis (BIT.IN.COLUMN BITMAP X)))
            (SETQ BOTTOM (for X from 0 to (SUB1 HEIGHT) thereis (BIT.IN.ROW BITMAP X)))
            (SETQ TOP (for X from (SUB1 HEIGHT) to 0 by −1 thereis (BIT.IN.ROW BITMAP X)))
            (RETURN NEW.BITMAP])
```

(**NOTEPAD.ON.GRID**

```
  [LAMBDA (PT GRID)                                           (* rrb "27-DEC-82 16:59")
     (create POSITION
             XCOORD _ (NOTEPAD.ON.GRID.X (fetch (POSITION XCOORD) of PT)
                             (fetch (REGION LEFT) of GRID)
                             (fetch (REGION WIDTH) of GRID))
             YCOORD _ (NOTEPAD.ON.GRID.X (fetch (POSITION YCOORD) of PT)
                             (fetch (REGION BOTTOM) of GRID)
                             (fetch (REGION HEIGHT) of GRID])
```

(**NOTEPAD.ON.GRID.X**

```
  [LAMBDA (X X0 DX)                                           (* edited%: " 7-DEC-82 17:30")
     (PROG (X1 NEGATIVE)
           (SETQ X1 (IDIFFERENCE X X0))
           (COND
              ((ILESSP X1 0)
               (SETQ X1 (MINUS X1))
               (SETQ NEGATIVE T)))
           (SETQ X1 (ITIMES DX (IQUOTIENT (IPLUS X1 (RSH DX 1))
                                     DX)))
           [COND
              (NEGATIVE (SETQ X1 (MINUS X1]
           (RETURN (IPLUS X0 X1])
```

(**NOTEPAD.OPERATION**

```
  [LAMBDA NIL                                                 (* edited%: " 4-DEC-82 15:55")
     (COND
        ((LASTMOUSESTATE LEFT)
         'PAINT)
        (T 'ERASE])
```

(**NOTEPAD.SOLID.AREA**

```
  [LAMBDA (WINDOW REGION OPERATION)                           (* edited%: " 4-DEC-82 17:31")
     (BITBLT NIL 0 0 WINDOW (fetch (REGION LEFT) of REGION)
             (fetch (REGION BOTTOM) of REGION)
```

```
                        (fetch (REGION WIDTH) of REGION)
                        (fetch (REGION HEIGHT) of REGION)
                        'TEXTURE
                        'REPLACE
                        (COND
                           ((EQ OPERATION 'PAINT)
                            BLACKSHADE)
                           (T WHITESHADE)])
```

(**BITMAP.EXTERIOR**
```
  [LAMBDA (BITMAP)                                                (* rrb "19-JAN-83 19:27")
     (PROG ((WIDTH (BITMAPWIDTH BITMAP))
            (HEIGHT (BITMAPHEIGHT BITMAP))
            INTERIOR
            (NBITS (BITSPERPIXEL BITMAP))
            FROMVALUE MAXVAL MAXX MAXY)
           (SETQ MAXVAL (SUB1 (EXPT 2 NBITS)))
           (SETQ INTERIOR (BITMAPCREATE WIDTH HEIGHT NBITS))
           (SETQ MAXX (SUB1 WIDTH))
           (SETQ MAXY (SUB1 HEIGHT))

           (* use the upper right corner bit as the from value in the color case, 0 in the b&w case.)

           [SETQ FROMVALUE (COND
                              ((EQ NBITS 1)
                               0)
                              (T (BITMAPBIT BITMAP MAXX MAXY]
           (for X from 0 to MAXX do (EXTEND.AREA BITMAP INTERIOR X 0 MAXX MAXY FROMVALUE MAXVAL)
                                    (EXTEND.AREA BITMAP INTERIOR X MAXY MAXX MAXY FROMVALUE MAXVAL))
           (for Y from 1 to (SUB1 MAXY) do (EXTEND.AREA BITMAP INTERIOR 0 Y MAXX MAXY FROMVALUE MAXVAL)
                                           (EXTEND.AREA BITMAP INTERIOR MAXX Y MAXX MAXY FROMVALUE MAXVAL))
           (RETURN INTERIOR])
```

(**NOTEPAD.TITLEBUTTONFN**
```
  [LAMBDA (WINDOW)                                                (* DAHJr " 4-APR-83 16:42")
     (PROG (CHARACTERISTICS DSP COMMAND.MENU COMMAND NEW.NOTEPAD STARTPT ENDPT REGION POSITION WINDOWBITMAP
                BITMAP.NAME NEW.BITMAP TEXT)
           (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
           (SETQ DSP (WINDOWPROP WINDOW 'DSP))
           [SETQ COMMAND.MENU (create MENU
                                      TITLE _ "Operations on the whole"
                                      ITEMS _ '(HELP NEW.NOTEPAD COPY.NOTEPAD SAVE.AS.BITMAP INSPECT.STYLE]
           (SETQ COMMAND (MENU COMMAND.MENU))
           (SELECTQ COMMAND
               (NIL NIL)
               (HELP (NOTEPAD.HELP))
               (NEW.NOTEPAD (SETQ NEW.NOTEPAD (NOTEPAD.CREATE))
                            [WINDOWPROP NEW.NOTEPAD 'CHARACTERISTICS (COPYALL (WINDOWPROP WINDOW
                                                                                     'CHARACTERISTICS])
               (COPY.NOTEPAD (SETQ REGION (DSPCLIPPINGREGION NIL DSP))
                             (SETQ NEW.BITMAP (PICKUP.BITMAP WINDOW REGION))
                             (SETQ NEW.NOTEPAD (NOTEPAD.CREATE NEW.BITMAP))
                             [WINDOWPROP NEW.NOTEPAD 'CHARACTERISTICS (COPYALL (WINDOWPROP WINDOW
                                                                                     'CHARACTERISTICS])
               (SAVE.AS.BITMAP
                   (printout PROMPTWINDOW T "Indicate region to be saved as a bitmap")
                   (SETQ REGION (GET.NOTEPAD.REGION WINDOW))
                   (SETQ BITMAP.NAME (READ.FROM.PROMPT.WINDOW "Name for new bitmap: "))
                   (SETQ NEW.BITMAP (PICKUP.BITMAP WINDOW REGION))
                   (SET BITMAP.NAME NEW.BITMAP))
               (INSPECT.STYLE
                   (INSPECT/PLIST CHARACTERISTICS))
               (SHOULDNT (CONCAT "Unrecognized COMMAND in NOTEPAD.BUTTONEVENTFN: " COMMAND])
```

(**PAINT.A.BITMAP**
```
  [LAMBDA (BRUSH MASK MIDX MIDY POSSIBLY.ROTATED)                 (* rrb "20-DEC-82 15:46")
                                                                  (* masks with mask and paints or erases with the brush
                                                                  bitmaps.)
     (COND
        ((AND POSSIBLY.ROTATED .NOTEPAD.USE.SYMMETRIC.BRUSH/MASK)
         (COND
            (.NOTEPAD.USE.MASK (BITBLT MASK 0 0 .NOTEPAD.WINDOW (IDIFFERENCE MIDX .NOTEPAD.MASK.HALF.HEIGHT)
                                       (IDIFFERENCE MIDY .NOTEPAD.MASK.HALF.WIDTH)
                                       .NOTEPAD.MASK.HEIGHT .NOTEPAD.MASK.WIDTH 'INPUT .NOTEPAD.INVERSE.OPERATION)
                ))
         (BITBLT BRUSH 0 0 .NOTEPAD.WINDOW (IDIFFERENCE MIDX .NOTEPAD.BRUSH.HALF.HEIGHT)
                 (IDIFFERENCE MIDY .NOTEPAD.BRUSH.HALF.WIDTH)
                 .NOTEPAD.BRUSH.HEIGHT .NOTEPAD.BRUSH.WIDTH 'INPUT .NOTEPAD.OPERATION))
        (T (COND
              (.NOTEPAD.USE.MASK (BITBLT MASK 0 0 .NOTEPAD.WINDOW (IDIFFERENCE MIDX .NOTEPAD.MASK.HALF.WIDTH)
                                         (IDIFFERENCE MIDY .NOTEPAD.MASK.HALF.HEIGHT)
                                         .NOTEPAD.MASK.WIDTH .NOTEPAD.MASK.HEIGHT 'INPUT
                                         .NOTEPAD.INVERSE.OPERATION)))
           (BITBLT BRUSH 0 0 .NOTEPAD.WINDOW (IDIFFERENCE MIDX .NOTEPAD.BRUSH.HALF.WIDTH)
```

```
                        (IDIFFERENCE MIDY .NOTEPAD.BRUSH.HALF.HEIGHT)
                        .NOTEPAD.BRUSH.WIDTH .NOTEPAD.BRUSH.HEIGHT 'INPUT .NOTEPAD.OPERATION])
```

(**PAINT.ALL.BITMAPS**
```
  [LAMBDA (MIDX MIDY)                                        (* edited%: "17-DEC-82 16:28")
    (SELECTQ .NOTEPAD.USE.SYMMETRY
        (NIL (PAINT.A.BITMAP .NOTEPAD.BRUSH.1 .NOTEPAD.MASK.1 MIDX MIDY))
        (LEFT/RIGHT (PAINT.A.BITMAP .NOTEPAD.BRUSH.1 .NOTEPAD.MASK.1 MIDX MIDY)
                    (PAINT.A.BITMAP .NOTEPAD.BRUSH.2 .NOTEPAD.MASK.2 (IDIFFERENCE (IPLUS .NOTEPAD.POSX
                                                                                         .NOTEPAD.POSX)
                                                                                 MIDX)
                        MIDY))
        (UP/DOWN (PAINT.A.BITMAP .NOTEPAD.BRUSH.1 .NOTEPAD.MASK.1 MIDX MIDY)
                 (PAINT.A.BITMAP .NOTEPAD.BRUSH.3 .NOTEPAD.MASK.3 MIDX (IDIFFERENCE (IPLUS .NOTEPAD.POSY
                                                                                          .NOTEPAD.POSY)
                                                                                  MIDY)))
        (4-FOLD (PROG (REF.MIDX REF.MIDY)
                      (SETQ REF.MIDX (IDIFFERENCE (IPLUS .NOTEPAD.POSX .NOTEPAD.POSX)
                                                  MIDX))
                      (SETQ REF.MIDY (IDIFFERENCE (IPLUS .NOTEPAD.POSY .NOTEPAD.POSY)
                                                  MIDY))
                      (PAINT.A.BITMAP .NOTEPAD.BRUSH.1 .NOTEPAD.MASK.1 MIDX MIDY)
                      (PAINT.A.BITMAP .NOTEPAD.BRUSH.2 .NOTEPAD.MASK.2 REF.MIDX MIDY)
                      (PAINT.A.BITMAP .NOTEPAD.BRUSH.3 .NOTEPAD.MASK.3 MIDX REF.MIDY)
                      (PAINT.A.BITMAP .NOTEPAD.BRUSH.4 .NOTEPAD.MASK.4 REF.MIDX REF.MIDY)))
        (8-FOLD (PROG (REF.MIDX REF.MIDY DIAG.MIDX DIAG.MIDY)
                      (SETQ REF.MIDX (IDIFFERENCE (IPLUS .NOTEPAD.POSX .NOTEPAD.POSX)
                                                  MIDX))
                      (SETQ REF.MIDY (IDIFFERENCE (IPLUS .NOTEPAD.POSY .NOTEPAD.POSY)
                                                  MIDY))
                      (PAINT.A.BITMAP .NOTEPAD.BRUSH.1 .NOTEPAD.MASK.1 MIDX MIDY)
                      (PAINT.A.BITMAP .NOTEPAD.BRUSH.2 .NOTEPAD.MASK.2 REF.MIDX MIDY)
                      (PAINT.A.BITMAP .NOTEPAD.BRUSH.3 .NOTEPAD.MASK.3 MIDX REF.MIDY)
                      (PAINT.A.BITMAP .NOTEPAD.BRUSH.4 .NOTEPAD.MASK.4 REF.MIDX REF.MIDY)
                      (SETQ DIAG.MIDX (IDIFFERENCE (IPLUS .NOTEPAD.POSX MIDY
                                                          .NOTEPAD.POSY))
                      (SETQ DIAG.MIDY (IDIFFERENCE (IPLUS .NOTEPAD.POSY MIDX)
                                                          .NOTEPAD.POSX))
                      (SETQ REF.MIDX (IDIFFERENCE (IPLUS .NOTEPAD.POSX .NOTEPAD.POSX)
                                                  DIAG.MIDX))
                      (SETQ REF.MIDY (IDIFFERENCE (IPLUS .NOTEPAD.POSY .NOTEPAD.POSY)
                                                  DIAG.MIDY))
                      (PAINT.A.BITMAP .NOTEPAD.BRUSH.5 .NOTEPAD.MASK.5 DIAG.MIDX DIAG.MIDY T)
                      (PAINT.A.BITMAP .NOTEPAD.BRUSH.6 .NOTEPAD.MASK.6 REF.MIDX DIAG.MIDY T)
                      (PAINT.A.BITMAP .NOTEPAD.BRUSH.7 .NOTEPAD.MASK.7 DIAG.MIDX REF.MIDY T)
                      (PAINT.A.BITMAP .NOTEPAD.BRUSH.8 .NOTEPAD.MASK.8 REF.MIDX REF.MIDY T)))
        (ERROR USE.SYMMETRY ": unrecognized symmetry type in PAINT.ALL.BITMAPS"])
```

(**PAINT.AT.POSSIBLE.POINT**
```
  [LAMBDA (MIDX MIDY)                                        (* edited%: " 7-DEC-82 17:34")
    (PROG (USABLE.MIDX USABLE.MIDY)
          (SETQ USABLE.MIDX (COND
                                (.NOTEPAD.USE.GRID (NOTEPAD.ON.GRID.X MIDX .NOTEPAD.GRID.X0 .NOTEPAD.GRID.DX))
                                (T MIDX)))
          (SETQ USABLE.MIDY (COND
                                (.NOTEPAD.USE.GRID (NOTEPAD.ON.GRID.X MIDY .NOTEPAD.GRID.Y0 .NOTEPAD.GRID.DY))
                                (T MIDY)))
          (COND
             ((OR (NEQ USABLE.MIDX .NOTEPAD.PREVIOUS.MIDX)
                  (NEQ USABLE.MIDY .NOTEPAD.PREVIOUS.MIDY))
              (PAINT.ALL.BITMAPS USABLE.MIDX USABLE.MIDY)
              (SETQ .NOTEPAD.PREVIOUS.MIDX USABLE.MIDX)
              (SETQ .NOTEPAD.PREVIOUS.MIDY USABLE.MIDY])
```

(**PAINT.WITH.BITMAP**
```
  [LAMBDA (WINDOW OPERATION)                                  (* edited%: "17-DEC-82 15:44")
    (PROG (BUFFER.BITMAP DOWN DONE MIDX MIDY NEW.MIDX NEW.MIDY)
          (NOTEPAD.SETUP.TO.PAINT WINDOW OPERATION)
          (SETQ BUFFER.BITMAP (BITMAPCREATE .NOTEPAD.BRUSH.WIDTH .NOTEPAD.BRUSH.HEIGHT))
          [until DONE do ((GETMOUSESTATE)
                          (COND
                             ((LASTMOUSESTATE (NOT UP))             (* RESTORE BITMAP)
                              (SETQ DOWN T)))
                          (COND
                             ((AND DOWN (LASTMOUSESTATE UP))        (* RESTORE BITMAP)
                              [COND
                                 (MIDX (BITBLT BUFFER.BITMAP 0 0 WINDOW (IDIFFERENCE MIDX
                                                                                     .NOTEPAD.BRUSH.HALF.WIDTH)
                                              (IDIFFERENCE MIDY .NOTEPAD.BRUSH.HALF.HEIGHT)
                                              .NOTEPAD.BRUSH.WIDTH .NOTEPAD.BRUSH.HEIGHT 'INPUT 'REPLACE]
                              (SETQ DONE T))
                             (T (SETQ NEW.MIDX (LASTMOUSEX WINDOW))
                                (SETQ NEW.MIDY (LASTMOUSEY WINDOW))
                                (COND
```

```
                                        ((OR (NEQ NEW.MIDX MIDX)
                                             (NEQ NEW.MIDY MIDY)
                                             (LASTMOUSESTATE (OR LEFT MIDDLE)))
                                                                    (* RESTORE BITMAP)
                                   [COND
                                      (MIDX (BITBLT BUFFER.BITMAP 0 0 WINDOW (IDIFFERENCE MIDX
                                                                                    .NOTEPAD.BRUSH.HALF.WIDTH)
                                                    (IDIFFERENCE MIDY .NOTEPAD.BRUSH.HALF.HEIGHT)
                                                    .NOTEPAD.BRUSH.WIDTH .NOTEPAD.BRUSH.HEIGHT 'INPUT
                                                    'REPLACE]
                                   (SETQ MIDX NEW.MIDX)
                                   (SETQ MIDY NEW.MIDY)
                                   (COND
                                      ((LASTMOUSESTATE (NOT UP))
                                       (PAINT.AT.POSSIBLE.POINT MIDX MIDY)))
                                   (BITBLT WINDOW (IDIFFERENCE MIDX .NOTEPAD.BRUSH.HALF.WIDTH)
                                           (IDIFFERENCE MIDY .NOTEPAD.BRUSH.HALF.HEIGHT)
                                           BUFFER.BITMAP 0 0 .NOTEPAD.BRUSH.WIDTH .NOTEPAD.BRUSH.HEIGHT
                                           'INPUT
                                           'REPLACE)
                                   (BITBLT BRUSH 0 0 WINDOW (IDIFFERENCE MIDX .NOTEPAD.BRUSH.HALF.WIDTH)
                                           (IDIFFERENCE MIDY .NOTEPAD.BRUSH.HALF.HEIGHT)
                                           .NOTEPAD.BRUSH.WIDTH .NOTEPAD.BRUSH.HEIGHT 'INPUT 'PAINT]
                   (RETURN])
```

(**PICKUP.BITMAP**
```
  [LAMBDA (WINDOW REGION)                                          (* rrb "22-DEC-82 12:14")
     (PROG (NEW.BITMAP WIDTH HEIGHT)
        (SETQ WIDTH (fetch (REGION WIDTH) of REGION))
        (SETQ HEIGHT (fetch (REGION HEIGHT) of REGION))
        [SETQ NEW.BITMAP (BITMAPCREATE WIDTH HEIGHT (COND
                                                       (WINDOW (fetch (BITMAP BITMAPBITSPERPIXEL)
                                                                  of (DSPDESTINATION NIL WINDOW)))
                                                       (T 1]
        (BITBLT (OR WINDOW (SCREENBITMAP))
                (fetch (REGION LEFT) of REGION)
                (fetch (REGION BOTTOM) of REGION)
                NEW.BITMAP 0 0 WIDTH HEIGHT 'INPUT 'REPLACE)
        (RETURN NEW.BITMAP])
```

(**PICKUP.SCREEN.BITMAP**
```
  [LAMBDA (REGION SCREEN)                                          (* rrb "22-DEC-82 10:35")
                                                                   (* pick up a piece of a screen.)

     (PROG (NEW.BITMAP WIDTH HEIGHT)
        (SETQ WIDTH (fetch (REGION WIDTH) of REGION))
        (SETQ HEIGHT (fetch (REGION HEIGHT) of REGION))
        (SETQ NEW.BITMAP (BITMAPCREATE WIDTH HEIGHT (BITSPERPIXEL SCREEN)))
        (BITBLT SCREEN (fetch (REGION LEFT) of REGION)
                (fetch (REGION BOTTOM) of REGION)
                NEW.BITMAP 0 0 WIDTH HEIGHT 'INPUT 'REPLACE)
        (RETURN NEW.BITMAP])
```

(**READ.FROM.PROMPT.WINDOW**
```
  [LAMBDA (PRMPT)                                                  (* edited%: "15-DEC-82 23:02")
     (CLEARBUF T T)
     (CAR (PROCESS.READ PROMPTWINDOW PRMPT T])
```

(**PUTBACK.BITMAP**
```
  [LAMBDA (WINDOW REGION BITMAP)                                   (* edited%: "18-OCT-82 11:57")
     (BITBLT BITMAP 0 0 WINDOW (fetch (REGION LEFT) of REGION)
             (fetch (REGION BOTTOM) of REGION)
             (fetch (REGION WIDTH) of REGION)
             (fetch (REGION HEIGHT) of REGION)
             'INPUT
             'REPLACE])
```

(**PUTDEF.NOTEPAD.STYLE**
```
  [LAMBDA (NAME TYPE NOTEPAD.STYLE)                                (* DAHJr "26-JAN-83 10:33")
     (COND
        (NOTEPAD.STYLES (PUTASSOC NAME NOTEPAD.STYLE NOTEPAD.STYLES)
                (SORT NOTEPAD.STYLES T))
        (T (SETQ NOTEPAD.STYLES (LIST (CONS NAME NOTEPAD.STYLE]
```

(**SAVE.STYLE**
```
  [LAMBDA (WINDOW)                                                 (* DAHJr "26-JAN-83 10:34")
     (PROG (CHARACTERISTICS NAME OLD.ENTRY)
        (SETQ CHARACTERISTICS (WINDOWPROP WINDOW 'CHARACTERISTICS))
        (SETQ NAME (READ.FROM.PROMPT.WINDOW "Give name under which to save the style: "))
        (COND
           (NAME (SETQ OLD.ENTRY (GETDEF.NOTEPAD.STYLE NAME 'NOTEPADSTYLE))
                 (COND
```

```
                        ((OR (NULL OLD.ENTRY)
                             (CONFIRM (CONCAT "Over-write old entry named " NAME)))
                          (PUTDEF.NOTEPAD.STYLE NAME 'NOTEPADSTYLE (COPYALL CHARACTERISTICS))
                         (MARKASCHANGED NAME 'NOTEPADSTYLE (NULL OLD.ENTRY))
                         (printout PROMPTWINDOW T "Style saved under the name " NAME])
```

(**SBIT**
```
  [LAMBDA (X Y)                                          (* edited%: " 5-DEC-82 12:39")
    (BITBLT NIL 0 0 WINDOW X Y 1 1 'TEXTURE 'INVERT BLACKSHADE])
```

(**TEST.AND.SET**
```
  [LAMBDA (DEFINING.BITMAP AREA.BITMAP X Y FROMVALUE TOVALUE)   (* rrb "19-JAN-83 19:21")
    (DECLARE (LOCALVARS . T))
    (COND
       ((EQ (BITMAPBIT AREA.BITMAP X Y)
            TOVALUE)                                      (* already been here)
        NIL)
       ((NEQ (BITMAPBIT DEFINING.BITMAP X Y)
             FROMVALUE)                                   (* hit a boundary point)
        NIL)
       (T (BITMAPBIT AREA.BITMAP X Y TOVALUE)
          T])
)
```

(DEFINEQ

(**GET.WINDOW.REGION**
```
  [LAMBDA (W)                                             (* DAHJr "23-OCT-83 11:13")

          (* gets a region from a window on either the color screen or the b&w screen.)

    (PROG (DS REG)
          (RETURN (COND
                      ((SETQ DS (WINDOWPROP W 'INCOLOR))
                       (SETQ REG (GETCOLORREGION))
                       (create REGION
                               LEFT _ (IDIFFERENCE (fetch LEFT of REG)
                                            (DSPXOFFSET NIL DS))
                               BOTTOM _ (IDIFFERENCE (fetch BOTTOM of REG)
                                            (DSPYOFFSET NIL DS))
                               WIDTH _ (fetch WIDTH of REG)
                               HEIGHT _ (fetch HEIGHT of REG)))
                      (T (SETQ REG (GETREGION))
                         (SETQ DS (WINDOWPROP W 'DSP))
                         (create REGION
                               LEFT _ (IDIFFERENCE (fetch LEFT of REG)
                                            (DSPXOFFSET NIL DS))
                               BOTTOM _ (IDIFFERENCE (fetch BOTTOM of REG)
                                            (DSPYOFFSET NIL DS))
                               WIDTH _ (fetch WIDTH of REG)
                               HEIGHT _ (fetch HEIGHT of REG])
```

(**GETCOLORPOSITION**
```
  [LAMBDA (DS CURSOR)                                     (* rrb "20-DEC-82 16:07")
                                                         (* gets a point from the color screen)

    (RESETLST
        (RESETSAVE (CHANGECURSORSCREEN (COLORSCREENBITMAP)))
        (GETPOSITION DS CURSOR])
```

(**COLORBITMAPP**
```
  [LAMBDA (BITMAP)                                        (* rrb "22-DEC-82 09:42")
    (AND (BITMAPP BITMAP)
         (NEQ (BITSPERPIXEL BITMAP)
              1)
         BITMAP])
)
```

(RPAQQ **NOTEPAD.DEFAULT.FONT** (HELVETICA 18 BOLD))

(RPAQQ **NOTEPAD.STYLE.REPRESENTATION.NUMBER** 1)

(RPAQQ **NOTEPAD.COMMAND.MENU** NIL)

(RPAQQ **NOTEPAD.SHOW.FILL** NIL)

(RPAQQ **NOTEPAD.USE.GRID.MENU** NIL)

(RPAQQ **.NOTEPAD.BRUSH.1** NIL)

(RPAQQ **.NOTEPAD.BRUSH.2** NIL)

(RPAQQ **.NOTEPAD.BRUSH.3** NIL)

(RPAQQ **.NOTEPAD.BRUSH.4** NIL)

(RPAQQ **.NOTEPAD.BRUSH.5** NIL)

(RPAQQ **.NOTEPAD.BRUSH.6** NIL)

(RPAQQ **.NOTEPAD.BRUSH.7** NIL)

(RPAQQ **.NOTEPAD.BRUSH.8** NIL)

(RPAQQ **.NOTEPAD.MASK.1** NIL)

(RPAQQ **.NOTEPAD.MASK.2** NIL)

(RPAQQ **.NOTEPAD.MASK.3** NIL)

(RPAQQ **.NOTEPAD.MASK.4** NIL)

(RPAQQ **.NOTEPAD.MASK.5** NIL)

(RPAQQ **.NOTEPAD.MASK.6** NIL)

(RPAQQ **.NOTEPAD.MASK.7** NIL)

(RPAQQ **.NOTEPAD.MASK.8** NIL)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NOTEPAD.COMMAND.MENU NOTEPAD.USE.GRID.MENU .NOTEPAD.WINDOW .NOTEPAD.OPERATION .NOTEPAD.BRUSH.1
        .NOTEPAD.BRUSH.2 .NOTEPAD.BRUSH.3 .NOTEPAD.BRUSH.4 .NOTEPAD.BRUSH.5 .NOTEPAD.BRUSH.6 .NOTEPAD.BRUSH.7
        .NOTEPAD.BRUSH.8 .NOTEPAD.MASK.1 .NOTEPAD.MASK.2 .NOTEPAD.MASK.3 .NOTEPAD.MASK.4 .NOTEPAD.MASK.5
        .NOTEPAD.MASK.6 .NOTEPAD.MASK.7 .NOTEPAD.MASK.8 .NOTEPAD.USE.GRID .NOTEPAD.GRID.X0 .NOTEPAD.GRID.Y0
        .NOTEPAD.GRID.DX .NOTEPAD.GRID.DY .NOTEPAD.USE.MASK .NOTEPAD.INVERSE.OPERATION
        .NOTEPAD.USE.SYMMETRIC.BRUSH/MASK .NOTEPAD.USE.SYMMETRY .NOTEPAD.POSX .NOTEPAD.POSY .NOTEPAD.BRUSH.WIDTH
        .NOTEPAD.BRUSH.HEIGHT .NOTEPAD.BRUSH.HALF.WIDTH .NOTEPAD.BRUSH.HALF.HEIGHT .NOTEPAD.MASK.WIDTH
        .NOTEPAD.MASK.HEIGHT .NOTEPAD.MASK.HALF.WIDTH .NOTEPAD.MASK.HALF.HEIGHT .NOTEPAD.PREVIOUS.MIDX
        .NOTEPAD.PREVIOUS.MIDY)
)

(RPAQQ **COLORSPOTMARKER** NIL)

(RPAQ? **NOTEPAD.STYLES**)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NOTEPAD.STYLES)
)

(RPAQQ **SPOTMARKER**   )

(RPAQQ **NOTEPAD.DEFAULT.BRUSH** .)

(RPAQQ **NOTEPAD.DEFAULT.MASK** ▪)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS SPOTMARKER NOTEPAD.DEFAULT.BRUSH NOTEPAD.DEFAULT.MASK)
)

(RPAQ **CIRCLE.CENTER** (CURSORCREATE '
                             'NIL 8 8))

(RPAQ **CIRCLE.EDGE** (CURSORCREATE '
                             'NIL 15 8))

(RPAQ **ELLIPSE.CENTER** (CURSORCREATE '
                             'NIL 8 8))

(RPAQ **ELLIPSE.SEMI.MAJOR** (CURSORCREATE '
                             'NIL 15 8))

(RPAQ **ELLIPSE.SEMI.MINOR** (CURSORCREATE '
                             'NIL 8 15))

(RPAQ **CURVE.KNOT** (CURSORCREATE '
                             'NIL 8 8))

```
(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS CIRCLE.CENTER CIRCLE.EDGE ELLIPSE.CENTER ELLIPSE.SEMI.MAJOR ELLIPSE.SEMI.MINOR CURVE.KNOT)
)

(FILESLOAD (FROM VALUEOF LISPUSERSDIRECTORIES)
       EDITBITMAP READNUMBER)

[PUTDEF 'NOTEPADSTYLE 'FILEPKGCOMS '((COM COM T)
                                    (TYPE DESCRIPTION "NOTEPAD styles" GETDEF GETDEF.NOTEPAD.STYLE DELDEF
                                        DELDEF.NOTEPAD.STYLE PUTDEF PUTDEF.NOTEPAD.STYLE]
```

(**ADD.NOTEPAD.TO.BACKGROUND.MENU**)

```
(FONTCREATE NOTEPAD.DEFAULT.FONT)

(COND
   ((NULL NOTEPAD.STYLES)
    (FILESLOAD NOTEPAD-CORESTYLES)))

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS

(ADDTOVAR NLAMA )

(ADDTOVAR NLAML READ.NOTEPAD.STYLE)

(ADDTOVAR LAMA )
)
```

(PUTPROPS **NOTEPAD COPYRIGHT** ("Xerox Corporation" 1982 1983 1988 1991))

FUNCTION INDEX

VARIABLE INDEX