File created: 30-Mar-89 09:18:42 **{ERINYES}<LISPUSERS>MEDLEY>POSTSCRIPT>POSTSCRIPT.;6**

changes to: (FNS POSTSCRIPT.INIT \BITBLT.PSC)

previous date: 22-Feb-89 15:28:19 {ERINYES}<LISPUSERS>MEDLEY>POSTSCRIPT>POSTSCRIPT.;5

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;

```
(RPAQQ POSTSCRIPTCOMS
       [(RECORDS BRUSH FONTID ARRAYP PSCFONT \POSTSCRIPTDATA)
        (FNS CLOSEPOSTSCRIPTSTREAM OPENPOSTSCRIPTSTREAM POSTSCRIPT.BITMAPSCALE POSTSCRIPT.CLOSESTRING
             POSTSCRIPT.FONTCREATE POSTSCRIPT.FONTSAVAILABLE POSTSCRIPT.GETFONTID POSTSCRIPT.HARDCOPYW
             POSTSCRIPT.INIT POSTSCRIPT.OUTSTR POSTSCRIPT.PUTBITMAPBYTES POSTSCRIPT.PUTCOMMAND
             POSTSCRIPT.SHOWACCUM POSTSCRIPT.STARTPAGE POSTSCRIPT.TEDIT POSTSCRIPT.TEXT POSTSCRIPTFILEP
             PSCFONT.READFONT PSCFONT.SPELLFILE PSCFONT.WRITEFONT READ-AFM-FILE \BITBLT.PSC \BLTSHADE.PSC
             \CHARWIDTH.PSC \DRAWARC.PSC \DRAWCIRCLE.PSC \DRAWCURVE.PSC \DRAWELLIPSE.PSC \DRAWLINE.PSC
             \DRAWPOLYGON.PSC \DSPBOTTOMMARGIN.PSC \DSPCLIPPINGREGION.PSC \DSPFONT.PSC \DSPLEFTMARGIN.PSC
             \DSPLINEFEED.PSC \DSPRESET.PSC \DSPRIGHTMARGIN.PSC \DSPSCALE.PSC \DSPSPACEFACTOR.PSC
             \DSPTOPMARGIN.PSC \DSPXPOSITION.PSC \DSPYPOSITION.PSC \FILLCIRCLE.PSC \FILLPOLYGON.PSC \MOVETO.PSC
             \NEWPAGE.PSC \POSTSCRIPT.OUTCHARFN \POSTSCRIPT.PUTCHAR \STRINGWIDTH.PSC \TERPRI.PSC \DSPROTATE.PSC
             \DSPTRANSLATE.PSC \DRAWPOINT.PSC)
        (VARS (\POSTSCRIPT.ORIENTATION.MENU (create MENU ITEMS _ '(("Landscape" T "Print this
                                                                          file/document/image in Landscape
                                                                          Orientation")
                                                                   ("Portrait" 'NIL "Print this
                                                                          file/document/image in Portrait
                                                                          Orientation"))
                                                   TITLE _ "Orientation" CENTERFLG _ T MENUOFFSET _
                                                   (create POSITION XCOORD _ -1 YCOORD _ 0)
                                                   CHANGEOFFSETFLG _ 'Y))
              PS.BITMAPARRAY \POSTSCRIPT.JOB.SETUP SlopeMenuItems WeightMenuItems)
        (CONSTANTS (GOLDEN.RATIO 1.618034))
        (INITVARS (POSTSCRIPT.BITMAP.SCALE 1)
               (POSTSCRIPT.IMAGESIZEFACTOR 1.0)
               (POSTSCRIPT.PREFER.LANDSCAPE NIL)
               (POSTSCRIPT.TEXTFILE.LANDSCAPE NIL)
               (POSTSCRIPT.TEXTURE.SCALE 4)
               (POSTSCRIPTFONTDIRECTORIES '("{DSK}<LISPFILES>FONTS>PSC>"))
               (\POSTSCRIPT.LONGEDGE.SHIFT 0)
               (\POSTSCRIPT.SHORTEDGE.SHIFT 0)
               (\POSTSCRIPT.LONGEDGE.PTS (+ (TIMES 72 10.92)
                                            \POSTSCRIPT.SHORTEDGE.SHIFT))
               (\POSTSCRIPT.SHORTEDGE.PTS (+ (TIMES 72 8.0)
                                             \POSTSCRIPT.LONGEDGE.SHIFT))
               (\POSTSCRIPT.MAX.WILD.FONTSIZE 72))
        [ADDVARS (POSTSCRIPT.FONT.ALIST (HELVETICA . HELVETICA)
                        (TIMESROMAN . TIMES)
                        (TIMESROMAND . TIMES)
                        (COURIER . COURIER)
                        (GACHA . COURIER)
                        (CLASSIC . TIMES)
                        (MODERN . HELVETICA)
                        (CREAM . HELVETICA)
                        (TERMINAL . COURIER)
                        (LOGO . HELVETICA))
               [PRINTERTYPES ((POSTSCRIPT)
                              (CANPRINT (POSTSCRIPT))
                              (STATUS TRUE)
                              (PROPERTIES NILL)
                              (SEND POSTSCRIPT.SEND)
                              (BITMAPSCALE POSTSCRIPT.BITMAPSCALE)
                              (BITMAPFILE (POSTSCRIPT.HARDCOPYW FILE BITMAP SCALEFACTOR REGION ROTATION TITLE]
               [PRINTFILETYPES (POSTSCRIPT (TEST POSTSCRIPTFILEP)
                                      (EXTENSION (PS PSC))
                                      (CONVERSION (TEXT POSTSCRIPT.TEXT TEDIT POSTSCRIPT.TEDIT]
               (IMAGESTREAMTYPES (POSTSCRIPT (OPENSTREAM OPENPOSTSCRIPTSTREAM)
                                        (FONTCREATE POSTSCRIPT.FONTCREATE)
                                        (FONTSAVAILABLE POSTSCRIPT.FONTSAVAILABLE)
                                        (CREATECHARSET NILL]
        (GLOBALVARS DEFAULTPRINTINGHOST POSTSCRIPT.BITMAP.SCALE POSTSCRIPT.FONT.ALIST
               POSTSCRIPT.PREFER.LANDSCAPE POSTSCRIPT.TEXTFILE.LANDSCAPE POSTSCRIPT.TEXTURE.SCALE
               POSTSCRIPTFONTDIRECTORIES \POSTSCRIPT.JOB.SETUP \POSTSCRIPT.LONGEDGE.PTS
               \POSTSCRIPT.LONGEDGE.SHIFT \POSTSCRIPT.MAX.WILD.FONTSIZE \POSTSCRIPT.ORIENTATION.MENU
               \POSTSCRIPT.SHORTEDGE.PTS \POSTSCRIPT.SHORTEDGE.SHIFT \POSTSCRIPTIMAGEOPS)
        (FILES PS-SEND)
        (P (POSTSCRIPT.INIT))
        (PROP (FILETYPE MAKEFILE-ENVIRONMENT)
              POSTSCRIPT)
        (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)
                                                                                  (NLAML)
                                                                                  (LAMA POSTSCRIPT.PUTCOMMAND])
(DECLARE%: EVAL@COMPILE

(RECORD BRUSH (BRUSHSHAPE BRUSHSIZE BRUSHCOLOR)
       BRUSHSHAPE _ 'ROUND BRUSHSIZE _ 1)

(RECORD FONTID (FONTIDNAME FONTXFACTOR FONTOBLIQUEFACTOR))

(DATATYPE ARRAYP ((ORIG BITS 1)
                  (NIL BITS 1)
                  (READONLY FLAG)                          ; probably no READONLY arrays now
                  (NIL BITS 1)
                  (TYP BITS 4)
                  (BASE POINTER)
                  (LENGTH WORD)
                  (OFFST WORD))

       ;; note that while ARRAYP is a DATATYPE, the allocation of it actually happens at MAKEINIT time under INITDATATYPE{NAMES}

       )

(RECORD PSCFONT (FID IL-FONTID FIRSTCHAR LASTCHAR ASCENT DESCENT WIDTHS))

(DATATYPE \POSTSCRIPTDATA (POSTSCRIPTFONT                  ; The fontdescriptor of the current font
                                 POSTSCRIPTX POSTSCRIPTY POSTSCRIPTLEFTMARGIN POSTSCRIPTRIGHTMARGIN
                                 POSTSCRIPTBOTTOMMARGIN POSTSCRIPTTOPMARGIN POSTSCRIPTLINESPACING
                                 POSTSCRIPTCOLOR POSTSCRIPTSCALE POSTSCRIPTOPERATION POSTSCRIPTCLIPPINGREGION
                                 POSTSCRIPTPAGENUM POSTSCRIPTHEADING POSTSCRIPTHEADINGFONT POSTSCRIPTSPACEFACTOR
                                                           ; The expansion factor for spaces (see DSPSPACEFACTOR)
                                 POSTSCRIPTLANDSCAPE       ; T means that the paper is in 'landscape' mode
                                 POSTSCRIPTCHARSTOSHOW     ; T means that the string of chars has already been started

                                 )
       POSTSCRIPTSPACEFACTOR _ 1 POSTSCRIPTPAGENUM _ 0)
)

(/DECLAREDATATYPE 'ARRAYP '((BITS 1)
                            (BITS 1)
                            FLAG
                            (BITS 1)
                            (BITS 4)
                            POINTER WORD WORD)

       ;; ---field descriptor list elided by lister---

       '4)

(/DECLAREDATATYPE '\POSTSCRIPTDATA
       '(POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
               POINTER POINTER POINTER POINTER POINTER)

       ;; ---field descriptor list elided by lister---

       '36)

(DEFINEQ

(CLOSEPOSTSCRIPTSTREAM
  [LAMBDA (VSTREAM)                                        ; Edited 20-Jan-88 17:43 by Matt Heffron
    (POSTSCRIPT.PUTCOMMAND VSTREAM "
            savepage restore
           showpage
           %%%%Trailer
           "])

(OPENPOSTSCRIPTSTREAM
  [LAMBDA (FILE OPTIONS)                                   ; Edited 20-Oct-88 14:45 by Matt Heffron
    (LET ([FP (OPENSTREAM (if (AND (STRING-EQUAL (UNPACKFILENAME.STRING FILE 'EXTENSION)
                                          "")
                                   (STRING-EQUAL (UNPACKFILENAME.STRING FILE 'HOST)
                                          "LPT"))
                              then (PACKFILENAME.STRING 'HOST "LPT" 'NAME (UNPACKFILENAME.STRING FILE
                                                                                 'NAME)
                                          'EXTENSION "PS")
                            else FILE)
                     'OUTPUT NIL '((TYPE POSTSCRIPT)
                                    (SEQUENTIAL T]
          (IMAGEDATA (create \POSTSCRIPTDATA))
          LANDSCAPE? FONT IMAGESIZEFACTOR SHORTEDGE LONGEDGE TEMP)
         (SETFILEINFO FP 'EOL 'CR)
         (replace (STREAM OUTCHARFN) of FP with '\FILEOUTCHARFN)
         (printout FP "%%!PS-Adobe-2.0" T "%%%%Title: " (MKSTRING (OR (LISTGET OPTIONS 'DOCUMENT.NAME)
                                                                       FILE))
                T "%%%%Creator: PostScript ImageStream Driver by Matt Heffron of Beckman Instruments" T
                "%%%%CreationDate: " (DATE)
                T "%%%%For: " (if (STRING-EQUAL INITIALS "Edited:")
                                  then (MKSTRING USERNAME)
```

```
                                else INITIALS)
                T "%%%%EndComments" T)
         (for PJS in \POSTSCRIPT.JOB.SETUP do (PRIN1 PJS FP)
                                             (TERPRI FP))
         [if (SETQ LANDSCAPE? (CL:GETF OPTIONS 'ROTATION 'DEFAULTNIL))
             then (if (EQ LANDSCAPE? 'DEFAULTNIL)
                      then (SETQ LANDSCAPE? (if (EQ POSTSCRIPT.PREFER.LANDSCAPE 'ASK)
                                               then (MENU \POSTSCRIPT.ORIENTATION.MENU)
                                             else POSTSCRIPT.PREFER.LANDSCAPE]
         (replace POSTSCRIPTLANDSCAPE of IMAGEDATA with LANDSCAPE?)
         (if (NOT (AND (SETQ IMAGESIZEFACTOR (LISTGET OPTIONS 'IMAGESIZEFACTOR))
                      (NUMBERP IMAGESIZEFACTOR)
                      (CL:PLUSP IMAGESIZEFACTOR)))
             then (SETQ IMAGESIZEFACTOR 1.0))
         (if (AND (NUMBERP POSTSCRIPT.IMAGESIZEFACTOR)
                  (CL:PLUSP POSTSCRIPT.IMAGESIZEFACTOR))
             then (SETQ IMAGESIZEFACTOR (TIMES IMAGESIZEFACTOR POSTSCRIPT.IMAGESIZEFACTOR)))
         (PRIN1 "/imagesizefactor " FP)
         (PRIN1 IMAGESIZEFACTOR FP)
         (PRIN1 " def" FP)
         (TERPRI FP)
         (PRIN1 "%%%%EndSetup" FP)
         (TERPRI FP)
         (replace POSTSCRIPTSCALE of IMAGEDATA with 100.0)
         (SETQ LONGEDGE (FQUOTIENT (FTIMES \POSTSCRIPT.LONGEDGE.PTS 100.0)
                               IMAGESIZEFACTOR))
         (SETQ SHORTEDGE (FQUOTIENT (FTIMES \POSTSCRIPT.SHORTEDGE.PTS 100.0)
                                IMAGESIZEFACTOR))
         (replace (STREAM IMAGEOPS) of FP with \POSTSCRIPTIMAGEOPS)
         (replace (STREAM IMAGEDATA) of FP with IMAGEDATA)
         (replace (STREAM LINELENGTH) of FP with MAX.SMALLP)
         (replace (STREAM CHARPOSITION) of FP with 0)
         (replace (STREAM OUTCHARFN) of FP with '\POSTSCRIPT.OUTCHARFN)
         (if LANDSCAPE?
             then (\DSPTOPMARGIN.PSC FP (FIXR SHORTEDGE))
                  (\DSPRIGHTMARGIN.PSC FP (FIXR LONGEDGE))
                  (replace POSTSCRIPTCLIPPINGREGION of IMAGEDATA
                     with (create REGION
                                 LEFT _ 0.0
                                 BOTTOM _ 0.0
                                 WIDTH _ LONGEDGE
                                 HEIGHT _ SHORTEDGE))
           else (\DSPTOPMARGIN.PSC FP (FIXR LONGEDGE))
                (\DSPRIGHTMARGIN.PSC FP (FIXR SHORTEDGE))
                (replace POSTSCRIPTCLIPPINGREGION of IMAGEDATA
                   with (create REGION
                               LEFT _ 0.0
                               BOTTOM _ 0.0
                               WIDTH _ SHORTEDGE
                               HEIGHT _ LONGEDGE)))
         (SETQ FONT (FONTCREATE (OR [CAR (MKLIST (LISTGET OPTIONS 'FONTS]
                                    DEFAULTFONT)
                           NIL NIL NIL FP))
         (if (SETQ TEMP (LISTGET OPTIONS 'HEADING))
             then (replace POSTSCRIPTHEADING of IMAGEDATA with TEMP)
                  (replace POSTSCRIPTHEADINGFONT of IMAGEDATA with FONT))
         (\DSPLEFTMARGIN.PSC FP 0)
         (\DSPBOTTOMMARGIN.PSC FP 0)
         (\DSPFONT.PSC FP FONT)
         (\DSPLINEFEED.PSC FP (IMINUS (fetch (FONTDESCRIPTOR \SFHeight) of FONT)))
         (POSTSCRIPT.STARTPAGE FP)
         FP])

(POSTSCRIPT.BITMAPSCALE
  [LAMBDA (WIDTH HEIGHT)                                           ; Edited 20-Oct-88 14:48 by Matt Heffron
    (LET* ([MINDIMP (MIN (FQUOTIENT \POSTSCRIPT.LONGEDGE.PTS (SETQ HEIGHT (TIMES HEIGHT POSTSCRIPT.BITMAP.SCALE)
                                                              ))
                         (FQUOTIENT \POSTSCRIPT.SHORTEDGE.PTS (SETQ WIDTH (TIMES WIDTH POSTSCRIPT.BITMAP.SCALE]
           (MINDIML (MIN (FQUOTIENT \POSTSCRIPT.SHORTEDGE.PTS HEIGHT)
                         (FQUOTIENT \POSTSCRIPT.LONGEDGE.PTS WIDTH)))
           (PPL (if (EQ POSTSCRIPT.PREFER.LANDSCAPE 'ASK)
                    then (MENU \POSTSCRIPT.ORIENTATION.MENU)
                  else POSTSCRIPT.PREFER.LANDSCAPE))
           MINDIM OTHERDIM SF1 SF2)
          (if PPL
              then (SETQ MINDIM MINDIML)
                   (SETQ OTHERDIM MINDIMP)
            else (SETQ MINDIM MINDIMP)
                 (SETQ OTHERDIM MINDIML))
          (SETQ SF1 (if (GREATERP MINDIM 1)
                          then 1
                        elseif (GREATERP MINDIM 0.75)
                          then 0.75
                        elseif (GREATERP MINDIM 0.5)
                          then 0.5
                        elseif (GREATERP MINDIM 0.25)
                          then 0.25
                        else MINDIM))
          (SETQ SF2 (if (GREATERP OTHERDIM 1)
                          then 1
                        elseif (GREATERP OTHERDIM 0.75)
                          then 0.75
                        elseif (GREATERP OTHERDIM 0.5)
                          then 0.5
                        elseif (GREATERP OTHERDIM 0.25)
                          then 0.25
                        else OTHERDIM))
          (if (AND (LESSP SF1 1)
                   (LESSP SF1 SF2))
              then (CONS SF2 (NOT PPL))
            else (CONS SF1 PPL])

(POSTSCRIPT.CLOSESTRING
  [LAMBDA (STREAM)                                              ; Edited 12-Jan-88 12:33 by Matt Heffron
    (LET ((IMAGEDATA (fetch IMAGEDATA of STREAM)))
         (if (fetch POSTSCRIPTCHARSTOSHOW of IMAGEDATA)
             then (POSTSCRIPT.OUTSTR STREAM ") ")
                  (replace POSTSCRIPTCHARSTOSHOW of IMAGEDATA with NIL)
                  T
           else NIL])

(POSTSCRIPT.FONTCREATE
  [LAMBDA (FAMILY SIZE FACE ROTATION DEVICE)                    ; Edited 12-Jan-88 12:36 by Matt Heffron
    (LET (UNITFONT FULLNAME SCALEFONTP PSCFD ASCENT DESCENT FIXPWIDTHS CHARSETINFO0 WIDTHSBLOCK FD FACECHANGED
                (WEIGHT (CAR FACE))
                (SLOPE (CADR FACE))
                (EXPANSION (CADDR FACE)))

         ;; Ignore rotations, it is **MUCH** easier to rotate the Postscript stream user space coordinates.

         (if (EQ SIZE 1)
             then
                  ;; Since a 1 point font is rediculously small, and it is the standard size for Postscript font info, a 1 point font is presumed to be the
                  ;; unit size Postscript font info

                  (if (SETQ FULLNAME (PSCFONT.SPELLFILE FAMILY SIZE FACE ROTATION DEVICE))
                      then (SETQ FACECHANGED NIL)
                   elseif (AND (NEQ EXPANSION 'REGULAR)
                               (SETQ FULLNAME (PSCFONT.SPELLFILE FAMILY SIZE (LIST WEIGHT SLOPE 'REGULAR)
                                                     ROTATION DEVICE)))
                      then (SETQ FACECHANGED T)
                   elseif (AND (EQ SLOPE 'ITALIC)
                               (SETQ FULLNAME (PSCFONT.SPELLFILE FAMILY SIZE (LIST WEIGHT 'REGULAR EXPANSION)
                                                     ROTATION DEVICE)))
                      then (SETQ FACECHANGED T)
                   elseif (AND (NEQ EXPANSION 'REGULAR)
                               (EQ SLOPE 'ITALIC)
                               (SETQ FULLNAME (PSCFONT.SPELLFILE FAMILY SIZE (LIST WEIGHT 'REGULAR 'REGULAR)
                                                     ROTATION DEVICE)))
                      then (SETQ FACECHANGED T)
                   elseif (AND (NEQ WEIGHT 'MEDIUM)
                               (SETQ FULLNAME (PSCFONT.SPELLFILE FAMILY SIZE (LIST 'MEDIUM SLOPE EXPANSION)
                                                     ROTATION DEVICE)))
                      then (SETQ FACECHANGED T)
                   elseif (AND (NEQ WEIGHT 'MEDIUM)
                               (NEQ EXPANSION 'REGULAR)
                               (SETQ FULLNAME (PSCFONT.SPELLFILE FAMILY SIZE (LIST 'MEDIUM SLOPE 'REGULAR)
                                                     ROTATION DEVICE)))
                      then (SETQ FACECHANGED T)
                   elseif (AND (NEQ WEIGHT 'MEDIUM)
                               (EQ SLOPE 'ITALIC)
                               (SETQ FULLNAME (PSCFONT.SPELLFILE FAMILY SIZE (LIST 'MEDIUM 'REGULAR EXPANSION)
                                                     ROTATION DEVICE)))
                      then (SETQ FACECHANGED T)
                   elseif (AND (NEQ WEIGHT 'MEDIUM)
                               (NEQ EXPANSION 'REGULAR)
                               (EQ SLOPE 'ITALIC)
                               (SETQ FULLNAME (PSCFONT.SPELLFILE FAMILY SIZE (LIST 'MEDIUM 'REGULAR 'REGULAR)
                                                     ROTATION DEVICE)))
                      then (SETQ FACECHANGED T))
                  [if FULLNAME
                      then (SETQ PSCFD (PSCFONT.READFONT FULLNAME))
                           (SETQ ASCENT (FIXR (TIMES (fetch (PSCFONT ASCENT) of PSCFD)
                                                     0.1)))
                           (SETQ DESCENT (FIXR (TIMES (fetch (PSCFONT DESCENT) of PSCFD)
                                                     0.1)))
                           (if FACECHANGED
                               then (replace (PSCFONT IL-FONTID) of PSCFD with (POSTSCRIPT.GETFONTID
                                                                             (fetch (PSCFONT FID) of PSCFD)
                                                                             WEIGHT SLOPE EXPANSION]
           elseif (SETQ UNITFONT (FONTCREATE FAMILY 1 FACE ROTATION DEVICE T))
             then (SETQ PSCFD (fetch (FONTDESCRIPTOR FONTDEVICESPEC) of UNITFONT))

                  ;; Scale the ASCENT and DESCENT
```

```
                  (SETQ ASCENT (FIXR (TIMES SIZE (fetch (PSCFONT ASCENT) of PSCFD)
                                           0.1)))
                  (SETQ DESCENT (FIXR (TIMES SIZE (fetch (PSCFONT DESCENT) of PSCFD)
                                            0.1)))
                  (SETQ SCALEFONTP T)
           else
               ;; Here for fonts that only come in specific sizes.  Their info is not scaled like built-in Postscript fonts, it is already correct for this
               ;; pointsize.

               (if (SETQ FULLNAME (PSCFONT.SPELLFILE FAMILY SIZE FACE ROTATION DEVICE))
                   then (SETQ PSCFD (PSCFONT.READFONT FULLNAME))
                        (SETQ ASCENT (fetch (PSCFONT ASCENT) of PSCFD))
                        (SETQ DESCENT (fetch (PSCFONT DESCENT) of PSCFD))
                        (SETQ SCALEFONTP NIL)))
         (if PSCFD
             then (SETQ FIXPWIDTHS (fetch (PSCFONT WIDTHS) of PSCFD))
                  (SETQ CHARSETINFO0 (create CHARSETINFO))
                  (SETQ WIDTHSBLOCK (fetch (CHARSETINFO WIDTHS) of CHARSETINFO0))
                  (SETQ FD
                   (create FONTDESCRIPTOR
                          FONTDEVICESPEC _ PSCFD
                          FONTSCALE _ 100.0
                          FONTDEVICE _ DEVICE
                          FONTFAMILY _ FAMILY
                          FONTSIZE _ SIZE
                          FONTFACE _ FACE
                          ROTATION _ 0
                          \SFHeight _ (IPLUS ASCENT DESCENT)
                          \SFAscent _ ASCENT
                          \SFDescent _ DESCENT
                          \SFRWidths _ WIDTHSBLOCK
                          FONTIMAGEWIDTHS _ WIDTHSBLOCK))
                  (replace (CHARSETINFO IMAGEWIDTHS) of CHARSETINFO0 with WIDTHSBLOCK)
                  (replace (CHARSETINFO CHARSETASCENT) of CHARSETINFO0 with ASCENT)
                  (replace (CHARSETINFO CHARSETDESCENT) of CHARSETINFO0 with DESCENT)
                  [if SCALEFONTP
                      then [for CH from 0 to 255 do (\FSETWIDTH WIDTHSBLOCK CH (FIXR (TIMES SIZE (ELT FIXPWIDTHS
                                                                                                   CH)
                                                                                         0.1]
                    else (for CH from 0 to 255 do (\FSETWIDTH WIDTHSBLOCK CH (ELT FIXPWIDTHS CH]
                  (\SETCHARSETINFO (fetch FONTCHARSETVECTOR of FD)
                         0 CHARSETINFO0)
                 FD
           else NIL])

(POSTSCRIPT.FONTSAVAILABLE
  [LAMBDA (FAMILY SIZE FACE ROTATION DEVICE)                    ; Edited 12-Jan-88 13:04 by Matt Heffron

    ;; the filtering code was borrowed from Richard Burton's \SEARCHINTERPRESSFONTS.  Note that without it [HELVETICA * (MEDIUM REGULAR
    ;; REGULAR)]  would pick up [HELVETICA-NARROW * (MEDIUM REGULAR REGULAR)] as well.

    (LET ((PATTERN (\FONTFILENAME (OR (CDR (ASSOC FAMILY POSTSCRIPT.FONT.ALIST))
                                      FAMILY)
                          SIZE FACE 'PSCFONT))
          [INVERSE.ALIST (for PAIR in POSTSCRIPT.FONT.ALIST collect (CONS (CDR PAIR)
                                                                           (CAR PAIR]
          FONTSAVAILABLE)
         (SETQ FONTSAVAILABLE
          (for FD in [for DIRECTORY in POSTSCRIPTFONTDIRECTORIES
                       join (for FILE in (DIRECTORY (CONCAT DIRECTORY PATTERN))
                             collect (LET* ((RAWFD (\FONTINFOFROMFILENAME FILE DEVICE))
                                            (RAWNAME (CAR RAWFD)))
                                           (RPLACA RAWFD (OR (CDR (ASSOC RAWNAME INVERSE.ALIST))
                                                             RAWNAME]
             when (AND (OR (EQ FAMILY '*)
                           (EQ FAMILY (CAR FD)))
                       (OR (EQ SIZE '*)
                           (EQ SIZE (CADR FD))
                           (EQ (CADR FD)
                               1))
                       (OR (EQ FACE '*)
                           (EQUAL FACE (CADDR FD))
                           (EQUAL [CDR (ASSOC FACE '((MRR MEDIUM REGULAR REGULAR)
                                                     (STANDARD MEDIUM REGULAR REGULAR)
                                                     (MIR MEDIUM ITALIC REGULAR)
                                                     (ITALIC MEDIUM ITALIC REGULAR)
                                                     (BRR BOLD REGULAR REGULAR)
                                                     (BOLD BOLD REGULAR REGULAR)
                                                     (BIR BOLD ITALIC REGULAR)
                                                     (BOLDITALIC BOLD ITALIC REGULAR]
                                  (CADDR FD)))
                       (NOT (MEMBER FD $$VAL)))
             collect FD))
         (if (EQ SIZE '*)
             then

;;; If SIZE was wildcarded, then provide list of pointsizes for Postscript scaled fonts (those with a 1 point descriptor file)
```

```
                 (for FD in FONTSAVAILABLE
                    join (if (EQ 1 (CADR FD))
                            then (CONS FD (for NF in (for S from 2 to \POSTSCRIPT.MAX.WILD.FONTSIZE
                                                      collect (LET ((NFD (COPY FD)))
                                                                  (RPLACA (CDR NFD)
                                                                        S)
                                                                 NFD))
                                            unless (MEMBER NF FONTSAVAILABLE) collect NF))
                          else (LIST FD)))
           else FONTSAVAILABLE])

(POSTSCRIPT.GETFONTID
  [LAMBDA (FID WEIGHT SLOPE EXPANSION)                          ; Edited 12-Jan-88 12:58 by Matt Heffron
    (LET (FONTID)
         (SETQ FONTID (create FONTID
                             FONTIDNAME _ (CAR FID)
                             FONTXFACTOR _ 1.0
                             FONTOBLIQUEFACTOR _ 0.0))
         [if (AND (NEQ (CADDR FID)
                       SLOPE)
                  (EQ SLOPE 'ITALIC))
             then (replace FONTOBLIQUEFACTOR of FONTID with (CONSTANT (TAN 7.0]
         (if (AND (NEQ (CADR FID)
                       WEIGHT)
                  (EQ WEIGHT 'BOLD))
             then                                               ; Fake bold by slight expansion.
                  (replace FONTXFACTOR of FONTID with 1.1))
         [if (NEQ EXPANSION 'REGULAR)
             then (replace FONTXFACTOR of FONTID with (TIMES (fetch FONTXFACTOR of FONTID)
                                                          (if (EQ EXPANSION 'COMPRESSED)
                                                              then (CONSTANT (QUOTIENT 1.0 GOLDEN.RATIO))
                                                            else GOLDEN.RATIO]
         FONTID])

(POSTSCRIPT.HARDCOPYW
  [LAMBDA (FILE BITMAP SCALEFACTOR REGION Landscape? TITLE)     ; Edited  4-Feb-88 13:18 by Matt Heffron
    (SPAWN.MOUSE)                                               ; (SETQ Landscape? T) ;Must be landscape to prevent printer
                                                                ; hang??
    (LET ((STREAM (OPENPOSTSCRIPTSTREAM FILE (LIST 'DOCUMENT.NAME TITLE 'ROTATION Landscape? 'IMAGESIZEFACTOR
                                                     SCALEFACTOR)))
          SCLIP W H SCALE)
         [SETQ W (fetch (REGION WIDTH) of (SETQ SCLIP (DSPCLIPPINGREGION NIL STREAM]
         (SETQ H (fetch (REGION HEIGHT) of SCLIP))
         [if REGION
             then (SETQ REGION (COPY REGION))                   ; In case we need to change it.
                  [if (< (BITMAPWIDTH BITMAP)
                         (+ (fetch (REGION LEFT) of REGION)
                            (fetch (REGION WIDTH) of REGION)))
                      then (replace (REGION WIDTH) of REGION with (- (BITMAPWIDTH BITMAP)
                                                                     (fetch (REGION LEFT) of REGION]
                  [if (< (BITMAPHEIGHT BITMAP)
                         (+ (fetch (REGION BOTTOM) of REGION)
                            (fetch (REGION HEIGHT) of REGION)))
                      then (replace (REGION HEIGHT) of REGION with (- (BITMAPHEIGHT BITMAP)
                                                                      (fetch (REGION BOTTOM) of REGION]
           else (SETQ REGION (create REGION
                                   LEFT _ 0
                                   BOTTOM _ 0
                                   WIDTH _ (BITMAPWIDTH BITMAP)
                                   HEIGHT _ (BITMAPHEIGHT BITMAP]
         (SETQ SCALE (TIMES POSTSCRIPT.BITMAP.SCALE (DSPSCALE NIL STREAM)))
         (BITBLT BITMAP (fetch (REGION LEFT) of REGION)
                (fetch (REGION BOTTOM) of REGION)
                STREAM
                (QUOTIENT (DIFFERENCE W (TIMES SCALE (fetch (REGION WIDTH) of REGION)))
                       2)
                (QUOTIENT (DIFFERENCE H (TIMES SCALE (fetch (REGION HEIGHT) of REGION)))
                       2)
                (fetch (REGION WIDTH) of REGION)
                (fetch (REGION HEIGHT) of REGION)
                'INPUT
                'REPLACE)
         (CLOSEF STREAM)
         (FULLNAME STREAM])

(POSTSCRIPT.INIT
  [LAMBDA NIL                                                   ; Edited 29-Mar-89 11:21 by snow
    [MAPC [CL:REMOVE-DUPLICATES (NCONC (for FD in FONTDEFS join (for FP in (CDR (ASSOC 'FONTPROFILE (CDR FD)))
                                                                collect (CAR FP)))
                                       '(FONT7 FONT6 FONT5 FONT4 FONT3 FONT2 FONT1 BOLDFONT LITTLEFONT BIGFONT
                                               PRETTYCOMFONT COMMENTFONT USERFONT SYSTEMFONT CLISPFONT
                                               LAMBDAFONT CHANGEFONT DEFAULTFONT]
          (FUNCTION (LAMBDA (CLASS)
                      (LET (COPYFD OLDPSCFD)
                           (if (BOUNDP CLASS)
                               then (SETQ CLASS (EVALV CLASS))
                                    (if (TYPEP CLASS 'FONTCLASS)
                                        then (SETQ COPYFD (OR (fetch (FONTCLASS PRESSFD) of CLASS)
                                                              (fetch (FONTCLASS INTERPRESSFD) of CLASS)
                                                              (fetch (FONTCLASS DISPLAYFD) of CLASS)))
                                             (if (SETQ OLDPSCFD (ASSOC 'POSTSCRIPT (fetch (FONTCLASS OTHERFDS)
                                                                                      of CLASS)))
                                                 then [if (NOT (CDR OLDPSCFD))
                                                          then (RPLACD OLDPSCFD (if (LISTP COPYFD)
                                                                                    then COPYFD
                                                                                  else (FONTUNPARSE COPYFD]
                                               else (push (fetch (FONTCLASS OTHERFDS) of CLASS)
                                                          (CONS 'POSTSCRIPT (if (LISTP COPYFD)
                                                                                then COPYFD
                                                                              else (FONTUNPARSE COPYFD]
    (SETQ \POSTSCRIPTIMAGEOPS (create IMAGEOPS
                                     IMAGETYPE _ 'POSTSCRIPT
                                     IMCLOSEFN _ (FUNCTION CLOSEPOSTSCRIPTSTREAM)
                                     IMXPOSITION _ (FUNCTION \DSPXPOSITION.PSC)
                                     IMYPOSITION _ (FUNCTION \DSPYPOSITION.PSC)
                                     IMMOVETO _ (FUNCTION \MOVETO.PSC)
                                     IMFONT _ (FUNCTION \DSPFONT.PSC)
                                     IMLEFTMARGIN _ (FUNCTION \DSPLEFTMARGIN.PSC)
                                     IMRIGHTMARGIN _ (FUNCTION \DSPRIGHTMARGIN.PSC)
                                     IMLINEFEED _ (FUNCTION \DSPLINEFEED.PSC)
                                     IMDRAWLINE _ (FUNCTION \DRAWLINE.PSC)
                                     IMDRAWCURVE _ (FUNCTION \DRAWCURVE.PSC)
                                     IMDRAWCIRCLE _ (FUNCTION \DRAWCIRCLE.PSC)
                                     IMDRAWELLIPSE _ (FUNCTION \DRAWELLIPSE.PSC)
                                     IMFILLCIRCLE _ (FUNCTION \FILLCIRCLE.PSC)
                                     IMBLTSHADE _ (FUNCTION \BLTSHADE.PSC)
                                     IMBITBLT _ (FUNCTION \BITBLT.PSC)
                                     IMNEWPAGE _ (FUNCTION \NEWPAGE.PSC)
                                     IMSCALE _ (FUNCTION \DSPSCALE.PSC)
                                     IMTERPRI _ (FUNCTION \TERPRI.PSC)
                                     IMTOPMARGIN _ (FUNCTION \DSPTOPMARGIN.PSC)
                                     IMBOTTOMMARGIN _ (FUNCTION \DSPBOTTOMMARGIN.PSC)
                                     IMSPACEFACTOR _ (FUNCTION \DSPSPACEFACTOR.PSC)
                                     IMFONTCREATE _ 'POSTSCRIPT
                                     IMCLIPPINGREGION _ (FUNCTION \DSPCLIPPINGREGION.PSC)
                                     IMRESET _ (FUNCTION \DSPRESET.PSC)
                                     IMDRAWPOLYGON _ (FUNCTION \DRAWPOLYGON.PSC)
                                     IMFILLPOLYGON _ (FUNCTION \FILLPOLYGON.PSC)
                                     IMSTRINGWIDTH _ (FUNCTION \STRINGWIDTH.PSC)
                                     IMCHARWIDTH _ (FUNCTION \CHARWIDTH.PSC)
                                     IMDRAWARC _ (FUNCTION \DRAWARC.PSC)
                                     IMROTATE _ (FUNCTION \DSPROTATE.PSC)
                                     IMTRANSLATE _ (FUNCTION \DSPTRANSLATE.PSC)
                                     IMDRAWPOINT _ (FUNCTION \DRAWPOINT.PSC)
                                     IMSCALEDBITBLT _ (FUNCTION \BITBLT.PSC])

(POSTSCRIPT.OUTSTR
  [LAMBDA (STREAM STRING)                                          ; Edited 13-Apr-88 16:33 by Matt Heffron
    (if (OR (LITATOM STRING)
           (STRINGP STRING)
           (AND (ZEROP STRING)
                (SETQ STRING "0.0")))
        then [for CI from 1 to (NCHARS STRING) do (BOUT STREAM (LOGAND 255 (NTHCHARCODE STRING CI]
      else (for CC in (CHCON STRING) do (BOUT STREAM (LOGAND 255 CC])

(POSTSCRIPT.PUTBITMAPBYTES
  [LAMBDA (STREAM BITMAP DELIMFLG)
    (DECLARE (GLOBALVARS PS.BITMAPARRAY)
           (LOCALVARS . T))                                        ; Edited 27-Jan-89 11:16 by Matt Heffron
    (LET ((BMBASE (fetch BITMAPBASE of BITMAP))
          (BYTESPERROW (LRSH (IPLUS (fetch BITMAPWIDTH of BITMAP)
                                    7)
                             3))
          (BYTEOFFSETPERROW (LSH (fetch BITMAPRASTERWIDTH of BITMAP)
                                 1))
          (HEIGHT (fetch BITMAPHEIGHT of BITMAP))
          (POS 0)
          (BYTE)
          (PS.BITMAPARRAYBASE (fetch (ARRAYP BASE) of PS.BITMAPARRAY)))

         ;; PS.BITMAPARRAY code speedup by Will Snow @ Envos

         (if DELIMFLG
             then (POSTSCRIPT.OUTSTR STREAM " <
                         "))
         (for R from (SUB1 HEIGHT) to 0 by -1 as ROWOFFSET from (ITIMES (SUB1 HEIGHT)
                                                                      BYTEOFFSETPERROW)
            by (IMINUS BYTEOFFSETPERROW)
            do (for B from 1 to BYTESPERROW as BYTEOFFSET from ROWOFFSET by 1
                  do (if (IGEQ POS 254)
                         then (\BUFFERED.BOUT STREAM (CHARCODE EOL))
                              (SETQ POS 0))
                     (SETQ BYTE (\GETBASEBYTE BMBASE BYTEOFFSET))
                     [\BUFFERED.BOUT STREAM (\GETBASEBYTE PS.BITMAPARRAYBASE (LOGAND 15 (LRSH BYTE 4]
                     (\BUFFERED.BOUT STREAM (\GETBASEBYTE PS.BITMAPARRAYBASE (LOGAND 15 BYTE)))
                     (SETQ POS (IPLUS POS 2)))
               (\BUFFERED.BOUT STREAM (CHARCODE EOL))
               (SETQ POS 0))
         (if DELIMFLG
             then (POSTSCRIPT.OUTSTR STREAM ">
                         "])
```

(POSTSCRIPT.PUTCOMMAND

```
  [LAMBDA S.STRS                                                ; Edited 12-Jan-88 13:01 by Matt Heffron
    (LET ((STREAM (ARG S.STRS 1)))
         (POSTSCRIPT.SHOWACCUM STREAM)
         (for STR# from 2 to S.STRS do (POSTSCRIPT.OUTSTR STREAM (ARG S.STRS STR#])
```

(POSTSCRIPT.SHOWACCUM

```
  [LAMBDA (STREAM)                                              ; Edited 12-Jan-88 16:06 by Matt Heffron
    (if (POSTSCRIPT.CLOSESTRING STREAM)
        then (LET* ((IMAGEDATA (fetch IMAGEDATA of STREAM))
                    (SPACEFACTOR (fetch POSTSCRIPTSPACEFACTOR of IMAGEDATA))
                    WIDTH)
                   (if (EQP SPACEFACTOR 1)
                       then (POSTSCRIPT.OUTSTR STREAM "show ")
                     else (replace POSTSCRIPTSPACEFACTOR of IMAGEDATA with 1)
                          (SETQ WIDTH (\CHARWIDTH.PSC STREAM (CHARCODE SPACE)))
                          (replace POSTSCRIPTSPACEFACTOR of IMAGEDATA with SPACEFACTOR)
                          (POSTSCRIPT.OUTSTR STREAM (TIMES WIDTH (DIFFERENCE SPACEFACTOR 1)))
                          (POSTSCRIPT.OUTSTR STREAM " 0 ")
                          (POSTSCRIPT.OUTSTR STREAM (CHARCODE SPACE))
                          (POSTSCRIPT.OUTSTR STREAM " 4 -1 roll widthshow "])
```

(POSTSCRIPT.STARTPAGE

```
  [LAMBDA (STREAM)                                              ; Edited  9-Sep-88 10:48 by Matt Heffron
    (LET* ((IMAGEDATA (fetch (STREAM IMAGEDATA) of STREAM))
           (CLIPREGN (fetch POSTSCRIPTCLIPPINGREGION of IMAGEDATA))
           (CFONT (fetch POSTSCRIPTFONT of IMAGEDATA))
           LEFT BOTTOM WIDTH HEIGHT)
          (POSTSCRIPT.PUTCOMMAND STREAM "
                 %%%%BeginPageSetup
                 ")
          (if (fetch POSTSCRIPTLANDSCAPE of IMAGEDATA)
              then (POSTSCRIPT.PUTCOMMAND STREAM "xmax ymin translate 90 rotate
                        ")
                   (if (OR (NOT (ZEROP \POSTSCRIPT.SHORTEDGE.SHIFT))
                           (NOT (ZEROP \POSTSCRIPT.LONGEDGE.SHIFT)))
                       then (POSTSCRIPT.PUTCOMMAND STREAM \POSTSCRIPT.SHORTEDGE.SHIFT " " (MINUS
                                                                                          \POSTSCRIPT.LONGEDGE.SHIFT
                                                                                                     )
                                 " translate
                                 "))
            else (if (AND (ZEROP \POSTSCRIPT.LONGEDGE.SHIFT)
                          (ZEROP \POSTSCRIPT.SHORTEDGE.SHIFT))
                     then (POSTSCRIPT.PUTCOMMAND STREAM "xmin ymin translate
                               ")
                   else (POSTSCRIPT.PUTCOMMAND STREAM "xmin " \POSTSCRIPT.LONGEDGE.SHIFT " add ymin "
                               \POSTSCRIPT.SHORTEDGE.SHIFT " add translate
                               ")))
          (POSTSCRIPT.PUTCOMMAND STREAM "0.01 imagesizefactor mul 0.01 imagesizefactor mul scale
                 %%%%EndPageSetup
                 /savepage save def")

          ;; Since the clipping region is per page in Postscript by virtue of the savepage ..., reset the current clipping region for this page.

          (SETQ LEFT (fetch LEFT of CLIPREGN))
          (SETQ BOTTOM (fetch BOTTOM of CLIPREGN))
          (SETQ WIDTH (fetch (REGION WIDTH) of CLIPREGN))
          (SETQ HEIGHT (fetch (REGION HEIGHT) of CLIPREGN))
          (POSTSCRIPT.PUTCOMMAND STREAM "
                 newpath " LEFT " " BOTTOM " mto " WIDTH " 0 rlineto 0 " HEIGHT " rlineto " (IMINUS WIDTH)
                 " 0 rlineto closepath
                  clip newpath
                 ")

          ;; It seems that Lisp depends on the current font being carried over from page to page, so reset it explicitly here.

          (replace POSTSCRIPTFONT of IMAGEDATA with NIL)          ; There is no FONT at the beginning of a page.
          (if (fetch POSTSCRIPTHEADING of IMAGEDATA)
              then
                   ;; Here we handle headings.  This imitates the INTERPRESS code.

                   (\DSPFONT.PSC STREAM (fetch POSTSCRIPTHEADINGFONT of IMAGEDATA))
                   (\DSPRESET.PSC STREAM)
                   (PRIN3 (fetch POSTSCRIPTHEADING of IMAGEDATA)
                                     STREAM)
                           (RELMOVETO 7200 0 STREAM)                                  ; Skip an inch before page number
                           (PRIN3 "Page " STREAM)
                           (PRIN3 (CL:INCF (fetch POSTSCRIPTPAGENUM of IMAGEDATA))
                                  STREAM)
                           (\TERPRI.PSC STREAM)                                       ; Skip 2 lines
                           (\TERPRI.PSC STREAM)
                           (\DSPFONT.PSC STREAM CFONT)
                 else (\DSPFONT.PSC STREAM CFONT)
                        (\DSPRESET.PSC STREAM])

(POSTSCRIPT.TEDIT
   [LAMBDA (FILE PFILE)                                                               ; Edited 12-Jan-88 13:03 by Matt Heffron
      (SETQ FILE (OPENTEXTSTREAM FILE))
      (TEDIT.FORMAT.HARDCOPY FILE PFILE T NIL NIL NIL 'POSTSCRIPT)
      (CLOSEF? FILE)
     PFILE])

(POSTSCRIPT.TEXT
   [LAMBDA (FILE PSCFILE FONTS HEADING TABS)                                          ; Edited 12-Jan-88 13:03 by Matt Heffron
      (TEXTTOIMAGEFILE FILE PSCFILE 'POSTSCRIPT FONTS HEADING TABS (if POSTSCRIPT.TEXTFILE.LANDSCAPE
                                                                        then '(ROTATION T)
                                                                      else NIL])

(POSTSCRIPTFILEP
   [LAMBDA (FILE)                                                                     ; Edited  4-Apr-88 16:31 by Matt Heffron
      (OR (CL:MEMBER (UNPACKFILENAME.STRING FILE 'EXTENSION)
                 '("PS" "PSC")
                 :TEST
                 (FUNCTION STRING-EQUAL))
          (CL:UNWIND-PROTECT
                [PROGN (SETQ FILE (OPENSTREAM FILE 'INPUT))
                          (AND (EQ (BIN FILE)
                                   (CHARCODE %%))
                               (EQ (BIN FILE)
                                   (CHARCODE !]
                (CLOSEF? FILE))])

(PSCFONT.READFONT
   [LAMBDA (FONTFILENAME)                                                             ; Edited 15-Oct-87 11:10 by Matt Heffron
      (LET ((PF (create PSCFONT))
              [S (OPENSTREAM FONTFILENAME 'INPUT NIL '((SEQUENTIAL T]
              FID W)
             [replace (PSCFONT FID) of PF with (SETQ FID (READ S (FIND-READTABLE "INTERLISP"]
             (CL:DO NIL
                      ((EQ (BIN S)
                             255))

                   ;; Body of the loop is empty, the test does all of the work

                   )
             (replace (PSCFONT IL-FONTID) of PF with (CAR FID))
             (replace (PSCFONT FIRSTCHAR) of PF with (\WIN S))
             (replace (PSCFONT LASTCHAR) of PF with (\WIN S))
             (replace (PSCFONT ASCENT) of PF with (\WIN S))
             (replace (PSCFONT DESCENT) of PF with (\WIN S))
             (replace (PSCFONT WIDTHS) of PF with (SETQ W (ARRAY 256 'SMALLPOSP 0 0)))
             (for C from 0 to 255 do (SETA W C (\WIN S)))
             (CLOSEF S)
             PF])

(PSCFONT.SPELLFILE
   [LAMBDA (FAMILY SIZE FACE ROTATION DEVICE)                                         ; Edited 12-Jan-88 13:04 by Matt Heffron
      (SETQ FAMILY (OR (CDR (ASSOC FAMILY POSTSCRIPT.FONT.ALIST))
                        FAMILY))
      (bind FULLNAME for PATH in POSTSCRIPTFONTDIRECTORIES thereis [SETQ FULLNAME (INFILEP (CONCAT
                                                                                                  PATH
                                                                                                  (\FONTFILENAME
                                                                                                   FAMILY SIZE FACE
                                                                                                   '.PSCFONT]
          finally (RETURN FULLNAME])

(PSCFONT.WRITEFONT
   [LAMBDA (FONTFILENAME PF)                                                          ; Edited 15-Oct-87 11:12 by Matt Heffron
      (LET ([S (OPENSTREAM FONTFILENAME 'OUTPUT NIL '((TYPE BINARY)
                                                          (SEQUENTIAL T]
              (W (fetch (PSCFONT WIDTHS) of PF))
              (*READTABLE* (FIND-READTABLE "INTERLISP")))
             (PRIN3 (fetch (PSCFONT FID) of PF)
                      S)
             (BOUT S 0)
             (BOUT S 255)
```

```
         (\WOUT S (fetch (PSCFONT FIRSTCHAR) of PF))
         (\WOUT S (fetch (PSCFONT LASTCHAR) of PF))
         (\WOUT S (fetch (PSCFONT ASCENT) of PF))
         (\WOUT S (fetch (PSCFONT DESCENT) of PF))
         (for C from 0 to 255 do (\WOUT S (ELT W C)))
         (CLOSEF S)
         FONTFILENAME])
```

(READ-AFM-FILE

```
  [LAMBDA (FILE)                                                  ; Edited 20-Jan-88 17:22 by Matt Heffron
    (LET ((IFILE (OPENSTREAM FILE 'INPUT))
          (PSCFONT (create PSCFONT))
          (FCHAR 1000)
          (LCHAR 0)
          (W (ARRAY 256 'SMALLPOSP 0 0))
          TOKEN WEIGHT SLOPE CMCOUNT FBBOX)
         (with PSCFONT PSCFONT (repeatuntil (STRING-EQUAL "FontName" (RSTRING IFILE)) do (READCCODE IFILE))
               (repeatwhile (STRING-EQUAL "" (SETQ TOKEN (RSTRING IFILE))) do (READCCODE IFILE))
               [if (NOT (AND (BOUNDP 'WeightMenu)
                             (type? MENU WeightMenu)))
                   then (SETQ WeightMenu (create MENU
                                                 ITEMS _ WeightMenuItems
                                                 MENUFONT _ (FONTCREATE 'HELVETICA 12]
               [if (NOT (AND (BOUNDP 'SlopeMenu)
                             (type? MENU SlopeMenu)))
                   then (SETQ SlopeMenu (create MENU
                                                ITEMS _ SlopeMenuItems
                                                MENUFONT _ (FONTCREATE 'HELVETICA 12]
               (printout T T "Font WEIGHT for " PSCFONT ": " (SETQ WEIGHT (MENU WeightMenu))
                      T)
               (printout T T "Font SLOPE for " PSCFONT ": " (SETQ SLOPE (MENU SlopeMenu))
                      T)
               (SETQ FID (LIST TOKEN WEIGHT SLOPE 'REGULAR))
               [SETQ IL-FONTID (if (AND (EQ SLOPE 'REGULAR)
                                        (EQ WEIGHT 'MEDIUM))
                                   then TOKEN
                                 else (POSTSCRIPT.GETFONTID FID WEIGHT SLOPE 'REGULAR]
               (repeatuntil (STRING-EQUAL "StartCharMetrics" TOKEN) do (SETQ TOKEN (RSTRING IFILE))
                                                                       (if (STRING-EQUAL "FontBBox" TOKEN)
                                                                           then (SETQ FBBOX (LIST (READ IFILE)
                                                                                                  (READ IFILE)
                                                                                                  (READ IFILE)
                                                                                                  (READ IFILE)))

                                                    ;; The Ascender and Descender properties from the AFM file are currently ignored,
                                                    ;; and the values from the FontBBox are used.

                                                                                (SETQ DESCENT (IABS (CADR FBBOX)))
                                                                                (SETQ ASCENT (CADDDR FBBOX))
                                                                         else (READCCODE IFILE)))
               (SETQ CMCOUNT (RATOM IFILE))
               (repeatuntil (EQ (CHARCODE EOL)
                                (READCCODE IFILE))
                  do)
               (SETQ WIDTHS W)
               (for CC from 1 to CMCOUNT do (LET (CCODE)
                                                 (repeatuntil (EQ 'C (RATOM IFILE)) do)
                                                 (SETQ CCODE (READ IFILE))
                                                 (if (CL:PLUSP CCODE)
                                                     then (if (ILESSP CCODE FCHAR)
                                                              then (SETQ FCHAR CCODE))
                                                          (if (IGREATERP CCODE LCHAR)
                                                              then (SETQ LCHAR CCODE))
                                                          (RATOMS 'WX IFILE)
                                                          (SETA W CCODE (READ IFILE)))
                                                 (repeatuntil (EQ (CHARCODE EOL)
                                                                  (READCCODE IFILE))
                                                    do)))
               (SETQ FIRSTCHAR FCHAR)
               (SETQ LASTCHAR LCHAR))
         (CLOSEF IFILE)
         PSCFONT])
```

(\BITBLT.PSC

```
  [LAMBDA (SOURCEBITMAP SOURCELEFT SOURCEBOTTOM STREAM DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT SOURCETYPE
                 OPERATION TEXTURE CLIPPINGREGION CLIPPEDSOURCELEFT CLIPPEDSOURCEBOTTOM SCALEFACTOR)
                                                                  ; Edited 29-Mar-89 18:31 by snow

    ;; Added SCALEFACTOR so this can be used by both IMBITBLT and IMSCALEDBITBLT.  --was 29-Mar-89

    ;; Postscript can only handle OPERATION REPLACE and PAINT.  SOURCETYPE = TEXTURE is converted to BLTSHADE before getting here (so
    ;; the TEXTURE argument can be ignored) (What are the CLIPPEDSOURCELEFT & CLIPPEDSOURCEBOTTOM arguments?  They are not
    ;; documented)

    (LET (RGN LEFT BOTTOM TEMPBM (SCALE (DSPSCALE NIL STREAM)))

         ;; scaledbitblt may pass nil as DESTINATIONLEFT or DESTINATIONBOTTOM, which means print this at the current position.

         (SETQ DESTINATIONLEFT (OR DESTINATIONLEFT (DSPXPOSITION NIL STREAM)))
         (SETQ DESTINATIONBOTTOM (OR DESTINATIONBOTTOM (DSPYPOSITION NIL STREAM)))
         (SETQ RGN (create REGION
                          LEFT _ (QUOTIENT DESTINATIONLEFT SCALE)
                          BOTTOM _ (QUOTIENT DESTINATIONBOTTOM SCALE)
                          WIDTH _ WIDTH
                          HEIGHT _ HEIGHT))
         (if CLIPPINGREGION
             then (SETQ RGN (INTERSECTREGIONS CLIPPINGREGION RGN))
                  (SETQ LEFT (TIMES (fetch (REGION LEFT) of RGN)
                                    SCALE))
                  (SETQ BOTTOM (TIMES (fetch (REGION BOTTOM) of RGN)
                                      SCALE))
                  (SETQ WIDTH (fetch (REGION WIDTH) of RGN))
                  (SETQ HEIGHT (fetch (REGION HEIGHT) of RGN))
           else (SETQ LEFT DESTINATIONLEFT)
                (SETQ BOTTOM DESTINATIONBOTTOM))
         (if RGN
             then (SETQ TEMPBM (BITMAPCREATE WIDTH HEIGHT 1))
                  (BITBLT SOURCEBITMAP SOURCELEFT SOURCEBOTTOM TEMPBM 0 0 WIDTH HEIGHT SOURCETYPE 'REPLACE)
                  (SETQ SCALE (TIMES SCALE (OR (AND (BOUNDP 'POSTSCRIPT.BITMAP.SCALE)
                                                    (NUMBERP POSTSCRIPT.BITMAP.SCALE))
                                               1)
                                     (OR SCALEFACTOR 1)))
                  (POSTSCRIPT.PUTCOMMAND STREAM "
                         /bitbltsave save def " LEFT " " BOTTOM " translate " (TIMES SCALE WIDTH)
                         " "
                         (TIMES SCALE HEIGHT)
                         " scale " WIDTH " " HEIGHT (if (EQ OPERATION 'PAINT)
                                                        then " true"
                                                      else " false")
                         " thebitimage
                         ")
                  (POSTSCRIPT.PUTBITMAPBYTES STREAM TEMPBM NIL)
                  (POSTSCRIPT.OUTSTR STREAM "
                         bitbltsave restore
                         ")
                  (\MOVETO.PSC STREAM DESTINATIONLEFT DESTINATIONBOTTOM)
                  T
           else NIL])

(\BLTSHADE.PSC
  [LAMBDA (TEXTURE STREAM DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT OPERATION CLIPPINGREGION)
                                                                  ; Edited  9-Sep-88 10:56 by Matt Heffron

    ;; Maybe we should do something with OPERATION

    (LET (TEXTUREBM TEXTUREWIDTH LEFT BOTTOM RGN)
         (if CLIPPINGREGION
             then (SETQ RGN (INTERSECTREGIONS CLIPPINGREGION
                                   (create REGION
                                          LEFT _ DESTINATIONLEFT
                                          BOTTOM _ DESTINATIONBOTTOM
                                          WIDTH _ WIDTH
                                          HEIGHT _ HEIGHT)))
                  (SETQ LEFT (fetch (REGION LEFT) of RGN))
                  (SETQ BOTTOM (fetch (REGION BOTTOM) of RGN))
                  (SETQ WIDTH (fetch (REGION WIDTH) of RGN))
                  (SETQ HEIGHT (fetch (REGION HEIGHT) of RGN))
           else (SETQ RGN T)
                (SETQ LEFT DESTINATIONLEFT)
                (SETQ BOTTOM DESTINATIONBOTTOM))
         (if RGN
             then (POSTSCRIPT.PUTCOMMAND STREAM "
                         gsave
                          newpath ")
                  (if (FIXP TEXTURE)
                      then (if (ZEROP TEXTURE)
                               then (SETQ TEXTURE 1.0)            ; The setgray version of white
                             elseif (OR (EQL TEXTURE 65535)
                                        (EQL TEXTURE -1))
                               then (SETQ TEXTURE 0.0)            ; The setgray version of black
                                    ))
                  (if (FLOATP TEXTURE)
                      then ;; If TEXTURE is a FLOATP, then it is specified in PostScript setgray notation.

                           (POSTSCRIPT.PUTCOMMAND STREAM TEXTURE " setgray ")
                    elseif (OR (TEXTUREP TEXTURE)
                               (NULL TEXTURE))
                      then (SETQ TEXTUREBM (BITMAPCREATE 16 16 1))
                           (SETQ TEXTUREWIDTH 16)
                           (BLTSHADE TEXTURE TEXTUREBM)
                    elseif (BITMAPP TEXTURE)
                      then (SETQ TEXTUREWIDTH (MIN (fetch BITMAPWIDTH of TEXTUREBM)
                                                   (fetch BITMAPHEIGHT of TEXTUREBM)))
                           (SETQ TEXTUREBM (BITMAPCREATE TEXTUREWIDTH TEXTUREWIDTH 1))
                           (BITBLT TEXTURE 0 0 TEXTUREBM 0 0 TEXTUREWIDTH TEXTUREWIDTH 'INPUT 'REPLACE))
                  (if TEXTUREBM
                     then (POSTSCRIPT.PUTCOMMAND STREAM "100 100 scale " (QUOTIENT LEFT 100.0)
                                 " "
                                 (QUOTIENT BOTTOM 100.0)
                                 " mto "
                                 (SETQ WIDTH (QUOTIENT WIDTH 100.0))
                                 " 0 rlineto 0 "
                                 (QUOTIENT HEIGHT 100.0)
                                 " rlineto "
                                 (MINUS WIDTH)
                                 " 0 rlineto closepath
                                 ")
                          (POSTSCRIPT.PUTBITMAPBYTES STREAM TEXTUREBM T)
                          (POSTSCRIPT.PUTCOMMAND STREAM TEXTUREWIDTH " " (LSH (fetch BITMAPRASTERWIDTH
                                                                                    of TEXTUREBM)
                                                                                 1)
                                 " 0 "
                                 (TIMES 72 (QUOTIENT (DSPSCALE NIL STREAM)
                                                  100.0))
                                 " findresolution " TEXTUREWIDTH " div div ceiling " POSTSCRIPT.TEXTURE.SCALE
                                 " mul setpattern eofill
                                 grestore
                                 ")
                   else (POSTSCRIPT.PUTCOMMAND STREAM LEFT " " BOTTOM " mto " WIDTH " 0 rlineto 0 " HEIGHT
                               " rlineto " (MINUS WIDTH)
                               " 0 rlineto closepath eofill
                               grestore
                               "))
                  (\MOVETO.PSC STREAM DESTINATIONLEFT DESTINATIONBOTTOM)
                  T
           else NIL])

(\CHARWIDTH.PSC
  [LAMBDA (STREAM CHARCODE)                                        ; Edited 12-Jan-88 15:54 by Matt Heffron
   ;; no NS character set treatment yet
    (LET (WID SPACEFACTOR (IMAGEDATA (fetch IMAGEDATA of STREAM)))
         (SETQ WID (\FGETWIDTH (fetch FONTIMAGEWIDTHS of (fetch POSTSCRIPTFONT of IMAGEDATA))
                          (LOGAND CHARCODE 255)))
         (if (AND (EQ CHARCODE (CHARCODE SPACE))
                 (NOT (EQP (SETQ SPACEFACTOR (fetch POSTSCRIPTSPACEFACTOR of IMAGEDATA))
                           1)))
             then (FIXR (TIMES WID SPACEFACTOR))
           else WID])

(\DRAWARC.PSC
  [LAMBDA (STREAM CENTERX CENTERY RADIUS STARTANGLE NDEGREES BRUSH DASHING)
                                                                   ; Edited  9-Sep-88 10:59 by Matt Heffron
    (LET (WIDTH COLOR)
         (if (NUMBERP BRUSH)
             then (SETQ WIDTH BRUSH)
           elseif (LISTP BRUSH)
             then (if (NEQ (fetch BRUSHSHAPE of BRUSH)
                          'ROUND)
                      then (printout T T "[In \DRAWARC.PSC: Non-ROUND BRUSH not supported.]
                                 [Using ROUND BRUSH]" T))
                  (SETQ WIDTH (fetch BRUSHSIZE of BRUSH))
                  (SETQ COLOR (fetch BRUSHCOLOR of BRUSH))
           else                                                    ; If FUNCTIONAL BRUSH big trouble!
               (printout T T "[In \DRAWARC.PSC: Functional BRUSH not supported.]
                      [Using ROUND 1 point BRUSH]" T)
               (SETQ WIDTH (DSPSCALE NIL STREAM)))
         (if (NOT (ZEROP WIDTH))
             then (POSTSCRIPT.PUTCOMMAND STREAM "
                        gsave newpath ")
                  (if (FLOATP COLOR)
                      then (POSTSCRIPT.PUTCOMMAND STREAM COLOR " setgray ")
                                                                   ; COLOR is specified in POSTSCRIPT setgray notation.
                          )
                  (if (LISTP DASHING)
                      then (POSTSCRIPT.OUTSTR STREAM " [")
                          (for D in DASHING do (POSTSCRIPT.PUTCOMMAND STREAM (TIMES D WIDTH)
                                                                       " "))
                          (POSTSCRIPT.OUTSTR STREAM "] 0 setdash
                                 ")                                ; Since Interlisp DASHING are in terms of BRUSH units, we must
                                                                   ; multiply by the brush size.

                          )
                  (POSTSCRIPT.PUTCOMMAND STREAM WIDTH " setlinewidth 1 setlinecap 1 setlinejoin " CENTERX " "
                        CENTERY " " RADIUS " " STARTANGLE " " (+ STARTANGLE NDEGREES)
                        " arc stroke
                        grestore"))
         (\MOVETO.PSC STREAM CENTERX CENTERY])

(\DRAWCIRCLE.PSC
  [LAMBDA (STREAM CENTERX CENTERY RADIUS BRUSH DASHING)          ; Edited  9-Sep-88 10:59 by Matt Heffron
    (LET (WIDTH COLOR)
         (if (NUMBERP BRUSH)
             then (SETQ WIDTH BRUSH)
           elseif (LISTP BRUSH)
             then (if (NEQ (fetch BRUSHSHAPE of BRUSH)
                              'ROUND)
                      then (printout T T "[In \DRAWCIRCLE.PSC: Non-ROUND BRUSH not supported.]
                                      [Using ROUND BRUSH]" T))
                  (SETQ WIDTH (fetch BRUSHSIZE of BRUSH))
                  (SETQ COLOR (fetch BRUSHCOLOR of BRUSH))
           else                                                     ; If FUNCTIONAL BRUSH big trouble!
               (printout T T "[In \DRAWCIRCLE.PSC: Functional BRUSH not supported.]
                      [Using (ROUND 1) BRUSH]" T)
               (SETQ WIDTH 1))
         (if (NOT (ZEROP WIDTH))
             then (POSTSCRIPT.PUTCOMMAND STREAM "
                        gsave newpath ")
                  (if (FLOATP COLOR)
                      then (POSTSCRIPT.PUTCOMMAND STREAM COLOR " setgray ")
                                                                    ; COLOR is specified in POSTSCRIPT setgray notation.
                           )
                  (if (LISTP DASHING)
                      then (POSTSCRIPT.OUTSTR STREAM " [")
                           (for D in DASHING do (POSTSCRIPT.PUTCOMMAND STREAM (TIMES D WIDTH)
                                                       " "))
                           (POSTSCRIPT.OUTSTR STREAM "] 0 setdash
                                  ")                                ; Since Interlisp DASHING are in terms of BRUSH units, we must
                                                                    ; multiply by the brush size.

                           )
                  (POSTSCRIPT.PUTCOMMAND STREAM WIDTH " setlinewidth 1 setlinecap 1 setlinejoin " CENTERX " "
                        CENTERY " " RADIUS " 0 360 arc stroke
                        grestore"))
         (\MOVETO.PSC STREAM CENTERX CENTERY])

(\DRAWCURVE.PSC
  [LAMBDA (STREAM KNOTS CLOSED BRUSH DASHING)                       ; Edited  9-Sep-88 10:56 by Matt Heffron
    (LET (WIDTH SHAPE COLOR PSPLINE XA YA DXA DYA N PREVX PREVY PREV-DX3 PREV-DY3)
         (if (NUMBERP BRUSH)
             then (SETQ WIDTH BRUSH)
                  (SETQ SHAPE 'ROUND)
           elseif (LISTP BRUSH)
             then (SETQ WIDTH (fetch BRUSHSIZE of BRUSH))
                  (SETQ SHAPE (fetch BRUSHSHAPE of BRUSH))
                  (SETQ COLOR (fetch BRUSHCOLOR of BRUSH))
           else ;; If FUNCTIONAL BRUSH then BIG trouble!

               (printout T T "[In \DRAWCURVE.PSC: Functional BRUSH not supported.]
                      [Using (ROUND 1) BRUSH]" T)
               (SETQ WIDTH (DSPSCALE NIL STREAM))
               (SETQ SHAPE 'ROUND))
         (if (NOT (ZEROP WIDTH))
             then (POSTSCRIPT.PUTCOMMAND STREAM "
                        gsave newpath ")
                  (if (FLOATP COLOR)
                      then (POSTSCRIPT.PUTCOMMAND STREAM COLOR " setgray ")

                           ;; COLOR is specified in POSTSCRIPT setgray notation.
                           )
                  (if (LISTP DASHING)
                      then (POSTSCRIPT.OUTSTR STREAM " [")
                           (for D in DASHING do (POSTSCRIPT.PUTCOMMAND STREAM (TIMES D WIDTH)
                                                       " "))
                           (POSTSCRIPT.OUTSTR STREAM "] 0 setdash
                                  ")

                           ;; Since Interlisp DASHING are in terms of BRUSH units, we must multiply by the brush size.

                           )
                  (SETQ PSPLINE (PARAMETRICSPLINE KNOTS CLOSED NIL))
                  (SETQ N (pop PSPLINE))
                  (SETQ XA (pop PSPLINE))
                  (SETQ YA (pop PSPLINE))
                  (SETQ DXA (pop PSPLINE))
                  (SETQ DYA (pop PSPLINE))
                  (POSTSCRIPT.PUTCOMMAND STREAM (SELECTQ SHAPE
                                                     (ROUND " 1 setlinecap 1 setlinejoin ")
                                                     (SQUARE " 2 setlinecap 0 setlinejoin ")
                                                     " 0 setlinecap 0 setlinejoin ")
                        WIDTH " setlinewidth " (SETQ PREVX (ELT XA 1))
                        " "
                        (SETQ PREVY (ELT YA 1))
                        " mto
                        ")
                  (SETQ PREV-DX3 (FQUOTIENT (ELT DXA 1)
                                        3.0))
                  (SETQ PREV-DY3 (FQUOTIENT (ELT DYA 1)
                                        3.0))
                  (for C from 2 to N do (POSTSCRIPT.PUTCOMMAND STREAM (FPLUS PREVX PREV-DX3)
                                              " "
                                              (FPLUS PREVY PREV-DY3)
                                              " "
                                              (FDIFFERENCE (SETQ PREVX (ELT XA C))
                                                     (SETQ PREV-DX3 (FQUOTIENT (ELT DXA C)
                                                                           3.0)))
                                              " "
                                              (FDIFFERENCE (SETQ PREVY (ELT YA C))
                                                     (SETQ PREV-DY3 (FQUOTIENT (ELT DYA C)
                                                                           3.0)))
                                              " " PREVX " " PREVY " curveto
                                              "))
                  (POSTSCRIPT.PUTCOMMAND STREAM "stroke
                         grestore"))
         (\MOVETO.PSC STREAM PREVX PREVY))
    NIL])
```

(\DRAWELLIPSE.PSC

```
  [LAMBDA (STREAM CENTERX CENTERY MINORRADIUS MAJORRADIUS ORIENTATION BRUSH DASHING)
                                                                 ; Edited  9-Sep-88 10:59 by Matt Heffron
    (LET (WIDTH COLOR)
         (if (NUMBERP BRUSH)
             then (SETQ WIDTH BRUSH)
           elseif (LISTP BRUSH)
             then (if (NEQ (fetch BRUSHSHAPE of BRUSH)
                           'ROUND)
                      then (printout T T "[In \DRAWELLIPSE.PSC: Non-ROUND BRUSH not supported.]
                                  [Using ROUND BRUSH]" T))
                  (SETQ WIDTH (fetch BRUSHSIZE of BRUSH))
                  (SETQ COLOR (fetch BRUSHCOLOR of BRUSH))
           else                                                  ; If FUNCTIONAL BRUSH, big trouble!
               (printout T T "[In \DRAWELLIPSE.PSC: Functional BRUSH not supported.]
                      [Using (ROUND 1) BRUSH]" T)
               (SETQ WIDTH 1))
         (if (NOT (ZEROP WIDTH))
             then (POSTSCRIPT.PUTCOMMAND STREAM "
                         gsave newpath ")
                  (if (FLOATP COLOR)
                      then (POSTSCRIPT.PUTCOMMAND STREAM COLOR " setgray ")
                                                                 ; COLOR is specified in POSTSCRIPT setgray notation.
                           )
                  (if (LISTP DASHING)
                      then (POSTSCRIPT.OUTSTR STREAM " [")
                           (for D in DASHING do (POSTSCRIPT.PUTCOMMAND STREAM (TIMES D WIDTH)
                                                       " "))
                           (POSTSCRIPT.OUTSTR STREAM "] 0 setdash
                                  ")                             ; Since Interlisp DASHING are in terms of BRUSH units, we must
                                                                 ; multiply by the brush size.

                           )
                  (POSTSCRIPT.PUTCOMMAND STREAM WIDTH " setlinewidth 1 setlinecap 1 setlinejoin " CENTERX " "
                         CENTERY " " MAJORRADIUS " " MINORRADIUS " " ORIENTATION " 0 360 ellipse stroke
                         grestore"))
         (\MOVETO.PSC STREAM CENTERX CENTERY])
```

(\DRAWLINE.PSC

```
  [LAMBDA (STREAM X1 Y1 X2 Y2 WIDTH OPERATION COLOR DASHING)     ; Edited 22-Feb-89 11:26 by snow
    (LET ((IMAGEDATA (fetch IMAGEDATA of STREAM)))
         (if (NOT (NUMBERP WIDTH))
             then
                 ;; The WIDTH = NIL should have been handled before here, but just in case!

                  (SETQ WIDTH (DSPSCALE NIL STREAM)))
         (if (NOT (ZEROP WIDTH))
             then (POSTSCRIPT.PUTCOMMAND STREAM "
                         gsave newpath ")
                  (if (FLOATP COLOR)
                      then (POSTSCRIPT.PUTCOMMAND STREAM COLOR " setgray ")

                           ;; COLOR is specified in POSTSCRIPT setgray notation.
                           )
                  (if (LISTP DASHING)
                      then (POSTSCRIPT.OUTSTR STREAM " [")
                           (for D in DASHING do (POSTSCRIPT.PUTCOMMAND STREAM (TIMES D WIDTH)
                                                       " "))
                           (POSTSCRIPT.OUTSTR STREAM "] 0 setdash
                                  ")

                           ;; Since Interlisp DASHING are in terms of BRUSH units, we must multiply by the brush size.

                           )
                  (POSTSCRIPT.PUTCOMMAND STREAM WIDTH " setlinewidth 0 setlinecap " X1 " " Y1 " mto " X2 " "
                         Y2 " lineto stroke
                         grestore " X2 " " Y2 " mto "))
         (replace POSTSCRIPTX of IMAGEDATA with X2)
         (replace POSTSCRIPTY of IMAGEDATA with Y2])

(\DRAWPOLYGON.PSC
  [LAMBDA (STREAM POINTS CLOSED BRUSH DASHING)                     ; Edited  9-Sep-88 11:00 by Matt Heffron
    (LET ((LASTPOINT (CAR (LAST POINTS)))
          WIDTH SHAPE COLOR)
         (if (NUMBERP BRUSH)
             then (SETQ WIDTH BRUSH)
                  (SETQ SHAPE 'ROUND)
           elseif (LISTP BRUSH)
             then (SETQ WIDTH (fetch BRUSHSIZE of BRUSH))
                  (SETQ SHAPE (fetch BRUSHSHAPE of BRUSH))
                  (SETQ COLOR (fetch BRUSHCOLOR of BRUSH))
           else ;; If FUNCTIONAL BRUSH then BIG trouble!

                (printout T T "[In \DRAWPOLYGON.PSC: Functional BRUSH not supported.]
                       [Using (ROUND 1) BRUSH]" T)
                (SETQ WIDTH (DSPSCALE NIL STREAM))
                (SETQ SHAPE 'ROUND))
         (if (NOT (ZEROP WIDTH))
             then (POSTSCRIPT.PUTCOMMAND STREAM "
                        gsave newpath ")
                  (if (FLOATP COLOR)
                      then (POSTSCRIPT.PUTCOMMAND STREAM COLOR " setgray ")

                           ;; COLOR is specified in POSTSCRIPT setgray notation.
                           )
                  (if (LISTP DASHING)
                      then (POSTSCRIPT.OUTSTR STREAM " [")
                           (for D in DASHING do (POSTSCRIPT.PUTCOMMAND STREAM (TIMES D WIDTH)
                                                       " "))
                           (POSTSCRIPT.OUTSTR STREAM "] 0 setdash
                                  ")                                 ; Since Interlisp DASHING are in terms of BRUSH units, we must
                                                                     ; multiply by the brush size.

                           )
                  (POSTSCRIPT.PUTCOMMAND STREAM (SELECTQ SHAPE
                                                     (ROUND " 1 setlinecap 1 setlinejoin ")
                                                     (SQUARE " 2 setlinecap 0 setlinejoin ")
                                                     " 0 setlinecap 0 setlinejoin ")
                        WIDTH " setlinewidth " (fetch XCOORD of (CAR POINTS))
                        " "
                        (fetch YCOORD of (CAR POINTS))
                        " mto
                        ")
                  (for P in (CDR POINTS) do (POSTSCRIPT.PUTCOMMAND STREAM (fetch XCOORD of P)
                                                   " "
                                                   (fetch YCOORD of P)
                                                   " lineto
                                                   "))
                  (if CLOSED
                      then (POSTSCRIPT.PUTCOMMAND STREAM " closepath"))
                  (POSTSCRIPT.PUTCOMMAND STREAM " stroke
                        grestore"))
         (\MOVETO.PSC STREAM (fetch XCOORD of LASTPOINT)
                (fetch YCOORD of LASTPOINT])

(\DSPBOTTOMMARGIN.PSC
  [LAMBDA (STREAM YPOSITION)                                       ; Edited 12-Jan-88 13:14 by Matt Heffron
    (PROG1 (fetch POSTSCRIPTBOTTOMMARGIN of (fetch IMAGEDATA of STREAM))
        (if YPOSITION
            then (replace POSTSCRIPTBOTTOMMARGIN of (fetch IMAGEDATA of STREAM) with YPOSITION)))])

(\DSPCLIPPINGREGION.PSC
  [LAMBDA (STREAM REGION)                                          ; Edited 12-Jan-88 13:15 by Matt Heffron
    (LET* ((IMAGEDATA (fetch IMAGEDATA of STREAM))
           (CURRGN (fetch POSTSCRIPTCLIPPINGREGION of IMAGEDATA))
           (SCALE (fetch POSTSCRIPTSCALE of IMAGEDATA))
           (LONGEDGE (TIMES \POSTSCRIPT.LONGEDGE.PTS (QUOTIENT 10000 SCALE)))
           (SHORTEDGE (TIMES \POSTSCRIPT.SHORTEDGE.PTS (QUOTIENT 10000 SCALE)))
           RGN WIDTH DEFREGION)
          (SETQ DEFREGION (if (fetch POSTSCRIPTLANDSCAPE of IMAGEDATA)
                              then (create REGION
                                          LEFT _ 0.0
                                          BOTTOM _ 0.0
                                          WIDTH _ LONGEDGE
                                          HEIGHT _ SHORTEDGE)
                            else (create REGION
                                        LEFT _ 0.0
                                        BOTTOM _ 0.0
                                        WIDTH _ SHORTEDGE
                                        HEIGHT _ LONGEDGE)))
          (if REGION
              then (SETQ RGN (INTERSECTREGIONS REGION DEFREGION))
```

```
                  ;; If the new clipping region doesn't intersect with the default for the appropriate page orientation, just ignore this and reset to
                  ;; the default.

                  (if RGN
                      then (replace POSTSCRIPTCLIPPINGREGION of IMAGEDATA with RGN)
                           (SETQ WIDTH (fetch (REGION WIDTH) of RGN))
                           (POSTSCRIPT.PUTCOMMAND STREAM " initclip newpath " (fetch LEFT of RGN)
                                  " "
                                  (fetch BOTTOM of RGN)
                                  " moveto " WIDTH " 0 rlineto 0 " (fetch (REGION HEIGHT) of RGN)
                                  " rlineto "
                                  (IMINUS WIDTH)
                                  " 0 rlineto closepath
                                   clip newpath")
                    else DEFREGION))
          CURRGN])

(\DSPFONT.PSC
  [LAMBDA (STREAM FONT)                                         ; Edited  9-Sep-88 10:57 by Matt Heffron
    (LET* ((IMAGEDATA (fetch IMAGEDATA of STREAM))
           FONTID)
          (PROG1 (fetch POSTSCRIPTFONT of IMAGEDATA)
              [if FONT
                 then (SETQ FONT (SELECTQ (TYPENAME FONT)
                                      (FONTDESCRIPTOR
                                          FONT)
                                      (FONTCLASS (FONTCREATE FONT NIL NIL NIL STREAM))
                                      (SHOULDNT "arg not FONT descriptor or class")))
                      (if (NEQ (IMAGESTREAMTYPE STREAM)
                               (fetch FONTDEVICE of FONT))
                          then (SETQ FONT (with FONTDESCRIPTOR FONT (FONTCREATE FONTFAMILY FONTSIZE FONTFACE NIL
                                                                         STREAM]
              [if (AND FONT (NEQ FONT (fetch POSTSCRIPTFONT of IMAGEDATA)))
                 then (SETQ FONTID (fetch (PSCFONT IL-FONTID) of (fetch (FONTDESCRIPTOR FONTDEVICESPEC)
                                                                  of FONT)))
                      (if (LISTP FONTID)
                          then (POSTSCRIPT.PUTCOMMAND STREAM "
                                     /" (fetch FONTIDNAME of FONTID)
                                     " findfont ["
                                     (TIMES (fetch FONTXFACTOR of FONTID)
                                            (fetch (FONTDESCRIPTOR FONTSIZE) of FONT)
                                            100)
                                     " 0 "
                                     (TIMES (fetch FONTOBLIQUEFACTOR of FONTID)
                                            (fetch (FONTDESCRIPTOR FONTSIZE) of FONT)
                                            100)
                                     " "
                                     (TIMES (fetch (FONTDESCRIPTOR FONTSIZE) of FONT)
                                            100)
                                     " 0 0] makefont setfont
                                     ")
                        else (POSTSCRIPT.PUTCOMMAND STREAM "
                                   /" FONTID " findfont " (TIMES (fetch (FONTDESCRIPTOR FONTSIZE) of FONT)
                                                                 100)
                                   " scalefont setfont
                                   "))
                      (replace POSTSCRIPTFONT of IMAGEDATA with FONT)
                      (\DSPLINEFEED.PSC STREAM (IMINUS (fetch (FONTDESCRIPTOR \SFHeight) of FONT])])

(\DSPLEFTMARGIN.PSC
  [LAMBDA (STREAM XPOSITION)                                    ; Edited 12-Jan-88 13:15 by Matt Heffron
    (PROG1 (fetch POSTSCRIPTLEFTMARGIN of (fetch IMAGEDATA of STREAM))
        (if XPOSITION
            then (replace POSTSCRIPTLEFTMARGIN of (fetch IMAGEDATA of STREAM) with XPOSITION)))])

(\DSPLINEFEED.PSC
  [LAMBDA (STREAM LINELEADING)                                  ; Edited 12-Jan-88 13:16 by Matt Heffron
    (PROG1 (fetch POSTSCRIPTLINESPACING of (fetch IMAGEDATA of STREAM))
        (if LINELEADING
            then (replace POSTSCRIPTLINESPACING of (fetch IMAGEDATA of STREAM) with LINELEADING)))])

(\DSPRESET.PSC
  [LAMBDA (STREAM)                                              ; Edited  9-Sep-88 11:00 by Matt Heffron
    (LET ((IMAGEDATA (fetch IMAGEDATA of STREAM)))
         (\MOVETO.PSC STREAM (fetch POSTSCRIPTLEFTMARGIN of IMAGEDATA)
                (DIFFERENCE (fetch POSTSCRIPTTOPMARGIN of IMAGEDATA)
                       (FONTPROP (fetch POSTSCRIPTFONT of IMAGEDATA)
                              'ASCENT])

(\DSPRIGHTMARGIN.PSC
  [LAMBDA (STREAM XPOSITION)                                    ; Edited 12-Jan-88 13:16 by Matt Heffron
    (PROG1 (fetch POSTSCRIPTRIGHTMARGIN of (fetch IMAGEDATA of STREAM))
        (if XPOSITION
            then (replace POSTSCRIPTRIGHTMARGIN of (fetch IMAGEDATA of STREAM) with XPOSITION)))])

(\DSPSCALE.PSC
  [LAMBDA (STREAM SCALE)                                             ; Edited 28-Sep-87 13:30 by Matt Heffron
    (LET* ((IMAGEDATA (fetch IMAGEDATA of STREAM))
           (OSCALE (fetch POSTSCRIPTSCALE of IMAGEDATA))
           NSCALE)
          (if (AND NIL

                   ;; Changing SCALE is not implemented.  According to IRM.

                   (NUMBERP SCALE)
                   (CL:PLUSP SCALE))
              then (SETQ NSCALE (QUOTIENT SCALE OSCALE))

                   ;; NSCALE is the adjustment for the fact that the scale operator takes RELATIVE scale changes.

                   (POSTSCRIPT.PUTCOMMAND STREAM " " NSCALE " " NSCALE " scale")
                   (replace POSTSCRIPTSCALE of IMAGEDATA with SCALE))
          OSCALE])

(\DSPSPACEFACTOR.PSC
  [LAMBDA (STREAM FACTOR)                                            ; Edited 12-Jan-88 13:49 by Matt Heffron
    (LET ((IMAGEDATA (fetch IMAGEDATA of STREAM)))
         (PROG1 (fetch POSTSCRIPTSPACEFACTOR of IMAGEDATA)
             (if FACTOR
                 then (POSTSCRIPT.SHOWACCUM STREAM)
                      (replace POSTSCRIPTSPACEFACTOR of IMAGEDATA with FACTOR)))])

(\DSPTOPMARGIN.PSC
  [LAMBDA (STREAM YPOSITION)                                         ; Edited 12-Jan-88 13:17 by Matt Heffron
    (PROG1 (fetch POSTSCRIPTTOPMARGIN of (fetch IMAGEDATA of STREAM))
        (if YPOSITION
            then (replace POSTSCRIPTTOPMARGIN of (fetch IMAGEDATA of STREAM) with YPOSITION)))])

(\DSPXPOSITION.PSC
  [LAMBDA (STREAM XPOSITION)                                         ; Edited  9-Sep-88 10:58 by Matt Heffron
    (LET ((IMAGEDATA (fetch IMAGEDATA of STREAM))
          OLDX)
         (PROG1 (SETQ OLDX (fetch POSTSCRIPTX of IMAGEDATA))
             (if (AND XPOSITION (NOT (EQUAL XPOSITION OLDX)))
                 then (\MOVETO.PSC STREAM XPOSITION (fetch POSTSCRIPTY of IMAGEDATA)))))])

(\DSPYPOSITION.PSC
  [LAMBDA (STREAM YPOSITION)                                         ; Edited  9-Sep-88 10:58 by Matt Heffron
    (LET ((IMAGEDATA (fetch IMAGEDATA of STREAM))
          OLDY)
         (PROG1 (SETQ OLDY (fetch POSTSCRIPTY of IMAGEDATA))
             (if (AND YPOSITION (NOT (EQUAL YPOSITION OLDY)))
                 then (\MOVETO.PSC STREAM (fetch POSTSCRIPTX of IMAGEDATA)
                                 YPOSITION))))])

(\FILLCIRCLE.PSC
  [LAMBDA (STREAM CENTERX CENTERY RADIUS TEXTURE)                    ; Edited  9-Sep-88 11:00 by Matt Heffron
    (LET (TEXTUREBM TEXTUREWIDTH)
         (POSTSCRIPT.PUTCOMMAND STREAM "
              gsave
               newpath ")
         (if (FIXP TEXTURE)
             then (if (ZEROP TEXTURE)
                      then (SETQ TEXTURE 1.0)                        ; The setgray version of white
                    elseif (OR (EQL TEXTURE 65535)
                               (EQL TEXTURE -1))
                      then (SETQ TEXTURE 0.0)                        ; The setgray version of black
                           ))
         (if (FLOATP TEXTURE)
             then
                  ;; If TEXTURE is a FLOATP, then it is specified in PostScript setgray notation.

                  (POSTSCRIPT.PUTCOMMAND STREAM TEXTURE " setgray ")
           elseif (OR (TEXTUREP TEXTURE)
                      (NULL TEXTURE))
             then (SETQ TEXTUREBM (BITMAPCREATE 16 16 1))
                  (SETQ TEXTUREWIDTH 16)
                  (BLTSHADE TEXTURE TEXTUREBM)
           elseif (BITMAPP TEXTURE)
             then (SETQ TEXTUREWIDTH (MIN (fetch BITMAPWIDTH of TEXTUREBM)
                                          (fetch BITMAPHEIGHT of TEXTUREBM)))
                  (SETQ TEXTUREBM (BITMAPCREATE TEXTUREWIDTH TEXTUREWIDTH 1))
                  (BITBLT TEXTURE 0 0 TEXTUREBM 0 0 TEXTUREWIDTH TEXTUREWIDTH 'INPUT 'REPLACE))
         (POSTSCRIPT.PUTCOMMAND STREAM " " CENTERX " " CENTERY " " RADIUS " 0 360 arc
              ")
         (if TEXTUREBM
             then (POSTSCRIPT.PUTCOMMAND STREAM "100 100 scale ")
                  (POSTSCRIPT.PUTBITMAPBYTES STREAM TEXTUREBM T)
                  (POSTSCRIPT.PUTCOMMAND STREAM TEXTUREWIDTH " " (LSH (fetch BITMAPRASTERWIDTH of TEXTUREBM)
                                                                         1)
                         " 0 "
                         (TIMES 72 (QUOTIENT (DSPSCALE NIL STREAM)
                                          100.0))
                         " findresolution " TEXTUREWIDTH " div div ceiling " POSTSCRIPT.TEXTURE.SCALE " mul
                         setpattern eofill
                         grestore
                         ")
           else (POSTSCRIPT.PUTCOMMAND STREAM " eofill
                      grestore
                      "))
         (\MOVETO.PSC STREAM CENTERX CENTERY])

(\FILLPOLYGON.PSC
  [LAMBDA (STREAM KNOTS TEXTURE OPERATION WINDNUMBER)             ; Edited  9-Sep-88 11:01 by Matt Heffron
    (DECLARE (SPECVARS FILL.WRULE))

    ;; OPERATION is ignored here

    (LET ((LASTPOINT (CAR (LAST KNOTS)))
          TEXTUREBM TEXTUREWIDTH)
         (POSTSCRIPT.PUTCOMMAND STREAM "
                gsave
                 newpath ")
         (if (NOT (OR (ZEROP WINDNUMBER)
                      (EQL WINDNUMBER 1)))
             then (SETQ WINDNUMBER FILL.WRULE))
         (if (FIXP TEXTURE)
             then (if (ZEROP TEXTURE)
                      then (SETQ TEXTURE 1.0)                       ; The setgray version of white
                    elseif (OR (EQL TEXTURE 65535)
                               (EQL TEXTURE -1))
                      then (SETQ TEXTURE 0.0)                       ; The setgray version of black
                          ))
         (if (FLOATP TEXTURE)
             then 
                  ;; If TEXTURE is a FLOATP, then it is specified in PostScript setgray notation.

                  (POSTSCRIPT.PUTCOMMAND STREAM TEXTURE " setgray ")
           elseif (OR (TEXTUREP TEXTURE)
                      (NULL TEXTURE))
             then (SETQ TEXTUREBM (BITMAPCREATE 16 16 1))
                  (SETQ TEXTUREWIDTH 16)
                  (BLTSHADE TEXTURE TEXTUREBM)
           elseif (BITMAPP TEXTURE)
             then (SETQ TEXTUREWIDTH (MIN (fetch BITMAPWIDTH of TEXTUREBM)
                                          (fetch BITMAPHEIGHT of TEXTUREBM)))
                  (SETQ TEXTUREBM (BITMAPCREATE TEXTUREWIDTH TEXTUREWIDTH 1))
                  (BITBLT TEXTURE 0 0 TEXTUREBM 0 0 TEXTUREWIDTH TEXTUREWIDTH 'INPUT 'REPLACE))
         (POSTSCRIPT.PUTCOMMAND STREAM (fetch XCOORD of (CAR KNOTS))
                " "
                (fetch YCOORD of (CAR KNOTS))
                " mto
                ")
         (for K in (CDR KNOTS) do (POSTSCRIPT.PUTCOMMAND STREAM (fetch XCOORD of K)
                                         " "
                                         (fetch YCOORD of K)
                                         " lineto
                                         "))
         (POSTSCRIPT.PUTCOMMAND STREAM " closepath
                ")
         (if TEXTUREBM
             then (POSTSCRIPT.PUTCOMMAND STREAM "100 100 scale ")
                  (POSTSCRIPT.PUTBITMAPBYTES STREAM TEXTUREBM T)
                  (POSTSCRIPT.PUTCOMMAND STREAM TEXTUREWIDTH " " (LSH (fetch BITMAPRASTERWIDTH of TEXTUREBM)
                                                                         1)
                         " 0 "
                         (TIMES 72 (QUOTIENT (DSPSCALE NIL STREAM)
                                          100.0))
                         " findresolution " TEXTUREWIDTH " div div ceiling " POSTSCRIPT.TEXTURE.SCALE " mul
                         setpattern"))
         (POSTSCRIPT.PUTCOMMAND STREAM (if (ZEROP WINDNUMBER)
                                           then " fill
                                                grestore
                                                  "
                                         else " eofill
                                              grestore
                                              "))
         (\MOVETO.PSC STREAM (fetch XCOORD of LASTPOINT)
                (fetch YCOORD of LASTPOINT])

(\MOVETO.PSC
  [LAMBDA (STREAM X Y)                                              ; Edited 12-Jan-88 13:18 by Matt Heffron
    (LET ((IMAGEDATA (fetch IMAGEDATA of STREAM)))
         (POSTSCRIPT.PUTCOMMAND STREAM " " X " " Y " mto
               ")
         (with \POSTSCRIPTDATA IMAGEDATA (SETQ POSTSCRIPTX X)
              (SETQ POSTSCRIPTY Y])

(\NEWPAGE.PSC
  [LAMBDA (STREAM)                                               ; Edited 20-Jan-88 17:36 by Matt Heffron
    (POSTSCRIPT.PUTCOMMAND STREAM "
            savepage restore
           showpage")
    (POSTSCRIPT.STARTPAGE STREAM])

(\POSTSCRIPT.OUTCHARFN
  [LAMBDA (STREAM CHAR)                                          ; Edited  9-Sep-88 11:02 by Matt Heffron
    (LET* ((POSTSCRIPTDATA (fetch IMAGEDATA of STREAM)))
          (SELCHARQ CHAR
               ((CR LF TENEXEOL)
                   (\TERPRI.PSC STREAM))
               (FF (\NEWPAGE.PSC STREAM))
               (PROGN (if (NOT (fetch POSTSCRIPTCHARSTOSHOW of POSTSCRIPTDATA))
                          then (POSTSCRIPT.OUTSTR STREAM " (")
                               (replace POSTSCRIPTCHARSTOSHOW of POSTSCRIPTDATA with T))
                      (\POSTSCRIPT.PUTCHAR STREAM CHAR])

(\POSTSCRIPT.PUTCHAR
  [LAMBDA (STREAM CHAR)                                          ; Edited  5-Feb-88 10:29 by Matt Heffron
    (LET* ((POSTSCRIPTDATA (fetch IMAGEDATA of STREAM))
           (FONT (fetch POSTSCRIPTFONT of POSTSCRIPTDATA))
           TEMP)
          (SETQ CHAR (LOGAND CHAR 255))                          ; no NS character set treatment yet
          (if (EQ CHAR (CHARCODE TAB))
              then (RPTQ 8 (\POSTSCRIPT.PUTCHAR STREAM (CHARCODE SPACE)))
                                                                 ; wimpy, but no better way yet.
            else (if (FMEMB CHAR (CHARCODE (%( %) \)))
                     then (BOUT STREAM (CHARCODE \))
                          (BOUT STREAM CHAR)
                   elseif (NOT (<= (CHARCODE SPACE)
                                   CHAR 126))
                     then (BOUT STREAM (CHARCODE \))
                          (SETQ TEMP (CHCON (OCTALSTRING CHAR)))
                          (if (< (LENGTH TEMP)
                                 3)
                              then (SETQ TEMP (APPEND [if (CDR TEMP)
                                                           then (CONSTANT (CHARCODE (0)))
                                                         else (CONSTANT (CHARCODE (0 0]
                                                     TEMP)))
                          (for CC in TEMP do (BOUT STREAM CC))
                   else (BOUT STREAM CHAR))
                 (add (fetch POSTSCRIPTX of POSTSCRIPTDATA)
                      (\FGETWIDTH (fetch FONTIMAGEWIDTHS of FONT)
                             CHAR)))
          CHAR])

(\STRINGWIDTH.PSC
  [LAMBDA (STREAM STR RDTBL)                                     (* DECLARATIONS%: INTEGER)
                                                                 ; Edited 12-Jan-88 13:27 by Matt Heffron
    (LET* [(FNT (DSPFONT NIL STREAM))
           (SPACEFACTOR (fetch POSTSCRIPTSPACEFACTOR of (fetch (STREAM IMAGEDATA) of STREAM)))
           (WA (fetch (PSCFONT WIDTHS) of (fetch (FONTDESCRIPTOR FONTDEVICESPEC) of FNT)))
           (W (for CI from 1 to (NCHARS STR) sum (LET* ((CC (LOGAND 255 (NTHCHARCODE STR CI NIL RDTBL)))
                                                        (WID (ELT WA CC)))
                                                       (if (EQ CC (CHARCODE SPACE))
                                                           then (TIMES WID SPACEFACTOR)
                                                         else WID]
          (FIXR (TIMES W (fetch (FONTDESCRIPTOR FONTSIZE) of FNT)
                       0.1])

(\TERPRI.PSC
  [LAMBDA (STREAM)                                               ; Edited  9-Sep-88 11:02 by Matt Heffron
    (LET ((IMAGEDATA (fetch IMAGEDATA of STREAM)))
         (with \POSTSCRIPTDATA IMAGEDATA (SETQ POSTSCRIPTX POSTSCRIPTLEFTMARGIN)
              (SETQ POSTSCRIPTY (IPLUS POSTSCRIPTY POSTSCRIPTLINESPACING))

              ;; IPLUS because POSTSCRIPTLINESPACING is -ve if correct.

               (if (LESSP POSTSCRIPTY (IPLUS (fetch (FONTDESCRIPTOR \SFDescent) of POSTSCRIPTFONT)
                                            POSTSCRIPTBOTTOMMARGIN))
                   then (\NEWPAGE.PSC STREAM)
                 else (\MOVETO.PSC STREAM POSTSCRIPTX POSTSCRIPTY])

(\DSPROTATE.PSC
  [LAMBDA (STREAM ROTATION)                                      ; Edited 22-Feb-89 13:47 by snow
```

```
     ;; rotate the postscript  stream by ROTATION

     ;; we only know 90 degrees of rotation for now.

     (LET ((IMAGEDATA (fetch (STREAM IMAGEDATA) of STREAM)))
          (replace POSTSCRIPTLANDSCAPE of IMAGEDATA with (IF (EQ ROTATION 0)
                                                             THEN NIL
                                                           ELSE T))
          (\NEWPAGE.PSC STREAM))
     1])

(\DSPTRANSLATE.PSC
  [LAMBDA (STREAM TX TY)                                     ; Edited 22-Feb-89 11:40 by snow

     ;; the translation happens automatically when we do a rotate.  This isn't really a translate function, but it works for the simple rotate by 90 case that
     ;; occurs most often.

     1])

(\DRAWPOINT.PSC
  [LAMBDA (STREAM X Y BRUSH OPERATION)                       ; Edited 22-Feb-89 15:24 by snow

     ;; draw a point on the stream

     (IF (BITMAPP BRUSH)
         THEN (LET ((WIDTH (BITMAPWIDTH BRUSH))
                    (HEIGHT (BITMAPHEIGHT BRUSH)))
                   (BITBLT BRUSH 0 0 STREAM (- X (IQUOTIENT WIDTH 2))
                          (- Y (IQUOTIENT HEIGHT 2))
                          WIDTH HEIGHT OPERATION))
       ELSE (\DRAWLINE.PSC STREAM X Y X Y BRUSH OPERATION])

)

(RPAQ \POSTSCRIPT.ORIENTATION.MENU
      (create MENU ITEMS _ '(("Landscape" T "Print this file/document/image in Landscape Orientation")
                             ("Portrait" 'NIL "Print this file/document/image in Portrait Orientation"))
             TITLE _ "Orientation" CENTERFLG _ T MENUOFFSET _ (create POSITION XCOORD _ -1 YCOORD _ 0)
             CHANGEOFFSETFLG _ 'Y))

(RPAQ PS.BITMAPARRAY (READARRAY-FROM-LIST 16 'BYTE 0
                            '(48 49 50 51 52 53 54 55 56 57 65 66 67 68 69 70 NIL)))

(RPAQQ \POSTSCRIPT.JOB.SETUP
       ("/s /show load def" "/mto /moveto load def" "/ellipsedict 9 dict def" "ellipsedict /mtrx matrix put"
              "/ellipse" " { ellipsedict begin" "  /endangle exch def" "  /startangle exch def" "  /orientation
              exch def" "  /minorrad exch def" "  /majorrad exch def" "  /y exch def" "  /x exch def" "
              /savematrix mtrx currentmatrix def" "  x y translate" "  orientation rotate" "  majorrad minorrad
              scale" "  0 0 1 startangle endangle arc" "  savematrix setmatrix" " end } bind def"
              "/concatprocs" " {/proc2 exch cvlit def" "  /proc1 exch cvlit def" "  /newproc proc1 length proc2
              length add array def" "  newproc 0 proc1 putinterval" "  newproc proc1 length proc2 putinterval"
              "  newproc cvx" " } bind def" "/resmatrix matrix def" "/findresolution" " {72 0 resmatrix
              defaultmatrix dtransform" "  /yres exch def /xres exch def" "  xres dup mul yres dup mul add
              sqrt" " } bind def" "/thebitimage" " {/maskp exch def" "  /bihgt exch def" "  /biwid exch def"
              "  /strbuf biwid 8 div ceiling cvi string def" "  {1 exch sub} currenttransfer concatprocs
              settransfer" "  biwid bihgt" "  maskp { false } { 1 } ifelse" "  [biwid 0 0 bihgt 0 0]" "  {
              currentfile strbuf readhexstring pop }" "  maskp { imagemask } { image } ifelse" " } bind def"
              "/setuserscreendict 22 dict def" "setuserscreendict begin" " /tempctm matrix def" " /temprot
              matrix def" " /tempscale matrix def" "end" "/setuserscreen" " {setuserscreendict begin" "
              /spotfunction exch def" "   /screenangle exch def" "   /cellsize exch def" "   /m tempctm
              currentmatrix def" "   /rm screenangle temprot rotate def" "   /sm cellsize dup tempscale scale
              def" "   sm rm m m concatmatrix m concatmatrix pop" "   1 0 m dtransform /y1 exch def /x1 exch
              def" "   /veclength x1 dup mul y1 dup mul add sqrt def" "   /frequency findresolution veclength
              div def" "   /newscreenangle y1 x1 atan def" "   m 2 get m 1 get mul m 0 get m 3 get mul sub" "
              0 gt { { neg } /spotfunction load concatprocs" "          /spotfunction exch def } if" "
              frequency newscreenangle /spotfunction load setscreen" "  end" " } bind def" "/setpatterndict 18
              dict def" "setpatterndict begin" " /bitison" "  {/ybit exch def /xbit exch def" "   /bytevalue
              bstring ybit bwidth mul xbit 8 idiv add get def" "   /mask 1 7 xbit 8 mod sub bitshift def" "
              bytevalue mask and 0 ne" "  } bind def" "end" "/bitpatternspotfunction" " {setpatterndict begin"
              "   /y exch def /x exch def" "   /xindex x 1 add 2 div bpside mul cvi def" "   /yindex y 1 add 2
              div bpside mul cvi def" "   xindex yindex bitison" "    {/onbits onbits 1 add def 1}" "
              {/offbits offbits 1 add def 0} ifelse" "  end" " } bind def" "/setpattern" " {setpatterndict
              begin" "   /cellsz exch def" "   /angle exch def" "   /bwidth exch def" "   /bpside exch def" "
              /bstring exch def" "   /onbits 0 def /offbits 0 def" "   cellsz angle /bitpatternspotfunction load
              setuserscreen" "   {} settransfer" "   offbits offbits onbits add div setgray" "  end" " } bind
              def" "%%%%EndProlog" "%%%%BeginSetup" "clippath pathbbox" "/ymax exch def /xmax exch def /ymin
              exch def /xmin exch def"))

(RPAQQ SlopeMenuItems ((Italic 'ITALIC "This is an Italic Slope font")
                       (Regular 'REGULAR "This is a Regular Slope font")))

(RPAQQ WeightMenuItems ((Bold 'BOLD "This is a Bold Weight font")
                        (Medium 'MEDIUM "This is a Medium Weight font")
                        (Light 'LIGHT "This is a Light Weight font")))

(DECLARE%: EVAL@COMPILE

(RPAQQ GOLDEN.RATIO 1.618034)
```

```
(CONSTANTS (GOLDEN.RATIO 1.618034))
)

(RPAQ? POSTSCRIPT.BITMAP.SCALE 1)

(RPAQ? POSTSCRIPT.IMAGESIZEFACTOR 1.0)

(RPAQ? POSTSCRIPT.PREFER.LANDSCAPE NIL)

(RPAQ? POSTSCRIPT.TEXTFILE.LANDSCAPE NIL)

(RPAQ? POSTSCRIPT.TEXTURE.SCALE 4)

(RPAQ? POSTSCRIPTFONTDIRECTORIES '("{DSK}<LISPFILES>FONTS>PSC>"))

(RPAQ? \POSTSCRIPT.LONGEDGE.SHIFT 0)

(RPAQ? \POSTSCRIPT.SHORTEDGE.SHIFT 0)

(RPAQ? \POSTSCRIPT.LONGEDGE.PTS (+ (TIMES 72 10.92)
                                   \POSTSCRIPT.SHORTEDGE.SHIFT))

(RPAQ? \POSTSCRIPT.SHORTEDGE.PTS (+ (TIMES 72 8.0)
                                    \POSTSCRIPT.LONGEDGE.SHIFT))

(RPAQ? \POSTSCRIPT.MAX.WILD.FONTSIZE 72)

(ADDTOVAR POSTSCRIPT.FONT.ALIST (HELVETICA . HELVETICA)
                                (TIMESROMAN . TIMES)
                                (TIMESROMAND . TIMES)
                                (COURIER . COURIER)
                                (GACHA . COURIER)
                                (CLASSIC . TIMES)
                                (MODERN . HELVETICA)
                                (CREAM . HELVETICA)
                                (TERMINAL . COURIER)
                                (LOGO . HELVETICA))

(ADDTOVAR PRINTERTYPES ((POSTSCRIPT)
                        (CANPRINT (POSTSCRIPT))
                        (STATUS TRUE)
                        (PROPERTIES NILL)
                        (SEND POSTSCRIPT.SEND)
                        (BITMAPSCALE POSTSCRIPT.BITMAPSCALE)
                        (BITMAPFILE (POSTSCRIPT.HARDCOPYW FILE BITMAP SCALEFACTOR REGION ROTATION TITLE))))

(ADDTOVAR PRINTFILETYPES (POSTSCRIPT (TEST POSTSCRIPTFILEP)
                                     (EXTENSION (PS PSC))
                                     (CONVERSION (TEXT POSTSCRIPT.TEXT TEDIT POSTSCRIPT.TEDIT))))

(ADDTOVAR IMAGESTREAMTYPES (POSTSCRIPT (OPENSTREAM OPENPOSTSCRIPTSTREAM)
                                       (FONTCREATE POSTSCRIPT.FONTCREATE)
                                       (FONTSAVAILABLE POSTSCRIPT.FONTSAVAILABLE)
                                       (CREATECHARSET NILL)))

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS DEFAULTPRINTINGHOST POSTSCRIPT.BITMAP.SCALE POSTSCRIPT.FONT.ALIST POSTSCRIPT.PREFER.LANDSCAPE
       POSTSCRIPT.TEXTFILE.LANDSCAPE POSTSCRIPT.TEXTURE.SCALE POSTSCRIPTFONTDIRECTORIES \POSTSCRIPT.JOB.SETUP
       \POSTSCRIPT.LONGEDGE.PTS \POSTSCRIPT.LONGEDGE.SHIFT \POSTSCRIPT.MAX.WILD.FONTSIZE
       \POSTSCRIPT.ORIENTATION.MENU \POSTSCRIPT.SHORTEDGE.PTS \POSTSCRIPT.SHORTEDGE.SHIFT \POSTSCRIPTIMAGEOPS)
)

(FILESLOAD PS-SEND)

(POSTSCRIPT.INIT)

(PUTPROPS POSTSCRIPT FILETYPE :TCOMPL)

(PUTPROPS POSTSCRIPT MAKEFILE-ENVIRONMENT (:PACKAGE "INTERLISP" :READTABLE "INTERLISP"))

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

(ADDTOVAR NLAMA )

(ADDTOVAR NLAML )

(ADDTOVAR LAMA POSTSCRIPT.PUTCOMMAND)
)

(PUTPROPS POSTSCRIPT COPYRIGHT ("Beckman Instruments, Inc" 1986 1987 1988 1989))
```

{MEDLEY}<obsolete>lispusers>POSTSCRIPT-old>POSTSCRIPT.;1 28-Jun-2024 18:34:03

-- Listed on 30-Jun-2024 13:29:04 --

FUNCTION INDEX

VARIABLE INDEX

RECORD INDEX

PROPERTY INDEX

CONSTANT INDEX