File created: 10-Apr-2023 07:15:37 **{DSK}<home>larry>il>medley>library>PRESS.;2**

edit by: lmm

changes to: (VARS PRESSCOMS)

previous date: 5-Feb-2021 22:18:06 {DSK}<home>larry>il>medley>library>PRESS.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
;;
;; Copyright (c) 1981-1987, 1990, 1993, 2021 by Venue & Xerox Corporation.

(RPAQQ PRESSCOMS
       [

;;; PRESS printing support module

        (COMS ;; Font creation functions
              (FNS \SEARCHPRESSFONTS \GETPRESSFONTNAMES \PRESSFAMILYCODELST \DECODEPRESSFACEBYTE
                   \CREATEPRESSFONT \CREATECHARSET.PRESS)
              (INITVARS (PRESSFONTWIDTHSFILES '{ERIS}<LISP>FONTS>FONTS.WIDTHS))
              (ALISTS (SYSTEMINITVARS PRESSFONTWIDTHSFILES))
              (DECLARE%: DONTCOPY (CONSTANTS noInfoCode)))
        ;; Bitmap printing support
        (FNS PRESSBITMAP FULLPRESSBITMAP SHOWREGION SHOWPRESSBITMAPREGION PRESSWINDOW \WRITEPRESSBITMAP)
        ;; Basic PRESS data structure output functions
        (FNS \BCPLSOUT.PRESS \PAGEPAD.PRESS \ENTITYEND.PRESS \PARTEND.PRESS \ENTITYSTART.PRESS SETX.PRESS
             SETXY.PRESS SETY.PRESS SHOW.PRESS)
        ;; Image stream support functions:
        (FNS OPENPRSTREAM \BITBLT.PRESS \BLTSHADE.PRESS \SCALEDBITBLT.PRESS \BITMAPSIZE.PRESS \CHARWIDTH.PRESS
             \CLOSEF.PRESS \DRAWLINE.PRESS \ENDPAGE.PRESS NEWLINE.PRESS NEWPAGE.PRESS SETUPFONTS.PRESS
             \DEFINEFONT.PRESS \DSPBOTTOMMARGIN.PRESS \DSPCLIPPINGREGION.PRESS \DSPFONT.PRESS
             \DSPLEFTMARGIN.PRESS \DSPLINEFEED.PRESS \DSPRIGHTMARGIN.PRESS \DSPSPACEFACTOR.PRESS
             \DSPTOPMARGIN.PRESS \DSPXPOSITION.PRESS \DSPYPOSITION.PRESS \FIXLINELENGTH.PRESS \OUTCHARFN.PRESS
             \SETSPACE.PRESS \STARTPAGE.PRESS \STRINGWIDTH.PRESS SHOWRECTANGLE.PRESS \PRESS.CONVERT.NSCHARACTER)
        [COMS                                                  ; Drawcurve code
              (FNS \ENDVECRUN \VECENCODE \VECPUT \VECSKIP \VECFONTINIT \DRAWCIRCLE.PRESS \DRAWARC.PRESS
                   \DRAWCURVE.PRESS \DRAWCURVE.PRESS.LINE \DRAWELLIPSE.PRESS \GETBRUSHFONT.PRESS \PRESSCURVE2)
              (INITVARS (\VecFontDir))
              (CONSTANTS (\MicasPerInch 2540))
              (DECLARE%: DONTCOPY (CONSTANTS (ScansPerIn 384)
                                             (PointsPerIn 72.27)
                                             (MicasPerScan (FQUOTIENT \MicasPerInch ScansPerIn))
                                             (ScansPerMica (FQUOTIENT ScansPerIn \MicasPerInch))
                                             (ScansPerPoint (FQUOTIENT ScansPerIn PointsPerIn))
                                             (PointsPerScan (FQUOTIENT PointsPerIn ScansPerIn))
                                             (MicasPerPoint (FQUOTIENT \MicasPerInch PointsPerIn))
                                             (PointsPerMica (FQUOTIENT PointsPerIn \MicasPerInch))
                                             (SPRUCEPAPERTOPSCANS 4096)
                                             (SPRUCEPAPERTOPMICAS (FIX (FQUOTIENT (FTIMES SPRUCEPAPERTOPSCANS
                                                                                          \MicasPerInch)
                                                                                  ScansPerIn)))
                                             (SPRUCEPAPERRIGHTMICAS (FIX (FTIMES 8.5 \MicasPerInch)))
                                             (SPRUCEPAPERRIGHTSCANS (FIX (FTIMES 8.5 ScansPerIn)))
                                             (SPRUCEPAPERBOTTOMSCANS 0)
                                             (SPRUCEPAPERBOTTOMMICAS 0)
                                             (SPRUCEPAPERLEFTSCANS 0)
                                             (SPRUCEPAPERLEFTMICAS 0]
        ;; Initialization code
        (FNS \PRESSINIT)
        (DECLARE%: DONTEVAL@LOAD DOCOPY (P (\PRESSINIT)))
        (DECLARE%: DONTCOPY (RECORDS PRESSDATA FONTDIRENTRY))
        (INITRECORDS PRESSDATA)
        [INITVARS (DEFAULTPAGEREGION (CREATEREGION 2794 1905 16256 24765))
               (PRESSBITMAPREGION (CREATEREGION 1270 1270 (FIX (TIMES 7.5 \MicasPerInch))
                                                (TIMES 10 \MicasPerInch]
        (GLOBALVARS DEFAULTPAGEREGION)
        (DECLARE%: DONTCOPY (CONSTANTS (BYTESPERRECORD 512)
                                       (LISPENTITYTYPE 6)
                                       (MICASPERINCH \MicasPerInch))
               (E (RESETSAVE (RADIX 8)))
               (CONSTANTS * PRESSOPS))
        ;; Hardcopy user interface connections:
        (COMS (FNS MAKEPRESS PRESSFILEP PRESS.BITMAPSCALE)
              (ALISTS (IMAGESTREAMTYPES PRESS))
```

```
            (ADDVARS [PRINTERTYPES ((PRESS SPRUCE PENGUIN DOVER)
                                    (CANPRINT (PRESS))
                                    (STATUS PUP.PRINTER.STATUS)
                                    (PROPERTIES PUP.PRINTER.PROPERTIES)
                                    (SEND EFTP)
                                    (BITMAPSCALE NIL)
                                    (BITMAPFILE (PRESSBITMAP FILE BITMAP SCALEFACTOR REGION ROTATION TITLE)))
                                ((FULLPRESS RAVEN)
                                                                  ; same as PRESS but can scale bitmaps
                                 (CANPRINT (PRESS))
                                 (STATUS TRUE)
                                 (PROPERTIES NILL)
                                 (SEND EFTP)
                                 (BITMAPSCALE PRESS.BITMAPSCALE)
                                 (BITMAPFILE (FULLPRESSBITMAP FILE BITMAP SCALEFACTOR REGION ROTATION TITLE]
                     (PRINTFILETYPES (PRESS (TEST PRESSFILEP)
                                            (EXTENSION (PRESS))
                                            (CONVERSION (TEXT MAKEPRESS TEDIT
                                                              (LAMBDA (FILE PFILE FONTS HEADING)
                                                                     (SETQ FILE (OPENTEXTSTREAM FILE))
                                                                     (TEDIT.FORMAT.HARDCOPY FILE PFILE T NIL NIL
                                                                            NIL 'PRESS)
                                                                     (CLOSEF? FILE)
                                                                     PFILE])
```

;;; PRESS printing support module

;; Font creation functions

```
(DEFINEQ
```

(\SEARCHPRESSFONTS

```
  [LAMBDA (FAMILY PSIZE FACE ROTATION DEVICE)                    (* rrb "26-Sep-84 16:35")
```

(* * returns a list of the form (family size face rotation PRESS) for any font matching the specs.
* is used as wildcard.)

```
    (DECLARE (GLOBALVARS PRESSFONTWIDTHSFILES))
    (RESETLST
        (bind FONTSFOUND WSTRM for F inside PRESSFONTWIDTHSFILES when (INFILEP F)
           do [COND
                 ((SETQ WSTRM (\GETSTREAM F 'INPUT T))
                  (RESETSAVE NIL (LIST 'SETFILEPTR WSTRM (GETFILEPTR WSTRM)))
                  (SETFILEPTR WSTRM 0))
                 (T (RESETSAVE (SETQ WSTRM (OPENSTREAM F 'INPUT 'OLD 8))
                               '(PROGN (CLOSEF? OLDVALUE]
              (SETQ FONTSFOUND (UNION (\GETPRESSFONTNAMES WSTRM FAMILY PSIZE FACE ROTATION)
                                      FONTSFOUND))
           finally (RETURN FONTSFOUND)))])
```

(\GETPRESSFONTNAMES

```
  [LAMBDA (WSTRM FAMILY PSIZE FACE ROTATION)                     (* rmk%: "17-Dec-84 13:55")
                                                                 (* finds the fonts that exist that match the args.
                                                                 * is used as wildcard.)
    (bind FONTSFOUND TYPE XFACE XFAMILY XSIZE XFACE XROTATION [XFACECODE _ (COND
                                                                              ((AND (LISTP FACE)
                                                                                    (NOT (MEMB '* FACE)))
                                                                 (* if complete face is specified, compute code so don't have to
                                                                 on each font.)
                                                                               (\FACECODE FACE]
          (FAMILYCODELST _ (\PRESSFAMILYCODELST WSTRM))
          (NEXT _ 0)
          (MICASIZE _ (AND (NEQ PSIZE '*)
                           (IQUOTIENT (ITIMES PSIZE 2540)
                                      72)))
       do (SETFILEPTR WSTRM NEXT)
          (SETQ TYPE (\BIN WSTRM))
          (add NEXT (LLSH (IPLUS (\BIN WSTRM)
                                 (LLSH (LOGAND TYPE 15)
                                       8))
                          1))
          (SELECTQ (LRSH TYPE 4)
               (4 (SETQ XFAMILY (OR (CDR (FASSOC (\BIN WSTRM)
                                                 FAMILYCODELST))
                                    (ERROR "unknown code number in widths file")))
                  [COND
                     ((OR (EQ FAMILY '*)
                          (EQ FAMILY XFAMILY))
                      (COND
                         ([AND (ILESSP (SETQ XFACE (\BIN WSTRM))
                                       18)
                               (COND
                                  (XFACECODE (AND (EQ XFACECODE XFACE)
                                                  (SETQ XFACE FACE)))
                                  ((PROGN (SETQ XFACE (\DECODEPRESSFACEBYTE XFACE))
                                          (OR (EQ FACE '*)
```

```
                                        (EQUAL FACE XFACE)
                                        (for SPEC in FACE as XFIELD in XFACE
                                           always (OR (EQ SPEC XFIELD)
                                                      (EQ SPEC '*]
                                                (* greater than 18 means either ASCII or other type of font,
                                                ignore it.)
                                                (* skip beg and end chars)
                        (\BIN WSTRM)
                        (\BIN WSTRM)
                        (SETQ XSIZE (\WIN WSTRM))
                        (COND
                           ((OR (EQ PSIZE '*)
                                (EQ MICASIZE XSIZE)
                                (AND (EQ XSIZE 0)
                                     (SETQ XSIZE MICASIZE)))
```

(* if XSIZE is 0, the font widths are relative and are to be used for all font sizes.
In this case, if the user asked about a particular size, claim that it is there.)

```
                         (SETQ XROTATION (\WIN WSTRM))
                         (COND
                            ((OR (EQ ROTATION '*)
                                 (EQ XROTATION ROTATION))
                             (push FONTSFOUND (LIST XFAMILY (FIXR (FQUOTIENT (ITIMES XSIZE 72)
                                                                             2540))
                                                            XFACE XROTATION 'PRESS])
              (0 (RETURN FONTSFOUND))
              NIL])
```

(\PRESSFAMILYCODELST

```
[LAMBDA (WSTRM)                                          (* rrb "26-Sep-84 09:55")
```

(* returns an ALIST of code -
family pairs from the press font widths file WSTRM.)

```
                                                         (* leaving the file positioned at the beginning of the next file
                                                         entry.)
    (bind PAIRS TYPE (NEXT _ 0) do (SETFILEPTR WSTRM NEXT)
                                   (SETQ TYPE (\BIN WSTRM))
                                   (add NEXT (LLSH (IPLUS (\BIN WSTRM)
                                                          (LLSH (LOGAND TYPE 15)
                                                                8))
                                                   1))
                                   (SELECTQ (LRSH TYPE 4)
                                        (1 (SETQ PAIRS (CONS [CONS (\WIN WSTRM)
                                                                   (PACKC (for I from 1 to (\BIN WSTRM)
                                                                             collect (\BIN WSTRM]
                                                             PAIRS)))
                                        (0 (RETURN PAIRS))
                                        NIL])
```

(\DECODEPRESSFACEBYTE

```
[LAMBDA (FACECODE)                                       (* rrb "26-Sep-84 14:28")
```

(* * returns a list of (weight slope expansion) from a press widths file byte code.)

```
   (COND
      [(ILESSP FACECODE 18)
       (PROG (EXP SLOPE WEIGHT)
             [SETQ EXP (COND
                          ((IGEQ FACECODE 12)
                           (SETQ FACECODE (IDIFFERENCE FACECODE 12))
                           'EXPANDED)
                          ((IGEQ FACECODE 6)
                           (SETQ FACECODE (IDIFFERENCE FACECODE 6))
                           'COMPRESSED)
                          (T 'REGULAR]
             [SETQ WEIGHT (COND
                             ((IGEQ FACECODE 4)
                              (SETQ FACECODE (IDIFFERENCE FACECODE 4))
                              'LIGHT)
                             ((IGEQ FACECODE 2)
                              (SETQ FACECODE (IDIFFERENCE FACECODE 2))
                              'BOLD)
                             (T 'MEDIUM]
             [SETQ SLOPE (COND
                            ((EQ FACECODE 1)
                             'ITALIC)
                            (T 'REGULAR]
             (RETURN (LIST WEIGHT SLOPE EXP]
      (T                                                 (* non xerox font)
         NIL])
```

(\CREATEPRESSFONT

```
[LAMBDA (FAMILY PSIZE FACE ROTATION DEVICE)              (* jds "10-Mar-86 16:35")
```

```
        (* Widths array is fully allocated, with zeroes for characters with no information.
        An array is not allocated for fixed WidthsY. DEVICE is PRESS or INTERPRESS)

    (DECLARE (GLOBALVARS PRESSFONTWIDTHSFILES))
    (RESETLST                                                    (* RESETLST to make sure the fontfiles get closed)
        (PROG ((FD (create FONTDESCRIPTOR
                           FONTDEVICE _ DEVICE
                           FONTFAMILY _ FAMILY
                           FONTSIZE _ PSIZE
                           FONTFACE _ FACE
                           \SFFACECODE _ (\FACECODE FACE)
                           ROTATION _ ROTATION
                           FONTSCALE _ (CONSTANT (FQUOTIENT 2540 72))
                           \SFHeight _ 0
                           \SFAscent _ 0
                           \SFDescent _ 0)))
              (\GETCHARSETINFO 0 FD T)
              (RETURN FD)))])
```

(**CREATECHARSET.PRESS**

```
  [LAMBDA (FAMILY SIZE FACE ROTATION DEVICE CHARSET FONTDESC NOSLUG?)
                                                                 ; Edited 29-Jul-87 14:15 by jds
```

;;; just a dummy definition. Press should not ever be trying to change character sets, since the fonts only contain charset 0 (roughly)

```
    (DECLARE (GLOBALVARS PRESSFONTWIDTHSFILES))
    (COND
       ((NEQ 0 CHARSET)
        (ERROR "Press does not support NS characters.")))
    (RESETLST                                                    ; RESETLST to make sure the fontfiles get closed
        (PROG* (WSTRM STRMCACHE XLATEDFAM FIXEDFLAGS RELFLAG FIRSTCHAR LASTCHAR TEM WIDTHSY WIDTHS
                  (PRESSMICASIZE (IQUOTIENT (ITIMES SIZE 2540)
                                            72))
                  (NSMICASIZE (FIXR (FQUOTIENT (ITIMES SIZE 2540)
                                               72)))
                  (FACECODE (\FACECODE FACE))
                  [FD (create FONTDESCRIPTOR
                              FONTDEVICE _ DEVICE
                              FONTFAMILY _ FAMILY
                              FONTSIZE _ SIZE
                              FONTFACE _ FACE
                              \SFFACECODE _ FACECODE
                              ROTATION _ ROTATION
                              FONTSCALE _ (CONSTANT (FQUOTIENT 2540 72]
                  (CSINFO (create CHARSETINFO)))
```

;;; Go look for the fonts.widths file that has this font's info in it.

```
                (OR [for F inside PRESSFONTWIDTHSFILES when (INFILEP F)
                       do                                        ; Look thru the candidate PRESSFONTWIDTHSFILES for a file
                                                                 ; that has a description for this font.
                          [COND
                             [(SETQ WSTRM (\GETSTREAM F 'INPUT T))
                              (COND
                                 ((RANDACCESSP WSTRM)
                                  (RESETSAVE NIL (LIST 'SETFILEPTR WSTRM (GETFILEPTR WSTRM)))
                                  (SETFILEPTR WSTRM 0]
                             (T (RESETSAVE (SETQ WSTRM (OPENSTREAM F 'INPUT 'OLD 8))
                                           '(PROGN (CLOSEF? OLDVALUE]
                          [OR (RANDACCESSP WSTRM)
                              (COPYBYTES WSTRM (SETQ WSTRM (OPENSTREAM '{NODIRCORE} 'BOTH 'NEW]
                          (push STRMCACHE WSTRM)                 ; Save for coercions below
                          (COND
                             ((SETQ RELFLAG (\POSITIONFONTFILE WSTRM NSMICASIZE FIRSTCHAR LASTCHAR FAMILY
                                                               FACECODE))     ; OK, we found this font described in this file.
                              (RETURN T]
                      [AND (SETQ XLATEDFAM (SELECTQ FAMILY
                                                    (MODERN 'HELVETICA)
                                                    (CLASSIC 'TIMESROMAN)
                                                    (LOGOTYPE 'LOGO)
                                                    (TERMINAL 'GACHA)
                                                    NIL))
                           (for old WSTRM in (SETQ STRMCACHE (DREVERSE STRMCACHE))
                              first (replace FONTFAMILY of FD with XLATEDFAM)
                              do                                 ; Now try coercing the family name
                                 ;; We know the file was left open and is randaccessp from the previous loop, which must have run off the end of
                                 ;; the file list
                                 (SETFILEPTR WSTRM 0)
                                 (COND
                                    ((SETQ RELFLAG (\POSITIONFONTFILE WSTRM NSMICASIZE FIRSTCHAR LASTCHAR XLATEDFAM
                                                                      FACECODE))
                                     (replace FONTDEVICESPEC of FD with (LIST XLATEDFAM SIZE FACE ROTATION DEVICE))
                                     (replace FONTFAMILY of FD with FAMILY)
                                     (RETURN T]
```

```
        [AND (SETQ XLATEDFAM (SELECTQ FAMILY
                                  (MODERN 'FRUTIGER)
                                  (CLASSIC 'CENTURY)
                                  NIL))
             (for old WSTRM in STRMCACHE first (replace FONTFAMILY of FD with XLATEDFAM)
                do (SETFILEPTR WSTRM 0)
                   (COND
                      ((SETQ RELFLAG (\POSITIONFONTFILE WSTRM NSMICASIZE FIRSTCHAR LASTCHAR XLATEDFAM
                                                FACECODE))
                       (replace FONTDEVICESPEC of FD with (LIST XLATEDFAM SIZE FACE ROTATION DEVICE))
                       (replace FONTFAMILY of FD with FAMILY)
                       (RETURN T]
        (RETURN NIL))
```

;;; Having found the font-widths file, now read the width info from it.

```
(SETQ RELFLAG (ZEROP RELFLAG))                    ; Actually, \POSITIONFONTFILE returns zero if the font metrics
                                                  ; are size-relative and must be scaled.
(SETQ WIDTHS (fetch (CHARSETINFO WIDTHS) of CSINFO))
(SETFILEPTR WSTRM (UNFOLD (\FIXPIN WSTRM)
                          BYTESPERWORD))
```

;; Read the location of the WD segment for this font (we're in the directory part of the file now), and go there.

```
(SETQ FBBOX (SIGNED (\WIN WSTRM)
                    BITSPERWORD))                 ; replace (FONTDESCRIPTOR FBBOX) of FD with (SIGNED
                                                  ; (\WIN WSTRM) BITSPERWORD)
                                                  ; Get the max bounding width for the font
(replace (CHARSETINFO CHARSETDESCENT) of CSINFO with (IMINUS (SIGNED (\WIN WSTRM)
                                                                     BITSPERWORD)))
                                                  ; Descent is -FBBOY
(SETQ FOO (\WIN WSTRM))                           ; replace (FONTDESCRIPTOR FBBDX) of FD with (SIGNED
                                                  ; (\WIN WSTRM) BITSPERWORD)
                                                  ; And the standard kern value (?)
(SETQ CHARSETHEIGHT (SIGNED (\WIN WSTRM)
                            BITSPERWORD))         ; replace \SFHeight of FD with (SIGNED (\WIN WSTRM)
                                                  ; BITSPERWORD)
                                                  ; Height is FBBDY
[COND
   (RELFLAG                                       ; Dimensions are relative, must be scaled
          ;; replace (FONTDESCRIPTOR FBBOX) of FD with (IQUOTIENT (ITIMES (fetch (FONTDESCRIPTOR FBBOX) of
          ;; FD) NSMICASIZE) 1000)
          (replace (CHARSETINFO CHARSETDESCENT) of CSINFO with (IQUOTIENT
                                                                 (ITIMES (fetch (CHARSETINFO
                                                                                  CHARSETDESCENT
                                                                                  )
                                                                                of CSINFO)
                                                                         NSMICASIZE)
                                                                 1000))
          ;; replace (FONTDESCRIPTOR FBBDX) of FD with (IQUOTIENT (ITIMES (fetch (FONTDESCRIPTOR FBBDX) of FD)
          ;; NSMICASIZE) 1000)
          (SETQ CHARSETHEIGHT (IQUOTIENT (ITIMES CHARSETHEIGHT NSMICASIZE)
                                         1000]
(replace (CHARSETINFO CHARSETASCENT) of CSINFO with (IDIFFERENCE CHARSETHEIGHT
                                                                 (fetch CHARSETDESCENT of CSINFO)))
(SETQ FIXEDFLAGS (LRSH (\BIN WSTRM)
                       6))                        ; The fixed flags
(\BIN WSTRM)                                      ; Skip the spares
[COND
   ((EQ 2 (LOGAND FIXEDFLAGS 2))                  ; This font is fixed width.
    (SETQ TEM (\WIN WSTRM))                       ; Read the fixed width for this font
    [COND
       ((AND RELFLAG (NOT (ZEROP TEM)))           ; If it's size relative, scale it.
        (SETQ TEM (IQUOTIENT (ITIMES TEM NSMICASIZE)
                             1000]
    (for I from FIRSTCHAR to LASTCHAR do          ; Fill in the char widths table with the width.
                                        (\FSETWIDTH WIDTHS I TEM)))
   (T                                             ; Variable width font, so we have to read widths.
                                                  ; AIN WIDTHS FIRSTCHAR (ADD1 (IDIFFERENCE LASTCHAR
                                                  ; FIRSTCHAR)) WSTRM
      (for I from FIRSTCHAR to LASTCHAR do (\FSETWIDTH WIDTHS I noInfoCode))
      (\BINS (\GETOFD WSTRM 'INPUT)
             WIDTHS
             (UNFOLD FIRSTCHAR BYTESPERWORD)
             (UNFOLD (ADD1 (IDIFFERENCE LASTCHAR FIRSTCHAR))
                     BYTESPERWORD))               ; Read the X widths.
      (for I from FIRSTCHAR to LASTCHAR when (EQ noInfoCode (\FGETWIDTH WIDTHS I))
         do                                       ; For chars that have no width info, let width be zero.
            (\FSETWIDTH WIDTHS I 0))
      (COND
         (RELFLAG                                 ; If the widths are size-relative, scale them.
                (for I from FIRSTCHAR to LASTCHAR
                   do (\FSETWIDTH WIDTHS I (IQUOTIENT (ITIMES (\FGETWIDTH WIDTHS I)
                                                              NSMICASIZE)
                                                      1000]
[COND
```

```
                  [(EQ 1 (LOGAND FIXEDFLAGS 1))
                   (COND
                      ((ILESSP (GETFILEPTR WSTRM)
                               (GETEOFPTR WSTRM))
                       (SETQ WIDTHSY (\WIN WSTRM)))
                      (T                                          ; STAR FONT FILES LIKE TO LEAVE OFF THE Y WIDTH.
                         (SETQ WIDTHSY 0)))                       ; The fixed width-Y for this font;  the width-Y field is a single
                                                                  ; integer in the FD
                    (replace (CHARSETINFO YWIDTHS) of CSINFO with (COND
                                                                    ((AND RELFLAG (NOT (ZEROP WIDTHSY)))
                                                                     (IQUOTIENT (ITIMES WIDTHSY NSMICASIZE)
                                                                            1000))
                                                                    (T WIDTHSY]
                  (T                                              ; Variable Y-width font.  Fill it in as above
                     (SETQ WIDTHSY (replace (CHARSETINFO YWIDTHS) of CSINFO with (\CREATECSINFOELEMENT)))
                     (for I from FIRSTCHAR to LASTCHAR do (\FSETWIDTH WIDTHSY I noInfoCode))
                     (\BINS (\GETOFD WSTRM 'INPUT)
                            WIDTHSY
                            (UNFOLD FIRSTCHAR BYTESPERWORD)
                            (UNFOLD (ADD1 (IDIFFERENCE LASTCHAR FIRSTCHAR))
                                    BYTESPERWORD))                ; Read the Y widths
                     (for I from FIRSTCHAR to LASTCHAR when (EQ noInfoCode (\FGETWIDTH WIDTHSY I))
                        do                                        ; Let any characters with no width info be zero height
                            (\FSETWIDTH WIDTHSY I 0))
                     (COND
                        (RELFLAG                                  ; If the widths are size-relative, scale them.
                                (for I from FIRSTCHAR to LASTCHAR
                                   do (\FSETWIDTH WIDTHSY I (IQUOTIENT (ITIMES (\FGETWIDTH WIDTHSY I)
                                                                               NSMICASIZE)
                                                                       1000]
                (RETURN CSINFO)))])

)

(RPAQ? PRESSFONTWIDTHSFILES '{ERIS}<LISP>FONTS>FONTS.WIDTHS)

(ADDTOVAR SYSTEMINITVARS (PRESSFONTWIDTHSFILES {DSK}FONTS.WIDTHS))

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

(RPAQQ noInfoCode 32768)

(CONSTANTS noInfoCode)
)
)
```

;; Bitmap printing support

```
(DEFINEQ
```

(**PRESSBITMAP**

```
  [LAMBDA (FILE BITMAP SCALEFACTOR CLIPPINGREGION)                ; Edited 12-Jun-90 10:39 by mitani
```

(* * This routine uses the whole page (ie PRTOP and PRRIGHT as opposed to PRWIDTH and PRHEIGHT) to produce a SPRUCE Press file. It will truncate if necessary since SPRUCE does not support scaling)

```
    (PROG ((PRSTREAM (OPENPRSTREAM FILE))
           WIDTH HEIGHT PRDATA XPOS YPOS (PRESSPAGEHEIGHT (fetch (REGION HEIGHT) of PRESSBITMAPREGION))
           (PRESSPAGEWIDTH (fetch (REGION WIDTH) of PRESSBITMAPREGION)))
          (SETQ PRDATA (fetch (STREAM IMAGEDATA) of PRSTREAM))
          (if (AND SCALEFACTOR (NOT (EQUAL SCALEFACTOR 1)))
              then (ERROR "Spruce cannot scale bitmaps.  Try pressing to a full press printer."))
                                                                  (* Get width and height in screen pts)
          [COND
             (CLIPPINGREGION (SETQ WIDTH (fetch (REGION WIDTH) of CLIPPINGREGION))
                    (SETQ HEIGHT (fetch (REGION HEIGHT) of CLIPPINGREGION)))
             (T (SETQ WIDTH (BITMAPWIDTH BITMAP))
                (SETQ HEIGHT (BITMAPHEIGHT BITMAP]
          (SETQ XPOS (IQUOTIENT (IDIFFERENCE PRESSPAGEWIDTH (FIX (TIMES MicasPerPoint WIDTH)))
                                2))
          (SETQ YPOS (IQUOTIENT (IDIFFERENCE PRESSPAGEHEIGHT (FIX (TIMES MicasPerPoint HEIGHT)))
                                2))
          [COND
             ((OR (ILESSP XPOS 0)
                  (ILESSP YPOS 0))
              (printout T "Warning:  Bitmap too large for Spruce PRESS page, will be clipped..." T)
              (SETQ XPOS (IMAX 0 XPOS))
              (SETQ YPOS (IMAX 0 YPOS))
              (SETQ CLIPPINGREGION (if CLIPPINGREGION
                                       then [CREATEREGION (fetch (REGION LEFT) of CLIPPINGREGION)
                                                          (fetch (REGION BOTTOM) of CLIPPINGREGION)
                                                          (FIX (MIN WIDTH (QUOTIENT PRESSPAGEWIDTH MicasPerPoint)))
                                                          (FIX (MIN HEIGHT (QUOTIENT PRESSPAGEHEIGHT MicasPerPoint]
                                     else (CREATEREGION 0 0 (FIX (MIN WIDTH (QUOTIENT PRESSPAGEWIDTH
                                                                                      MicasPerPoint)))
```

```
                                                (FIX (MIN HEIGHT (QUOTIENT PRESSPAGEHEIGHT MicasPerPoint]
        (\WRITEPRESSBITMAP BITMAP (IPLUS (fetch (REGION LEFT) of PRESSBITMAPREGION)
                                         XPOS)
               (IPLUS (fetch (REGION BOTTOM) of PRESSBITMAPREGION)
                      YPOS)
               SCALEFACTOR CLIPPINGREGION PRSTREAM)
        (RETURN (CLOSEF PRSTREAM])
```

(FULLPRESSBITMAP

```
  [LAMBDA (FILE BITMAP SCALEFACTOR CLIPPINGREGION)              ; Edited 12-Jun-90 10:39 by mitani
```

(* * This routine uses the whole page (ie PRTOP and PRRIGHT as opposed to PRWIDTH and PRHEIGHT) to produce a full Press file. It will scale if necessary)

(* * When this fn is called from HARDCOPYW, the scalefactor should already be correct.
On a direct call, it will handle it itself)

```
    (PROG ((PRSTREAM (OPENPRSTREAM FILE))
           WIDTH HEIGHT PRDATA XPOS YPOS (PRESSPAGEHEIGHT (fetch (REGION HEIGHT) of PRESSBITMAPREGION))
           (PRESSPAGEWIDTH (fetch (REGION WIDTH) of PRESSBITMAPREGION)))
          (SETQ PRDATA (fetch (STREAM IMAGEDATA) of PRSTREAM))
          (if (NOT SCALEFACTOR)
              then (SETQ SCALEFACTOR 1.0))                      (* Get width and height in screen pts)
          [COND
             (CLIPPINGREGION (SETQ WIDTH (fetch (REGION WIDTH) of CLIPPINGREGION))
                    (SETQ HEIGHT (fetch (REGION HEIGHT) of CLIPPINGREGION)))
             (T (SETQ WIDTH (BITMAPWIDTH BITMAP))
                (SETQ HEIGHT (BITMAPHEIGHT BITMAP]
          (SETQ XPOS (IQUOTIENT (IDIFFERENCE PRESSPAGEWIDTH (FIX (TIMES MicasPerPoint WIDTH SCALEFACTOR)))
                                2))
          (SETQ YPOS (IQUOTIENT (IDIFFERENCE PRESSPAGEHEIGHT (FIX (TIMES MicasPerPoint HEIGHT SCALEFACTOR)))
                                2))
          [COND
             ((OR (ILESSP XPOS 0)
                  (ILESSP YPOS 0))
              (printout T "Warning:  Bitmap too large for PRESS page, will be scaled..." T)
              (SETQ SCALEFACTOR (PRESS.BITMAPSCALE WIDTH HEIGHT))
              (SETQ XPOS (IQUOTIENT (IDIFFERENCE PRESSPAGEWIDTH (FIX (TIMES MicasPerPoint WIDTH SCALEFACTOR)))
                                    2))
              (SETQ YPOS (IQUOTIENT (IDIFFERENCE PRESSPAGEHEIGHT (FIX (TIMES MicasPerPoint HEIGHT SCALEFACTOR)))
                                    2))
              (if (OR (ILESSP XPOS 0)
                      (ILESSP YPOS 0))
                  then (ERROR "Internal consistency check failed in FULLPRESSBITMAP."]
          (\WRITEPRESSBITMAP BITMAP (IPLUS (fetch (REGION LEFT) of PRESSBITMAPREGION)
                                           XPOS)
                 (IPLUS (fetch (REGION BOTTOM) of PRESSBITMAPREGION)
                        YPOS)
                 SCALEFACTOR CLIPPINGREGION PRSTREAM)
          (RETURN (CLOSEF PRSTREAM])
```

(SHOWREGION

```
  [LAMBDA (REGION STREAM)                                       ; Edited 12-Jun-90 10:38 by mitani
```

(* * comment)

```
    (PROG NIL
          (MOVETO (fetch (REGION LEFT) of REGION)
                  (fetch (REGION BOTTOM) of REGION)
                  STREAM)
          (RELDRAWTO (fetch (REGION WIDTH) of REGION)
                 0 NIL NIL STREAM)
          (RELDRAWTO 0 (fetch (REGION HEIGHT) of REGION)
                 NIL NIL STREAM)
          (RELDRAWTO (MINUS (fetch (REGION WIDTH) of REGION))
                 0 NIL NIL STREAM)
          (RELDRAWTO 0 (MINUS (fetch (REGION HEIGHT) of REGION))
                 NIL NIL STREAM)
          (RETURN STREAM])
```

(SHOWPRESSBITMAPREGION

```
  [LAMBDA NIL                                                   (* gbn "16-Sep-84 19:18")
```

(* * comment)

```
    (PROG [(STR (OPENIMAGESTREAM '{LPT} 'PRESS]
          (SHOWREGION PRESSBITMAPREGION STR)
          (RETURN (CLOSEF STR])
```

(PRESSWINDOW

```
  [LAMBDA (W)                                                   ; Edited 12-Jun-90 10:39 by mitani
                                                                (* First Try)
    (PROG ((PRSTREAM (OPENPRSTREAM '{CORE}WINDOW.PRESS (LIST 'HEADING "Press Stream Window Image"
```

```
                                                                    'BREAKPAGEFILENAME "Press Stream Window Image")))
          [BITMAP (WINDOW.BITMAP (OR W (WHICHW]
          WIDTH HEIGHT (PTSTOMICAS 35))
         (SETQ WIDTH (BITMAPWIDTH BITMAP))
         (SETQ HEIGHT (BITMAPHEIGHT BITMAP))
         (DSPXPOSITION (IPLUS (fetch PRLEFT of (fetch (STREAM IMAGEDATA) of PRSTREAM))
                              (IQUOTIENT (IDIFFERENCE (fetch PRWIDTH of (fetch (STREAM IMAGEDATA) of PRSTREAM))
                                                      (ITIMES PTSTOMICAS WIDTH))
                                    2))
                PRSTREAM)
         (DSPYPOSITION (IPLUS (fetch PRBOTTOM of (fetch (STREAM IMAGEDATA) of PRSTREAM))
                              (IQUOTIENT (IDIFFERENCE (fetch PRHEIGHT of (fetch (STREAM IMAGEDATA) of PRSTREAM))
                                                      (ITIMES PTSTOMICAS HEIGHT))
                                    2))
                PRSTREAM)
         (\WRITEPRESSBITMAP BITMAP NIL NIL PRSTREAM)
         (RETURN (CLOSEF PRSTREAM])
```

(\WRITEPRESSBITMAP

```
  [LAMBDA (BITMAP XPOS YPOS SCALEFACTOR CLIPPINGREGION PRSTREAM) ; Edited 12-Jun-90 10:39 by mitani
                                                                 (* This should define the origin of the bitmap on the page)
    [COND
       (CLIPPINGREGION                                           (* UGH)
              (SETQ BITMAP (PROG [(BM (BITMAPCREATE (fetch (REGION WIDTH) of CLIPPINGREGION)
                                                    (fetch (REGION HEIGHT) of CLIPPINGREGION]
                                 (with REGION CLIPPINGREGION (BITBLT BITMAP LEFT BOTTOM BM NIL NIL WIDTH HEIGHT))
                                 (RETURN BM]
    (PROG ((PRDATA (fetch (STREAM IMAGEDATA) of PRSTREAM))
           (WW (fetch BITMAPRASTERWIDTH of BITMAP))
           (HT (fetch BITMAPHEIGHT of BITMAP))
           ELSTREAM TOTCOUNT CURX CURY)
          (SETQ ELSTREAM (fetch ELSTREAM of PRDATA))
          (SETQ CURX (fetch PRXPOS of PRDATA))
          (SETQ CURY (fetch PRYPOS of PRDATA))
          (SHOW.PRESS PRSTREAM)                                  (* flush chars before ending entity)
          (\ENTITYEND.PRESS PRSTREAM)

          (* Close previous entity because we used to specify a translation for the bitmap entity.
          But now we are using the current x and y position. All this stuff might therefore be unnecessary)

          (\ENTITYSTART.PRESS PRSTREAM)
          (SETXY.PRESS PRSTREAM XPOS YPOS)
          (COND
             ((NULL SCALEFACTOR)
              (SETQ SCALEFACTOR 1.0)))
          (\WOUT PRSTREAM 256)                                   (* Output <<Set-Coding>>. (0 notates bitmap, followed by 2byte
     width (in dots) and height (in dots)))
          (\WOUT PRSTREAM (UNFOLD WW BITSPERWORD))               (* Width)
          (\WOUT PRSTREAM HT)                                    (* Height)
          (\WOUT PRSTREAM (IPLUS 512 3))                         (* <<Set-Mode>> notates that the Lisp bitmap is stored
                                                                 left-to-right and top-to-bottom)
          (\WOUT PRSTREAM 2)

          (* you might think it should be MicasPerPoint -
          ha ha ha! Only the value 32 works! Oops!)

          [\WOUT PRSTREAM (FIXR (FTIMES SCALEFACTOR (TIMES 32 (UNFOLD WW BITSPERWORD]
          [\WOUT PRSTREAM (FIXR (FTIMES SCALEFACTOR (TIMES 32 HT]
          (\WOUT PRSTREAM 1)

          (* Set Window. 2 bytes of how many bytes to skip, 2 bytes of how many dots wide to display followed by the same for lines)

          (\WOUT PRSTREAM 0)                                     (* skip 0 dots)
          (\WOUT PRSTREAM (UNFOLD WW BITSPERWORD))
          (\WOUT PRSTREAM 0)                                     (* skip 0 lines)
          (\WOUT PRSTREAM HT)
          (\WOUT PRSTREAM 3)                                     (* <<Dots-Follow>>)
                                                                 (* TOTCOUNT is a word count.)
          (\BOUTS PRSTREAM (fetch BITMAPBASE of BITMAP)
                 0
                 (UNFOLD (SETQ TOTCOUNT (ITIMES HT WW))
                         BYTESPERWORD))
          (\BOUT ELSTREAM ShowDotsCode)
          (\FIXPOUT ELSTREAM (IPLUS TOTCOUNT 13))                (* Number of DL bytes)
          (\ENTITYEND.PRESS PRSTREAM)
          (\ENTITYSTART.PRESS PRSTREAM)                          (* Since START reestablishes X and Y, following might not be
                                                                 necessary)
          (SETXY.PRESS PRSTREAM CURX CURY])

)

;; Basic PRESS data structure output functions

(DEFINEQ
```

(\BCPLSOUT.PRESS

```
  [LAMBDA (STRM X N)                                                    (* rmk%: "14-Jun-84 19:36")

          (* Puts out a Bcpl string X in N bytes, filling with zeroes or truncating if needed.)

    (PROG [(NC (IMIN (NCHARS X)
                     (SETQ N (SUB1 N]
          (\BOUT STRM NC)
          (for I from 1 to NC do (\BOUT STRM (NTHCHARCODE X I)))
          (for I from (ADD1 NC) to N do (\BOUT STRM 0])
```

(**PAGEPAD.PRESS**

```
  [LAMBDA (STRM)                                                        (* rmk%: "14-Jun-84 18:30")
                                                                        (* Move the fileptr to the next record boundary, returning the
                                                                        number of words skipped.)
    (PROG (PADDING (P (GETFILEPTR STRM)))
          (SETQ PADDING (MODUP P BYTESPERRECORD))
          (COND
             ((IGREATERP PADDING 0)                                     (* SETFILEPTR for all but 1, then \BOUT to make sure the file
                                                                        gets extended.)
              [AND (NEQ PADDING 1)
                   (SETFILEPTR STRM (IPLUS P (SUB1 PADDING]
              (\BOUT STRM 0)))
          (RETURN (FOLDLO PADDING BYTESPERWORD])
```

(**ENTITYEND.PRESS**

```
  [LAMBDA (PRSTREAM XOFFSET YOFFSET ETYPE)                              ; Edited 12-Jun-90 10:39 by mitani
    (PROG (ELSTREAM DLLENGTH (PRDATA (fetch (STREAM IMAGEDATA) of PRSTREAM)))
          (SETQ ELSTREAM (fetch ELSTREAM of PRDATA))
          (SETQ DLLENGTH (IDIFFERENCE (\GETFILEPTR PRSTREAM)
                                      (fetch DLSTARTBYTE of PRDATA)))
          (COND
             ((ODDP (GETFILEPTR ELSTREAM))
              (\BOUT ELSTREAM NopCode)))
          (\BOUT ELSTREAM (OR ETYPE LISPENTITYTYPE))
          (\BOUT ELSTREAM (OR (fetch FONTSET# of (fetch PRCURRFDE of PRDATA))
                              0))                                       (* fontset)
          (\FIXPOUT ELSTREAM (IDIFFERENCE (fetch DLSTARTBYTE of PRDATA)
                                          (UNFOLD (fetch PRPARTSTART of PRDATA)
                                                  BYTESPERRECORD)))     (* (IDIFFERENCE (fetch DLSTARTBYTE of PRDATA)
                                                                        (UNFOLD (fetch PRPARTSTART of PRDATA)
                                                                        BYTESPERRECORD)))
                                                                        (* part relative start of data list for this entity)
          (\FIXPOUT ELSTREAM DLLENGTH)                                  (* length of data)
          (\WOUT ELSTREAM (OR XOFFSET 0))                               (* Entity origin)
          (\WOUT ELSTREAM (OR YOFFSET 0))
          (\WOUT ELSTREAM (fetch PRLEFT of PRDATA))                     (* The bounding box for this entity -
                                                                        MAYBE LEFT AND BOTTOM ARE SIGNED?)
          (\WOUT ELSTREAM (fetch PRBOTTOM of PRDATA))
          (\WOUT ELSTREAM (IDIFFERENCE (fetch PRRIGHT of PRDATA)
                                       (fetch PRLEFT of PRDATA)))       (* width)
          (\WOUT ELSTREAM (IDIFFERENCE (fetch PRTOP of PRDATA)
                                       (fetch PRBOTTOM of PRDATA)))     (* height)
          (\WOUT ELSTREAM (ADD1 (FOLDLO (IDIFFERENCE (GETFILEPTR ELSTREAM)
                                                     (fetch ELSTARTBYTE of PRDATA))
                                        BYTESPERWORD)))                 (* Length in words--ADD1 for the length itself)
      ])
```

(**PARTEND.PRESS**

```
  [LAMBDA (PRSTREAM PARTTYPE)                                           ; Edited 12-Jun-90 10:39 by mitani

          (* Closes one part and sets up for the next, by saving the partstart and emptying the entitylist stream)

    (PROG (START PDSTREAM (PRDATA (fetch (STREAM IMAGEDATA) of PRSTREAM)))
          (SETQ PDSTREAM (fetch PDSTREAM of PRDATA))
          (SETQ START (fetch PRPARTSTART of PRDATA))
          (\WOUT PDSTREAM PARTTYPE)
          (\WOUT PDSTREAM START)                                        (* Starting record)
                                                                        (* Update starting record for next part, and record length in
                                                                        records of this part)
          (\WOUT PDSTREAM (IDIFFERENCE (replace PRPARTSTART of PRDATA with (FOLDHI (GETFILEPTR PRSTREAM)
                                                                                   BYTESPERRECORD))
                                       START))
          (\WOUT PDSTREAM (\PAGEPAD.PRESS PRSTREAM))
          (SETFILEPTR (fetch ELSTREAM of PRDATA)
                 0])
```

(**ENTITYSTART.PRESS**

```
  [LAMBDA (PRSTREAM)                                                    ; Edited 12-Jun-90 10:39 by mitani
    (PROG ((PRDATA (fetch (STREAM IMAGEDATA) of PRSTREAM)))
          (freplace PRSPACEWIDTH of PRDATA with NIL)

          (* This really should be the spacewidth of the current font. But then, if we switch fonts to one whose space*spacefactor
          comes out the same, we won't know to put out a setspace command.
          So when we actually set up the first font in this entity, we will end up putting out an explicit setspace
```

```
        (even if the space factor is 1))

        (freplace PRFONT of PRDATA with NIL)

        (* We set the font to NIL, knowing that the current font can be recoverd from the PRCURRFDE.
        This font will be set in the press file before the first show, if no explicit dspfont intervenes.
        Note, however, that up until the first dspfont, the widthscache still corresponds to what was the PRFONT.)

        (freplace DLSTARTBYTE of PRDATA with (\GETFILEPTR PRSTREAM))
        (freplace ELSTARTBYTE of PRDATA with (\GETFILEPTR (fetch ELSTREAM of PRDATA)))
        (freplace STARTCHARBYTE of PRDATA with (\GETFILEPTR PRSTREAM))
                                                        (* Entity starts with position at 0,0 so must re-establish current
                                                        position (?))
        (SETXY.PRESS PRSTREAM (fetch PRXPOS of PRDATA)
               (fetch PRYPOS of PRDATA])
```

(SETX.PRESS

```
  [LAMBDA (PRSTREAM X)                                  ; Edited 12-Jun-90 10:39 by mitani
    (PROG [(ELSTREAM (fetch ELSTREAM of (fetch (STREAM IMAGEDATA) of PRSTREAM]
          (COND
             ([AND (IGEQ X SPRUCEPAPERLEFTMICAS)
                   (ILEQ X SPRUCEPAPERRIGHTMICAS)
                   (NOT (IEQP X (fetch PRXPOS of (fetch (STREAM IMAGEDATA) of PRSTREAM]
              (\BOUT ELSTREAM SetXCode)                 (* Outcharfn ignores characters that are not in the clipping
                                                        region)
              (\WOUT ELSTREAM X)))
          (replace PRXPOS of (fetch (STREAM IMAGEDATA) of PRSTREAM) with X])
```

(SETXY.PRESS

```
  [LAMBDA (PRSTREAM X Y)                                ; Edited 12-Jun-90 10:39 by mitani
    (PROG (ELSTREAM (PRDATA (fetch (STREAM IMAGEDATA) of PRSTREAM)))
          (SETQ ELSTREAM (fetch ELSTREAM of PRDATA))
          (COND
             ((AND (IGEQ X SPRUCEPAPERLEFTMICAS)
                   (ILEQ X SPRUCEPAPERRIGHTMICAS))

        (* this clause could be part of the above test to avoid putting out set x when the position is in the right place.
        There is a place that Ron thinks is in endvecrun where setxy is called to get the printer and the streams idea of where the
        position is back into step. Thus if this test is included, that setxy is not put out when it should be.
        rrb (NOT (IEQP X (fetch PRXPOS of PRDATA))))

              (\BOUT ELSTREAM SetXCode)
              (\WOUT ELSTREAM X)))
          (replace PRXPOS of PRDATA with X)
          (COND
             ((AND (IGEQ Y SPRUCEPAPERBOTTOMMICAS)
                   (ILEQ Y SPRUCEPAPERTOPMICAS))

        (* see above comment (NOT (IEQP Y (fetch PRYPOS of PRDATA))) This clause should NOT be reinserted, because
        functions like \ENTITYSTART.PRESS call this function and need to really have the commands emitted, even tho the
        PRXPOS and PRYPOS fields claim to be real.)

              (\BOUT ELSTREAM SetYCode)
              (\WOUT ELSTREAM Y)))
          (RETURN (replace PRYPOS of PRDATA with Y])
```

(SETY.PRESS

```
  [LAMBDA (PRSTREAM Y)                                  ; Edited 12-Jun-90 10:39 by mitani
    (PROG [(ELSTREAM (fetch ELSTREAM of (fetch (STREAM IMAGEDATA) of PRSTREAM]
          (COND
             ([AND (IGEQ Y SPRUCEPAPERBOTTOMMICAS)
                   (ILEQ Y SPRUCEPAPERTOPMICAS)
                   (NOT (IEQP Y (fetch PRYPOS of (ffetch (STREAM IMAGEDATA) of PRSTREAM]
              (\BOUT ELSTREAM SetYCode)
              (\WOUT ELSTREAM Y)))
          (freplace PRYPOS of (ffetch (STREAM IMAGEDATA) of PRSTREAM) with Y])
```

(SHOW.PRESS

```
  [LAMBDA (PRSTREAM)                                    ; Edited 12-Jun-90 10:39 by mitani
    (PROG (CNT ELSTREAM (PRDATA (fetch (STREAM IMAGEDATA) of PRSTREAM))
               (CURBYTE (\GETFILEPTR PRSTREAM)))
          (SETQ ELSTREAM (fetch ELSTREAM of PRDATA))
          (SETQ CNT (IDIFFERENCE CURBYTE (fetch STARTCHARBYTE of PRDATA)))
          [COND
             ((IGREATERP CNT 0)
              [COND
                 ((NULL (fetch PRFONT of PRDATA))

        (* This is the first run of characters in this entity, and there has been no explicit dspfont.
        We therefore re-establish the current font as of the end of the last entity)

                  (replace PRFONT of PRDATA with (fetch DESCR of (fetch PRCURRFDE of PRDATA)))
                  (\BOUT (fetch ELSTREAM of PRDATA)
```

```
                        (LOGOR FontCode (fetch (FONTDIRENTRY FONT#) of (fetch PRCURRFDE of PRDATA]
            (COND
               ((ILESSP CNT 33)                                    (* short form)
                (\BOUT ELSTREAM (IPLUS ShowCharactersShortCode CNT -1)))
               (T                                                  (* Break up every 255)
                  (while (IGREATERP CNT 255) do (\BOUT ELSTREAM ShowCharactersCode)
                                                (\BOUT ELSTREAM 255)
                                                (SETQ CNT (IDIFFERENCE CNT 255))
                     finally (\BOUT ELSTREAM ShowCharactersCode)
                             (\BOUT ELSTREAM CNT]
         (replace STARTCHARBYTE of PRDATA with CURBYTE])

)
```

;; Image stream support functions:

```
(DEFINEQ
```

(OPENPRSTREAM

```
  [LAMBDA (PRFILE OPTIONS)                                         (* rmk%: "17-Dec-84 10:34")
```

(* Opens a Press stream, to which user can do OUTCHAR. OPTIONS can include a REGION, HEADING, BREAKPAGEFILENAME, and FONTS. FONTS is a list of fonts to be set up initially.
Headings will be printed in the first font in FONTS. If FONTS is NIL, then the stream is initialized with the PRESS DEFAULTFONT)

```
    (DECLARE (GLOBALVARS DEFAULTPAGEREGION \PRESSIMAGEOPS))
    (PROG [OPT PRDATA (PRSTREAM (OPENSTREAM PRFILE 'OUTPUT 'NEW 8 '((TYPE BINARY]
          [SETQ PRDATA (create PRESSDATA
                               PRPAGEREGION _ (COND
                                                  ([type? REGION (SETQ OPT (LISTGET OPTIONS 'REGION]
                                                   OPT)
                                                  (T DEFAULTPAGEREGION))
                               PDSTREAM _ (PROG1 (OPENSTREAM '{NODIRCORE} 'BOTH 'OLD/NEW)
                                                                  (* Make sure the fileptr of the following is zero
                                                                  (GETRESOURCE \PRESSPDSTREAM)
                                                                  (and free this in \CLOSE.PRESS))
                                                 )
                               ELSTREAM _ (PROG1 (OPENSTREAM '{NODIRCORE} 'BOTH 'OLD/NEW)
                                                                  (* Make sure the fileptr of the following is zero
                                                                  (GETRESOURCE \PRESSELSTREAM)
                                                                  (and free this in \CLOSE.PRESS))
                                                 )
                               PRDOCNAME _ (LISTGET OPTIONS 'DOCUMENT.NAME]
          (COND
             ((OR (NEQ \NOIMAGEOPS (fetch (STREAM IMAGEOPS) of PRSTREAM))
                  (NEQ 0 (GETEOFPTR PRSTREAM)))
              (ERROR "can't convert existing file to Press" (FULLNAME PRSTREAM))
                                                                  (* GETEOFPTR might bomb on some streams)
              ))
          (replace (STREAM OUTCHARFN) of PRSTREAM with (FUNCTION \OUTCHARFN.PRESS))
          (replace (STREAM IMAGEOPS) of PRSTREAM with \PRESSIMAGEOPS)
          (replace (STREAM IMAGEDATA) of PRSTREAM with PRDATA)
          (COND
             ((SETQ OPT (LISTGET OPTIONS 'HEADING))
              (replace PRHEADING of PRDATA with OPT)))
          (SETUPFONTS.PRESS PRSTREAM (LISTGET OPTIONS 'FONTS))
          (\STARTPAGE.PRESS PRSTREAM)
          (RETURN PRSTREAM])
```

(\BITBLT.PRESS

```
  [LAMBDA (SOURCEBITMAP SOURCELEFT SOURCEBOTTOM DESTINATION DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT
                SOURCETYPE OPERATION TEXTURE CLIPPINGREGION CLIPPEDSOURCELEFT CLIPPEDSOURCEBOTTOM)
                                                                  (* hdj " 5-Dec-84 18:39")
    (LET* ((OLDX (\DSPXPOSITION.PRESS DESTINATION))
           (OLDY (\DSPYPOSITION.PRESS DESTINATION))
           (DESTINATIONLEFT (OR DESTINATIONLEFT OLDX))
           (DESTINATIONBOTTOM (OR DESTINATIONBOTTOM OLDY)))
          (\DSPXPOSITION.PRESS DESTINATION DESTINATIONLEFT)
          (\DSPYPOSITION.PRESS DESTINATION DESTINATIONBOTTOM)
          (\WRITEPRESSBITMAP SOURCEBITMAP DESTINATIONLEFT DESTINATIONBOTTOM 1
                 (COND
                    (CLIPPINGREGION (INTERSECTREGIONS CLIPPINGREGION (CREATEREGION CLIPPEDSOURCELEFT
                                                                                  CLIPPEDSOURCEBOTTOM WIDTH HEIGHT)))
                    (T (CREATEREGION CLIPPEDSOURCELEFT CLIPPEDSOURCEBOTTOM WIDTH HEIGHT)))
                 DESTINATION)
          (\DSPXPOSITION.PRESS DESTINATION OLDX)
          (\DSPYPOSITION.PRESS DESTINATION OLDY))
    T])
```

(\BLTSHADE.PRESS

```
  [LAMBDA (TEXTURE STREAM DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT OPERATION CLIPPINGREGION)
                                                                  (* hdj "12-Mar-85 12:30")
    (LET* ((REGION (CREATEREGION DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT))
           (DESTREGION (if CLIPPINGREGION
```

```
                        then (INTERSECTREGIONS REGION CLIPPINGREGION)
                      else REGION)))

        (* * (SHOWSHADE.IP STREAM TEXTURE DESTREGION OPERATION))
                                                                    (* Dovers print at 32 micas per point)
        (\BLTSHADE.GENERICPRINTER TEXTURE STREAM DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT OPERATION
               CLIPPINGREGION 32])
```

(\SCALEDBITBLT.PRESS

```
  [LAMBDA (SOURCEBITMAP SOURCELEFT SOURCEBOTTOM DESTINATION DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT
                 SOURCETYPE OPERATION TEXTURE CLIPPINGREGION CLIPPEDSOURCELEFT CLIPPEDSOURCEBOTTOM SCALE)
                                                                    (* hdj "14-Feb-85 14:33")
    (LET* ((OLDX (\DSPXPOSITION.PRESS DESTINATION))
           (OLDY (\DSPYPOSITION.PRESS DESTINATION))
           (DESTINATIONLEFT (OR DESTINATIONLEFT OLDX))
           (DESTINATIONBOTTOM (OR DESTINATIONBOTTOM OLDY)))
          (\DSPXPOSITION.PRESS DESTINATION DESTINATIONLEFT)
          (\DSPYPOSITION.PRESS DESTINATION DESTINATIONBOTTOM)
          (\WRITEPRESSBITMAP SOURCEBITMAP DESTINATIONLEFT DESTINATIONBOTTOM SCALE
                 (COND
                    (CLIPPINGREGION (INTERSECTREGIONS CLIPPINGREGION (CREATEREGION CLIPPEDSOURCELEFT
                                                                          CLIPPEDSOURCEBOTTOM WIDTH HEIGHT)))
                    (T (CREATEREGION CLIPPEDSOURCELEFT CLIPPEDSOURCEBOTTOM WIDTH HEIGHT)))
                 DESTINATION)
          (\DSPXPOSITION.PRESS DESTINATION OLDX)
          (\DSPYPOSITION.PRESS DESTINATION OLDY))
    T])
```

(\BITMAPSIZE.PRESS

```
  [LAMBDA (STREAM BITMAP DIMENSION)                                 (* rmk%: "17-Dec-84 10:22")
    (SELECTQ DIMENSION
         (WIDTH (UNFOLD (BITMAPWIDTH BITMAP)
                        32))
         (HEIGHT (UNFOLD (BITMAPHEIGHT BITMAP)
                         32))
         (NIL (CONS (UNFOLD (BITMAPWIDTH BITMAP)
                            32)
                    (UNFOLD (BITMAPHEIGHT BITMAP)
                            32)))
         (\ILLEGAL.ARG DIMENSION])
```

(\CHARWIDTH.PRESS

```
  [LAMBDA (STREAM CHARCODE)                                         ; Edited 12-Jun-90 10:39 by mitani
                                                                    (* Gets the width of CHARCODE in an Interpress STREAM,
                                                                    observing spacefactor)

        (* * Convert from NS characters back to old PARC-internal coding for PRESS fonts)

    (SETQ CHARCODE (\PRESS.CONVERT.NSCHARACTER CHARCODE))

        (* * Then compute the character's width.)

    (COND
       ((EQ CHARCODE (CHARCODE SPACE))                              (* If it's a SPACE, use the declared space width from the stream)
        (ffetch PRSPACEWIDTH of (ffetch (STREAM IMAGEDATA) of STREAM)))
       (T (\FGETCHARWIDTH (ffetch PRFONT of (ffetch (STREAM IMAGEDATA) of STREAM))
                 (LOGAND CHARCODE \CHARMASK])
```

(\CLOSEF.PRESS

```
  [LAMBDA (PRSTREAM)                                                ; Edited 12-Jun-90 10:39 by mitani
                                                                    (* FILENAME is for the printer break page)
    (\ENDPAGE.PRESS PRSTREAM)
    (PROG (PDSTREAM (PRDATA (fetch (STREAM IMAGEDATA) of PRSTREAM)))
          (SETQ PDSTREAM (fetch PDSTREAM of PRDATA))
          (COND
             ((NEQ 0 (GETFILEPTR PDSTREAM))
              (for FDE DESCR in (fetch PRESSFONTDIR of PRDATA) as I from 0
                 do (SETQ DESCR (fetch DESCR of FDE))
                     (\WOUT PRSTREAM 16)
                     (\BOUT PRSTREAM (fetch FONTSET# of FDE))       (* Fontset)
                     (\BOUT PRSTREAM (fetch FONT# of FDE))          (* font#)
                     (\BOUT PRSTREAM 3)                             (* (\BOUT PRSTREAM (fetch FIRSTCHAR of DESCR)))
                     (\BOUT PRSTREAM 254)                           (* (\BOUT PRSTREAM (fetch LASTCHAR of DESCR)))
                     (\BCPLSOUT.PRESS PRSTREAM (FONTPROP DESCR 'DEVICEFAMILY)
                            20)
                     [\BOUT PRSTREAM (\FACECODE (FONTPROP DESCR 'DEVICEFACE]
                     (\BOUT PRSTREAM 3)                             (* (\BOUT PRSTREAM (fetch FIRSTCHAR of DESCR)))
                     (\WOUT PRSTREAM (FONTPROP DESCR 'DEVICESIZE))
                     (\WOUT PRSTREAM (fetch ROTATION of DESCR)))
              (\WOUT PRSTREAM 0)                                    (* Font part ends with 0 word)
              (\PARTEND.PRESS PRSTREAM 1)
              (COPYBYTES PDSTREAM PRSTREAM 0 (GETFILEPTR PDSTREAM))
              (\PAGEPAD.PRESS PRSTREAM)
```

```
            (PROG (DDRECORD (DDFILEPTR (GETFILEPTR PRSTREAM))) (* Write document directory)
                  (SETQ DDRECORD (FOLDLO DDFILEPTR BYTESPERRECORD))
                  (\WOUT PRSTREAM 27183)                        (* password)
                  (\WOUT PRSTREAM (ADD1 DDRECORD))
                  (\WOUT PRSTREAM (FOLDLO (GETFILEPTR PDSTREAM)
                                          8))                   (* number of parts, since each occupies 8 bytes in PD)
                  (\WOUT PRSTREAM (fetch PRPARTSTART of PRDATA))
                                                                (* part directory)
                  (\WOUT PRSTREAM (IDIFFERENCE DDRECORD (fetch PRPARTSTART of PRDATA)))
                  (\SIGNEDWOUT PRSTREAM -1)                     (* obselete)
                  (\FIXPOUT PRSTREAM (LISP.TO.ALTO.DATE (IDATE)))
                  (\WOUT PRSTREAM 1)
                  (\WOUT PRSTREAM 1)                            (* copies)
                  (\SIGNEDWOUT PRSTREAM -1)
                  (\SIGNEDWOUT PRSTREAM -1)                     (* first and last pages)
                  (\SIGNEDWOUT PRSTREAM -1)                     (* printing mode default)
                  (SETFILEPTR PRSTREAM (IPLUS DDFILEPTR 256))
                  (\BCPLSOUT.PRESS PRSTREAM (OR (fetch PRDOCNAME of PRDATA)
                                                (FULLNAME PRSTREAM))
                         52)
                  (\BCPLSOUT.PRESS PRSTREAM USERNAME 32)
                  (\BCPLSOUT.PRESS PRSTREAM (GETFILEINFO PRSTREAM 'CREATIONDATE)
                          40)
                  (\PAGEPAD.PRESS PRSTREAM])
```

(\DRAWLINE.PRESS

```
  [LAMBDA (PRSTREAM X1 Y1 X2 Y2 WIDTH OPERATION COLOR DASHING)   (* rrb "27-Sep-85 18:15")
    (COND
       (DASHING

           (* hack to handle dashing by breaking into small lines. Should be removed if \DRAWCURVE.PRESS is ever updated to
           handle dashing. rrb -
           27-sept-85)

           (DRAWDASHEDLINE X1 Y1 X2 Y2 WIDTH OPERATION PRSTREAM COLOR DASHING))
       (T (\DRAWCURVE.PRESS PRSTREAM (LIST (CREATEPOSITION X1 Y1)
                                           (CREATEPOSITION X2 Y2))
                  NIL
                  (LIST 'BUTT WIDTH)
                  DASHING)))
    Y2])
```

(\ENDPAGE.PRESS

```
  [LAMBDA (PRSTREAM)                                             ; Edited 12-Jun-90 10:39 by mitani
    (PROG [(ELSTREAM (fetch ELSTREAM of (fetch (STREAM IMAGEDATA) of PRSTREAM]
          (SHOW.PRESS PRSTREAM)
          (\ENTITYEND.PRESS PRSTREAM)
          (COND
             ((NEQ 0 (\GETFILEPTR ELSTREAM))
              (COND
                 ((ODDP (\GETFILEPTR PRSTREAM))
                  (\BOUT PRSTREAM 0)))
              (\WOUT PRSTREAM 0)                                 (* 0 word to separate DL from EL)
              (COPYBYTES ELSTREAM PRSTREAM 0 (\GETFILEPTR ELSTREAM))
              (\PARTEND.PRESS PRSTREAM 0])
```

(NEWLINE.PRESS

```
  [LAMBDA (PRSTREAM)                                             ; Edited 12-Jun-90 10:39 by mitani
                                                                 (* Go to next line (or next page))
    (PROG (NEWYPOS (PRDATA (ffetch (STREAM IMAGEDATA) of PRSTREAM)))
          (SETQ NEWYPOS (IPLUS (ffetch PRYPOS of PRDATA)
                               (ffetch PRLINEFEED of PRDATA)))
          (COND
             ((ILESSP NEWYPOS (ffetch PRBOTTOM of PRDATA))
              (NEWPAGE.PRESS PRSTREAM))
             (T (SHOW.PRESS PRSTREAM)
                (SETXY.PRESS PRSTREAM (ffetch PRLEFT of PRDATA)
                       NEWYPOS])
```

(NEWPAGE.PRESS

```
  [LAMBDA (PRSTREAM)                                             (* rmk%: "16-Jun-84 14:29")
    (\ENDPAGE.PRESS PRSTREAM)
    (\STARTPAGE.PRESS PRSTREAM])
```

(SETUPFONTS.PRESS

```
  [LAMBDA (PRSTREAM FONTS)                                       ; Edited 12-Jun-90 10:40 by mitani

          (* Sets up fonts in the initial fontset. and sets heading font. Leaves PRFONT as NIL.
          This means that \DSPFONT.PRESS of the heading font will establish that as the current font when the first page opens.)

    (for F FLG inside (OR FONTS DEFAULTFONT) do (SETQ F (FONTCREATE F NIL NIL NIL 'PRESS))
                                                (COND
```

```
                                        (FLG (\DEFINEFONT.PRESS PRSTREAM F))
                                        (T (\DSPFONT.PRESS PRSTREAM F)
                                                (* Install first font as current font and heading font.
                                                font.)
                                           (\ENTITYEND.PRESS PRSTREAM)
                                           (replace PRHEADINGFONT of (fetch (STREAM IMAGEDATA)
                                                                            of PRSTREAM)
                                              with F)
                                           (SETQ FLG T])
```

(\DEFINEFONT.PRESS

```
[LAMBDA (PRSTREAM FONT)                                    ; Edited 12-Jun-90 10:40 by mitani
  (PROG ((PRDATA (fetch (STREAM IMAGEDATA) of PRSTREAM)))
        (RETURN (OR (FASSOC FONT (fetch PRESSFONTDIR of PRDATA))
                    (CAR (push (fetch PRESSFONTDIR of PRDATA)
                               (PROG1 (create FONTDIRENTRY
                                              DESCR _ FONT
                                              FONT# _ (fetch PRNEXTFONT# of PRDATA)
                                              FONTSET# _ (fetch PRMAXFONTSET of PRDATA))
                                   (COND
                                      ((EQ 16 (add (fetch PRNEXTFONT# of PRDATA)
                                                   1))
                                       (add (fetch PRMAXFONTSET of PRDATA)
                                            1)
                                       (replace PRNEXTFONT# of PRDATA with 0)))))])
```

(\DSPBOTTOMMARGIN.PRESS

```
[LAMBDA (PRSTREAM YPOSITION)                               ; Edited 12-Jun-90 10:40 by mitani
  (PROG1 (fetch PRBOTTOM of (fetch (STREAM IMAGEDATA) of PRSTREAM))
      (COND
         (YPOSITION (replace PRBOTTOM of (fetch (STREAM IMAGEDATA) of PRSTREAM) with YPOSITION)))])
```

(\DSPCLIPPINGREGION.PRESS

```
[LAMBDA (STREAM REGION)                                    ; Edited 12-Jun-90 10:40 by mitani
                                                           (* sets the clipping region of a PRESS stream.)
  (PROG ((PRDATA (FETCH (STREAM IMAGEDATA) OF STREAM)))
        (RETURN (PROG1 (ffetch PRClippingRegion of PRDATA)
                       [COND
                          (REGION (OR (type? REGION REGION)
                                      (ERROR REGION " is not a REGION."))
                                  (UNINTERRUPTABLY
                                      (freplace PRClippingRegion of PRDATA with REGION))])])
```

(\DSPFONT.PRESS

```
[LAMBDA (PRSTREAM FONT)                                    ; Edited 12-Jun-90 10:40 by mitani
```

(* * The DSPFONT method for PRESS-type image streams -- change the stream's current font to FONT)

```
  (PROG ((PRDATA (ffetch (STREAM IMAGEDATA) of PRSTREAM))
         CSINFO OLDFONT FDENTRY)
        (SETQ OLDFONT (ffetch PRFONT of PRDATA))
        (COND
           ([OR (NULL FONT)
                (EQ OLDFONT (SETQ FONT (OR (\GETFONTDESC FONT 'PRESS T)
                                           (FONTCOPY OLDFONT FONT]
```

(* If no new font was specified, or it's the same font, don't bother with it.)

```
               (RETURN OLDFONT)))
        (SHOW.PRESS PRSTREAM)
        (SETQ CSINFO (\GETCHARSETINFO 0 FONT T))           (* Since PRESS only uses charset 0 for now....)
        (SETQ FDENTRY (\DEFINEFONT.PRESS PRSTREAM FONT))
        (COND
           ((NEQ (ffetch FONTSET# of FDENTRY)
                 (ffetch FONTSET# of (ffetch PRCURRFDE of PRDATA)))
                                                           (* Swtich font sets)
            (\ENTITYEND.PRESS PRSTREAM)
            (\ENTITYSTART.PRESS PRSTREAM)))
        (freplace PRCURRFDE of PRDATA with FDENTRY)
        (freplace PRFONT of PRDATA with FONT)
        (\BOUT (ffetch ELSTREAM of PRDATA)
               (LOGOR FontCode (ffetch FONT# of FDENTRY)))
        (freplace PRWIDTHSCACHE of PRDATA with (fetch (CHARSETINFO WIDTHS) OF CSINFO))
        [\SETSPACE.PRESS PRSTREAM (FIXR (TIMES (ffetch PRSPACEFACTOR of PRDATA)
                                               (\FGETWIDTH (ffetch PRWIDTHSCACHE of PRDATA)
                                                           (CHARCODE SPACE]
        [freplace PRLINEFEED of PRDATA with (IDIFFERENCE (CONSTANT (IMINUS MicasPerPoint))
                                                         (FONTPROP FONT 'HEIGHT]
        (\FIXLINELENGTH.PRESS PRSTREAM)
        (RETURN OLDFONT])
```

(\DSPLEFTMARGIN.PRESS

```
[LAMBDA (PRSTREAM XPOSITION)                                    ; Edited 12-Jun-90 10:40 by mitani
  (PROG1 (ffetch PRLEFT of (ffetch (STREAM IMAGEDATA) of PRSTREAM))
      (COND
         (XPOSITION (freplace PRLEFT of (ffetch (STREAM IMAGEDATA) of PRSTREAM) with XPOSITION)
                (\FIXLINELENGTH.PRESS PRSTREAM)))))])
```

(\DSPLINEFEED.PRESS

```
[LAMBDA (PRSTREAM DELTAY)                                       ; Edited 12-Jun-90 10:40 by mitani
                                                                (* sets the amount that a line feed increases the y coordinate by.)
  (PROG ((PRDATA (ffetch (STREAM IMAGEDATA) of PRSTREAM)))
        (RETURN (PROG1 (ffetch PRLINEFEED of PRDATA)
                       [AND DELTAY (COND
                                      ((NUMBERP DELTAY)
                                       (freplace PRLINEFEED of PRDATA with DELTAY))
                                      (T (\ILLEGAL.ARG DELTAY])])
```

(\DSPRIGHTMARGIN.PRESS

```
[LAMBDA (PRSTREAM XPOSITION)                                    ; Edited 12-Jun-90 10:40 by mitani
  (PROG1 (ffetch PRRIGHT of (ffetch (STREAM IMAGEDATA) of PRSTREAM))
      (COND
         (XPOSITION (freplace PRRIGHT of (ffetch (STREAM IMAGEDATA) of PRSTREAM) with XPOSITION)
                (\FIXLINELENGTH.PRESS PRSTREAM)))))])
```

(\DSPSPACEFACTOR.PRESS

```
[LAMBDA (STREAM FACTOR)                                         ; Edited 12-Jun-90 10:40 by mitani
  (PROG ((PRDATA (ffetch (STREAM IMAGEDATA) of STREAM)))
        (RETURN (PROG1 (ffetch PRSPACEFACTOR of PRDATA)
                       [COND
                          (FACTOR (SHOW.PRESS STREAM)
                                  (freplace PRSPACEFACTOR of PRDATA with FACTOR)
                                  (\SETSPACE.PRESS STREAM (FIXR (TIMES FACTOR (\FGETWIDTH (ffetch PRWIDTHSCACHE
                                                                                             of PRDATA)
                                                                                    (CHARCODE SPACE])])
```

(\DSPTOPMARGIN.PRESS

```
[LAMBDA (PRSTREAM YPOSITION)                                    ; Edited 12-Jun-90 10:40 by mitani
  (PROG1 (fetch PRTOP of (fetch (STREAM IMAGEDATA) of PRSTREAM))
      (COND
         (YPOSITION (replace PRTOP of (fetch (STREAM IMAGEDATA) of PRSTREAM) with YPOSITION)))))])
```

(\DSPXPOSITION.PRESS

```
[LAMBDA (PRSTREAM XPOSITION)                                    ; Edited 12-Jun-90 10:40 by mitani
  (PROG1 (fetch PRXPOS of (fetch (STREAM IMAGEDATA) of PRSTREAM))
      (COND
         (XPOSITION (SHOW.PRESS PRSTREAM)
                (SETX.PRESS PRSTREAM XPOSITION)))))])
```

(\DSPYPOSITION.PRESS

```
[LAMBDA (PRSTREAM YPOSITION)                                    ; Edited 12-Jun-90 10:40 by mitani
  (PROG1 (fetch PRYPOS of (fetch (STREAM IMAGEDATA) of PRSTREAM))
      (COND
         (YPOSITION (SHOW.PRESS PRSTREAM)
                (SETY.PRESS PRSTREAM YPOSITION)))))])
```

(\FIXLINELENGTH.PRESS

```
[LAMBDA (PRSTREAM)                                              ; Edited 12-Jun-90 10:40 by mitani

        (* PRSTREAM is known to be a stream of type press. Called by RIGHTMARGIN LEFTMARGIN and \DSPFONT.PRESS to update the LINELENGTH field in the stream. also called when the stream is created.)

  (PROG (LLEN (PRDATA (ffetch (STREAM IMAGEDATA) of PRSTREAM)))
        (freplace (STREAM LINELENGTH) of PRSTREAM with (COND
                                                          ((IGREATERP [SETQ LLEN
                                                                       (IQUOTIENT (IDIFFERENCE (ffetch PRRIGHT
                                                                                                  of PRDATA)
                                                                                               (ffetch PRLEFT
                                                                                                  of PRDATA))
                                                                                  (fetch FONTAVGCHARWIDTH
                                                                                     of (fetch PRFONT of PRDATA]
                                                                       1)
                                                           LLEN)
                                                          (T 10])
```

(\OUTCHARFN.PRESS

```
[LAMBDA (PRSTREAM CHARCODE)                                     ; Edited 12-Jun-90 10:40 by mitani
                                                                (* Handle all the special-purpose characters going to a PRESS file)

  (SELCHARQ CHARCODE
```

```
        (EOL                                                    (* New Line)
             (NEWLINE.PRESS PRSTREAM)
             (replace (STREAM CHARPOSITION) of PRSTREAM with 0))
        (LF                                                     (* Line feed--move down, but not over)
            (\DSPXPOSITION.PRESS PRSTREAM (PROG1 (DSPXPOSITION NIL PRSTREAM)
                                                 (NEWLINE.PRESS PRSTREAM))))
        (^L                                                     (* Form Feed)
            (replace (STREAM CHARPOSITION) of PRSTREAM with 0)
            (NEWPAGE.PRESS PRSTREAM))
        (PROG (XPOS NEWXPOS CLIPPINGREGION (PRDATA (fetch (STREAM IMAGEDATA) of PRSTREAM)))
              (SETQ XPOS (fetch PRXPOS of PRDATA))
              (SETQ CHARCODE (\PRESS.CONVERT.NSCHARACTER CHARCODE))
              [SETQ NEWXPOS (IPLUS XPOS (COND
                                          ((EQ CHARCODE (CHARCODE SPACE))
                                           (ffetch PRSPACEWIDTH of PRDATA))
                                          (T (\FGETWIDTH (ffetch PRWIDTHSCACHE of PRDATA)
                                                         CHARCODE]
              (COND
                 ((AND [IGEQ XPOS (fetch (REGION LEFT) of (SETQ CLIPPINGREGION (fetch PRClippingRegion
                                                                                  of PRDATA]
                       (ILEQ NEWXPOS (fetch (REGION RIGHT) of CLIPPINGREGION))
                       (IGEQ (fetch PRYPOS of PRDATA)
                             (fetch (REGION BOTTOM) of CLIPPINGREGION)))
                                                                (* Bottom test should really subtract off the descent, and also
                                                                should do a top-test)
                                                                (* The Y-tests can probably be done inside SETXY, SETY, and
                  DSPFONT.)
                  [COND
                     ((NOT (ffetch CHARWASDISPLAYING of PRDATA))  (* Was being clipped, now not)
                      (freplace CHARWASDISPLAYING of PRDATA with T)
                      (SHOW.PRESS PRSTREAM)                     (* SHOW shouldn't be necessary, but |...|)
                      (SETXY.PRESS PRSTREAM XPOS (fetch PRYPOS of PRDATA]
                  (\BOUT PRSTREAM CHARCODE))
                 (T (SHOW.PRESS PRSTREAM)                       (* Don't put out any characters if out of the clipping region)
                    (freplace CHARWASDISPLAYING of PRDATA with NIL)))
              (replace PRXPOS of PRDATA with NEWXPOS])
```

(\SETSPACE.PRESS

```
  [LAMBDA (PRSTREAM S)                                          ; Edited 12-Jun-90 10:40 by mitani
    (PROG (ELSTREAM (PRDATA (fetch (STREAM IMAGEDATA) of PRSTREAM)))
          (AND (EQ S (ffetch PRSPACEWIDTH of PRDATA))
               (RETURN))
          (SETQ ELSTREAM (fetch ELSTREAM of (fetch (STREAM IMAGEDATA) of PRSTREAM)))
          (if (ILEQ S 2047)
              then (\WOUT ELSTREAM (IPLUS (LLSH SetSpaceXShortCode 8)
                                          S))
            else (\BOUT ELSTREAM SetSpaceXCode)
                 (\WOUT ELSTREAM S))
          (freplace PRSPACEWIDTH of PRDATA with S])
```

(\STARTPAGE.PRESS

```
  [LAMBDA (PRSTREAM)                                            ; Edited 12-Jun-90 10:40 by mitani
                                                                (* Should be called only when no previous page is open)
    (PROG (CFONT HFONT SPACEFACTOR (PRDATA (ffetch (STREAM IMAGEDATA) of PRSTREAM)))
          (SETQ CFONT (ffetch PRFONT of PRDATA))

          (* Save current font so that \ENTITYSTART.PRESS can make PRFONT be NIL, indicating that there is no actual font at the
          beginning of a page)

          (\ENTITYSTART.PRESS PRSTREAM)
          [COND
             ((ffetch PRHEADING of PRDATA)
              (SETQ SPACEFACTOR (ffetch PRSPACEFACTOR of PRDATA))
              (freplace PRSPACEFACTOR of PRDATA with 1)
              (SETQ HFONT (ffetch PRHEADINGFONT of PRDATA))
              (\DSPFONT.PRESS PRSTREAM HFONT)                   (* Set up heading font)
              [SETXY.PRESS PRSTREAM (ffetch PRLEFT of PRDATA)
                     (IDIFFERENCE (ffetch PRTOP of PRDATA)
                                  (FONTPROP HFONT 'ASCENT]
              (PRIN3 (ffetch PRHEADING of PRDATA)
                     PRSTREAM)                                  (* Skip an inch before page number)
              (SHOW.PRESS PRSTREAM)
              (SETX.PRESS PRSTREAM (IPLUS MICASPERINCH (ffetch PRXPOS of PRDATA)))
              (PRIN3 "Page " PRSTREAM)
              (PRIN3 (add (ffetch PRPAGENUM of PRDATA)
                          1)
                     PRSTREAM)
              (NEWLINE.PRESS PRSTREAM)                          (* Skip 2 lines)
              (NEWLINE.PRESS PRSTREAM)
              (freplace PRSPACEFACTOR of PRDATA with SPACEFACTOR))
             (T (SETXY.PRESS PRSTREAM (ffetch PRLEFT of PRDATA)
                       (IDIFFERENCE (ffetch PRTOP of PRDATA)
                                    (FONTPROP CFONT 'ASCENT]    (* Now we set the font to our (previous) current font)
          (\DSPFONT.PRESS PRSTREAM CFONT])
```

(\STRINGWIDTH.PRESS

```
[LAMBDA (STREAM STRING RDTBL)                                    ; Edited 12-Jun-90 10:40 by mitani

        (* * Returns the width of STRING in the press STREAM, observing spacefactor)

        (* * This is based on the code in \STRINGWIDTH.GENERIC)

   (PROG [(PRFONT (ffetch PRFONT of (ffetch (STREAM IMAGEDATA) of STREAM]
         [COND
            [(LITATOM STRING)                                    (* It's an atom. Loop thru its characters.)
             (if RDTBL
                 then (GO SLOW)
               else                                              (* Only doing pname, much simpler task)
                    (RETURN (LET ((WIDTHSBASE (ffetch (CHARSETINFO WIDTHS) of (\GETCHARSETINFO 0 PRFONT)))
                                  CSET)
                                 (for C inatom STRING sum (SETQ C (\PRESS.CONVERT.NSCHARACTER C))
                                                                 (* CONVERT from NS characters back to old PARC-internal
                                                                 coding for PRESS fonts)
                                                        (COND
                                                           ((EQ C (CHARCODE SPACE))
                                                            (ffetch PRSPACEWIDTH of (ffetch (STREAM IMAGEDATA)
                                                                                            of STREAM)))
                                                           (T (\FGETWIDTH WIDTHSBASE (\CHAR8CODE C]
            ((STRINGP STRING)                                    (* It's a string; we know how to loop thru its chars quickly)
             (RETURN (LET ((TOTAL 0)
                           (WIDTHSBASE (ffetch (CHARSETINFO WIDTHS) of (\GETCHARSETINFO 0 PRFONT)))
                           ESCWIDTH ESC CSET)
                          [COND
                             (RDTBL                              (* Count delimiting quotes and internal escapes)
                                    (SETQ TOTAL (UNFOLD (\FGETWIDTH WIDTHSBASE (CHARCODE %"))
                                                        2))
                                    (SETQ ESC (fetch (READTABLEP ESCAPECHAR) of RDTBL))
                                    (SETQ ESCWIDTH (\FGETWIDTH WIDTHSBASE ESC]
                          [for C instring STRING
                             do (SETQ C (\PRESS.CONVERT.NSCHARACTER C))
                                                                 (* CONVERT from NS characters back to old PARC-internal
                                                                 coding for PRESS fonts)
                                (add TOTAL (COND
                                              ((EQ C (CHARCODE SPACE))
                                               (ffetch PRSPACEWIDTH of (ffetch (STREAM IMAGEDATA) of STREAM)))
                                              (T (IPLUS (\FGETWIDTH WIDTHSBASE (\CHAR8CODE C))
                                                        (COND
                                                           ((AND RDTBL (OR (EQ C (CHARCODE %"))
                                                                           (EQ C ESC)))
                                                                 (* String char must be escaped)
                                                            ESCWIDTH)
                                                           (T 0]
                          TOTAL]
      SLOW
          (RETURN (LET ((TOTALWIDTH 0)
                        (WIDTHSBASE (ffetch (CHARSETINFO WIDTHS) of (\GETCHARSETINFO 0 PRFONT)))
                        CSET)

        (* * Neither atom nor string; we have to use \MAPPNAME to do the job.)

                       (\MAPPNAME [FUNCTION (LAMBDA (DUMMY CC)
                                            (SETQ CC (\PRESS.CONVERT.NSCHARACTER CC))
                                                                 (* Convert from NS characters back to old PARC-internal coding
                                                                 for PRESS fonts)
                                            (add TOTALWIDTH (COND
                                                               ((EQ CC (CHARCODE SPACE))
                                                                (ffetch PRSPACEWIDTH of (ffetch (STREAM IMAGEDATA
                                                                                                )
                                                                                                of STREAM)))
                                                               (T (\FGETWIDTH WIDTHSBASE (\CHAR8CODE CC]
                                  STRING RDTBL RDTBL)
                          TOTALWIDTH])
```

(SHOWRECTANGLE.PRESS

```
[LAMBDA (PRSTREAM WIDTH HEIGHT)                                  ; Edited 12-Jun-90 10:40 by mitani
  (PROG [(ELSTREAM (fetch ELSTREAM of (fetch (STREAM IMAGEDATA) of PRSTREAM]
        (\BOUT ELSTREAM ShowRectangleCode)
        (\WOUT ELSTREAM WIDTH)
        (\WOUT ELSTREAM HEIGHT])
```

(\PRESS.CONVERT.NSCHARACTER

```
[LAMBDA (CHARCODE)                                               (* jds " 4-Nov-85 08:02")

        (* Provide backward compatibility for extended-language characters in the PRESS printing environment.
        Converts certain of the NS characters into their equivalent PARC-internal charcodes)

   (SELCHARQ CHARCODE
        (357,55                                                  (* em quad)
                153)
```

```
         (357,54                                                    (* en quad)
                152)
         (357,57                                                    (* Thin space)
                159)
         (357,44                                                    (* en dash / figure dash)
                155)
         (357,45                                                    (* em dash)
                156)
         (357,146                                                   (* bullet)
                 183)
         (0,251                                                     (* left single quote)
               96)
         (0,271                                                     (* right single quote)
               39)
         (\CHAR8CODE CHARCODE])

)

;; Drawcurve code

(DEFINEQ

(\ENDVECRUN
  [LAMBDA (PRSTREAM HALFVECWIDTH)                                   ; Edited 12-Jun-90 10:40 by mitani
    (SHOW.PRESS PRSTREAM)
    (PROG ((PRDATA (fetch (STREAM IMAGEDATA) of PRSTREAM))
           ORIGXPOS ORIGYPOS XPOS YPOS WASDISPLAYING ORIGWASDISPLAYING)
          (COND
             ((NOT (fetch VECMOVINGRIGHT of PRDATA))

          (* We've been moving to the left, so it's time to uncache those characters we saved.)

              (SETQ XPOS (fetch VECCURX of PRDATA))
              (SETQ YPOS (fetch VECCURY of PRDATA))
              (SETQ ORIGXPOS (FIXR (FTIMES MicasPerScan XPOS)))  (* Remember where the end of the line is, so we can come back
                                                                 here.)
              (SETQ ORIGYPOS (FIXR (FTIMES MicasPerScan YPOS)))
              [SETQ ORIGWASDISPLAYING (AND (IGEQ XPOS (IPLUS SPRUCEPAPERLEFTSCANS HALFVECWIDTH))
                                           (IGEQ YPOS (IPLUS SPRUCEPAPERBOTTOMSCANS HALFVECWIDTH))
                                           (ILESSP YPOS (IDIFFERENCE SPRUCEPAPERTOPSCANS HALFVECWIDTH))
                                           (ILESSP XPOS (IDIFFERENCE SPRUCEPAPERRIGHTSCANS HALFVECWIDTH]
              (SETQ WASDISPLAYING ORIGWASDISPLAYING)             (* Decide whether to start out by displaying any characters or
                                                                 not.)
              (COND
                 (WASDISPLAYING (SETXY.PRESS PRSTREAM ORIGXPOS ORIGYPOS)))

          (* We may have been adjusting the X and Y position in the PRDATA without actually putting out the file commands)

              [for CH in (fetch VECSEGCHARS of PRDATA) do (COND
                                                            [(AND (IGEQ XPOS (IPLUS SPRUCEPAPERLEFTSCANS
                                                                                    HALFVECWIDTH))
                                                                  (IGEQ YPOS (IPLUS SPRUCEPAPERBOTTOMSCANS
                                                                                    HALFVECWIDTH))
                                                                  (ILESSP YPOS (IDIFFERENCE SPRUCEPAPERTOPSCANS
                                                                                      HALFVECWIDTH))
                                                                  (ILESSP XPOS (IDIFFERENCE SPRUCEPAPERRIGHTSCANS
                                                                                      HALFVECWIDTH)))
                                                                  (* We're on-paper. Go ahead and display the character.)
                                                             (COND
                                                                ((NOT WASDISPLAYING)
                                                                  (* We haven't really been displaying characters up to now--we
                                                                  need to reposition.)
                                                                  (SHOW.PRESS PRSTREAM)
                                                                  (SETXY.PRESS PRSTREAM (FIXR (FTIMES MicasPerScan
                                                                                                      XPOS))
                                                                         (FIXR (FTIMES MicasPerScan YPOS)))
                                                                  (SETQ WASDISPLAYING T)))
                                                             (\BOUT PRSTREAM (\VECENCODE (IMINUS (CAR CH))
                                                                                         (IMINUS (CDR CH]
                                                            (T    (* We are off-paper. Stop displaying, and remember that we
                                                                  took a hiatus)
                                                               (SETQ WASDISPLAYING NIL)))
                                                          (SETQ XPOS (IDIFFERENCE XPOS (CAR CH)))
                                                          (SETQ YPOS (IDIFFERENCE YPOS (CDR CH]
              (SHOW.PRESS PRSTREAM)
              (SETXY.PRESS PRSTREAM ORIGXPOS ORIGYPOS)
              (replace VECWASDISPLAYING of PRDATA with ORIGWASDISPLAYING)))
          (replace VECSEGCHARS of PRDATA with NIL])

(\VECENCODE
  [LAMBDA (DX DY)                                                   (* jds "18-DEC-81 15:48")

          (* Given dx and dy in dover pixels, decide which Vector Font character represents that move, and return it.)

    (if (ILESSP 0 DY)
        then (IDIFFERENCE (IPLUS 160 DX (IMINUS DY))
```

```
                    (ITIMES 9 (IMAX DX DY)))
      else (IDIFFERENCE (IDIFFERENCE (IDIFFERENCE 160 DX)
                                DY)
                        (ITIMES 7 (IMAX DX (IMINUS DY])
```

(\VECPUT

```
[LAMBDA (PRSTREAM DX DY HALFVECWIDTH)                    ; Edited 12-Jun-90 10:40 by mitani

        (* Send this dx,dy pair to the press file; hold and reverse any strings which run right-to-left on the page.)

   (PROG ((PRDATA (fetch (STREAM IMAGEDATA) of PRSTREAM))
          XPOS YPOS)
         (COND
            ((OR (AND (fetch VECMOVINGRIGHT of PRDATA)
                      (ILESSP DX 0))
                 (AND (NOT (fetch VECMOVINGRIGHT of PRDATA))
                      (ILESSP 0 DX)))

        (* We switched direction (LEFT->RIGHT or RIGHT->LEFT)%. Put out what we've got, and start the new run.)

             (\ENDVECRUN PRSTREAM HALFVECWIDTH)
             (replace VECMOVINGRIGHT of PRDATA with (NOT (fetch VECMOVINGRIGHT of PRDATA)))
                                                         (* Switch the direction we think we're moving.)
             ))
         (SETQ XPOS (fetch VECCURX of PRDATA))           (* In DOVER spots)
         (SETQ YPOS (fetch VECCURY of PRDATA))
         (replace VECCURX of PRDATA with (IPLUS XPOS DX))
         (replace VECCURY of PRDATA with (IPLUS YPOS DY))
         (COND
            [(fetch VECMOVINGRIGHT of PRDATA)            (* We're moving right, and are really putting out characters.)
                                                         (* SPRUCEPAPERTOPSCANS is in dover points)
             (COND
                ((AND (IGEQ YPOS (IPLUS SPRUCEPAPERBOTTOMSCANS HALFVECWIDTH))
                      (ILESSP YPOS (IDIFFERENCE SPRUCEPAPERTOPSCANS HALFVECWIDTH))
                      (IGEQ XPOS (IPLUS SPRUCEPAPERLEFTSCANS HALFVECWIDTH))
                      (ILESSP XPOS (IDIFFERENCE SPRUCEPAPERRIGHTSCANS HALFVECWIDTH)))
                                                         (* We're on-paper. Go ahead and display this character.)
                 (COND
                    ((NOT (fetch VECWASDISPLAYING of PRDATA))
                                                         (* We haven't been displaying. before really putting out the
                                                         character,)
                     (SHOW.PRESS PRSTREAM)
                     (SETXY.PRESS PRSTREAM (FIXR (FTIMES MicasPerScan XPOS))
                            (FIXR (FTIMES MicasPerScan YPOS)))
                                                         (* So move to where we're emerging onto the paper.)
                     (replace VECWASDISPLAYING of PRDATA with T)))
                 (\BOUT PRSTREAM (\VECENCODE DX DY)))
                (T                                       (* We're off-page. Remember to do a SETXY when we get back
                                                         on.)
                   (replace VECWASDISPLAYING of PRDATA with NIL]
            (T

        (* We're moving left--and so caching characters for later. Don't bother making any checks going this way.)

               (push (fetch VECSEGCHARS of PRDATA)
                     (CONS DX DY))                       (* Just cache the DX,DY pair)
              ])
```

(\VECSKIP

```
[LAMBDA (PRSTREAM DX DY)                                 (* rmk%: "17-Dec-84 10:10")
                                                         (* Put out blank space for DX, DY)
   (\ENDVECRUN PRSTREAM)
   (SETQ VecCurX (IPLUS VecCurX DX))
   (SETQ VecCurY (IPLUS VecCurY DY))
   (\ENDVECRUN PRSTREAM])
```

(\VECFONTINIT

```
[LAMBDA NIL                                              (* jds " 2-Jan-86 14:24")

        (* Initialize \VecFontDir, a list of lists of dummy font descriptors for the ReDraw vector fonts.
        The structure is ((round brushes) (square brushes) (horizontal brushes)
        (vertical brushes)))

   (DECLARE (GLOBALVARS \VecFontDir))

        (* WIDTHS is a dummy array descriptor so that \DSPFONT.PRESS doesn't get confused.
        If any real character output were done with this descriptor in force, the results would be disastrous.
        But the RESETSAVE in \PRESSCURVE2 should prevent this.)
                                                         (* NOTE%: Perhaps we should just use the unit widths vector
                                                         for this)
   (OR \VecFontDir (SETQ \VecFontDir (BIND FD CSINFO for FMLY (WIDTHS _ (ARRAY 256 'SMALLP 1 0))
                                        in '(NEWVEC SNEWVEC HNEWVEC VNEWVEC)
                                        collect (for BRUSH in '(4 8 16 32 64)
                                                   collect (SETQ FD (create FONTDESCRIPTOR
                                                                            FONTDEVICE _ 'PRESS
                                                                            FONTFAMILY _ FMLY
```

```
                                                    FONTSIZE _ BRUSH
                                                    FONTFACE _ '(MEDIUM REGULAR REGULAR)
                                                    ROTATION _ 0
                                                    FONTAVGCHARWIDTH _ 1))
                                                (* Create a dummy font descriptor for this dummy font)
                                          (SETQ CSINFO (CREATE CHARSETINFO
                                                               WIDTHS _ (FETCH (ARRAYP BASE)
                                                                               OF WIDTHS)))
                                                (* And a CHARSETINFO that claims the characters are all 0
                                                wide)
                                          (\RPLPTR (FETCH (FONTDESCRIPTOR FONTCHARSETVECTOR)
                                                          OF FD)
                                                0 CSINFO)
                                                (* And Smash it into the charset vector.)

        (* * Now collect the font descriptors for the directory)

                                          FD])
```

(**DRAWCIRCLE.PRESS**

```
[LAMBDA (STREAM CENTERX CENTERY RADIUS BRUSH DASHING)          (* rmk%: "27-Sep-84 17:23")
  (PROG [(R2RAD (FIXR (FTIMES RADIUS (CONSTANT (FQUOTIENT (SQRT 2)
                                                          2]
        (DRAWCURVE (LIST (CREATEPOSITION (IPLUS CENTERX RADIUS)
                                 CENTERY)
                         (CREATEPOSITION (IPLUS CENTERX R2RAD)
                                (IPLUS CENTERY R2RAD))
                         (CREATEPOSITION CENTERX (IPLUS CENTERY RADIUS))
                         (CREATEPOSITION (IDIFFERENCE CENTERX R2RAD)
                                (IPLUS CENTERY R2RAD))
                         (CREATEPOSITION (IDIFFERENCE CENTERX RADIUS)
                                CENTERY)
                         (CREATEPOSITION (IDIFFERENCE CENTERX R2RAD)
                                (IDIFFERENCE CENTERY R2RAD))
                         (CREATEPOSITION CENTERX (IDIFFERENCE CENTERY RADIUS))
                         (CREATEPOSITION (IPLUS CENTERX R2RAD)
                                (IDIFFERENCE CENTERY R2RAD)))
                T BRUSH DASHING STREAM))
  (MOVETO CENTERX CENTERY STREAM])
```

(**DRAWARC.PRESS**

```
[LAMBDA (STREAM CENTERX CENTERY RADIUS STARTANGLE NDEGREES BRUSH DASHING)
                                                (* rrb " 4-Oct-85 17:27")
                                                (* draws an arc on an press file)
  (\DRAWARC.GENERIC STREAM CENTERX CENTERY RADIUS STARTANGLE NDEGREES BRUSH DASHING])
```

(**DRAWCURVE.PRESS**

```
[LAMBDA (PRSTREAM KNOTS CLOSED BRUSH DASHING)                  (* rmk%: "20-Nov-84 13:59")

        (* draws a spline curve with a given brush brush. Knots and brushwidth assumed to be in micas)

  [COND
     ((LISTP KNOTS)
      (SHOW.PRESS PRSTREAM)
      (PROG [LASTKNOT (DASHLST (AND DASHING (OR (AND (LISTP DASHING)
                                                     (EVERY DASHING (FUNCTION FIXP))
                                                DASHING)
                                           (\ILLEGAL.ARG DASHING]
                                                (* The above makes sure that DASHING is a list of numbers.)
            [OR (CDR KNOTS)
                (SETQ KNOTS (LIST (CAR KNOTS)
                                  (CAR KNOTS]   (* Handle the trival one-knot case.)
            (COND
               ((AND (NULL DASHING)
                     (EQ 2 (LENGTH KNOTS))
                     (\DRAWCURVE.PRESS.LINE PRSTREAM (fetch XCOORD of (CAR KNOTS))
                            (fetch YCOORD of (CAR KNOTS))
                            (fetch XCOORD of (CADR KNOTS))
                            (fetch YCOORD of (CADR KNOTS))
                            BRUSH DASHING))

        (* There were only two knots, and no dashing. \DRAWCURVE.PRESS.LINE returned T if it managed to draw the line the
        fast way.)
                                                (* Have to move to the endpoint of the line.)
                )
               (T                               (* Otherwise, use the full-strength curve drawer.)
                  (\PRESSCURVE2 PRSTREAM (PARAMETRICSPLINE [for KNOT in KNOTS
                                                           collect (CREATEPOSITION
                                                                   (FIXR (FTIMES (fetch XCOORD of KNOT)
                                                                                 ScansPerMica))
                                                                   (FIXR (FTIMES (fetch YCOORD of KNOT)
                                                                                 ScansPerMica]
                                                      CLOSED)
                                DASHING
```

```
                        (\GETBRUSHFONT.PRESS BRUSH))          (* This already leaves the current position at the endpoint of the
                  curve.)
                  ))
              (SETQ LASTKNOT (CAR (LAST KNOTS)))
              (SETXY.PRESS PRSTREAM (fetch XCOORD of LASTKNOT)
                     (fetch YCOORD of LASTKNOT]
    PRSTREAM])
```

(\DRAWCURVE.PRESS.LINE

```
  [LAMBDA (PRSTREAM X1 Y1 X2 Y2 BRUSH DASHING)                (* rmk%: "17-Dec-84 10:05")

          (* Returns T if this is a horizontal or vertical line, hence can be drawn as a rectangle.)

    (PROG (WIDTH BACKOFF LEFT BOTTOM DIST LB TR (SHAPE 'ROUND))
          (SETQ WIDTH (OR (COND
                             ((LISTP BRUSH)
                              (SETQ SHAPE (CAR BRUSH))
                              (CADR BRUSH))
                             (T BRUSH))
                          1))
          [SELECTQ SHAPE
               (BUTT (SETQ BACKOFF 0))
               (ROUND (RETURN NIL))
               (PROGN (SETQ BACKOFF (IQUOTIENT WIDTH 2]          (* For butt ends, we want the line to end at the given coordinate
     position)                                                   (* LB is left or bottom, TR is top or right, depending on
                                                                 orientation)
          (COND
             ((EQP X1 X2)                                       (* Vertical line)
              (SETQ LEFT (IDIFFERENCE X1 (IQUOTIENT WIDTH 2)))  (* Off to the left or right?)
              (AND (OR (ILESSP LEFT SPRUCEPAPERLEFTMICAS)
                       (IGREATERP (IPLUS LEFT WIDTH)
                                  SPRUCEPAPERRIGHTMICAS))
                   (RETURN T))
              (COND
                 ((IGREATERP Y1 Y2)
                  (SETQ LB Y2)
                  (SETQ TR Y1))
                 (T (SETQ LB Y1)
                    (SETQ TR Y2)))
              (SETQ LB (IMAX SPRUCEPAPERBOTTOMMICAS (IDIFFERENCE LB BACKOFF)))
                                                                 (* Clip to page)
              (SETQ TR (IMIN SPRUCEPAPERTOPMICAS (IPLUS TR BACKOFF)))
              (SETQ DIST (IDIFFERENCE TR LB))
              (OR (IGREATERP DIST 0)
                  (RETURN T))
              (SETXY.PRESS PRSTREAM LEFT LB)                    (* Move to where the line starts)
              (SHOWRECTANGLE.PRESS PRSTREAM WIDTH DIST)         (* Draw the rectangle that will do the job.)
              (RETURN T))
             ((EQP Y1 Y2)                                       (* Horizontal line)
              (SETQ BOTTOM (IDIFFERENCE Y1 (IQUOTIENT WIDTH 2))) (* Off to the bottom or top?)
              (AND (OR (ILESSP BOTTOM SPRUCEPAPERBOTTOMMICAS)
                       (IGREATERP (IPLUS BOTTOM WIDTH)
                                  SPRUCEPAPERTOPMICAS))
                   (RETURN T))
              (COND
                 ((IGREATERP X1 X2)
                  (SETQ LB X2)
                  (SETQ TR X1))
                 (T (SETQ LB X1)
                    (SETQ TR X2)))
              (SETQ LB (IMAX SPRUCEPAPERLEFTMICAS (IDIFFERENCE LB BACKOFF)))
                                                                 (* Clip to page)
              (SETQ TR (IMIN SPRUCEPAPERRIGHTMICAS (IPLUS TR BACKOFF)))
              (SETQ DIST (IDIFFERENCE TR LB))
              (OR (IGREATERP DIST 0)
                  (RETURN T))
              (SETXY.PRESS PRSTREAM LB BOTTOM)                  (* Move to where the line starts)
              (SHOWRECTANGLE.PRESS PRSTREAM DIST WIDTH)         (* Draw the rectangle that will do the job.)
              (RETURN T])
```

(\DRAWELLIPSE.PRESS

```
  [LAMBDA (PRSTREAM CENTERX CENTERY SEMIMINORRADIUS SEMIMAJORRADIUS ORIENTATION BRUSH DASHING)
                                                              (* rmk%: "23-Aug-84 10:51")
    (PROG [(SINOR (COND
                     (ORIENTATION (SIN ORIENTATION))
                     (T 0.0)))
           (COSOR (COND
                     (ORIENTATION (COS ORIENTATION))
                     (T 1.0]
          (\DRAWCURVE.PRESS PRSTREAM [LIST (CREATEPOSITION (PLUS CENTERX (FTIMES COSOR SEMIMAJORRADIUS))
                                                  (PLUS CENTERY (FTIMES SINOR SEMIMAJORRADIUS)))
                                           (CREATEPOSITION (DIFFERENCE CENTERX (FTIMES SINOR SEMIMINORRADIUS))
                                                  (PLUS CENTERY (FTIMES COSOR SEMIMINORRADIUS)))
                                           (CREATEPOSITION (DIFFERENCE CENTERX (FTIMES COSOR SEMIMAJORRADIUS))
                                                  (DIFFERENCE CENTERY (FTIMES SINOR SEMIMAJORRADIUS)))
```

```
                                        (CREATEPOSITION (PLUS CENTERX (FTIMES SINOR SEMIMINORRADIUS))
                                               (DIFFERENCE CENTERY (FTIMES COSOR SEMIMINORRADIUS]
              T BRUSH DASHING)
         (MOVETO CENTERX CENTERY PRSTREAM])
```

(\GETBRUSHFONT.PRESS

```
[LAMBDA (BRUSH)                                                    (* rmk%: "17-Dec-84 10:13")
  (\VECFONTINIT)
  (PROG [(LIST1 (SELECTQ (CAR (LISTP BRUSH))
                     (ROUND (CAR \VecFontDir))
                     (SQUARE (CADR \VecFontDir))
                     (HORIZONTAL (CADDR \VecFontDir))
                     (VERTICAL (CADDDR \VecFontDir))
                     (BUTT (CAR \VecFontDir))
                     (CAR \VecFontDir]
        (AND (LISTP BRUSH)
             (SETQ BRUSH (CADR BRUSH)))
        (RETURN (SELECTQ (FIXR (FTIMES (OR BRUSH 1)
                                       PointsPerMica))
                     ((0 1)
                          (CAR LIST1))
                     (2 (CADR LIST1))
                     ((3 4 5)
                          (CADDR LIST1))
                     ((6 7 8)
                          (CADDDR LIST1))
                     (CADDDR LIST1])
```

(\PRESSCURVE2

```
[LAMBDA (PRSTREAM SPLINE DASHING BRUSHFONT)                        ; Edited 12-Jun-90 10:40 by mitani
                                                                   (* Given a spline curve and a font, draw the lines to PRSTREAM)
  (RESETLST
      (RESETSAVE NIL (LIST 'DSPFONT (DSPFONT BRUSHFONT PRSTREAM)
                           PRSTREAM))
      [PROG ((PRDATA (fetch (STREAM IMAGEDATA) of PRSTREAM)))
            (COND
               ((IGREATERP (IDIFFERENCE (GETFILEPTR (fetch ELSTREAM of PRDATA))
                                        (fetch ELSTARTBYTE of PRDATA))
                           25000)
                (\ENTITYEND.PRESS PRSTREAM)                        (* Hack to prevent mysterious overflow in length of entities)
                (\ENTITYSTART.PRESS PRSTREAM]
      (\BOUT (fetch ELSTREAM of (fetch (STREAM IMAGEDATA) of PRSTREAM))
             ResetSpaceCode)

        (* because the space code shouldn't be interpreted specially when we are drawing in the vector font)

      (PROG ((XPOLY (create POLYNOMIAL))
             (X'POLY (create POLYNOMIAL))
             (YPOLY (create POLYNOMIAL))
             (Y'POLY (create POLYNOMIAL))
             (X (fetch (SPLINE SPLINEX) of SPLINE))
             (Y (fetch (SPLINE SPLINEY) of SPLINE))
             (X' (fetch (SPLINE SPLINEDX) of SPLINE))
             (Y' (fetch (SPLINE SPLINEDY) of SPLINE))
             (X'' (fetch (SPLINE SPLINEDDX) of SPLINE))
             (Y'' (fetch (SPLINE SPLINEDDY) of SPLINE))
             (X''' (fetch (SPLINE SPLINEDDDX) of SPLINE))
             (Y''' (fetch (SPLINE SPLINEDDDY) of SPLINE))
             (%#KNOTS (fetch %#KNOTS of SPLINE))
             (X0 (ELT (fetch (SPLINE SPLINEX) of SPLINE)
                      1))
             (Y0 (ELT (fetch (SPLINE SPLINEY) of SPLINE)
                      1))
             IX IY DX DY XT YT X'T Y'T NEWXT NEWYT XDIFF YDIFF XWALLDT YWALLDT DUPLICATEKNOT EXTRANEOUS TT
             NEWT DELTA DASHON DASHLST DASHCNT HALFVECWIDTH PUTDX EXTRADX PUTDY EXTRADY)
            (SETQ HALFVECWIDTH (FONTPROP BRUSHFONT 'SIZE))

        (* Half the width of the brush, in dots. Used to help decide when the line we're drawing goes off-paper.)

            (SETQ DASHON T)

        (* These are initialized outside the prog-bindings cause the compiler can't hack so many initialized variables)

            (SETQ DASHLST DASHING)
            (SETQ DASHCNT (CAR DASHING))
            (SETXY.PRESS PRSTREAM (FIXR (FTIMES X0 MicasPerScan))
                   (FIXR (FTIMES Y0 MicasPerScan)))                (* Move to the first knot on the curve)
            (replace VECMOVINGRIGHT of (fetch (STREAM IMAGEDATA) of PRSTREAM) with T)
                                                                   (* Start by assuming we're moving in increasing X
                                                                   (since the vector fonts only have strokes that work in that
                                                                   direction))
            (replace VECWASDISPLAYING of (fetch (STREAM IMAGEDATA) of PRSTREAM) with (AND (GEQ X0 0)
                                                                                         (GEQ Y0 0)))
            (replace VECSEGCHARS of (fetch (STREAM IMAGEDATA) of PRSTREAM) with NIL)
            (replace VECCURX of (fetch (STREAM IMAGEDATA) of PRSTREAM) with X0)
```

```
                                                    (* And set the current X and Y positions, denominated in dover
                                                    spots)
    (replace VECCURY of (fetch (STREAM IMAGEDATA) of PRSTREAM) with Y0)
                                                    (* Set up initial values in vec variables, perform SetX/SetY.)
    (SETQ TT 0.0)
    (SETQ DELTA 16)
    (SETQ IX (FIXR X0))
    (SETQ IY (FIXR Y0))
    [for KNOT# from 1 to (SUB1 %#KNOTS)
       do (LOADPOLY XPOLY X'POLY (ELT X''' KNOT#)
                  (ELT X'' KNOT#)
                  (ELT X' KNOT#)
                  (ELT X KNOT#))                    (* Set up the polynomials that describe X and X' over this
                                                    segment)
          (LOADPOLY YPOLY Y'POLY (ELT Y''' KNOT#)
                  (ELT Y'' KNOT#)
                  (ELT Y' KNOT#)
                  (ELT Y KNOT#))                    (* Set up the polynomials that describe Y and Y' over this
                                                    segment)
          (SETQ XT (POLYEVAL TT XPOLY 3))           (* XT _ X (t) --Evaluate the next point)
          (SETQ YT (POLYEVAL TT YPOLY 3))           (* YT _ Y (t))
          (COND
             [(NOT (IEQP KNOT# (SUB1 %#KNOTS)))
```

(* This isn't the last knot. Check to see if the next knot in line is a duplicated knot.)

```
               (SETQ DUPLICATEKNOT (AND (EQP (ELT X (ADD1 KNOT#))
                                             (ELT X (IPLUS KNOT# 2)))
                                        (EQP (ELT Y (ADD1 KNOT#))
                                             (ELT Y (IPLUS KNOT# 2]
             (T (SETQ DUPLICATEKNOT NIL)))
          [until (GEQ TT 1.0) do
```

(* Run the parameter, TT, from 0.0 up to 1.0. That moves the X and Y locations smoothly from this knot to the next one.)

```
                                    (SETQ X'T (POLYEVAL TT X'POLY 2))
                                                    (* X'T _ X' (t))
                                    (SETQ Y'T (POLYEVAL TT Y'POLY 2))
                                                    (* Y'T _ Y' (t))
                                    (COND
                                       ((EQP X'T 0.0)       (* Never let X' really get to 0.0 -- things become ill-conditioned
                                                            there.)
                                        (SETQ X'T 5.0E-4)))
                                    (COND
                                       ((EQP Y'T 0.0)       (* Likewise Y'.)
                                        (SETQ Y'T 5.0E-4)))
                                    [COND
                                       ((FGTP X'T 0.0)      (* If X' is positive, we'll try moving in the +X direction)
                                        (SETQ DX DELTA))
                                       (T                   (* If not, we'll try the -X direction.)
                                          (SETQ DX (IMINUS DELTA]
                                    [COND
                                       ((FGTP Y'T 0.0)      (* Likewise, if Y' is positive, try moving by DELTA in the +Y
                                                            direction)
                                        (SETQ DY DELTA))
                                       (T (SETQ DY (IMINUS DELTA]
                                    (SETQ XWALLDT (FQUOTIENT (FDIFFERENCE (IPLUS IX DX)
                                                                          XT)
                                                             X'T))
                                                    (* Compute a dT, based on moving by DELTA in X.)
                                    (SETQ YWALLDT (FQUOTIENT (FDIFFERENCE (IPLUS IY DY)
                                                                          YT)
                                                             Y'T))
                                                    (* And a dT based on moving by DELTA in Y.)
                                    [COND
                                       ((FLESSP XWALLDT YWALLDT)
```

(* Use the smaller of the two dT's. In this case, dT for X was smaller, so compute a new DY as depending on DX.)

```
                                        (SETQ NEWT (FPLUS TT XWALLDT))
                                        (SETQ DY (IDIFFERENCE (FIXR (FPLUS YT (FTIMES XWALLDT Y'T)))
                                                              IY)))
                                       (T
```

(* Changing Y gave the smaller dT. Compute a new DX, as though it depended on DY.)

```
                                          (SETQ NEWT (FPLUS TT YWALLDT))
                                          (SETQ DX (IDIFFERENCE (FIXR (FPLUS XT (FTIMES YWALLDT X'T)))
                                                                IX]
                                    (SETQ PUTDX DX)
                                    (SETQ EXTRADX 0)
                                    (SETQ PUTDY DY)
                                    (SETQ EXTRADY 0)
                                    [COND
                                       ((IGREATERP DX 16)
                                        (SETQ PUTDX 16)
                                        (SETQ EXTRADX (IDIFFERENCE DX 16]
```

```
                                        [COND
                                           ((IGREATERP -16 DX)
                                            (SETQ PUTDX -16)
                                            (SETQ EXTRADX (IPLUS DX 16]
                                        [COND
                                           ((IGREATERP DY 16)
                                            (SETQ PUTDY 16)
                                            (SETQ EXTRADY (IDIFFERENCE DY 16]
                                        [COND
                                           ((IGREATERP -16 DY)
                                            (SETQ PUTDY -16)
                                            (SETQ EXTRADY (IPLUS DY 16]
                                        (COND
                                           ([AND (FGTP NEWT 1.0)
                                                 (OR DUPLICATEKNOT (EQ KNOT# (SUB1 %#KNOTS]
                                            (SETQ NEWT 1.0)))
                                        (SETQ NEWXT (POLYEVAL NEWT XPOLY 3))
                                                                (* New XT _ X (new t))
                                        (SETQ NEWYT (POLYEVAL NEWT YPOLY 3))
                                                                (* New YT _ Y (new t))
                                        (SETQ XDIFF (ABS (FDIFFERENCE (IPLUS IX DX)
                                                                      NEWXT)))
                                        (SETQ YDIFF (ABS (FDIFFERENCE (IPLUS IY DY)
                                                                      NEWYT)))
                                        (COND
                                           ((AND (IGREATERP DELTA 1)
                                                 (OR (FGTP XDIFF 1.0)
                                                     (FGTP YDIFF 1.0)))
```

(* If we're more than a dover spot off where we'd expect to be because of the size of DELTA--and if there's room to make DELTA smaller--then try DELTA_DELTA/2)

```
                                            (SETQ DELTA (LRSH DELTA 1)))
                                           (T
```

(* No, this estimate is close enough. Put out a vector segment based on it, and move to the new TT.)

```
                                                 (\VECPUT PRSTREAM PUTDX PUTDY HALFVECWIDTH)
                                                                (* Print out a stroke using the vector font.)
                                                 (COND
                                                    ((OR (NEQ EXTRADX 0)
                                                         (NEQ EXTRADY 0))
                                                                (* If, actually, it was too big for one stroke, use another.)
                                                     (\VECPUT PRSTREAM EXTRADX EXTRADY HALFVECWIDTH)))
                                                 (SETQ IX (IPLUS IX DX))
                                                                (* Our new current location, in Dover spots)
                                                 (SETQ IY (IPLUS IY DY))
                                                 (SETQ TT NEWT)   (* Set TT to its new value)
                                                 (SETQ XT NEWXT)  (* And set the new floating-point values for X
                                                                  (t) and Y (t)%.)
                                                 (SETQ YT NEWYT)
                                                 (COND
                                                    ((AND (ILESSP DELTA 16)
                                                          (OR (FLESSP XDIFF 0.5)
                                                              (FLESSP YDIFF 0.5)))
                                                                (* If we were especially close, try making DELTA larger for the
                                                                next go round.)
                                                     (SETQ DELTA (LLSH DELTA 1]
                                        (SETQ TT (FDIFFERENCE TT 1.0))
```

(* Having moved past a knot, back the value of the parameter TT back down.
However, don't set it to 0.0--let's try to keep the line going from where it got to in passing the last knot.)

```
                                        (COND
                                           (DUPLICATEKNOT
```

(* This next knot is a duplicate. Skip over it, and start from the following knot.
This will avoid odd problems trying to go nowhere while obeying the constraints of X' and Y' at that knot--since it's a duplicate, X' and Y' are discontinuous there.)

```
                                                    (add KNOT# 1]
                                        (\ENDVECRUN PRSTREAM HALFVECWIDTH)))])

)

(RPAQ? \VecFontDir )

(DECLARE%: EVAL@COMPILE

(RPAQQ \MicasPerInch 2540)

(CONSTANTS (\MicasPerInch 2540))
)

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE
```

```
(RPAQQ ScansPerIn 384)

(RPAQQ PointsPerIn 72.27)

(RPAQ MicasPerScan (FQUOTIENT \MicasPerInch ScansPerIn))

(RPAQ ScansPerMica (FQUOTIENT ScansPerIn \MicasPerInch))

(RPAQ ScansPerPoint (FQUOTIENT ScansPerIn PointsPerIn))

(RPAQ PointsPerScan (FQUOTIENT PointsPerIn ScansPerIn))

(RPAQ MicasPerPoint (FQUOTIENT \MicasPerInch PointsPerIn))

(RPAQ PointsPerMica (FQUOTIENT PointsPerIn \MicasPerInch))

(RPAQQ SPRUCEPAPERTOPSCANS 4096)

(RPAQ SPRUCEPAPERTOPMICAS (FIX (FQUOTIENT (FTIMES SPRUCEPAPERTOPSCANS \MicasPerInch)
                                          ScansPerIn)))

(RPAQ SPRUCEPAPERRIGHTMICAS (FIX (FTIMES 8.5 \MicasPerInch)))

(RPAQ SPRUCEPAPERRIGHTSCANS (FIX (FTIMES 8.5 ScansPerIn)))

(RPAQQ SPRUCEPAPERBOTTOMSCANS 0)

(RPAQQ SPRUCEPAPERBOTTOMMICAS 0)

(RPAQQ SPRUCEPAPERLEFTSCANS 0)

(RPAQQ SPRUCEPAPERLEFTMICAS 0)

(CONSTANTS (ScansPerIn 384)
       (PointsPerIn 72.27)
       (MicasPerScan (FQUOTIENT \MicasPerInch ScansPerIn))
       (ScansPerMica (FQUOTIENT ScansPerIn \MicasPerInch))
       (ScansPerPoint (FQUOTIENT ScansPerIn PointsPerIn))
       (PointsPerScan (FQUOTIENT PointsPerIn ScansPerIn))
       (MicasPerPoint (FQUOTIENT \MicasPerInch PointsPerIn))
       (PointsPerMica (FQUOTIENT PointsPerIn \MicasPerInch))
       (SPRUCEPAPERTOPSCANS 4096)
       (SPRUCEPAPERTOPMICAS (FIX (FQUOTIENT (FTIMES SPRUCEPAPERTOPSCANS \MicasPerInch)
                                            ScansPerIn)))
       (SPRUCEPAPERRIGHTMICAS (FIX (FTIMES 8.5 \MicasPerInch)))
       (SPRUCEPAPERRIGHTSCANS (FIX (FTIMES 8.5 ScansPerIn)))
       (SPRUCEPAPERBOTTOMSCANS 0)
       (SPRUCEPAPERBOTTOMMICAS 0)
       (SPRUCEPAPERLEFTSCANS 0)
       (SPRUCEPAPERLEFTMICAS 0))
)
)
```

;; Initialization code

```
(DEFINEQ

(\PRESSINIT
  [LAMBDA NIL                                                    (* rrb " 4-Oct-85 17:27")
    (DECLARE (GLOBALVARS \PRESSIMAGEOPS))
    (SETQ \PRESSIMAGEOPS (create IMAGEOPS
                                 IMAGETYPE _ 'PRESS
                                 IMCLOSEFN _ (FUNCTION \CLOSEF.PRESS)
                                 IMXPOSITION _ (FUNCTION \DSPXPOSITION.PRESS)
                                 IMYPOSITION _ (FUNCTION \DSPYPOSITION.PRESS)
                                 IMFONT _ (FUNCTION \DSPFONT.PRESS)
                                 IMLEFTMARGIN _ (FUNCTION \DSPLEFTMARGIN.PRESS)
                                 IMRIGHTMARGIN _ (FUNCTION \DSPRIGHTMARGIN.PRESS)
                                 IMLINEFEED _ (FUNCTION \DSPLINEFEED.PRESS)
                                 IMDRAWLINE _ (FUNCTION \DRAWLINE.PRESS)
                                 IMDRAWCURVE _ (FUNCTION \DRAWCURVE.PRESS)
                                 IMDRAWCIRCLE _ (FUNCTION \DRAWCIRCLE.PRESS)
                                 IMDRAWELLIPSE _ (FUNCTION \DRAWELLIPSE.PRESS)
                                 IMFILLCIRCLE _ [FUNCTION (LAMBDA (STREAM)
                                                            (\UNIMPIMAGEOP STREAM 'FILLCIRCLE]
                                 IMBLTSHADE _ (FUNCTION \BLTSHADE.PRESS)
                                 IMBITBLT _ (FUNCTION \BITBLT.PRESS)
                                 IMSCALE _ [FUNCTION (LAMBDA NIL
                                                       (CONSTANT (FQUOTIENT MICASPERINCH 72]
                                 IMTERPRI _ (FUNCTION NEWLINE.PRESS)
                                 IMBOTTOMMARGIN _ (FUNCTION \DSPBOTTOMMARGIN.PRESS)
                                 IMTOPMARGIN _ (FUNCTION \DSPTOPMARGIN.PRESS)
                                 IMFONTCREATE _ 'PRESS
                                 IMNEWPAGE _ (FUNCTION NEWPAGE.PRESS)
                                 IMSPACEFACTOR _ (FUNCTION \DSPSPACEFACTOR.PRESS)
                                 IMSTRINGWIDTH _ (FUNCTION \STRINGWIDTH.PRESS)
```

```
                                IMCHARWIDTH _ (FUNCTION \CHARWIDTH.PRESS)
                                IMBITMAPSIZE _ (FUNCTION \BITMAPSIZE.PRESS)
                                IMCLIPPINGREGION _ (FUNCTION \DSPCLIPPINGREGION.PRESS)
                                IMSCALEDBITBLT _ (FUNCTION \SCALEDBITBLT.PRESS)
                                IMDRAWARC _ (FUNCTION \DRAWARC.PRESS])

)

(DECLARE%: DONTEVAL@LOAD DOCOPY

(\PRESSINIT)
)

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

[DATATYPE PRESSDATA (PRHEADING                                    (* The string to be printed atop each page.)
                           PRHEADINGFONT                          (* Font to print the heading in)
                           PRXPOS                                 (* Current X position)
                           PRYPOS                                 (* Current Y position)
                           PRFONT                                 (* Current font)
                           PRCURRFDE PRESSFONTDIR PRWIDTHSCACHE PRCOLOR PRLINEFEED PRPAGESTATE PDSTREAM ELSTREAM
                           XPRPAGEREGION PRDOCNAME (PRLEFT WORD)  (* Page left margin)
                           (PRBOTTOM WORD)                        (* Page bottom margin)
                           (PRRIGHT WORD)                         (* Page right margin)
                           (PRTOP WORD)                           (* Page top margin)
                           (PRPAGENUM WORD)                       (* Current Page number)
                           (PRNEXTFONT# BYTE)
                           (PRMAXFONTSET BYTE)
                           (PRPARTSTART INTEGER)
                           (DLSTARTBYTE INTEGER)
                           (ELSTARTBYTE INTEGER)
                           (STARTCHARBYTE INTEGER)
                           (VECMOVINGRIGHT FLAG)                  (* If we're drawing a curve with vector fonts, are we moving to
                                                                  the right?)
                           (VECWASDISPLAYING FLAG)

           (* Used during curve/line clipping to remember whether we were on-screen or not, so we know when to force a SETXY.)

                           VECSEGCHARS                            (* Cache for vector characters while we're moving to the left.)
                           VECCURX                                (* Current X position within vector code, in Dover spots)
                           VECCURY                                (* Current Y position with vector code, in Dover spots)
                           PRSPACEFACTOR PRSPACEWIDTH (CHARWASDISPLAYING FLAG)
                                                                  (* Says whether we have been printing characters inside the
                                                                  clipping region)
                           PRClippingRegion

           (* The edges of the paper, as far as PRESS is concerned. Used to protect SPRUCE users who get killed when the image
           goes off-paper)

                           )
       PRSPACEFACTOR _ 1 PRXPOS _ 0 PRYPOS _ 0                    (* We assume that the origin is translated to the bottom-left of
                                                                  the page region)
       PRClippingRegion _ (create REGION
                                  LEFT _ SPRUCEPAPERLEFTMICAS
                                  BOTTOM _ SPRUCEPAPERBOTTOMMICAS
                                  WIDTH _ (DIFFERENCE SPRUCEPAPERRIGHTMICAS SPRUCEPAPERLEFTMICAS)
                                  HEIGHT _ 29210)
       (ACCESSFNS ((PRWIDTH (IDIFFERENCE (fetch (PRESSDATA PRRIGHT) of DATUM)
                                         (fetch (PRESSDATA PRLEFT) of DATUM)))
                   (PRHEIGHT (IDIFFERENCE (fetch (PRESSDATA PRTOP) of DATUM)
                                          (fetch (PRESSDATA PRBOTTOM) of DATUM)))
                   (PRPAGEREGION (fetch (PRESSDATA XPRPAGEREGION) of DATUM)
                          (PROGN (replace (PRESSDATA XPRPAGEREGION) of DATUM with NEWVALUE)
                                 (replace (PRESSDATA PRLEFT) of DATUM with (fetch (REGION LEFT) of NEWVALUE))
                                 (replace (PRESSDATA PRBOTTOM) of DATUM with (fetch (REGION BOTTOM) of NEWVALUE))
                                 (replace (PRESSDATA PRRIGHT) of DATUM with (IPLUS (fetch (REGION LEFT)
                                                                                          of NEWVALUE)
                                                                                   (fetch (REGION WIDTH)
                                                                                          of NEWVALUE)))
                                 (replace (PRESSDATA PRTOP) of DATUM with (IPLUS (fetch (REGION BOTTOM)
                                                                                        of NEWVALUE)
                                                                                 (fetch (REGION HEIGHT)
                                                                                        of NEWVALUE]

(RECORD FONTDIRENTRY (DESCR FONT# FONTSET#))
)

(/DECLAREDATATYPE 'PRESSDATA
       '(POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
              POINTER POINTER WORD WORD WORD WORD WORD BYTE BYTE FIXP FIXP FIXP FIXP FLAG FLAG POINTER POINTER
              POINTER POINTER POINTER FLAG POINTER)

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

       '56)
)
```

```
(/DECLAREDATATYPE 'PRESSDATA
       '(POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
              POINTER POINTER WORD WORD WORD WORD WORD BYTE BYTE FIXP FIXP FIXP FIXP FLAG FLAG POINTER POINTER
              POINTER POINTER POINTER FLAG POINTER)
       ;; ---field descriptor list elided by lister---
       '56)

(RPAQ? DEFAULTPAGEREGION (CREATEREGION 2794 1905 16256 24765))

(RPAQ? PRESSBITMAPREGION (CREATEREGION 1270 1270 (FIX (TIMES 7.5 \MicasPerInch))
                                (TIMES 10 \MicasPerInch)))

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS DEFAULTPAGEREGION)
)

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

(RPAQQ BYTESPERRECORD 512)

(RPAQQ LISPENTITYTYPE 6)

(RPAQ MICASPERINCH \MicasPerInch)

(CONSTANTS (BYTESPERRECORD 512)
       (LISPENTITYTYPE 6)
       (MICASPERINCH \MicasPerInch))
)

(RPAQQ PRESSOPS
       (SetX SetY ShowCharacters ShowCharactersShortCode SkipCharactersShortCode ShowCharactersAndSkipCode
             SetSpaceXShortCode SetSpaceYShortCode FontCode SkipControlBytesImmediateCode AlternativeCode
             OnlyOnCopyCode SetXCode SetYCode ShowCharactersCode SkipCharactersCode SkipControlBytesCode
             ShowCharacterImmediateCode SetSpaceXCode SetSpaceYCode ResetSpaceCode SpaceCode SetBrightnessCode
             SetHueCode SetSaturationCode ShowObjectCode ShowDotsCode ShowDotsOpaqueCode ShowRectangleCode
             NopCode))

(DECLARE%: EVAL@COMPILE

(RPAQQ SetX 0)

(RPAQQ SetY 1)

(RPAQQ ShowCharacters 2)

(RPAQQ ShowCharactersShortCode 0)

(RPAQQ SkipCharactersShortCode 32)

(RPAQQ ShowCharactersAndSkipCode 64)

(RPAQQ SetSpaceXShortCode 96)

(RPAQQ SetSpaceYShortCode 104)

(RPAQQ FontCode 112)

(RPAQQ SkipControlBytesImmediateCode 235)

(RPAQQ AlternativeCode 236)

(RPAQQ OnlyOnCopyCode 237)

(RPAQQ SetXCode 238)

(RPAQQ SetYCode 239)

(RPAQQ ShowCharactersCode 240)

(RPAQQ SkipCharactersCode 241)

(RPAQQ SkipControlBytesCode 242)

(RPAQQ ShowCharacterImmediateCode 243)

(RPAQQ SetSpaceXCode 244)

(RPAQQ SetSpaceYCode 245)

(RPAQQ ResetSpaceCode 246)

(RPAQQ SpaceCode 247)
```

```
(RPAQQ SetBrightnessCode 248)

(RPAQQ SetHueCode 249)

(RPAQQ SetSaturationCode 250)

(RPAQQ ShowObjectCode 251)

(RPAQQ ShowDotsCode 252)

(RPAQQ ShowDotsOpaqueCode 253)

(RPAQQ ShowRectangleCode 254)

(RPAQQ NopCode 255)

(CONSTANTS SetX SetY ShowCharacters ShowCharactersShortCode SkipCharactersShortCode ShowCharactersAndSkipCode
       SetSpaceXShortCode SetSpaceYShortCode FontCode SkipControlBytesImmediateCode AlternativeCode
       OnlyOnCopyCode SetXCode SetYCode ShowCharactersCode SkipCharactersCode SkipControlBytesCode
       ShowCharacterImmediateCode SetSpaceXCode SetSpaceYCode ResetSpaceCode SpaceCode SetBrightnessCode
       SetHueCode SetSaturationCode ShowObjectCode ShowDotsCode ShowDotsOpaqueCode ShowRectangleCode NopCode)
)
)

;; Hardcopy user interface connections:

(DEFINEQ

(MAKEPRESS
  [LAMBDA (FILE PFILE FONTS HEADING TABS PRINTOPTIONS)          ; Edited 26-Aug-87 13:57 by Snow
    (TEXTTOIMAGEFILE FILE PFILE 'PRESS FONTS HEADING TABS PRINTOPTIONS])

(PRESSFILEP
  [LAMBDA (FILE)                                               ; Edited 20-Feb-87 18:41 by jds
    ;; Returns FILE if it looks like a Press file
    (AND (SETQ FILE (OR (STREAMP FILE)
                        (FINDFILE FILE)))
         (PROG [(LEN (GETFILEINFO FILE 'LENGTH]
               (AND (NOT (ZEROP LEN))
                    (EVENP LEN BYTESPERRECORD)
                    (RESETLST
                        [COND
                           (T (RESETSAVE (SETQ PRESS-STREAM (OPENSTREAM FILE 'INPUT 'OLD 8))
                                      '(PROGN (CLOSEF? OLDVALUE]
                        (SETFILEPTR PRESS-STREAM (IDIFFERENCE LEN BYTESPERRECORD))
                        (IEQP 27183 (\WIN PRESS-STREAM)))
                    (RETURN FILE])

(PRESS.BITMAPSCALE
  [LAMBDA (WIDTH HEIGHT)                                       ; Edited 12-Jun-90 10:38 by mitani
    (MIN (FQUOTIENT (TIMES (fetch (REGION HEIGHT) of PRESSBITMAPREGION)
                           PointsPerMica)
                    HEIGHT)
         (FQUOTIENT (TIMES (fetch (REGION WIDTH) of PRESSBITMAPREGION)
                           PointsPerMica)
                    WIDTH)
         (PROG1 2                                              (* MAXPRESSRATIO)])

)

(ADDTOVAR IMAGESTREAMTYPES (PRESS (OPENSTREAM OPENPRSTREAM)
                                  (FONTCREATE \CREATEPRESSFONT)
                                  (CREATECHARSET \CREATECHARSET.PRESS)
                                  (FONTSAVAILABLE \SEARCHPRESSFONTS)))

(ADDTOVAR PRINTERTYPES
          ((PRESS SPRUCE PENGUIN DOVER)
           (CANPRINT (PRESS))
           (STATUS PUP.PRINTER.STATUS)
           (PROPERTIES PUP.PRINTER.PROPERTIES)
           (SEND EFTP)
           (BITMAPSCALE NIL)
           (BITMAPFILE (PRESSBITMAP FILE BITMAP SCALEFACTOR REGION ROTATION TITLE)))
          ((FULLPRESS RAVEN)
                                                               ; same as PRESS but can scale bitmaps
           (CANPRINT (PRESS))
           (STATUS TRUE)
           (PROPERTIES NILL)
           (SEND EFTP)
           (BITMAPSCALE PRESS.BITMAPSCALE)
           (BITMAPFILE (FULLPRESSBITMAP FILE BITMAP SCALEFACTOR REGION ROTATION TITLE))))

(ADDTOVAR PRINTFILETYPES
```

```
          [PRESS (TEST PRESSFILEP)
                 (EXTENSION (PRESS))
                 (CONVERSION (TEXT MAKEPRESS TEDIT (LAMBDA (FILE PFILE FONTS HEADING)
                                                           (SETQ FILE (OPENTEXTSTREAM FILE))
                                                           (TEDIT.FORMAT.HARDCOPY FILE PFILE T NIL NIL NIL
                                                                  'PRESS)
                                                           (CLOSEF? FILE)
                                                           PFILE])
```

(PUTPROPS **PRESS COPYRIGHT** ("Venue & Xerox Corporation" 1981 1982 1983 1984 1985 1986 1987 1990 1993 2021))

FUNCTION INDEX

CONSTANT INDEX

VARIABLE INDEX

RECORD INDEX