



```

        (HOSTNAMEP NSPRINTER.HOSTNAMEP)
        (STATUS NSPRINTER.STATUS)
        (PROPERTIES NSPRINTER.PROPERTIES)
        (SEND NSPRINT)
        (BITMAPSCALE INTERPRESS.BITMAPSCALE)
        (BITMAPFILE (INTERPRESSBITMAP FILE BITMAP SCALEFACTOR REGION ROTATION TITLE])
    (PRINTFILETYPES (INTERPRESS (TEST INTERPRESSFILEP)
        (EXTENSION (IP IPR INTERPRESS))
        (CONVERSION (TEXT MAKEINTERPRESS TEDIT \TEDIT.HARDCOPY])
    (INITVARS (DEFAULT.INTERPRESS.BITMAP.ROTATION 90))
    (ALISTS (SYSTEMINITVARS INTERPRESSFONTDIRECTORIES))
    [INITVARS (INTERPRESSFONTEXTENSIONS ' (WD))
        (INTERPRESSFONTDIRECTORIES ' ("Erinyes}<Lyric>Fonts>"))
        (INTERPRESSPRINTWHEELFAMILIES ' (BOLDPS ELITE LETTERGOTHIC MASTER PICA PSBOLD SCIENTIFIC SPOKESMAN
            TITAN TREND TRENDPS TROJAN VINTAGE))
        (INTERPRESSFAMILYALIASES ' (LOGO LOGOTYPES-XEROX]
    (COMS ; NS Character Encoding
        (FNS NSMAP \COERCEASCIITONSFONT \CREATEINTERPRESSFONT \SEARCHINTERPRESSFONTS)
        (DECLARE%: EVAL@COMPILE DONTCOPY (CONSTANTS (noInfoCode 32768)))
        (INITVARS (ASCIITONSTRANSLATIONS))
        ;; These are in priority order: if an early one doesn't find a font for a family, the later ones are tried (essentially going to MODERN as
        ;; the default).
        (ADDVARS (ASCIITONSTRANSLATIONS (TIMESROMAN NIL CLASSIC)
            (GACHA NIL TERMINAL)
            (HELVETICA NIL MODERN)
            (CLASSIC NIL MODERN)
            (GACHA NIL MODERN)
            (TIMESROMAN NIL MODERN)
            (LOGO NIL LOGOTYPES)
            (HIPPO HIPPOTONSARRAY CLASSIC)
            (CYRILLIC CYRILLICTONSARRAY CLASSIC)
            (SYMBOL \SYMBOLTONSARRAY MODERN)
            (MATH \MATHTONSARRAY CLASSIC)))
        (UGLYVARS \SYMBOLTONSARRAY HIPPOTONSARRAY CYRILLICTONSARRAY \MATHTONSARRAY))
    (DECLARE%: DONTVAL@LOAD DOCOPY (P (\INTERPRESSINIT))
    (DECLARE%: EVAL@COMPILE DONTCOPY (P (LOADDEF 'SYSTEMBRUSH 'RESOURCES 'IMAGEIO)
        (LOADDEF 'BRUSH 'RECORDS 'IMAGEIO])

```

;; Literal interface

```

(RPAQ? CHARACTERCODEVERSION 'XC1-1-1)
(RPAQ? INTERPRESSVERSION "2.1")
(RPAQ? PRINTSERVICE 10.0)
(RPAQ? DEFAULTINTERPRESSMEDIUM ' (PAPER (KNOWN.SIZE "US.LETTER")))
(RPAQQ KNOWN.MEDIA.SIZES
    (("US.LETTER" (216 279))
     ("US.LEGAL" (216 356))
     ("A0" (841 1189))
     ("A1" (594 841))
     ("A2" (420 594))
     ("A3" (297 420))
     ("A4" (210 297))
     ("A5" (148 210))
     ("A6" (105 148))
     ("A7" (74 105))
     ("A8" (52 74))
     ("A9" (37 52))
     ("A10" (26 37))
     ("ISO.B0" (1000 1414))
     ("ISO.B1" (707 1000))
     ("ISO.B2" (500 707))
     ("ISO.B3" (353 500))
     ("ISO.B4" (250 353))
     ("ISO.B5" (176 250))
     ("ISO.B6" (125 176))
     ("ISO.B7" (88 125))
     ("ISO.B8" (62 88))
     ("ISO.B9" (44 62))
     ("ISO.B10" (31 44))
     ("JIS.B0" (1030 1456))
     ("JIS.B1" (728 1030))
     ("JIS.B2" (515 728))
     ("JIS.B3" (364 515))
     ("JIS.B4" (257 364))
     ("JIS.B5" (182 257))
     ("JIS.B6" (128 182))
     ("JIS.B7" (91 128))
     ("JIS.B8" (64 91))
     ("JIS.B9" (45 64))
     ("JIS.B10" (32 45))))

```

```

(DECLARE%: DONTCOPY EVAL@COMPILE

```

(RPAQQ **IPCONSTANTS** (IPVALUES RATIONALS NONPRIMS SEQUENCETYPES IPTYPES OPERATORS TOKENFORMATS IMAGERVARIABLES STROKEENDS IP82CONSTANTS))

(RPAQQ **IPVALUES** ((ENCODING 'IP-82)  
(\INTERPRESSSCALE (FQUOTIENT MICASPERINCH POINTSPERINCH))  
(MicasToDev (FQUOTIENT 300 MICASPERINCH))))

(RPAQQ **RATIONALS**  
( (METERSPERRAVENSPOT 1/11811)  
(MICASPERSCREENPOINT 127/4)  
(SCREENPOINTSPERMICA 4/127)  
(MICASPERPOINT 635/18)  
(POINTSPERINCH 72)  
(POINTSPERMICA 18/635)  
(POINTSPERMETER 360000/127)  
(METERSPERPOINT 127/360000)  
(MICASPERMETER 100000)  
(METERSPERMICA 1/100000)  
(RATZERO 0)  
(RATONE 1)  
(RAVENSPOTSPERINCH 300)  
(MICASPERRAVENSPOT 127/15)  
(RAVENSPOTSPERMICA 15/127)  
(ONEHALF 1/2)))

(RPAQQ **NONPRIMS** ((BEGINMASTER 102)  
(ENDMASTER 103)  
(PAGEINSTRUCTIONS 105)  
( { 106)  
( } 107)))

(RPAQQ **SEQUENCETYPES**  
( (SEQADAPTIVEPIXELVECTOR 12)  
(SEQCOMMENT 6)  
(SEQCOMPRESSPIXELVECTOR 10)  
(SEQCONTINUED 7)  
(SEQIDENTIFIER 5)  
(SEQINSERTFILE 11)  
(SEQINTEGER 2)  
(SEQLARGEVECTOR 8)  
(SEQPACKEDPIXELVECTOR 9)  
(SEQRATIONAL 4)  
(SEQSTRING 1)))

(RPAQQ **IPTYPES** ((COLOR.IPTYPE 7)  
(IDENTIFIER.IPTYPE 2)  
(NUMBER.IPTYPE 1)  
(OPERATOR.IPTYPE 4)  
(OUTLINE.IPTYPE 9)  
(PIXELARRAY.IPTYPE 6)  
(TRAJECTORY.IPTYPE 8)  
(TRANSFORMATION.IPTYPE 5)  
(VECTOR.IPTYPE 3)))

(RPAQQ **OPERATORS**  
( (ABS 200)  
(ADD 201)  
(AND 202)  
(ARCTO 403)  
(CEILING 203)  
(CLIPRECTANGLE 419)  
(CONCAT 165)  
(CONCATT 168)  
(COPY 183)  
(CORRECT 110)  
(CORRECTMASK 156)  
(CORRECTSPACE 157)  
(COUNT 188)  
(DIV 204)  
(DO 231)  
(DOSAVE 232)  
(DOSAVEALL 233)  
(DOSAVESIMPLEBODY 120)  
(DUP 181)  
(EQ 205)  
(ERROR.IPOP 600)  
(EXCH 185)  
(FGET 20)  
(FINDCOLOR 423)  
(FINDCOLORMODELOPERATOR 422)  
(FINDCOLOROPERATOR 421)  
(FINDDECOMPRESSOR 149)  
(FINDFONT 147)  
(FLOOR 206)  
(FSET 21)  
(GE 207)

```

(GETCP 159)
(GETPROP 287)
(GT 208)
(IF 239)
(IFCOPY 240)
(IFELSE 241)
(IGET 18)
(ISET 19)
(LINETO 23)
(LINETOX 14)
(LINETOY 15)
(MAKEGRAY 425)
(MAKEOUTLINE 417)
(MAKEOUTLINEODD 416)
(MAKEPIXELARRAY 450)
(MAKESAMPLEDBLACK 426)
(MAKESAMPLEDCOLOR 427)
(MAKESIMPLECO 114)
(MAKEPIXELARRAY 450)
(MAKEVEC 283)
(MAKEVECLU 282)
(MARK 186)
(MASKFILL 409)
(MASKPIXEL 452)
(MASKRECTANGLE 410)
(MASKSTROKE 24)
(MASKTRAPEZOIDX 411)
(MASKTRAPEZOIDY 412)
(MASKUNDERLINE 414)
(MASKVECTOR 441)
(MERGEPROP 288)
(MOD 209)
(MODIFYFONT 148)
(MOVE 169)
(MOVETO 25)
(MUL 210)
(NEG.IPOP 211)
(NOP 1)
(NOT 212)
(OR 213)
(POP 180)
(REM 216)
(ROLL 184)
(ROTATE 163)
(ROUND.IPOP 217)
(SCALE.OP 164)
(SCALE2 166)
(SETCORRECTMEASURE 154)
(SETCORRECTTOLERANCE 155)
(SETFONT 151)
(SETGRAY 424)
(SETXREL 12)
(SETXY 10)
(SETXYREL 11)
(SETYREL 13)
(SHAPE.IPOP 285)
(SHOW 22)
(SHOWANDXREL 146)
(SPACE 16)
(STARTUNDERLINE 413)
(SUB 214)
(TRANS.IPOP 170)
(TRANSLATE 162)
(TRUNC 215)
(TYPE.OP 220)
(UNMARK 187)
(UNMARK0 192))

```

```

(RPAQQ TOKENFORMATS ((SHORTOP 128)
(LONGOP 160)
(SHORTNUMBER 0)
(SHORTSEQUENCE 192)
(LONGSEQUENCE 224)))

```

```

(RPAQQ IMAGERVARIABLES
(DCSCPX 0)
(DCSCPY 1)
(CORRECTMX 2)
(CORRECTMY 3)
(CURRENTTRANS 4)
(PRIORITYIMPORTANT 5)
(MEDIUMXSIZE 6)
(MEDIUMYSIZE 7)
(FIELDXMIN 8)
(FIELDYMIN 9)
(FIELDXMAX 10)
(FIELDYMAX 11)

```

(SHOWVEC 12)
(COLOR.IMVAR 13)
(NOIMAGE 14)
(STROKEWIDTH 15)
(STROKEEND 16)
(UNDERLINESTART 17)
(AMPLIFYSPACE 18)
(CORRECTPASS 19)
(CORRECTSHRINK 20)
(CORRECTTX 21)
(CORRECTTY 22))

(RPAQQ STROKEENDS ((SQUARE 0)
(BUTT 1)
(ROUND 2)))

(RPAQQ IP82CONSTANTS ((BEGINPREAMBLE {}
(ENDPREAMBLE })
(BEGINPAGE {}
(ENDPAGE })
(ENCODINGSTRING "Interpress/Xerox/1.0 ")
(NOVERSIONENCODINGSTRING "Interpress/Xerox/")
(MAXLONGSEQUENCEBYTES (SUB1 (EXPT 2 16)))
(FILETYPE.INTERPRESS 4361)))

(DEFMACRO \IPC (X) ; Edited 2-May-2023 08:33 by Imm
(DECLARE (SPECIAL X))
[OR (AND (BOUNDP '\IPCONSTANDS)
(LISTP \IPCONSTANTS))
(SETQ \IPCONSTANTS (FOR X IN IPCONSTANTS JOIN (FOR Y IN (EVAL X) COLLECT (CONS (CAR Y)
(CADR Y))
(FOR I FROM 1 TO 10 DO (IF (EQUAL X (SETQ X (SUBLIS \IPCONSTANTS X)))
THEN (RETURN (LIST 'CONSTANT X)))
FINALLY (ERROR "too many \IPC levels" X)))]

:: FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE
(RPAQQ MICASPERINCH 2540)
(RPAQQ MICASPERMILLIMETER 100)
(CONSTANTS (MICASPERINCH 2540)
(MICASPERMILLIMETER 100))
)

:: END EXPORTED DEFINITIONS

(DEFINEQ
(APPENDBYTE.IP (\* rmk%: "21-JUN-82 23:30")
[LAMBDA (STREAM BYTE)
(\BOUT STREAM BYTE)]

(APPENDIDENTIFIER.IP ; Edited 2-May-2023 08:52 by Imm
(\* jds "14-Mar-84 10:42")
[LAMBDA (STREAM STRING)
:: Put an identifier into the IP file. NB that the characters in the identifier are ASCII, NOT NS CHARACTERS!!!!
(APPENDSEQUENCEDESCRIPTOR.IP STREAM (\IPC SEQIDENTIFIER)
(NCHARS STRING))
(for C instring (MKSTRING STRING) do (\BOUT STREAM C])

(APPENDINT.IP (\* Imm " 2-May-85 21:13")
[LAMBDA (STREAM NUM LENGTH)
(for I from (SUB1 LENGTH) to 0 by -1 do (APPENDBYTE.IP STREAM (LOADBYTE NUM (UNFOLD I BITSPERBYTE)
BITSPERBYTE))

(APPENDINTEGER.IP ; Edited 2-May-2023 08:52 by Imm
; Edited 13-Jan-88 01:32 by FS
[LAMBDA (STREAM N)
(COND
((AND (ILEQ -4000 N)
(ILEQ N 28767))
(APPENDINT.IPSTREAM (IPLUS N 4000)
2))
(T (PROG ((LEN (BYTESININT.IP N)))
(APPENDSEQUENCEDESCRIPTOR.IP STREAM (\IPC SEQINTEGER)
LEN)
(APPENDINT.IP STREAM N LEN))

**(APPENDLARGEVECTOR.IP**

[LAMBDA (STREAM ARRAY)

; Edited 2-May-2023 08:53 by Imm  
(\* rmk%: "25-JUN-82 22:26")

;; Appends a large vector stored as an Interlisp array. NUMELEMENTS is not an argument, since we assume that the caller can pass a  
;; SUBARRAY if he so intends.

```
(PROG (INTSIZE (ASIZE (ARRAYSIZE ARRAY))
      (AORIG (ARRAYORIG ARRAY)))
  [SETQ INTSIZE (for I from AORIG to (SUB1 (IPLUS ASIZE AORIG)) largest (BYTESININT.IP (ELT ARRAY I]
  (APPENDSEQUENCEDESCRIPTOR.IP STREAM (\IPC SEQLARGEVECTOR)
    (ADD1 (ITIMES ASIZE INTSIZE)))
  (for I from AORIG to (SUB1 (IPLUS ASIZE AORIG)) do (APPENDINT.IP STREAM (ELT ARRAY I)
    INTSIZE])
```

**(APPENDNUMBER.IP**

[LAMBDA (STREAM R)

; Edited 2-May-2023 09:12 by Imm  
; Edited 13-Jan-88 01:22 by FS

```
(COND
  ((FIXP R)
   (APPENDINTEGER.IPMACRO STREAM R))
  (T (OR (TYPEP R 'RATIO)
        (SETQ R (CL:RATIONAL R)))
   (APPENDRATIONAL.IP STREAM (CL:NUMERATOR R)
    (CL:DENOMINATOR R])
```

**(APPENDOP.IP**

[LAMBDA (STREAM OP)

; Edited 2-May-2023 09:00 by Imm  
(\* rmk%: "22-JUN-82 01:28")

```
(COND
  ((OR (ILESSP OP 0)
       (IGREATERP OP 8191))
   (ERROR "Invalid Interpress operator code:" OP))
  (COND
    ((ILEQ OP 31)
     (APPENDBYTE.IP STREAM (LOGOR (\IPC SHORTOP)
                                   OP)))
    (T (APPENDBYTE.IP STREAM (LOGOR (\IPC LONGOP)
                                     (FOLDLO OP 256)))
     (APPENDBYTE.IP STREAM (MOD OP 256])
```

**(APPENDRATIONAL.IP**

[LAMBDA (STREAM N D)

; Edited 2-May-2023 08:54 by Imm  
(\* rmk%: "20-JUL-82 23:45")

```
(PROG [(I (IMAX (BYTESININT.IP N)
               (BYTESININT.IP D])
  (APPENDSEQUENCEDESCRIPTOR.IP STREAM (\IPC SEQRATIONAL)
    (UNFOLD I 2))
  (APPENDINT.IP STREAM N I)
  (APPENDINT.IP STREAM D I])
```

**(APPENDSEQUENCEDESCRIPTOR.IP**

[LAMBDA (STREAM TYPE LENGTH)

; Edited 2-May-2023 09:00 by Imm  
(\* edited%: "30-MAY-83 23:19")

```
(COND
  ((OR (ILESSP TYPE 0)
       (IGREATERP TYPE 31))
   (ERROR "Invalid Interpress type" TYPE))
  (COND
    ([OR (ILESSP LENGTH 0)
         (IGREATERP LENGTH (CONSTANT (SUB1 (EXPT 2 24]
         (ERROR "Interpress sequence length too long" LENGTH))
    (COND
      ((ILESSP LENGTH 256)
       (APPENDBYTE.IP STREAM (LOGOR (\IPC SHORTSEQUENCE)
                                     TYPE))
       (APPENDBYTE.IP STREAM LENGTH))
      (T (APPENDBYTE.IP STREAM (LOGOR (\IPC LONGSEQUENCE)
                                       TYPE))
        (APPENDINT.IP STREAM LENGTH 3])
```

; Short sequence, with one byte of length

; Long sequence, with 3 bytes of length

**(BYTESININT.IP**

[LAMBDA (N)

(\* rmk%: "20-OCT-82 17:28")

```
(FOLDHI (ADD1 (INTEGERLENGTH N)
            BITSPERBYTE])
```

)

;; Operator interface

(DEFINEQ

**(ARCTO.IP**

[LAMBDA (IPSTREAM X1 Y1 X2 Y2)

; Edited 2-May-2023 08:54 by Imm  
; Edited 1-Feb-89 15:42 by FS

:: Relative (like MOVETO) circular (in world coordinates) arc, passing through current x, y, and x1,y1 and x2,y2.

::

:: This operation may not be supported in most Xerox implementations of Interpress, I believe this is not part of Interpress2.1  
:: (INTERPRESSVERSION).

(APPENDNUMBER.IP IPSTREAM (COND  
((FLOATP X1)  
(FIXR X1))  
(T X1)))

(APPENDNUMBER.IP IPSTREAM (COND  
((FLOATP Y1)  
(FIXR Y1))  
(T Y1)))

(APPENDNUMBER.IP IPSTREAM (COND  
((FLOATP X2)  
(FIXR X2))  
(T X2)))

(APPENDNUMBER.IP IPSTREAM (COND  
((FLOATP Y2)  
(FIXR Y2))  
(T Y2)))

(APPENDOP.IP IPSTREAM (\IPC ARCTO])

**(BEGINMASTER.IP**

[LAMBDA (IPSTREAM)

; Edited 2-May-2023 08:44 by Imm  
(\* jds " 4-Dec-84 17:58")

(APPENDOP.IP IPSTREAM (\IPC BEGINMASTER])

**(BEGINPAGE.IP**

[LAMBDA (IPSTREAM)

; Edited 2-May-2023 09:01 by Imm  
(\* FS " 4-Mar-86 14:23")

(APPENDOP.IP IPSTREAM (\IPC BEGINPAGE))  
(replace IPPAGESTATE of (fetch IPDATA of IPSTREAM) with 'PAGE])

**(BEGINPREAMBLE.IP**

[LAMBDA (IPSTREAM)

; Edited 2-May-2023 09:01 by Imm  
(\* rmk%: "13-JUL-82 17:39")

(APPENDOP.IP IPSTREAM (\IPC BEGINPREAMBLE))  
(replace IPPAGESTATE of (fetch IPDATA of IPSTREAM) with 'PREAMBLE])

**(CLIPRECTANGLE.IP**

[LAMBDA (IPSTREAM X Y W H)

; Edited 2-May-2023 08:54 by Imm  
; Edited 1-Feb-89 16:39 by FS

:: Not supported in Interpress2.1

(APPENDNUMBER.IP IPSTREAM X)  
(APPENDNUMBER.IP IPSTREAM Y)  
(APPENDNUMBER.IP IPSTREAM W)  
(APPENDNUMBER.IP IPSTREAM H)  
(APPENDOP.IP IPSTREAM (\IPC CLIPRECTANGLE])

**(CONCAT.IP**

[LAMBDA (IPSTREAM)

; Edited 2-May-2023 08:54 by Imm  
(\* rmk%: " 7-JUN-83 17:41")

(APPENDOP.IP IPSTREAM (\IPC CONCAT])

**(CONCATT.IP**

[LAMBDA (IPSTREAM)

; Edited 2-May-2023 08:54 by Imm  
(\* rmk%: " 7-JUL-82 00:08")

(APPENDOP.IP IPSTREAM (\IPC CONCATT])

**(ENDMASTER.IP**

[LAMBDA (IPSTREAM)

; Edited 2-May-2023 08:45 by Imm  
(\* jds " 4-Dec-84 17:58")  
; Put out the token to end the master

(APPENDOP.IP IPSTREAM (\IPC ENDMASTER])

**(ENDPAGE.IP**

[LAMBDA (IPSTREAM)

; Edited 2-May-2023 09:01 by Imm  
(\* FS " 4-Mar-86 14:23")

(SHOW.IP IPSTREAM)  
(APPENDOP.IP IPSTREAM (\IPC ENDPAGE))  
(replace IPPAGESTATE of (fetch IPDATA of IPSTREAM) with NIL])

**(ENDPREAMBLE.IP**

[LAMBDA (IPSTREAM)

; Edited 2-May-2023 09:01 by Imm  
(\* FS " 4-Mar-86 14:24")

(PROG ((IPDATA (fetch IPDATA of IPSTREAM)))

(replace IPPREAMBLEFONTS of IPDATA with (DREVERSE (fetch IPPAGEFONTS of IPDATA)))

; Reverse on tenuous assumption that first fonts are more  
; frequent

(replace IPPREAMBLENEXTFRAMEVAR of IPDATA with (fetch IPNEXTFRAMEVAR of IPDATA))

(APPENDOP.IP IPSTREAM (\IPC ENDPREAMBLE))

(replace IPPAGESTATE of IPDATA with NIL])

**(FGET.IP**

[LAMBDA (IPSTREAM FINDEX)

; Edited 2-May-2023 08:56 by Imm  
(\* rmk%: " 7-JUL-82 00:09")

(APPENDNUMBER.IP IPSTREAM FINDEX)

(APPENDOP.IP IPSTREAM (\IPC FGET])

**(FILLRECTANGLE.IP**

[LAMBDA (IPSTREAM LEFT BOTTOM WIDTH HEIGHT)

; Edited 2-May-2023 07:54 by Imm  
; Edited 1-Feb-89 16:04 by FS

::: Append clipped rectangle description using current Interpress state

:: FS: This clipping code is wrong. You aren't guaranteed this functions args are device units (300dpi), so converting micras to device units is  
:: wrong. They happen to be so (from CIRC SHADE.IP & POLY SHADE.IP), but there may be other callers.

(LET\* ((IPDATA (fetch (STREAM IMAGEDATA) of IPSTREAM))

[SCALED-VISTOP (FIXR (TIMES (\IPC MicrasToDev)

(fetch (INTERPRESSDATA IPVISTOP) of IPDATA]

[SCALED-VISBOTTOM (FIXR (TIMES (\IPC MicrasToDev)

(fetch (INTERPRESSDATA IPVISBOTTOM) of IPDATA]

[SCALED-VISLEFT (FIXR (TIMES (\IPC MicrasToDev)

(fetch (INTERPRESSDATA IPVISLEFT) of IPDATA]

[SCALED-VISRIGHT (FIXR (TIMES (\IPC MicrasToDev)

(fetch (INTERPRESSDATA IPVISRIGHT) of IPDATA]

TOP RIGHT)

[if (> WIDTH 0)

then (SETQ RIGHT (IMIN SCALED-VISRIGHT (+ LEFT WIDTH)))

(SETQ LEFT (IMAX LEFT SCALED-VISLEFT))

else (SETQ RIGHT (IMIN LEFT SCALED-VISRIGHT))

(SETQ LEFT (IMAX SCALED-VISLEFT (+ WIDTH LEFT))

[if (> HEIGHT 0)

then (SETQ TOP (IMIN SCALED-VISTOP (+ BOTTOM HEIGHT)))

(SETQ BOTTOM (IMAX BOTTOM SCALED-VISBOTTOM))

else (SETQ TOP (IMIN BOTTOM SCALED-VISTOP))

(SETQ BOTTOM (IMAX SCALED-VISBOTTOM (+ HEIGHT BOTTOM))

(SETQ WIDTH (- RIGHT LEFT))

(SETQ HEIGHT (- TOP BOTTOM))

[if (AND (> WIDTH 0)

(> HEIGHT 0))

then (APPENDINTEGER.IP IPSTREAM LEFT)

(APPENDINTEGER.IP IPSTREAM BOTTOM)

(APPENDINTEGER.IP IPSTREAM WIDTH)

(APPENDINTEGER.IP IPSTREAM HEIGHT)

(APPENDOP.IP IPSTREAM (\IPC MASKRECTANGLE])

**(FILLTRAJECTORY.IP**

[LAMBDA (IPSTREAM POINTS)

; Edited 2-May-2023 08:57 by Imm  
; Edited 2-Feb-89 17:38 by FS

:: Fills a single trajectory. This is not a particularly useful or interesting function, you should be calling \FILLPOLYGON.IP instead.

(TRAJECTORY.IP IPSTREAM POINTS)

(APPENDINTEGER.IP IPSTREAM 1)

; number of trajectories

(APPENDOP.IP IPSTREAM (\IPC MAKEOUTLINE))

(APPENDOP.IP IPSTREAM (\IPC MASKFILL])

**(FILLNGON.IP**

[LAMBDA (IPSTREAM NPOINTS RADIUS CENTERX CENTERY TEXTURE OPERATION)

; Edited 2-May-2023 08:46 by Imm  
; Edited 1-Feb-89 17:19 by FS

:: Create and fill a regular polygon (standing on its tip). Since its convex, we can use the primitive IP operator to do the job. Note there is no  
:: clipping in this routine.

:: Could have used FILLTRAJECTORY.IP, but this function CONSeS less. Could have walked 1/8 of circle and used symmetry, but what the  
:: heck.....

(LET (BASEANGLE ANGLE X Y)

:: Try to avoid limitations of printers. Anything more than 64 or so looks for all intents and purposes like a circle anyway.

(if (IGREATERP NPOINTS MAXSEGSPERTRAJECTORY)

then (SETQ NPOINTS MAXSEGSPERTRAJECTORY))

(SETQ BASEANGLE (FQUOTIENT 360 NPOINTS))

(APPENDOP.IP IPSTREAM (\IPC DOSAVESIMPLEBODY))

; Save state (to undo SETCOLOR)



```
(APPENDOP.IP IPSTREAM (\IPC {}))
(SETCOLOR.IP IPSTREAM TEXTURE OPERATION)
(MOVETO.IP IPSTREAM CENTERX (IPLUS CENTERY RADIUS)) ; handle 0 point specially
;; Note that the trajectory is not closed, IP spec says outlines get closed anyway.
(for I from 1 to (SUB1 NPOINTS) do (SETQ ANGLE (TIMES I BASEANGLE))
; Since these are micas, we can avoid some floating point by
; forcing values to be integer
[SETQ X (IPLUS CENTERX (TIMES RADIUS (SIN ANGLE)
[SETQ Y (IPLUS CENTERY (TIMES RADIUS (COS ANGLE)
(LINETO.IP IPSTREAM X Y))
; number of trajectories
(APPENDINTEGER.IP IPSTREAM 1)
(APPENDOP.IP IPSTREAM (\IPC MAKEOUTLINE))
(APPENDOP.IP IPSTREAM (\IPC MASKFILL))
(APPENDOP.IP IPSTREAM (\IPC {})) ; restore state
NIL])
```

```
(FSET.IP
[LAMBDA (IPSTREAM FINDEX) ; Edited 2-May-2023 08:56 by Imm
(* rmk%: "7-JUL-82 00:08")
(APPENDNUMBER.IP IPSTREAM FINDEX)
(APPENDOP.IP IPSTREAM (\IPC FSET])
```

```
(GETFRAMEVAR.IP
[LAMBDA (IPSTREAM) (* rmk%: "18-AUG-83 17:50")
(PROG [(FV (fetch IPNEXTFRAMEVAR of (fetch IPDATA of IPSTREAM)
(replace IPNEXTFRAMEVAR of (fetch IPDATA of IPSTREAM) with (ADD1 FV))
(RETURN FV)])
```

```
(INITIALIZEMASTER.IP
[LAMBDA (IPSTREAM) ; Edited 2-May-2023 09:02 by Imm
(* jds "10-Jan-85 15:48")
[for I from 1 do (\BOUT IPSTREAM (OR (NTHCHARCODE (\IPC NOVERSIONENCODINGSTRING)
I)
(RETURN)
[for I from 1 do (\BOUT IPSTREAM (OR (NTHCHARCODE INTERPRESSVERSION I)
(RETURN)
(\BOUT IPSTREAM (CHARCODE SPACE])
```

```
(INITIALIZECOLOR.IP
[LAMBDA (IPSTREAM) ; Edited 2-May-2023 08:55 by Imm
(* hdj "23-Jan-86 19:20")
(LET ((COLORMODELOP.FVAR (GETFRAMEVAR.IP IPSTREAM))
(IPDATA (fetch (STREAM IMAGEDATA) of IPSTREAM)))
;; create data for the color model operator --- colors will range from 0 to 255
(APPENDINTEGER.IP IPSTREAM 255)
(APPENDINTEGER.IP IPSTREAM 1)
(APPENDOP.IP IPSTREAM (\IPC MAKEVEC))
;; name of color model
(APPENDIDENTIFIER.IP IPSTREAM "Xerox")
(APPENDIDENTIFIER.IP IPSTREAM "Research")
(APPENDIDENTIFIER.IP IPSTREAM "RGBLinear")
(APPENDINTEGER.IP IPSTREAM 3)
(APPENDOP.IP IPSTREAM (\IPC MAKEVEC))
;; create the color model
(APPENDOP.IP IPSTREAM (\IPC FINDCOLORMODELOPERATOR))
(APPENDOP.IP IPSTREAM (\IPC DO))
;; store it in the preamble's frame
(FSET.IP IPSTREAM COLORMODELOP.FVAR)
;; remember which fvar it is in
(replace (INTERPRESSDATA IPCOLORMODEL) of IPDATA with COLORMODELOP.FVAR])
```

```
(ISET.IP
[LAMBDA (IPSTREAM IVAR) ; Edited 2-May-2023 08:56 by Imm
(* rmk%: "18-Oct-84 12:52")
;; Sets the imager variable IVAR to the top of stack
(APPENDINTEGER.IP IPSTREAM IVAR)
(APPENDOP.IP IPSTREAM (\IPC ISET])
```

```
(GETTCP.IP
[LAMBDA (IPSTREAM) ; Edited 2-May-2023 08:56 by Imm
(* hdj "27-Nov-85 17:30")
```

;;; Pushes current X & Y onto stack

(APPENDOP.IP IPSTREAM (\IPC GETCP])

(LINETO.IP

[LAMBDA (IPSTREAM X Y)

; Edited 2-May-2023 08:56 by Imm  
(\* rmk%: "19-Oct-84 08:50")

(APPENDNUMBER.IP IPSTREAM (COND  
((FLOATP X)  
(FIXR X))  
(T X)))  
(APPENDNUMBER.IP IPSTREAM (COND  
((FLOATP Y)  
(FIXR Y))  
(T Y)))  
(APPENDOP.IP IPSTREAM (\IPC LINETO])

(MASKSTROKE.IP

[LAMBDA (IPSTREAM)

; Edited 2-May-2023 08:57 by Imm  
(\* rmk%: "14-Jun-84 16:00")

(APPENDOP.IP IPSTREAM (\IPC MASKSTROKE])

(MOVETO.IP

[LAMBDA (IPSTREAM X Y)

; Edited 2-May-2023 08:57 by Imm  
(\* hdj "18-Oct-85 15:58")

(APPENDNUMBER.IP IPSTREAM X)  
(APPENDNUMBER.IP IPSTREAM Y)  
(APPENDOP.IP IPSTREAM (\IPC MOVETO])

(ROTATE.IP

[LAMBDA (IPSTREAM S)

; Edited 2-May-2023 08:57 by Imm  
(\* rmk%: " 6-JUN-83 18:02")

(APPENDNUMBER.IP IPSTREAM S)  
(APPENDOP.IP IPSTREAM (\IPC ROTATE])

(SCALE.IP

[LAMBDA (IPSTREAM S)

; Edited 2-May-2023 08:57 by Imm  
(\* rmk%: "15-Jun-84 12:21")

(APPENDNUMBER.IP IPSTREAM S)  
(APPENDOP.IP IPSTREAM (\IPC SCALE.OP])

(SCALE2.IP

[LAMBDA (IPSTREAM X Y)

; Edited 2-May-2023 08:57 by Imm  
(\* Imm "10-JUN-83 15:28")

(APPENDNUMBER.IP IPSTREAM X)  
(APPENDNUMBER.IP IPSTREAM Y)  
(APPENDOP.IP IPSTREAM (\IPC SCALE2])

(SETCOLOR.IP

[LAMBDA (IPSTREAM SHADE OPERATION SCALE ANGLE)

; Edited 2-May-2023 08:58 by Imm  
; Edited 21-Sep-88 14:41 by jds

(if (AND (STREAMPROP IPSTREAM 'COLOR)  
(LISTP SHADE)  
(RGBP (CADR SHADE)))  
then  
(SETRGB.IP IPSTREAM (CAADR SHADE)  
(CADR (CADR SHADE))  
(CADDR (CADR SHADE)))  
(SETQ SHADE (CAR SHADE)))  
(if (LITATOM SHADE)  
then  
;; Not sure what to do in LITATOM case  
(SETQ SHADE BLACKSHADE))

; the dosavesimplebody is in POLYSHADE.IP. For now, insist  
; that the CDR be RGB if color is desired

[COND  
((NOT OPERATION)  
  
(SETQ OPERATION (DSOPERATION NIL IPSTREAM)

; OPERATION got defaulted to whatever the stream's op is, but  
; we need to know here.

;; FS: Below this point, integers are considered TEXTURES, not COLORS.

(if [AND (OR (EQ SHADE BLACKSHADE)  
(EQ (NEGSHADE SHADE)  
BLACKSHADE))  
(OR (EQ OPERATION 'REPLACE)  
(EQ OPERATION 'PAINT])  
then  
;; Most common case, optimized  
(APPENDINTEGER.IP IPSTREAM 1)  
(APPENDOP.IP IPSTREAM (\IPC SETGRAY))  
elseif [AND (OR (EQ SHADE WHITESHADE)  
(EQ (NEGSHADE SHADE)  
WHITESHADE))

```

      (OR (EQ OPERATION 'REPLACE)
          (EQ OPERATION 'PAINT])
  then ;; Probably rare, but optimize anyway
      (APPENDINTEGER.IP IPSTREAM 0)
      (APPENDOP.IP IPSTREAM (\IPC SETGRAY))
  else ;; Patch around Print Service 8.0 bugs
      (if (EQUAL PRINTSERVICE 8.0)
          then (SETCOLOR16.IP IPSTREAM SHADE OPERATION SCALE ANGLE)
          else (SETCOLORLV.IP IPSTREAM SHADE OPERATION SCALE ANGLE))

```

(SETRGB.IP

```

[LAMBDA (IPSTREAM RED GREEN BLUE)
  (LET [(COLORMODEL.FVAR (fetch IPCOLORMODEL of (fetch IMAGEDATA of IPSTREAM]
      ;; force out any stored chars so they get colored
      (SHOW.IP IPSTREAM)
      ;; push RED GREEN BLUE vector
      (APPENDINTEGER.IP IPSTREAM RED)
      (APPENDINTEGER.IP IPSTREAM GREEN)
      (APPENDINTEGER.IP IPSTREAM BLUE)
      (APPENDINTEGER.IP IPSTREAM 3)
      (APPENDOP.IP IPSTREAM (\IPC MAKEVEC))
      ;; apply the color operator
      (FGET.IP IPSTREAM COLORMODEL.FVAR)
      (APPENDOP.IP IPSTREAM (\IPC DO))
      ;; set current color to result
      (ISET.IP IPSTREAM (\IPC COLOR.IMVAR)))
  NIL])

```

; Edited 2-May-2023 08:56 by Imm  
 (\* hdj "3-Feb-86 12:00")  
 (\* hdj "23-Jan-86 19:21")

(SETCOLORLV.IP

```

[LAMBDA (IPSTREAM SHADE OPERATION SCALE ANGLE)
  ;; OSD's Print Service 9.0 supports large vector arrays for MAKESAMPLEDBLACK, with power-of-2 scale factors up to eight, Also note that bitmap
  ;; gets rotated -90 degrees, Non-power-of-two values are rounded.
  ;; Note that OSD's Print Service 9.0 has an INCOMPATIBLE change to MAKESAMPLEDBLACK.
  ;; I changed this to set SCALE and ANGLE from texture if they are not given. The 8044 only allows 4x4 textures at the same scale at the screen.
  ;; A 4x4 will get a scale of 4 so that it looks like it does on the screen. A 16x16 will get a scale of 1 so that all of it appears albeit at 1/4 the size. rrb
  ;; 7-mar-86
  ;; FS- Note this is a general method; Common optimizations probably should be performed outside of here (e.g. SETCOLOR.IP)
  (PROG (SCRATCHBM (DIM 16))
    (COND
      ((EQ OPERATION 'ERASE) ; for now, simulate ERASE by painting white
        (SETQ SCRATCHBM (BITMAPCREATE DIM DIM))
        (SETQ OPERATION 'REPLACE))
      ((AND (BITMAPP SHADE)
            (EQ (BITMAPWIDTH SHADE) 16)
            (EQ (BITMAPHEIGHT SHADE) 16)) ; 16x16 texture case.
        (SETQ SCRATCHBM SHADE))
      (T ; all other textures
        [COND
          ((NOT (NUMBERP SCALE))
            (COND
              ((NUMBERP SHADE)
                ;; make numbered textures be at screen scale and bitmap textures be at closer to printer scale. This at least allows
                ;; ways of users getting different effects.
                (SETQ SCALE 4) ; Move the shade into the scratch bitmap, that's dim wide, so we
                ; can tell Interpress about it
              (SETQ SCRATCHBM (BITMAPCREATE DIM DIM))
              (BITBLT NIL 0 0 SCRATCHBM 0 0 DIM DIM 'TEXTURE 'REPLACE SHADE)))
          (APPENDNUMBER.IP IPSTREAM DIM) ; X Pixels
          (APPENDNUMBER.IP IPSTREAM DIM) ; Y Pixels
          (APPENDINTEGER.IP IPSTREAM 1) ; Samples per pixel
          (APPENDINTEGER.IP IPSTREAM 1) ; Max Sample Value
          (APPENDINTEGER.IP IPSTREAM 1) ; 'Interleaved' samples
          (SCALE.IP IPSTREAM 1) ; Transform datum to pixel array
          (APPENDSEQUENCEDESCRIPTOR.IP IPSTREAM (\IPC SEQLARGEVECTOR)
            (IPLUS 1 (ITIMES DIM DIM))) ; Header for Vector type
          (APPENDBYTE.IP IPSTREAM 1) ; bytes / sample
          ; samples / scanline
        ]))
  )

```

```

;; Now put put the bitmap -- each line must be a 32-bit multiple long
[for Y from (SUB1 DIM) to 0 by -1 do (for X from 0 to (SUB1 DIM)
  do (\BOUT IPSTREAM (BITMAPBIT SCRATCHBM X Y)

```

```

(APPENDOP.IP IPSTREAM (\IPC MAKEPIXELARRAY))
(SCALE.IP IPSTREAM (OR (NUMBERP SCALE)
1))
(ROTATE.IP IPSTREAM (OR (NUMBERP ANGLE)
-90))
(CONCAT.IP IPSTREAM)
(APPENDINTEGER.IP IPSTREAM (SELECTQ OPERATION
(REPLACE 0)
(PAINT 1)
1))
(APPENDOP.IP IPSTREAM (\IPC MAKESAMPLEDBLACK))
(ISET.IP IPSTREAM (\IPC COLOR.IMVAR))
(RETURN NIL)

```

; put out the bits  
; make the pixel array  
; the 8044 scans bitmaps from top to bottom rather than left to  
; right so rotate it.  
; 0 is white bits opaque, 1 is white bits clear

(SETCOLOR16.IP

[LAMBDA (IPSTREAM SHADE OPERATION SCALE ANGLE)

; Edited 2-May-2023 08:54 by Imm  
(\* FS " 2-Aug-85 00:54")

;;; OSD's Print Service 8.0 only supports 16x16 pixel arrays for MAKESAMPLEDBLACK, with power-of-2 scale factors up to eight, Also note that bitmap  
;;; gets rotated -90 degrees, Non-power-of-two values are rounded, PSD's interpress is allegedly more restrictive

;;; Note this version is correct for PS 8.0, by implementing the incorrect PS 8.0 method. Won't work for later versions

```

(PROG (SCRATCHBM BMBASE NBYTES (DIM 16))
(COND
((NOT (NUMBERP SCALE))
(SETQ SCALE 1)))
(COND
((NOT (NUMBERP ANGLE))
(SETQ ANGLE 0)))
(SETQ NBYTES (IQUOTIENT (ITIMES DIM DIM)
8))
(SETQ SCRATCHBM (BITMAPCREATE DIM DIM))
(SETQ BMBASE (fetch (BITMAP BITMAPBASE) of SCRATCHBM))
(BITBLT NIL 0 0 SCRATCHBM 0 0 DIM DIM 'TEXTURE 'REPLACE SHADE)
(APPENDNUMBER.IP IPSTREAM DIM)
(APPENDNUMBER.IP IPSTREAM DIM)
(APPENDINTEGER.IP IPSTREAM 1)
(APPENDINTEGER.IP IPSTREAM 1)
(APPENDINTEGER.IP IPSTREAM 1)
(SCALE.IP IPSTREAM 1)
(APPENDSEQUENCEDESCRIPTOR.IP IPSTREAM (\IPC SEQPACKEDPIXELVECTOR)
(IPLUS 4 NBYTES))
(APPENDINT.IP IPSTREAM 1 2)
(APPENDINT.IP IPSTREAM DIM 2)

```

; Move the shade into the scratch bitmap, that's dim wide, so we  
; can tell Interpress about it  
; X Pixels  
; Y Pixels  
; Samples per pixel  
; Max Sample Value  
; 'Interleaved' samples  
; Transform datum to pixel array  
; Header for Vector type  
; bits / sample  
; samples / scanline

:: Now put put the bitmap -- each line must be a 32-bit multiple long

```

(\BOUITS IPSTREAM BMBASE 0 NBYTES)
(APPENDOP.IP IPSTREAM (\IPC MAKEPIXELARRAY))
(SCALE.IP IPSTREAM SCALE)
(ROTATE.IP IPSTREAM ANGLE)
(CONCAT.IP IPSTREAM)
(APPENDINTEGER.IP IPSTREAM (SELECTQ OPERATION
(REPLACE 0)
(PAINT 1)
1))
(APPENDOP.IP IPSTREAM (\IPC MAKESAMPLEDBLACK))
(ISET.IP IPSTREAM (\IPC COLOR.IMVAR))
(RETURN NIL)

```

; put out the bits  
; make the pixel array  
; 0 is white bits opaque, 1 is white bits clear

(SETFONT.IP

[LAMBDA (IPSTREAM FONTNUM)

; Edited 2-May-2023 08:57 by Imm  
(\* rmk%: "20-AUG-83 14:03")

```

(APPENDNUMBER.IP IPSTREAM FONTNUM)
(APPENDOP.IP IPSTREAM (\IPC SETFONT))
(PROG ((IPDATA (fetch IPDATA of IPSTREAM))
(replace IPFONT of IPDATA with (for X in (fetch IPPAGEFONTS of IPDATA) when (EQ FONTNUM (CDR X))
do (RETURN (CAR X)) finally (ERROR "Undefined font number"))

```

(SETSPACE.IP

[LAMBDA (IPSTREAM SPACEWIDTH)

; Edited 1-May-2023 19:38 by Imm  
(\* rmk%: "11-Dec-83 21:12")

```

(APPENDNUMBER.IP IPSTREAM SPACEWIDTH)
(APPENDOP.IP IPSTREAM (\IPC SPACE])

```

(SETXREL.IP

[LAMBDA (IPSTREAM DX)

; Edited 2-May-2023 08:58 by Imm  
; Edited 11-Aug-88 15:24 by rmk:  
; Move by DX in the X direction



(>= Y (fetch IPMINVISIBLEBASELINE of IPDATA))
(<= Y (fetch IPMAXVISIBLEBASELINE of IPDATA])

(replace IPYPOS of IPDATA with Y])

(SETYREL.IP

[LAMBDA (IPSTREAM DY)

; Edited 2-May-2023 08:58 by Imm
; Edited 11-Aug-88 15:26 by rmk:

(LET ((IPDATA (fetch IPDATA of IPSTREAM))
(APPENDNUMBER.IP IPSTREAM DY)
(APPENDOP.IP IPSTREAM (\IPC SETYREL))
(SETQ DY (change (fetch IPYPOS of IPDATA)
(+ DY DATUM))))

(replace IPCHARVISIBLEP of IPDATA with (AND (>= (fetch IPXPOS of IPDATA)
(fetch IPVISLEFT of IPDATA))
(>= DY (fetch IPMINVISIBLEBASELINE of IPDATA))
(<= DY (fetch IPMAXVISIBLEBASELINE of IPDATA])))

(SHOW.IP

[LAMBDA (IPSTREAM MOVING?)

; Edited 2-May-2023 08:47 by Imm
; Edited 9-Dec-87 19:02 by jds

:: Shows a string buffered away in SHOWSTREAM
:: If MOVING? is true, we're going to be doing a positioning operation, so there's no point to correcting single characters.

(PROG ((IPDATA (ffetch IPDATA of IPSTREAM))
LEN SHOWSTREAM)
(SETQ SHOWSTREAM (ffetch IPSHOWSTREAM of IPDATA))
(SETQ LEN (\GETFILEPTR SHOWSTREAM))
(COND

((IGREATERP LEN 0) ; Only bother if there ARE characters to put out.
(COND

((OR (IGREATERP LEN 1) ; Let's assume that a single character won't get too far off.
(NOT MOVING?))

(APPENDNUMBER.IP IPSTREAM (- (ffetch IPXPOS of IPDATA) ; Set up the measures for the CORRECT op, so the characters
(fetch IPCORRECTSTARTX of IPDATA))) ; come out the right width

(APPENDINTEGER.IP IPSTREAM 0)
(APPENDOP.IP IPSTREAM (\IPC SETCORRECTMEASURE))
(APPENDOP.IP IPSTREAM (\IPC CORRECT)) ; Put the SHOW inside a block, so the CORRECT will affect it.
(APPENDOP.IP IPSTREAM (\IPC {}))

)
(APPENDSEQUENCEDESRIPTOR.IP IPSTREAM (\IPC SEQSTRING)
LEN)

(COPYBYTES SHOWSTREAM IPSTREAM 0 LEN)
(APPENDOP.IP IPSTREAM (\IPC SHOW))
(COND

((OR (IGREATERP LEN 1) ; Let's assume that a single character won't get too far off.
(NOT MOVING?)) ; End of the block affected by the CORRECT

(APPENDOP.IP IPSTREAM (\IPC {}))

)
(\SETFILEPTR SHOWSTREAM 0) ; Clear out the holding stream for characters
(COND

((NOT (IEQP (fetch NSCHARSET of IPDATA) ; If we're not in charset zero, change back to it.
0)))
(\CHANGECHARSET.IP IPDATA 0)))

(replace IPCORRECTSTARTX of IPDATA with (ffetch IPXPOS of IPDATA)) ; And notice our new real location for future CORRECTS.

)

(TRAJECTORY.IP

[LAMBDA (IPSTREAM POINTS)

(\* FS "19-Jul-85 11:53")

(MOVETO.IP IPSTREAM (fetch XCOORD of (CAR POINTS))
(fetch YCOORD of (CAR POINTS)))
(for P in (CDR POINTS) do (LINETO.IP IPSTREAM (fetch XCOORD of P)
(fetch YCOORD of P)))

(TRANS.IP

[LAMBDA (IPSTREAM)

; Edited 1-May-2023 19:36 by Imm
(\* rmk%: "27-Mar-85 14:24")

:: This translates the origin to the current position.

(APPENDOP.IP IPSTREAM (\IPC TRANS.IPOP])

(TRANSLATE.IP

[LAMBDA (IPSTREAM X Y)

; Edited 1-May-2023 19:30 by Imm
(\* rmk%: "21-JUL-82 13:23")

(APPENDNUMBER.IP IPSTREAM X)
(APPENDNUMBER.IP IPSTREAM Y)
(APPENDOP.IP IPSTREAM (\IPC TRANSLATE])

)

:: DIG interface

(DEFINEQ

(\CHANGE-VISIBLE-REGION.IP

[LAMBDA (IPDATA VISIBLE-REGION)

; Edited 18-Aug-88 16:17 by hdj

:: Unpacks parameters of the visible region

```
(LET ((FONT (ffetch IPFONT of IPDATA)))
  (freplace (INTERPRESSDATA IPVISLEFT) of IPDATA with (ffetch (REGION LEFT) of VISIBLE-REGION))
  (freplace (INTERPRESSDATA IPVISRIGHT) of IPDATA with (ffetch (REGION RIGHT) of VISIBLE-REGION))
  (freplace (INTERPRESSDATA IPVISTOP) of IPDATA with (ffetch (REGION TOP) of VISIBLE-REGION))
  (freplace (INTERPRESSDATA IPVISBOTTOM) of IPDATA with (ffetch (REGION BOTTOM) of VISIBLE-REGION))
  (freplace (INTERPRESSDATA IPVISBLEREGION) of IPDATA with VISIBLE-REGION)
  (freplace IPMAXVISIBLEBASELINE of IPDATA with (- (ffetch IPVISTOP of IPDATA)
                                                    (ffetch (FONTDESCRIPTOR \SFAscent) of FONT)))
```

[if (ffetch IPCLIPINCLUSIVE of IPDATA) then ; include characters that cross the bottom of the clipping region

```
[freplace IPMINVISIBLEBASELINE of IPDATA with (ADD1 (- (ffetch IPVISBOTTOM of IPDATA)
                                                         (ffetch (FONTDESCRIPTOR \SFAscent)
                                                         of FONT))
  else (freplace IPMINVISIBLEBASELINE of IPDATA with (+ (ffetch IPVISBOTTOM of IPDATA)
                                                         (ffetch (FONTDESCRIPTOR \SFDescent) of FONT))
```

```
[replace IPCHARVISIBLEP of IPDATA with (AND (>= (ffetch IPXPOS of IPDATA)
                                                (ffetch IPVISLEFT of IPDATA))
      (>= (ffetch IPYPOS of IPDATA)
         (ffetch IPMINVISIBLEBASELINE of IPDATA))
      (<= (ffetch IPYPOS of IPDATA)
         (ffetch IPMAXVISIBLEBASELINE of IPDATA))
      (ffetch IPVISRIGHT of IPDATA))
  (freplace IPMINCHARRIGHT of IPDATA with (MIN (ffetch IPVISRIGHT of IPDATA)
                                               (ffetch IPRIGHT of IPDATA))
```

(\PAPERSIZE.IP

[LAMBDA (IPSTREAM MEDIUM) (OR MEDIUM (SETQ MEDIUM DEFAULTINTERPRESSMEDIUM)) (LET [(PSIZE (COND

; Edited 15-Aug-88 09:28 by rmk:

```
((AND (EQ (CAR MEDIUM) 'PAPER)
      (SELECTQ (CAR (SETQ MEDIUM (CADR MEDIUM)))
               (KNOWN.SIZE (CADR (CL:ASSOC (CADR MEDIUM) KNOWN.MEDIA.SIZES :TEST 'STRING-EQUAL)))
               (OTHER.SIZE (CADR MEDIUM) NIL)))
  (T (ERROR "UNRECOGNIZED PRINTING MEDIUM"] ; Scale millimeters to micas
```

```
(LIST (TIMES MICASPERMILLIMETER (CAR PSIZE))
      (TIMES MICASPERMILLIMETER (CADR PSIZE]))
```

(\HEADINGOP.IP

[LAMBDA (IPSTREAM HEADING)

; Edited 2-May-2023 08:48 by Imm (\* hdj "18-Oct-85 15:46")

:: Stores the HEADINGOP operator as frame-variable 0 in the preamble.

```
(PROG ((IPDATA (ffetch IPDATA of IPSTREAM)))
  (APPENDOP.IP IPSTREAM (\IPC MAKESIMPLECO))
  (APPENDOP.IP IPSTREAM (\IPC {}))
  (COND
    (HEADING [SETXY.IP IPSTREAM (ffetch IPLEFT of IPDATA)
              (DIFFERENCE (ffetch IPTOP of IPDATA)
                          (FONTPROP (ffetch IPHEADINGFONT of IPDATA)
                                     'ASCENT])
              (SETFONT.IP IPSTREAM HEADINGFONTNUMBER)
              (PRIN3 HEADING IPSTREAM)
              (SHOW.IP IPSTREAM)
              (RELMOVETO MICASPERINCH 0 IPSTREAM) ; Skip an inch before page number
              (PRIN3 "Page " IPSTREAM) ; Show the page number argument (from stack)
              (TERPRI IPSTREAM) ; Skip 2 lines--have to pick up the linefeed from the heading font
              (TERPRI IPSTREAM)))
    (APPENDOP.IP IPSTREAM (\IPC {}))
    (FSET.IP IPSTREAM (replace IPHEADINGOPVAR of IPDATA with (GETFRAMEVAR.IP IPSTREAM)]
```

)

(DEFINEQ

(\DEFINEFONT.IP

[LAMBDA (IPSTREAM FONT)

; Edited 2-May-2023 07:57 by Imm (\* bvm%: "22-Oct-86 13:20")

```
(LET ((IPDATA (ffetch IPDATA of IPSTREAM))
      FRAMEVAR)
  (for N from 0 as ID in (FONTNAME.IP FONT) do (APPENDIDENTIFIER.IP IPSTREAM ID)
  finally (APPENDINTEGER.IP IPSTREAM N)
```

```

      (APPENDOP.IP IPSTREAM (\IPC MAKEVEC))
    (APPENDOP.IP IPSTREAM (\IPC FINDFONT))
  [SCALE.IP IPSTREAM (TIMES (\IPC MICASPERPOINT)
    (FONTPROP FONT 'DEVICESTRIDE))
  (APPENDOP.IP IPSTREAM (\IPC MODIFYFONT))
  (SETQ FRAMEVAR (GETFRAMEVAR.IP IPSTREAM))
  (FSET.IP IPSTREAM FRAMEVAR)
  (CAR (push (fetch IPPAGEFONTS of IPDATA)
    (CONS FONT FRAMEVAR]))

```

(FONTNAME.IP

```

[LAMBDA (FONTDESC)
  ;; Convert a Lisp font name to the proper NS font name
  (DECLARE (GLOBALVARS INTERPRESSPRINTWHEELFAMILIES INTERPRESSFAMILYALIASES))
  (PROG (FACE NAME)
    [COND
      ((EQ 'ITALIC (FONTPROP FONTDESC 'DEVICESLOPE))
        (SETQ FACE '(-Italic)
      [COND
        ((EQ 'BOLD (FONTPROP FONTDESC 'DEVICESTRIDE))
          (push FACE '(-Bold)
        (SETQ NAME (FONTPROP FONTDESC 'DEVICEFAMILY))
        [AND (MEMB NAME INTERPRESSPRINTWHEELFAMILIES)
          (SETQ NAME (PACK* NAME '-PRINTWHEEL)
        [COND
          ((MEMB NAME INTERPRESSFAMILYALIASES)
            (SETQ NAME (LISTGET INTERPRESSFAMILYALIASES NAME)
          [COND
            (FACE (SETQ NAME (PACK (CONS NAME FACE)
            (RETURN (LIST 'XEROX CHARACTERCODEVERSION NAME))

```

(\* jds "17-Jul-85 11:00")

(INTERPRESS.BITMAPSCALE

```

[LAMBDA (WIDTH HEIGHT)
  (PROG [(RATIO (MIN (FQUOTIENT (\IPC (TIMES (\IPC POINTSPERINCH)
    WIDTH)
    (FQUOTIENT (\IPC (TIMES (\IPC POINTSPERINCH)
    HEIGHT)
  (RETURN (COND
    ((GEQ RATIO 1)
      1)
    ((GEQ RATIO 0.5)
      0.5)
    ((GEQ RATIO 0.25)
      0.25)
    (T RATIO))

```

; Edited 2-May-2023 08:37 by Imm (\* Imm " 3-OCT-83 21:31")

(INTERPRESS.OUTCHARFN

```

[LAMBDA (IPSTREAM CHARCODE)
  ;; The \OUTCHAR method for interpress streams. Print a character, taking account of margins and visible region, and things like ^L.
  (LET* ((IPDATA (ffetch IPDATA of IPSTREAM))
    [NSCODE (COND
      ((\FATCHARCODEP CHARCODE)
        CHARCODE)
      (T (\GETBASE (ffetch NSTRANSTABLE of IPDATA)
        CHARCODE)
    (OLD-CSET (ffetch NSCHARSET of IPDATA)))
  [COND
    ((NEQ (\CHARSET NSCODE)
      OLD-CSET)
    ;; Switch character set so that we get the right char width, but DON'T write out the charset-shift sequence, in case the character gets
    ;; clipped.
    (\CHANGECHARSET.IP IPDATA (\CHARSET NSCODE])
  ;; Select on NSCODE, since ^L etc might be graphic in some ascii fonts:
  (SELCHARQ NSCODE
    (EOL (NEWLINE.IP IPSTREAM))
    (LF (\DSPXPOSITION.IP IPSTREAM (PROG1 (\DSPXPOSITION.IP IPSTREAM)
      (NEWLINE.IP IPSTREAM))))
    (^L (DSPNEWPAGE IPSTREAM))
    (PROG (CHAR-WIDTH NEWXPOS)
      [SETQ CHAR-WIDTH (COND
        ((EQ NSCODE (CHARCODE SPACE))
          (ffetch IPSPACEWIDTH of IPDATA))
        (T (\FGETWIDTH (ffetch IPWIDTHSCACHE of IPDATA)
          (\CHAR8CODE NSCODE)
        (SETQ NEWXPOS (+ (ffetch IPXPOS of IPDATA)
          CHAR-WIDTH))

```

; Edited 6-Jan-89 23:03 by jds

; Have to switch charset before fetching width from cache, even though we might later clip



```

RETRY
; Return to here if we have to emit a newline before printing
(COND
  ((AND (fetch IPCHARVISIBLEP of IPDATA)
        (<= NEWXPOS (fetch IPMINCHARRIGHT of IPDATA)))
   ;; Char vis means starting pos is inside the character clipping region. Minright is the min of the right margin and
   ;; clipping right, so we're OK if we end up left of that ; This is the common case we've optimized for: char starts and
   ;; ends visible and before right margin
   (freplace IPXPOS of IPDATA with NEWXPOS)
   [COND
     ((NEQ (\CHARSET NSCODE)
            OLD-CSET)
      (\BOUT (ffetch IPSHOWSTREAM of IPDATA)
              NSCHARSETSHIFT) ; Switch character set
      (\BOUT (ffetch IPSHOWSTREAM of IPDATA)
              (\CHARSET NSCODE)))
     ;; have to repeat this, since we may have done a CR before printing it.
     (\CHANGECHARSET.IP IPDATA (\CHARSET NSCODE])
     (\BOUT (ffetch IPSHOWSTREAM of IPDATA)
            (\CHAR8CODE NSCODE))
     (RETURN))
    (> NEWXPOS (ffetch IPRIGHT of IPDATA))
    ;; Failed visible or micharright, if over right margin, do newline and try again, otherwise clip
    (NEWLINE.IP IPSTREAM) ; This will reset the IPCHARVISIBLEP
    (SETQ NEWXPOS (+ (ffetch IPXPOS of IPDATA)
                    CHAR-WIDTH)) ; Retry to print if we ended up unclipped and within the margin,
    ; otherwise fall thru to clip
    (AND (<= NEWXPOS (fetch IPMINCHARRIGHT of IPDATA))
         (GO RETRY)))
    (AND (ffetch IPCLIPINCLUSIVE of IPDATA)
         (< (ffetch IPXPOS of IPDATA)
            (ffetch IPVISRIGHT of IPDATA))
         (>= NEWXPOS (ffetch IPVISRIGHT of IPDATA)))
    ;; We're clipping him, but he wants the straddling character left visible. Print it.
    (freplace IPXPOS of IPDATA with NEWXPOS)
    [COND
      ((NEQ (\CHARSET NSCODE)
            (ffetch NSCHARSET of IPDATA))
       (\BOUT (ffetch IPSHOWSTREAM of IPDATA)
               NSCHARSETSHIFT) ; Switch character set
       (\BOUT (ffetch IPSHOWSTREAM of IPDATA)
               (\CHARSET NSCODE)))
      ;; have to repeat this, since we may have done a CR before printing it.
      (\CHANGECHARSET.IP IPDATA (\CHARSET NSCODE])
      (\BOUT (ffetch IPSHOWSTREAM of IPDATA)
              (\CHAR8CODE NSCODE))
      (RETURN))
      (T ;; Nothing printed; have to reset the charset.
       (\CHANGECHARSET.IP IPDATA OLD-CSET)))
    (SHOW.IP IPSTREAM T) ; Either failed CHARVIS, or failed both VISRIGHT and IPRIGHT,
    ; so not in clipping region. Just move X position
    (SETX.IP IPSTREAM NEWXPOS])

```

**(INTERPRESSFILEP**

[LAMBDA (FILE NOOPEN)

; Edited 2-May-2023 09:09 by Imm  
 (\* jds "18-Feb-85 09:41")

```

;; Returns fullname of FILE if it looks like an Interpress file
(OR (EQ (GETFILEINFO FILE 'FILETYPE)
      (\IPC FILETYPE.INTERPRESS))
    (RESETLST
     [PROG (STRM)
      [COND
        ((SETQ STRM (\GETSTREAM FILE 'INPUT T))
         (OR (RANDACCESSP STRM)
              (RETURN)))
         (RESETSAVE NIL (LIST 'SETFILEPTR STRM (GETFILEPTR STRM)))
         (SETFILEPTR STRM 0))
        (NOOPEN (RETURN))
        (T (RESETSAVE (SETQ STRM (OPENSTREAM FILE 'INPUT 'OLD 8))
                     ' (PROGN (CLOSEP? OLDVALUE)
                               (RETURN (for I from 1 to (\IPC (NCHARS NOVERSIONENCODINGSTRING))
                                       when (OR (EOFP STRM)
                                                (NEQ (NTHCHARCODE (\IPC NOVERSIONENCODINGSTRING)
                                                                I)
                                                    (BIN STRM))))
                                       do (RETURN NIL) finally (RETURN (FULLNAME STRM))))))

```

**(MAKEINTERPRESS**

[LAMBDA (FILE IPFILE FONTS HEADING TABS OPTIONS)

(\* jds "9-May-85 16:28")

(TEXTTOIMAGEFILE FILE IPFILE 'INTERPRESS FONTS HEADING TABS OPTIONS])

(NEWLINE.IP

[LAMBDA (IPSTREAM)

(\* jds " 9-Feb-86 17:37")

; Doesn't check for page overflow--wait until something is  
; actually shown.

(SHOW.IP IPSTREAM)

(PROG (NEWYPOS (IPDATA (ffetch IPDATA of IPSTREAM)))  
(SETQ NEWYPOS (PLUS (ffetch IPYPOS of IPDATA)  
(ffetch IPLINEFEED of IPDATA))))

(COND

((LESSP NEWYPOS (fetch IPBOTTOM of IPDATA))  
(DSPNEWPAGE IPSTREAM))  
(T (SETXY.IP IPSTREAM (fetch IPLEFT of IPDATA)  
NEWYPOS]))

(NEWPAGE.IP

[LAMBDA (IPSTREAM)

; Edited 2-May-2023 08:34 by Imm  
; Edited 25-Nov-87 18:20 by jds

:: Start a new page in an interpress stream

(PROG (CFONT HFONT ROTATION XOFFSET YOFFSET (IPDATA (fetch IPDATA of IPSTREAM)))  
(SETQ CFONT (fetch IPFONT of IPDATA)))

:: Save current font and make IPFONT be NIL, indicating that there is no actual font at the beginning of a page

(replace IPFONT of IPDATA with NIL)  
(SELECTQ (fetch IPPAGESTATE of IPDATA)  
(PAGE (ENDPAGE.IP IPSTREAM))  
(PREAMBLE (ENDPREAMBLE.IP IPSTREAM))  
NIL)

(BEGINPAGE.IP IPSTREAM)

(replace IPPAGEFONTS of IPDATA with (fetch IPPREAMBLEFONTS of IPDATA))  
(replace IPNEXTFRAMEVAR of IPDATA with (fetch IPPREAMBLENEXTFRAMEVAR of IPDATA))  
(SCALE.IP IPSTREAM (\IPC METERSPERMICA)) ; Establish mica page coordinate system  
(CONCATT.IP IPSTREAM)

(COND

([NOT (ZEROP (SETQ ROTATION (fetch IPROTATION of IPDATA)  
(ROTATE.IP IPSTREAM ROTATION)  
(CONCATT.IP IPSTREAM)))] ; Take care of any rotation

(COND

([OR [NOT (ZEROP (SETQ XOFFSET (fetch IPXOFFSET of IPDATA)  
(NOT (ZEROP (SETQ YOFFSET (fetch IPYOFFSET of IPDATA)  
(TRANSLATE.IP IPSTREAM XOFFSET YOFFSET)  
(CONCATT.IP IPSTREAM)))] ; Take care of any translations

[COND

([fetch IPHEADING of IPDATA] ; If there's a page heading, do something about it.  
(SETQ HFONT (fetch IPHEADINGFONT of IPDATA))  
(DSPFONT.IP IPSTREAM HFONT) ; Set up heading font  
(SELECTQ (\IPC ENCODING)

(FULLIP-82 (PRIN3 (add (fetch IPPAGENUM of IPDATA)  
1)  
IPSTREAM)  
(FGET.IP IPSTREAM (fetch IPHEADINGOPVAR of (fetch IPDATA of IPSTREAM))) ; Get the heading operator  
(APPENDOP.IP IPSTREAM (\IPC DOSAVE)))

(IP-82 [SETXY.IP IPSTREAM (fetch IPLEFT of IPDATA)  
(DIFFERENCE (fetch IPTOP of IPDATA)  
(FONTPROP HFONT 'ASCENT]  
(DSPFONT HFONT IPSTREAM)  
(PRIN3 (fetch IPHEADING of IPDATA)  
IPSTREAM)

(REMOVETO MICASPERINCH 0 IPSTREAM) ; Skip an inch before page number

(PRIN3 "Page " IPSTREAM)  
(PRIN3 (add (fetch IPPAGENUM of IPDATA)  
1)  
IPSTREAM)

(NEWLINE.IP IPSTREAM) ; Skip 2 lines

(NEWLINE.IP IPSTREAM)

(SHOULDNT))

:: SETXY can't be done in HEADINGOP, cause the ascent of the current font is not known at image-time. We set it in terms of our  
:: current font, even though that hasn't yet be re-setup in the imager.

(SETYREL.IP IPSTREAM (IMINUS (FONTPROP CFONT 'ASCENT])  
(T (SETXY.IP IPSTREAM (fetch IPLEFT of IPDATA)  
(DIFFERENCE (fetch IPTOP of IPDATA)  
(FONTPROP CFONT 'ASCENT])

; Now we set the imagers font to our (previous) current font, to  
; override heading  
; Set up so that CORRECTs don't have to be exact.

(APPENDINTEGER.IP IPSTREAM 25)

(APPENDINTEGER.IP IPSTREAM 0)

(APPENDOP.IP IPSTREAM (\IPC SETCORRECTTOLERANCE))

(COND

([NOT (EQP 1 (ffetch IPSPACEFACTOR of IPDATA)))] ; Imager variables revert to initial values

```
(APPENDNUMBER.IP IPSTREAM (ffetch IPSPACEFACTOR of IPDATA))
(ISET.IP IPSTREAM (\IPC AMPLIFYSPACE])
(DSPFONT.IP IPSTREAM CFONT])
```

(NEWPAGE?.IP

```
[LAMBDA (IPSTREAM)
```

```
(* hdj "18-Oct-85 15:38")
; Are we about to overflow the page?
```

```
(COND
((LESSP (fetch IPYPOS of (fetch IPDATA of IPSTREAM))
(fetch IPBOTTOM of (fetch IPDATA of IPSTREAM)))
(NEWPAGE.IP IPSTREAM])
```

(OPENIPSTREAM

```
[LAMBDA (IPFILE OPTIONS)
```

```
; Edited 1-May-2023 22:09 by Imm
; Edited 27-Jun-2021 23:50 by rmk:
; Edited 18-Aug-88 16:13 by hdj
```

```
:: Opens an interpress stream, which user can OUTCHAR to. The FONTS option can be a list of fonts to be set up in the preamble. Headings will
;; be printed in the first font in that list. If that list is NIL, then the stream is initialized with the INTERPRESS DEFAULTFONT
```

```
(DECLARE (GLOBALVARS DEFAULTPAGEREGION \IPIMAGEOPS \NOIMAGEOPS PRINTER.DEFAULT.SCAN.DIRECTION
PRINTER.SCAN.DIRECTIONS.LIST)
(USEDFREE SERVER)) ; FVAR SERVER may be appeared in TEDIT.HARDCOPY
```

```
(LET* [(OPTION NIL)
(IPSTREAM (OPENSTREAM IPFILE 'OUTPUT 'NEW NIL ' ((TYPE INTERPRESS)
(MARGINREGION (COND
```

```
((type? REGION (SETQ OPTION (LISTGET OPTIONS 'REGION)
OPTION)
(LISTGET OPTIONS 'LANDSCAPE) ; Landscape printing: Set up things sideways.
DEFAULTLANDPAGEREGION)
(T DEFAULTPAGEREGION)))
```

```
[IPDATA (create INTERPRESSDATA
IPPAGEREGION _ MARGINREGION
IPLEFT _ (fetch (REGION LEFT) of MARGINREGION)
IPRIGHT _ (fetch (REGION RIGHT) of MARGINREGION)
IPTOP _ (fetch (REGION TOP) of MARGINREGION)
IPBOTTOM _ (fetch (REGION BOTTOM) of MARGINREGION)
IPSHOWSTREAM _ (PROG1 (OPENSTREAM '{NODIRCORE} 'BOTH 'OLD/NEW)
```

```
:: Make sure the fileptr of the following is zero (GETRESOURCE \IPSHOWSTREAM)
;; (and free this in CLOSEIPSTREAM)
```

```
)
IPDOCNAME _ (LISTGET OPTIONS 'DOCUMENT.NAME)
IPCLIPINCLUSIVE _ (LISTGET OPTIONS 'CLIP.INCLUSIVE]
(PAPERSIZE (\PAPERSIZE.IP IPSTREAM (LISTGET OPTIONS 'MEDIUM]
```

```
; Set up initial margins without calling functions to insure
; coercions and side-effects until everything is initialized. Note
; that linelength is initialized when font is set
```

```
(COND
((OR (NEQ \NOIMAGEOPS (fetch (IPSTREAM IMAGEOPS) of IPSTREAM))
(NEQ 0 (GETEOFPTR IPSTREAM)))
(ERROR "can't convert existing file to Interpress" (FULLNAME IPSTREAM))
; GETEOFPTR might bomb on some streams
```

```
))
;; We install a special external format to ensure that COPYCHARS won't do COPYBYTES when copying a from a text file to an IP
;; stream. Really only the outcharfn matters.
```

```
(\EXTERNALFORMAT IPSTREAM (create EXTERNALFORMAT
NAME _ 'INTERPRESS
OUTCHARFN _ (FUNCTION INTERPRESS.OUTCHARFN)
EOL _ (fetch (STREAM EOLCONVENTION) of IPSTREAM)))
```

```
(replace (IPSTREAM IMAGEOPS) of IPSTREAM with \IPIMAGEOPS)
(replace (IPSTREAM IPDATA) of IPSTREAM with IPDATA)
```

```
[COND
((LISTGET OPTIONS 'LANDSCAPE) ; For landscape printing, set up the default rotation and Y
; translate, and swap the papersize width and height
```

```
(replace (INTERPRESSDATA IPROTATION) of IPDATA with 90)
(replace (INTERPRESSDATA IPYOFFSET) of IPDATA with -21590)
(swap (CAR PAPERSIZE)
(CADR PAPERSIZE])
```

```
(STREAMPROP IPSTREAM 'PAPERSIZE (COPY PAPERSIZE))
(STREAMPROP IPSTREAM 'CLIP.INCLUSIVE (LISTGET OPTIONS 'CLIP.INCLUSIVE))
(replace IPPAGEFRAME of IPDATA with (create REGION
```

```
LEFT _ 0
BOTTOM _ 0
WIDTH _ (CAR PAPERSIZE)
HEIGHT _ (CADR PAPERSIZE)))
; Region created so can use INTERSECTREGIONS to compute
; visible region
```

```
(INITIALIZEMASTER.IP IPSTREAM)
(BEGINMASTER.IP IPSTREAM)
(BEGINPREAMBLE.IP IPSTREAM)
```

```
(COND
((SETQ OPTION (LISTGET OPTIONS 'HEADING))
(replace IPHEADING of IPDATA with OPTION)
```

```

(SELECTQ (\IPC ENCODING)
(FULLIP-82 (HEADINGOP.IP IPSTREAM OPTION))
(GETFRAMEVAR.IP IPSTREAM))
(T (GETFRAMEVAR.IP IPSTREAM)) ; initialize the stack
;; Allocate framevar 0, for heading op if there is one, otherwise for nothing. This means that the fonts will be in framevars that correspond
;; to their position in PREAMBLEFONTS. MAKEINTERPRESS relies on this.
(SETUPFONTS.IP IPSTREAM (LISTGET OPTIONS 'FONTS)) ; Initially clips to the page, after font installed
(DSPCLIPPINGREGION.IP IPSTREAM (fetch (INTERPRESSDATA IPPAGEFRAME) of IPDATA))
(COND
  ((LISTGET OPTIONS 'COLOR)
  (INITIALIZECOLOR.IP IPSTREAM)
  (STREAMPROP IPSTREAM 'COLOR T)))
(PUSH-IP-STACK IPSTREAM (create IPSTATE))
(NEWPAGE.IP IPSTREAM) ; NEWPAGE automatically closes the preamble
;; We need to set up the scan direction spec, so that polygon filling doesn't crash printers.
[LET [(PRINTSERVERNAME (OR (AND (BOUNDP 'SERVER)
SERVER)
(LISTGET OPTIONS 'SERVER)
(AND (EQ 'LPT (FILENAMEFIELD IPSTREAM 'HOST))
(LET (POS (FILE (FULLNAME IPSTREAM)))
;; This should be (FILENAMEFIELD FILE 'NAME) except that FILENAMEFIELD won't accept : as part
;; of the name, thinks it marks a device field. This code is borrowed from PRINTERDEVICE
(AND (SETQ POS (STRPOS "]" FILE))
(SUBSTRING FILE (ADD1 POS)
(SUB1 (OR (STRPOS "." FILE (ADD1 POS))
0]
; Puts the printer's scan direction into the stream.
(CL:WHEN PRINTSERVERNAME
(STREAMPROP IPSTREAM 'P.SCAN.DIRECTION (OR (CDR (CL:ASSOC (NSNAME.TO.STRING (PARSE.NSNAME
PRINTSERVERNAME)
)
PRINTER.SCAN.DIRECTIONS.LIST :TEST
#'STRING-EQUAL))
PRINTER.DEFAULT.SCAN.DIRECTION)))]
IPSTREAM])

```

(SETUPFONTS.IP

```

[LAMBDA (IPSTREAM FONTS) (* rmk%: "15-Sep-84 02:16")
;; Sets up preamble fonts, and sets heading font. Leaves IPFONT as NIL. This means that \DSPFONT.IP of the heading font will establish that as
;; the current font when the preamble is closed and the first page opens. NIL. Note that the preamble can't set the font imager variable.
(for F (IPDATA _ (fetch IPDATA of IPSTREAM)) inside (OR FONTS DEFAULTFONT)
do (SETQ F (FONTCREATE F NIL NIL NIL 'INTERPRESS))
(DEFINEFONT.IP IPSTREAM F)
(COND
  (IPDATA ; Take first font as heading font, and make it look like old current
; font on first NEWPAGE
(replace IPFONT of IPDATA with F)
(replace IPHEADINGFONT of IPDATA with F)
(SETQ IPDATA NIL]))

```

(SHOWBITMAP.IP

```

[LAMBDA (IPSTREAM BITMAP REGION SCALE ROTATION) ; Edited 2-May-2023 09:06 by Imm
; Edited 14-Jan-88 01:09 by FS
;; Puts out bit map with lower-left corner at current position. If given, REGION is a clipping region on the bitmap.
;; Brain damaged, "lower-left corner"?! What does rotation mean then, is the resulting image always (viewed from static observer holding paper) in
;; the NorthEast quadrant wrt x,y (rotated about its center and output), or not (rotated about x,y)?? It didn't work either way, so I rewrote it (in
;; showbitmap1.ip) to do the former. -FS.
(SHOW.IP IPSTREAM)
(PROG (XPIXELS YPIXELS XBYTES)
[COND
  [REGION ; Clip the incoming bitmap to the specified region.
(COND
  ([SETQ REGION (INTERSECTREGIONS REGION
(create REGION
LEFT _ 0
BOTTOM _ 0
WIDTH _ (fetch BITMAPWIDTH of BITMAP)
HEIGHT _ (fetch BITMAPHEIGHT of BITMAP)
(SETQ XPIXELS (fetch WIDTH of REGION))
(SETQ YPIXELS (fetch HEIGHT of REGION))
(T
(RETURN) ; The clipping region doesn't overlap this bitmap. Punt.
(T (SETQ XPIXELS (fetch BITMAPWIDTH of BITMAP))
(SETQ YPIXELS (fetch BITMAPHEIGHT of BITMAP))
(SETQ XBYTES (CEIL (FOLDHI XPIXELS BITS PER BYTE)
BYTESPERCELL)) ; Lines must be padded to multiples of 32bits (cells)
(COND
  ((IGREATERP XBYTES (\IPC MAXLONGSEQUENCEBYTES)) ; We should really start breaking it up in the X direction as well
(ERROR "Bitmap line too long for Interpress printing"))
((ZEROP XBYTES) ; Don't want to do anything if the bitmap is zero wide or high.

```

```

    (RETURN))
    ((ZEROP YPIXELS) ; Don't want to do anything if the bitmap is zero wide or high.
    (RETURN))) ; put out to avoid moire patterns
    (SETQ SCALE (COND
      (SCALE (TIMES SCALE (FQUOTIENT 2540 75)))
      (T (FQUOTIENT 2540 75))) ; Go to unit of 4 raven spots ~= 1 screen point
  )
  (bind LEFT (NEXTROW _ 0)
        (BOTTOM _ 0)
        (HEIGHT _ YPIXELS)
        (MAXYPIXELSPERCHUNK _ (IQUOTIENT (\IPC MAXLONGSEQUENCEBYTES)
                                         XBYTES))
  while (IGREATERP YPIXELS 0) first [COND
    (REGION
      ;; We're displaying a subsection of the bitmap. Set up the fields that let
      ;; SHOWBITMAP1.IP pick bits from the right place
      (SETQ LEFT (fetch LEFT of REGION))
      (SETQ BOTTOM (fetch BOTTOM of REGION))
    do ;; The bitmap is put out in chunks, from top to bottom -- corresponding to the order that the bits appear in memory.
      (SHOWBITMAP1.IP IPSTREAM BITMAP LEFT NEXTROW XPIXELS (IMIN YPIXELS MAXYPIXELSPERCHUNK)
        SCALE ROTATION HEIGHT XBYTES BOTTOM)
      (SETQ YPIXELS (IDIFFERENCE YPIXELS MAXYPIXELSPERCHUNK))
      (SETQ NEXTROW (IPLUS NEXTROW MAXYPIXELSPERCHUNK))
      ;; This is the next row of the bitmap (counting from the top of the region to be displayed) to go to the file.
    ])
  ]

```

(BITMAPSIZE.IP

```

[LAMBDA (STREAM BITMAP DIMENSION) ; (* rrb "11-Mar-86 10:03")
  ;; returns the height a bitmap will have on an interpress device. This is reduced in scale by 4 to avoid moire patterns on the 8044 by using
  ;; (FQUOTIENT 2540 75) rather than MICASPERPT
  (SELECTQ DIMENSION
    (WIDTH (TIMES (BITMAPWIDTH BITMAP)
                  (CONSTANT (FQUOTIENT 2540 75))))
    (HEIGHT (TIMES (BITMAPHEIGHT BITMAP)
                   (CONSTANT (FQUOTIENT 2540 75))))
    (NIL [CONS (TIMES (BITMAPWIDTH BITMAP)
                      (CONSTANT (FQUOTIENT 2540 75)))
              (TIMES (BITMAPHEIGHT BITMAP)
                      (CONSTANT (FQUOTIENT 2540 75))])
    (\ILLEGAL.ARG DIMENSION])

```

(SHOWBITMAP1.IP

```

[LAMBDA (IPSTREAM BITMAP LEFT FIRSTROW XPIXELS YPIXELS SCALEFACTOR ROTATION HEIGHT XBYTES REGIONBOTTOM)
  ; Edited 2-May-2023 08:49 by Imm
  ; Edited 14-Jan-88 00:52 by FS

  ;; Move a segment of bitmap to an INTERPRESS file.
  ;; FIRSTROW is the row count -- STARTING FROM THE TOP OF THE BITMAP AS ZERO -- for the first row to be displayed.
  ;; By the time we get here, XBYTES should have been raised to the next multiple of 32-bits-worth, since that's the required width of packed pixel
  ;; vectors.
  (PROG [(TOTALBYTES (ITIMES XBYTES YPIXELS))
        (SCRATCHBM (BITMAPCREATE (CEIL XPIXELS BITSPERCELL)
                                  1))
        (BMBASE (\ADDBASE (fetch (BITMAP BITMAPBASE) of BITMAP)
                             (ITIMES (IDIFFERENCE (IPLUS HEIGHT (OR REGIONBOTTOM 0))
                                                    (IPLUS FIRSTROW YPIXELS))
                                       (fetch (BITMAP BITMAPRSTERWIDTH) of BITMAP))
        (APPENDOP.IP IPSTREAM (\IPC DOSAVESIMPLEBODY))
        (APPENDOP.IP IPSTREAM (\IPC {})) ; Start the SIMPLEBODY for displaying this part of the bitmap.
        (TRANS.IP IPSTREAM) ; Translate to the current position
        (APPENDNUMBER.IP IPSTREAM YPIXELS) ; For the master, this is the number of pixels in the slow direction
        (APPENDNUMBER.IP IPSTREAM (CEIL XPIXELS BITSPERCELL)) ; Number of pixels in the master's fast direction
        (APPENDINTEGER.IP IPSTREAM 1) ; Reserved for future expansion
        (APPENDINTEGER.IP IPSTREAM 1)
        (APPENDINTEGER.IP IPSTREAM 1)

  ;; Adjusts segment (move in X because bitmap is rotated (see below)). Push this segment up to its 'true' height -- i.e., The first segment gets
  ;; pushed up all the way (since it's the top of the bitmap), the next segment gets pushed up HEIGHT-#ofRowsIn1stSeg (to account for the first
  ;; segment), and so on.
  (TRANSLATE.IP IPSTREAM (IDIFFERENCE 0 (IPLUS FIRSTROW YPIXELS))
    0)

  ;; Bitmaps are really shown on their sides (fast scan direction), hanging from the upper left corner.
  (SETQ ROTATION (IMOD (OR ROTATION 0)
                       360))
  (if (EQL ROTATION 90)
    elseif (OR (EQL ROTATION 0)
              (EQL ROTATION 180)
              (EQL ROTATION 270))
    then (ROTATE.IP IPSTREAM (- ROTATION 90))

```

```
(CONCAT.IP IPSTREAM)
else (ERROR ROTATION "rotation by other than multiples of 90 degrees not implemented")
(SCALE.IP IPSTREAM SCALEFACTOR) ; Scale the bitmap to its final size
(CONCAT.IP IPSTREAM)
(APPENDSEQUENCEDESCRIPTOR.IP IPSTREAM (\IPC SEQPACKEDPIXELVECTOR)
(IPLUS 4 TOTALBYTES))
(APPENDINT.IP IPSTREAM 1 2)
(APPENDINT.IP IPSTREAM (CELL XPIXELS BITSPERCELL)
2)
```

:: Now put out the bitmap -- each line must be a 32-bit multiple long

```
(for Y (XWORDS _ (FOLDHI XBYTES BYTESPERWORD)) from 1 to YPIXELS
do (BITBLT BITMAP (OR LEFT 0)
(IDIFFERENCE (IPLUS (OR REGIONBOTTOM 0)
FIRSTROW YPIXELS)
Y)
SCRATCHBM 0 0 XPIXELS 1 'INPUT 'REPLACE)
(\ABOUTS IPSTREAM (fetch (BITMAP BITMAPBASE) of SCRATCHBM)
0
(CELL XBYTES BYTESPERCELL)))
(APPENDOP.IP IPSTREAM (\IPC MAKEPIXELARRAY))
(APPENDOP.IP IPSTREAM (\IPC MASKPIXEL))
(APPENDOP.IP IPSTREAM (\IPC )))
```

**(SHOWSHADE.IP**

```
[LAMBDA (IPSTREAM SHADE REGION OPERATION SCALE ANGLE) ; Edited 2-May-2023 08:50 by Imm
; Edited 15-Aug-88 09:30 by rmk:
```

::: Puts out bit map with lower-left corner at current position. REGION is a clipping region on the bitmap.

```
(SHOW.IP IPSTREAM)
(APPENDOP.IP IPSTREAM (\IPC DOSAVESIMPLEBODY))
(APPENDOP.IP IPSTREAM (\IPC {}))
(SETCOLOR.IP IPSTREAM SHADE OPERATION SCALE ANGLE)
(APPENDINTEGER.IP IPSTREAM (fetch (REGION LEFT) of REGION))
(APPENDINTEGER.IP IPSTREAM (fetch (REGION BOTTOM) of REGION))
(APPENDINTEGER.IP IPSTREAM (fetch (REGION WIDTH) of REGION))
(APPENDINTEGER.IP IPSTREAM (fetch (REGION HEIGHT) of REGION))
(APPENDOP.IP IPSTREAM (\IPC MASKRECTANGLE))
(APPENDOP.IP IPSTREAM (\IPC )))
```

**(\BITBLT.IP**

```
[LAMBDA (SOURCEBITMAP SOURCELEFT SOURCEBOTTOM DESTINATION DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT
SOURCECETYPE OPERATION TEXTURE CLIPPINGREGION CLIPPEDSOURCELEFT CLIPPEDSOURCEBOTTOM)
; Edited 5-Aug-88 14:37 by rmk:
```

::: what this does: because there is no device-supported clipping in IP2.1, we are forced to do it ourselves. We transform the bitmap region into IP space, do the clipping there, then transform it back. Most of the ugliness comes from doing arithmetic on regions, which is always big and messy

```
(LET* [(OLDX (\DSPXPOSITION.IP DESTINATION))
(OLDY (\DSPYPOSITION.IP DESTINATION))
(DESTINATIONLEFT (OR DESTINATIONLEFT OLDX))
(DESTINATIONBOTTOM (OR DESTINATIONBOTTOM OLDY))
(SOURCE-REGION NIL)
(STREAMSCALE (DSPSCALE NIL DESTINATION))
(DESTWIDTH (TIMES STREAMSCALE WIDTH))
(DESTHEIGHT (TIMES STREAMSCALE HEIGHT))
(DESTINATIONREGION (INTERSECTREGIONS (CREATEREGION DESTINATIONLEFT DESTINATIONBOTTOM DESTWIDTH
DESTHEIGHT)
(fetch (INTERPRESSDATA IPVISIBLEREGION) of (fetch (IPSTREAM IMAGEDATA)
of DESTINATION))
(if CLIPPINGREGION
then (SETQ DESTINATIONREGION (INTERSECTREGIONS DESTINATIONREGION CLIPPINGREGION))
; transform the clipping region into source coord space
(if DESTINATIONREGION
then (\MOVETO.IP DESTINATION (fetch (REGION LEFT) of DESTINATIONREGION)
(fetch (REGION BOTTOM) of DESTINATIONREGION))
[SETQ SOURCE-REGION (CREATEREGION (PLUS CLIPPEDSOURCELEFT
(FIXR (QUOTIENT (DIFFERENCE (fetch (REGION LEFT)
of DESTINATIONREGION)
DESTINATIONLEFT)
STREAMSCALE)))
(PLUS CLIPPEDSOURCEBOTTOM (FIXR (QUOTIENT
(DIFFERENCE (fetch (REGION BOTTOM)
of
DESTINATIONREGION)
DESTINATIONBOTTOM)
STREAMSCALE)))]
(FIXR (QUOTIENT (fetch (REGION WIDTH) of DESTINATIONREGION)
STREAMSCALE))
(FIXR (QUOTIENT (fetch (REGION HEIGHT) of DESTINATIONREGION)
STREAMSCALE)]
(SHOWBITMAP.IP DESTINATION SOURCEBITMAP SOURCE-REGION 1)
```

```

(\MOVETO.IP DESTINATION OLDX OLDY) ;
T
else NIL])

```

**(\SCALED BITBLT.IP**

```

[LAMBDA (SOURCEBITMAP SOURCELEFT SOURCEBOTTOM DESTINATION DESTINATION-LEFT DESTINATION-BOTTOM WIDTH HEIGHT
SOURCE TYPE OPERATION TEXTURE CLIPPINGREGION CLIPPEDSOURCELEFT CLIPPEDSOURCEBOTTOM SCALE)
; Edited 19-Aug-88 11:02 by hdj

```

;; Print a clipped and scaled bitmap.

```

(LET* [(OLDX (\DSPXPOSITION.IP DESTINATION))
(OLDY (\DSPYPOSITION.IP DESTINATION))
(DESTINATION-LEFT (OR DESTINATION-LEFT OLDX))
(DESTINATION-BOTTOM (OR DESTINATION-BOTTOM OLDY))
(SOURCE-REGION NIL)
(STREAM-SCALE (DSPSCALE NIL DESTINATION))
(DESTINATION-REGION (INTERSECTREGIONS (CREATEREGION DESTINATION-LEFT DESTINATION-BOTTOM
(TIMES SCALE STREAM-SCALE WIDTH)
(TIMES SCALE STREAM-SCALE HEIGHT))
(fetch (INTERPRESSDATA IPVISIBLEREGION) of (fetch (IPSTREAM IMAGEDATA)
of DESTINATION))
(if CLIPPINGREGION
then (SETQ DESTINATION-REGION (INTERSECTREGIONS DESTINATION-REGION CLIPPINGREGION)))
;; transform the clipping region into source coord space
(if DESTINATION-REGION
then (\MOVETO.IP DESTINATION (fetch (REGION LEFT) of DESTINATION-REGION)
(fetch (REGION BOTTOM) of DESTINATION-REGION))
[SETQ SOURCE-REGION (CREATEREGION (+ CLIPPEDSOURCELEFT
(FIXR (QUOTIENT (- (fetch (REGION LEFT) of
DESTINATION-REGION)
)
DESTINATION-LEFT)
STREAM-SCALE))
(+ CLIPPEDSOURCEBOTTOM (FIXR (QUOTIENT (- (fetch (REGION BOTTOM)
of DESTINATION-REGION)
DESTINATION-BOTTOM)
STREAM-SCALE))
(FIXR (QUOTIENT (fetch (REGION WIDTH) of DESTINATION-REGION)
(TIMES SCALE STREAM-SCALE))
(FIXR (QUOTIENT (fetch (REGION HEIGHT) of DESTINATION-REGION)
(TIMES SCALE STREAM-SCALE))
(SHOWBITMAP.IP DESTINATION SOURCEBITMAP SOURCE-REGION SCALE)
(\MOVETO.IP DESTINATION OLDX OLDY) ;
T
else NIL])

```

**(\BLTSHADE.IP**

```

[LAMBDA (TEXTURE STREAM DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT OPERATION CLIPPINGREGION)
; Edited 2-May-2023 08:35 by lmm
; Edited 5-Aug-88 14:37 by rmk:

```

```

(PROG [(DESTREGION (INTERSECTREGIONS (fetch (INTERPRESSDATA IPVISIBLEREGION) of (fetch (IPSTREAM IMAGEDATA)
of STREAM))
(CREATEREGION DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT)
(if (NOT DESTREGION)
then (RETURN))
(if CLIPPINGREGION
then (SETQ DESTREGION (INTERSECTREGIONS DESTREGION CLIPPINGREGION)))
(if (NOT DESTREGION)
then (RETURN))
(OR OPERATION (SETQ OPERATION (DSPOPERATION NIL STREAM)))
(COND
(> PRINTSERVICE 8.0)
(SHOWSHADE.IP STREAM (INSURE.B&W.TEXTURE TEXTURE)
DESTREGION OPERATION))
(T
; until 8044s can print scaled textures without crashing
(\BLTSHADE.GENERICPRINTER TEXTURE STREAM DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT
OPERATION CLIPPINGREGION (\IPC \INTERPRESSSCALE])

```

**(\CHARWIDTH.IP**

```

[LAMBDA (STREAM CHARCODE) (* rmk%: "12-Apr-85 09:42")
```

;; Gets the width of CHARCODE in an Interpress STREAM, observing spacefactor

```

(COND
((EQ CHARCODE (CHARCODE SPACE))
(fetch IPSPACEWIDTH of (fetch IMAGEDATA of STREAM)))
(T (\FGETCHARWIDTH (fetch IPFONT of (fetch IMAGEDATA of STREAM))
CHARCODE])

```

**(\CLOSEIPSTREAM**

```

[LAMBDA (IPSTREAM) (* rmk%: "27-JUL-83 19:48")
(SELECTQ (fetch IPPAGESTATE of (fetch IPDATA of IPSTREAM))
(PAGE (ENDPAGE.IP IPSTREAM))

```

```
(PREAMBLE (ENDPREAMBLE.IP IPSTREAM))
NIL
(ENDMASTER.IP IPSTREAM])
```

(DRAWARC.IP

```
[LAMBDA (STREAM CENTERX CENTERY RADIUS STARTANGLE NDEGREES BRUSH DASHING)
(* rrb "4-Oct-85 17:24")
; draws an arc on an interpress file
(\DRAWARC.GENERIC STREAM CENTERX CENTERY RADIUS STARTANGLE NDEGREES BRUSH DASHING])
```

(DRAWCURVE.IP

```
[LAMBDA (IPSTREAM KNOTS CLOSED BRUSH DASHING) ; Edited 2-May-2023 08:51 by Imm
; Edited 5-Aug-88 16:45 by rmk:
```

;; draws a spline curve with a given brush--except that dashing is currently ignored, and the curve is done with straight lines.

```
[COND
((LISTP KNOTS) ; to allow the brush color to have the correct scope
(LET (K)
[OR (CDR KNOTS)
(SETQ KNOTS (LIST (CAR KNOTS)
(CAR KNOTS)) ; The funny case of a single knot
(COND
((AND (NULL DASHING)
(EQ 2 (LENGTH KNOTS))) ; There were only two knots, and no dashing.
(OR (type? POSITION (SETQ K (CAR KNOTS)))
(ERROR "bad knot" K))
(\DRAWLINE.IP IPSTREAM (fetch XCOORD of K)
(fetch YCOORD of K)
[fetch XCOORD of (COND
((type? POSITION (SETQ K (CADR KNOTS)))
K)
(T (ERROR "bad knot" K))
(fetch YCOORD of K)
BRUSH))
(T ; Otherwise, use the full-strength curve drawer.
(SHOW.IP IPSTREAM T)
(APPENDOP.IP IPSTREAM (\IPC DOSAVESIMPLEBODY))
(APPENDOP.IP IPSTREAM (\IPC (\IPC {})))
(IPCURVE2 IPSTREAM (PARAMETRICSPLINE KNOTS CLOSED)
DASHING BRUSH) ; This leaves the current position at the endpoint of the curve.
(APPENDOP.IP IPSTREAM (\IPC {}))
(SETQ K (CAR (LAST KNOTS)))
(SETXY.IP IPSTREAM (fetch XCOORD of K)
(fetch YCOORD of K)
IPSTREAM])
```

(DRAWPOINT.IP

```
[LAMBDA (IPSTREAM X Y BRUSH OPERATION) ; Edited 8-Aug-88 15:55 by rmk:
; draws a single point.
; to allow the brush color to have the correct scope
(SHOW.IP IPSTREAM)
(if (BITMAPP BRUSH)
then ;; Awful crufty case, must support it because it's documented.
(LET ((WIDTH (BITMAPWIDTH BRUSH))
(HEIGHT (BITMAPHEIGHT BRUSH)))
;; Call toplevel guy so don't need to set up clipping nonsense
(BITBLT BRUSH 0 0 IPSTREAM [- X (ITIMES WIDTH (CONSTANT (IQUOTIENT MICASPERPT 2)
[- Y (ITIMES HEIGHT (CONSTANT (IQUOTIENT MICASPERPT 2)
WIDTH HEIGHT OPERATION))
else (\DRAWLINE.IP IPSTREAM X Y X Y BRUSH OPERATION))
IPSTREAM])
```

(DSPCOLOR.IP

```
[LAMBDA (IPSTREAM COLOR) (* edited%: "31-Mar-86 15:36")
(if (STREAMPROP IPSTREAM 'COLOR)
then ; this is an interpress stream which can interpret color, otherwise
; dspcolor is a no-op
(if COLOR
then (LET* ((IPDATA (fetch IPDATA of IPSTREAM))
(RGB (ENSURE.RGB COLOR)))
(replace (INTERPRESSDATA IPCOLOR) of IPDATA with RGB)
(SETRGB.IP IPSTREAM (CAR RGB)
(CADR RGB)
(CADDR RGB)))
else (fetch (INTERPRESSDATA IPCOLOR) of (fetch IPDATA of IPSTREAM]))
```

(ENSURE.RGB

```
[LAMBDA (COLOR NOERRORFLG?) (* edited%: "31-Mar-86 21:41")
;; returns an rgb triple or errors (NIL if NOERRORFLG). Acceptable input is RGB, HLS, or litatom on COLORNAMES
(LET ((RGB COLOR))
```



```
(COND
  ((LITATOM COLOR)
   (if (SETQ RGB (\LOOKUPCOLORNAME COLOR))
       then (pop RGB)))
  ((HLSP RGB)
   (HLSTORGB RGB)))
(if (NOT (RGBP RGB))
  then (if NOERRORFLG?
         then NIL
         else (ERROR "Illegal color" COLOR))
  else RGB])
```

(\IPCURVE2

```
[LAMBDA (IPSTREAM SPLINE DASHING BRUSH)
```

```
; Edited 2-May-2023 07:57 by Imm
; Edited 8-Aug-88 15:13 by rmk:
```

;; Given an Interpress stream, and a spline in the form of derivatives for each segment, and a brush to draw with, draw line segments to paint the curve.

;; NB: The endpoints of line segments are placed only to 1/300in accuracy, since that's all the accuracy our printers have. This speeds things up by a factor of 8 or more.

;; Changed to step in micas \SPLINESTEP.IP, initially 16 (approx. 1/2 pt.). Used to be 8 (approx. screen resolution)

```
(PROG ( (XPOLY (create POLYNOMIAL))
        (X'POLY (create POLYNOMIAL))
        (YPOLY (create POLYNOMIAL))
        (Y'POLY (create POLYNOMIAL))
        (X (fetch (SPLINE SPLINEX) of SPLINE))
        (Y (ffetch (SPLINE SPLINEY) of SPLINE))
        (X' (ffetch (SPLINE SPLINEDX) of SPLINE))
        (Y' (ffetch (SPLINE SPLINEDY) of SPLINE))
        (X'' (ffetch (SPLINE SPLINEDDX) of SPLINE))
        (Y'' (ffetch (SPLINE SPLINEDDY) of SPLINE))
        (X''' (ffetch (SPLINE SPLINEDDDX) of SPLINE))
        (Y''' (ffetch (SPLINE SPLINEDDDY) of SPLINE))
        (%#KNOTS (ffetch %#KNOTS of SPLINE))
        (IPXPOS (ELT (ffetch (SPLINE SPLINEX) of SPLINE)
                    1))
        (IPYPOS (ELT (ffetch (SPLINE SPLINEY) of SPLINE)
                    1))
        IX IY DX DY XT YT X'T Y'T NEWXT NEWYT XDIFF YDIFF XWALLDT YWALLDT DUPLICATEKNOT EXTRANEIOUS TT NEWT
        DELTA DASHON DASHLST DASHCNT IPDATA SEG# SPLINESTEP HALFWIDTH LEFT RIGHT BOTTOM TOP SPLINEDIFF
        VISIBLEP PREVX PREVY)
  (SETQ SPLINESTEP (FIX \SPLINESTEP.IP))
  (SETQ HALFWIDTH (FQUOTIENT (\WIDTHFROMBRUSH BRUSH (\IPC MICASPERPOINT))
                              2))
  (SETQ SPLINEDIFF \SPLINESTEP.IP)
  (SETQ DASHON T)
```

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

```
(SETQ DASHLST DASHING) ; Make a circular list of dashing intervals, so that we can just
                        ; CDR down it to find dashings.

(SETQ DASHCNT (CAR DASHING))
(SETQ SEG# 0)
(SETQ IPDATA (fetch IMAGEDATA of IPSTREAM))
(SETQ LEFT (+ (fetch IPVISLEFT of IPDATA)
              HALFWIDTH))
(SETQ RIGHT (- (fetch IPVISRIGHT of IPDATA)
               HALFWIDTH))
(SETQ BOTTOM (+ (fetch IPVISBOTTOM of IPDATA)
               HALFWIDTH))
(SETQ TOP (- (fetch IPVISTOP of IPDATA)
             HALFWIDTH)) ; NOTE: Don't need to keep IPDATA up to date
(SETQ VISIBLEP (\VISIBLE.IP IPXPOS IPYPOS LEFT RIGHT TOP BOTTOM))
(if VISIBLEP
  then (MOVETO.IP IPSTREAM IPXPOS IPYPOS) ; Move to the curve's starting point
        ; We paint each segment by walking the parameter TT from 0.0
        ; to 1.0
  (SETQ TT 0.0)

  (SETQ DELTA 1024)
  (SETQ IX (FIXR IPXPOS))
  (SETQ IY (FIXR IPYPOS))
  [for KNOT# from 1 to (SUB1 %#KNOTS)
  do ; Draw each segment in turn
    (LOADPOLY XPOLY X'POLY (ELT X''' KNOT#)
              (ELT X'' KNOT#)
              (ELT X' KNOT#)
              (ELT X KNOT#))
    (LOADPOLY YPOLY Y'POLY (ELT Y''' KNOT#)
              (ELT Y'' KNOT#)
              (ELT Y' KNOT#)
              (ELT Y KNOT#))
    (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
    (SETQ X'T (POLYEVAL TT X'POLY 2)) ; Run the parameter TT from 0 to 1 for this segment
    (SETQ Y'T (POLYEVAL TT Y'POLY 2)) ; X'T _ X' (t)
    (COND ; Y'T _ Y' (t)
        ((EQP X'T 0.0) ; Prevent divide-by-zero
         (SETQ X'T 5.0E-4))
        ((EQP Y'T 0.0) ; Prevent divide-by-zero
         (SETQ Y'T 5.0E-4))
    )
    [COND
        ((FGREATERP X'T 0.0)
         (SETQ DX DELTA)
         (T (SETQ DX (IMINUS DELTA)
              (SETQ DX DELTA)
              (FGREATERP Y'T 0.0)
              (SETQ DY DELTA)
              (T (SETQ DY (IMINUS DELTA)
                   (SETQ DY DELTA)
                   (SETQ XWALLDT (FQUOTIENT (FDIFFERENCE (IPLUS IX DX)
                                                         XT)
                                                         X'T))
                   (SETQ YWALLDT (FQUOTIENT (FDIFFERENCE (IPLUS IY DY)
                                                         YT)
                                                         Y'T))
                   ; Decide which of dX or dY is changing faster, and use that as
                   ; the limiting value
                   [COND
                       ((FLESSP XWALLDT YWALLDT)
                        (SETQ NEWT (FPLUS TT XWALLDT))
                        (SETQ DY (IDIFFERENCE (FIXR (FPLUS YT (FTIMES XWALLDT Y'T))
                                                IY)))
                        (T (SETQ NEWT (FPLUS TT YWALLDT))
                           (SETQ DX (IDIFFERENCE (FIXR (FPLUS XT (FTIMES YWALLDT X'T))
                                                    IX)
                                                    IY)))
                        (COND
                            ([AND (FGTP NEWT 1.0)
                                 (OR DUPLICATEKNOT (EQ KNOT# (SUB1 %KNOTS))
                                     ; If we've run TT past 1, or if this knot is duplicated (meaning make a discontinuity in x' & y') then draw straight to the
                                     ; end point.
                                     (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))) ; Find out how close we come to the ideal
                             (SETQ YDIFF (ABS (FDIFFERENCE (IPLUS IY DY)
                                                            NEWYT)))
                             (COND
                                 ((AND (IGREATERP DELTA 8)
                                      (OR (FGREATERP XDIFF SPLINESTEP)
                                           (FGREATERP YDIFF SPLINESTEP)))
                                  ; We're more than a printer dot off, and we still have room to make the DX or DY smaller. Do so & try again.
                                  (SETQ DELTA (LRSH DELTA 1)))
                                 (T
                                  (COND
                                      ((IGREATERP (add SEG# 1)
                                       MAXSEGSPERTRAJECTORY)
                                       ; This is as close as we can come. Draw the line segment.
                                       ; Our printers limit the number of segments in a single TRAJECTORY; make sure we respect their
                                       ; limitations
                                       (\IMAGEPATH.IP BRUSH IPSTREAM)
                                       (SETQ SEG# 0)
                                       (MOVETO.IP IPSTREAM IPXPOS IPYPOS))
                                      (SETQ PREVX IPXPOS)
                                      (SETQ IPXPOS (PLUS IPXPOS DX))
                                      (SETQ PREVY IPYPOS)
                                      (SETQ IPYPOS (PLUS IPYPOS DY)) ; Now check clipping
                                      (if VISIBLEP
                                       then (if (SETQ VISIBLEP (\VISIBLE.IP IPXPOS IPYPOS LEFT RIGHT TOP BOTTOM))
                                               then
                                               (LINETO.IP IPSTREAM IPXPOS IPYPOS) ; Super-common case: both ends visible, draw the line
                                               else
                                               (\CLIPCURVELINE.IP PREVX PREVY IPXPOS IPYPOS LEFT RIGHT TOP BOTTOM T
                                                IPSTREAM) ; Starts visible, goes out
                                               (\IMAGEPATH.IP BRUSH IPSTREAM) ; Curve is now invisible
                                               (SETQ SEG# 0))
                                       else (if (SETQ VISIBLEP (\VISIBLE.IP IPXPOS IPYPOS LEFT RIGHT TOP BOTTOM))
                                               then
                                               ; Starts invisible, comes in. MOVETO is done in
                                               ; \CLIPCURVELINE.IP
                                               (\CLIPCURVELINE.IP PREVX PREVY IPXPOS IPYPOS LEFT RIGHT TOP BOTTOM NIL
                                                IPSTREAM)
                                               else
                                               ; Both ends invisible, could be visible in middle

```

```

      (if (\CLIPCURVELINE.IP PREVX PREVY IPXPOS IPYPOS LEFT RIGHT TOP BOTTOM
          NIL IPSTREAM)
        then
          (\IMAGEPATH.IP BRUSH IPSTREAM)
          (SETQ SEG# 0)
          ))
      (SETQ IX (IPLUS IX DX))
      (SETQ IY (IPLUS IY DY))
      (SETQ TT NEWT)
      (SETQ XT NEWXT)
      (SETQ YT NEWYT)
      (COND
        ((AND (ILESSP DELTA 1024)
              (OR (FLESSP XDIFF 4.0)
                  (FLESSP YDIFF 4.0)))
          ; If we were REAL close, we can relax a bit, and try moving
          ; farther next time.
          (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]
            (if VISIBLEP
              then
                ; Only need to clean up if we're now inside
                ; FS- Unfortunately no OPER.
                (\IMAGEPATH.IP BRUSH IPSTREAM])
            ))
      )

```

(\CLIPCURVELINE.IP

[LAMBDA (X1 Y1 X2 Y2 LEFT RIGHT TOP BOTTOM PT1VISP IPSTREAM) ; Edited 8-Aug-88 12:48 by rmk:

;; Called when the line between X1,Y1 X2,Y2 is known not to be entirely in the clipping region defined by LEFT RIGHT TOP BOTTOM, which have
;; already been adjusted by the halfwidth of the brush. If any part of the line is visible, it shows that segment, returns T if anything was shown for
;; any cleanup operators.

;; If PT1VISP and some part is visible, it knows that the initial part of the segment is visible and the final part is invisible. If not PT1VISP and
;; something is shown, then it knows that a MOVETO is necessary to the beginning of the segment.

(PROG (CA1 CA2 DX DY SWAPPED)

;; switch points so that X1 is less than X2.

```

      (if (> X1 X2)
        then
          (SETQ CA1 X1)
          (SETQ X1 X2)
          (SETQ X2 CA1)
          (SETQ CA1 Y1)
          (SETQ Y1 Y2)
          (SETQ Y2 CA1)
          (SETQ SWAPPED T))

```

```

      (SETQ DX (- X2 X1))
      (SETQ DY (- Y2 Y1))
      (SETQ CA1 (\CLIPCODE X1 Y1 LEFT RIGHT TOP BOTTOM))
      (SETQ CA2 (\CLIPCODE X2 Y2 LEFT RIGHT TOP BOTTOM))

```

; determine the sectors in which the points fall.

CLIPLP

```

      (COND
        ((NEQ 0 (LOGAND CA1 CA2))
          (RETURN NIL))
        ((EQ 0 (PLUS CA1 CA2))
          (if SWAPPED
            then
              (OR PT1VISP (MOVETO.IP IPSTREAM X2 Y2))
              (LINETO.IP IPSTREAM X1 Y1)
            else
              (OR PT1VISP (MOVETO.IP IPSTREAM X1 Y1))
              (LINETO.IP IPSTREAM X2 Y2))
          ))

```

; line is entirely out of clipping region

; line is now completely visible

; If PT1 wasn't visible, then we have to move to the point where
; the line enters the region. We can also assume that we are at
; the start of the trajectory, since caller does the setup

```

      (RETURN T))

```

[COND

```

      ((NEQ CA1 0)

```

;; now move point X1 Y1 so that one of the coordinates is on one of the boundaries. Which boundary is done first was copied from
;; BCPL.

(COND

```

      ((GREATERP CA1 7)
        ; y1 less than bottom
        ; calculate the least X for which Y will be at bottom.
        [SETQ X1 (PLUS X1 (FTIMES DX (FQUOTIENT (- BOTTOM Y1)
          DY]

```

```

      ((GREATERP CA1 3)
        ; y1 is greater than top
        [SETQ X1 (PLUS X1 (FTIMES DX (FQUOTIENT (- TOP Y1)
          DY]

```

```

      (SETQ Y1 TOP))
      (T
        ; x1 is less than left
        [SETQ Y1 (PLUS Y1 (FTIMES DY (FQUOTIENT (- LEFT X1)

```

```

DX]
  (SETQ X1 LEFT))
  (SETQ CA1 (\CLIPCODE X1 Y1 LEFT RIGHT TOP BOTTOM))
(T ; now move point X2 Y2 so that one of the coordinates is on one
; of the boundaries
(COND
  ((GREATERP CA2 7) ; y2 less than bottom
  [SETQ X2 (PLUS X2 (FTIMES DX (FQUOTIENT (- BOTTOM Y2)
  DY]
  (SETQ Y2 BOTTOM))
  ((GREATERP CA2 3) ; y2 is greater than top
  [SETQ X2 (- X2 (FTIMES DX (FQUOTIENT (- Y2 TOP)
  DY]
  (SETQ Y2 TOP))
(T ; x2 is greater than right
  [SETQ Y2 (- Y2 (FTIMES DY (FQUOTIENT (- X2 RIGHT)
  DX]
  (SETQ X2 RIGHT)))
  (SETQ CA2 (\CLIPCODE X2 Y2 LEFT RIGHT TOP BOTTOM]
(GO CLIPLP])

```

(DRAWLINE.IP

```

[LAMBDA (IPSTREAM X1 Y1 X2 Y2 WIDTH OPERATION COLOR DASHING) ; Edited 2-May-2023 07:59 by Imm
; Edited 8-Aug-88 15:15 by rmk:
(COND
  (DASHING ; added dashing hack --- rrb 27-sept-85
  (DRAWDASHEDLINE X1 Y1 X2 Y2 WIDTH OPERATION IPSTREAM COLOR DASHING))
(T ; RRB: A temporary interface function until we resolve the color/endshape/operation conflicts in the D.I.G. argument structure. Arguments
; are assumed to be in micas.
  (SHOW.IP IPSTREAM T)
  [LET ((IPDATA (ffetch (IPSTREAM IMAGEDATA) of IPSTREAM))
  (W (\WIDTHFROMBRUSH WIDTH (\IPC MICASPERPOINT)))
  HALFWIDTH)
  ;; FS: do quick and dirty test to avoid consing in the common case. Since Interpress line ends cannot extend past WIDTH, and
  ;; since line joints presumably cannot be made this way (not a polyline), simply grow line by WIDTH (which is conservatively more
  ;; than actual WIDTH/2)
  (APPENDOP.IP IPSTREAM (\IPC DOSAVESIMPLEBODY))
  (APPENDOP.IP IPSTREAM (\IPC {})) ; If totally clipped, this is a waste
  (COND
    ((AND (< (fetch (INTERPRESSDATA IPVISLEFT) of IPDATA)
    (- (MIN X1 X2)
    W))
    (< (fetch (INTERPRESSDATA IPVISBOTTOM) of IPDATA)
    (- (MIN Y1 Y2)
    W))
    (< (+ (MAX X1 X2)
    W)
    (fetch (INTERPRESSDATA IPVISRIGHT) of IPDATA))
    (< (+ (MAX Y1 Y2)
    W)
    (fetch (INTERPRESSDATA IPVISTOP) of IPDATA)))
  ;; Completely in clip region, common simple case.
  (MOVETO.IP IPSTREAM X1 Y1)
  (LINETO.IP IPSTREAM X2 Y2)
  (\IMAGEPATH.IP (COND
    ((BRUSHP WIDTH)
    WIDTH)
    (T (LIST 'BUTT WIDTH COLOR)))
  IPSTREAM OPERATION))
(T ; Must do more careful clipping in this case.
  (SETQ HALFWIDTH (FQUOTIENT W 2))
  (COND
    ((\CLIPCURVELINE.IP X1 Y1 X2 Y2 (+ (fetch IPVISLEFT of IPDATA)
    HALFWIDTH)
    (- (fetch IPVISRIGHT of IPDATA)
    HALFWIDTH)
    (- (fetch IPVISTOP of IPDATA)
    HALFWIDTH)
    (+ (fetch IPVISBOTTOM of IPDATA)
    HALFWIDTH)
    NIL IPSTREAM)
    (\IMAGEPATH.IP (COND
      ((BRUSHP WIDTH)
      WIDTH)
      (T (LIST 'BUTT WIDTH COLOR)))
      IPSTREAM OPERATION]
  (APPENDOP.IP IPSTREAM (\IPC {}))
  (SETXY.IP IPSTREAM X2 Y2])

```

(CLIPLINE

```

[LAMBDA (X1 Y1 X2 Y2 WIDTH CLIPREG) ; Edited 8-Aug-88 11:18 by rmk:

```

;; No longer called by Interpress, but may be called by someone else.

;; Clips the line X1 Y1 to X2 Y2 to the region CLIPREG leaving room for a brush WIDTH wide. If any part of the line is visible, it returns (LIST  
;; newX1 NewY1 NewX2 NewY2)

```
(PROG ((HALFWIDTH (FQUOTIENT WIDTH 2))
      LEFT RIGHT BOTTOM TOP CA1 CA2 DX DY)
```

;; set LEFT, RIGHT, BOTTOM, TOP to the boundaries of the clipping region compensating for the brush width.

```
(SETQ LEFT (+ (fetch (REGION LEFT) of CLIPREG)
              HALFWIDTH))
(SETQ RIGHT (- (fetch (REGION RIGHT) of CLIPREG)
              HALFWIDTH))
(SETQ BOTTOM (+ (fetch (REGION BOTTOM) of CLIPREG)
              HALFWIDTH))
(SETQ TOP (- (fetch (REGION TOP) of CLIPREG)
            HALFWIDTH))
```

; switch points so that X1 is less than X2.

```
(COND
  ((GREATERP X1 X2)
   (SETQ CA1 X1)
   (SETQ X1 X2)
   (SETQ X2 CA1)
   (SETQ CA1 Y1)
   (SETQ Y1 Y2)
   (SETQ Y2 CA1)))
(SETQ DX (DIFFERENCE X2 X1))
(SETQ DY (DIFFERENCE Y2 Y1))
(SETQ CA1 (\CLIPCODE X1 Y1 LEFT RIGHT TOP BOTTOM))
(SETQ CA2 (\CLIPCODE X2 Y2 LEFT RIGHT TOP BOTTOM))
```

; determine the sectors in which the points fall.

CLIPLP

```
[COND
  ((NOT (EQ 0 (LOGAND CA1 CA2)))
   (RETURN NIL))
  ((EQ 0 (PLUS CA1 CA2))
   (RETURN (LIST (FIXR X1)
                 (FIXR Y1)
                 (FIXR X2)
                 (FIXR Y2))
```

; line is entirely out of clipping region

; line is completely visible  
; reuse the variable CA1

[COND

```
( (NEQ CA1 0)
```

;; now move point X1 Y1 so that one of the coordinates is on one of the boundaries. Which boundary is done first was copied from  
;; BCPL.

```
(COND
  ((GREATERP CA1 7)
   [SETQ X1 (PLUS X1 (FTIMES DX (FQUOTIENT (DIFFERENCE BOTTOM Y1)
                                           DY))
              (SETQ Y1 BOTTOM))
   ((GREATERP CA1 3)
    [SETQ X1 (PLUS X1 (FTIMES DX (FQUOTIENT (DIFFERENCE TOP Y1)
                                           DY))
                  (SETQ Y1 TOP))
```

; y1 less than bottom  
; calculate the least X for which Y will be at bottom.

; y1 is greater than top

```
(T
  [SETQ Y1 (PLUS Y1 (FTIMES DY (FQUOTIENT (DIFFERENCE LEFT X1)
                                           DX))
            (SETQ X1 LEFT)))
(SETQ CA1 (\CLIPCODE X1 Y1 LEFT RIGHT TOP BOTTOM))
```

; x1 is less than left

(T  
; now move point X2 Y2 so that one of the coordinates is on one  
; of the boundaries

```
(COND
  ((GREATERP CA2 7)
   [SETQ X2 (PLUS X2 (FTIMES DX (FQUOTIENT (DIFFERENCE BOTTOM Y2)
                                           DY))
              (SETQ Y2 BOTTOM))
   ((GREATERP CA2 3)
    [SETQ X2 (DIFFERENCE X2 (FTIMES DX (FQUOTIENT (DIFFERENCE Y2 TOP)
                                                    DY))
                  (SETQ Y2 TOP))
   (T
    [SETQ Y2 (DIFFERENCE Y2 (FTIMES DY (FQUOTIENT (DIFFERENCE X2 RIGHT)
                                                    DX))
              (SETQ X2 RIGHT)))
  (SETQ CA2 (\CLIPCODE X2 Y2 LEFT RIGHT TOP BOTTOM])
```

; y2 less than bottom

; y2 is greater than top

; x2 is greater than right

(GO CLIPLP])

(\DSPBOTTOMMARGIN.IP

```
[LAMBDA (IPSTREAM YPOSITION)
  (PROG1 (fetch IPBOTTOM of (fetch IMAGEDATA of IPSTREAM))
    (COND
      (YPOSITION (replace IPBOTTOM of (fetch IMAGEDATA of IPSTREAM) with YPOSITION))))])
```

(\* rmk%: "26-Jun-84 14:01")

(\DSPFONT.IP

```
[LAMBDA (IPSTREAM FONT)
  ; Edited 2-May-2023 08:38 by Imm
  ; Edited 21-Aug-91 16:33 by jds
```

;; Change fonts (or return the current font) for an IP stream

```
(PROG (OLDFONT FRAMEVAR (IPDATA (ffetch IMAGEDATA of IPSTREAM)))
  (SETQ OLDFONT (ffetch IPFONT of IPDATA))
  (AND (NULL FONT)
    (RETURN OLDFONT))
  (SHOW.IP IPSTREAM) ; ALWAYS do the show, so that font changes force
                    ; recomputation of the exact position in the printer.
  (COND
    ([EQ OLDFONT (SETQ FONT (OR (\GETFONTDESC FONT 'INTERPRESS)
      (FONTCOPY OLDFONT FONT))] ; There was no change, or he was only asking for the old font.
      ; Just return it.
    (RETURN OLDFONT)))
  [SETQ FRAMEVAR (CDR (OR (ASSOC FONT (ffetch IPPAGEFONTS of IPDATA))
    (DEFINEFONT.IP IPSTREAM FONT))] ; Get the font number to go in the file
  (APPENDINTEGER.IP IPSTREAM FRAMEVAR)
  (APPENDOP.IP IPSTREAM (\IPC SETFONT))
  (freplace IPFONT of IPDATA with FONT) ; Remember the new font
  (\CHANGECHARSET.IP IPDATA \DEFAULTCHARSET)
  [freplace IPSPACEWIDTH of IPDATA with (FIXR (TIMES (ffetch IPSPACEFACTOR of IPDATA)
    (\FGETWIDTH (ffetch IPWIDTHSCACHE of IPDATA)
      (CHARCODE SPACE)
      ; Set the linefeed distance to be one point more than the font
      ; height
    (freplace IPLINEFEED of IPDATA with (IDIFFERENCE (\IPC (IMINUS (IQUOTIENT MICASPERINCH POINTSPERINCH))
      (FONTPROP FONT 'HEIGHT]
    (freplace NSTRANSTABLE of IPDATA with (ffetch OTHERDEVICEFONTPROPS of FONT))
  (\FIXLINELENGTH.IP IPSTREAM)
  (freplace IPMAXVISIBLEBASELINE of IPDATA with (- (ffetch IPVISTOP of IPDATA)
    (ffetch (FONTDESCRIPTOR \SFAscent) of FONT)))
  (freplace IPMINVISIBLEBASELINE of IPDATA with (+ (ffetch IPVISBOTTOM of IPDATA)
    (ffetch (FONTDESCRIPTOR \SFDescent) of FONT)))
  [replace IPCHARVISIBLEP of IPDATA with (AND (>= (ffetch IPXPOS of IPDATA)
    (ffetch IPVISLEFT of IPDATA))
    (>= (ffetch IPYPOS of IPDATA)
    (ffetch IPMINVISIBLEBASELINE of IPDATA))
    (<= (ffetch IPYPOS of IPDATA)
    (ffetch IPMAXVISIBLEBASELINE of IPDATA))
  (AND *INTERPRESS-PRINTER-DSPFONT-PATCH* (\MOVETO.IP IPSTREAM (ffetch IPXPOS of IPDATA)
    (ffetch IPYPOS of IPDATA)))
  (RETURN OLDFONT]))
```

(\DSPLEFTMARGIN.IP

```
[LAMBDA (IPSTREAM XPOSITION) ; (* rmk%: " 4-Oct-84 10:34")
  (PROG1 (ffetch IPLEFT of (ffetch IMAGEDATA of IPSTREAM))
    (COND
      (XPOSITION (freplace IPLEFT of (ffetch IMAGEDATA of IPSTREAM) with XPOSITION)
        (\FIXLINELENGTH.IP IPSTREAM))))]
```

(\DSPLINEFEED.IP

```
[LAMBDA (IPSTREAM DELTAY) ; (* rmk%: " 4-Oct-84 09:26")
  ; sets the amount that a line feed increases the y coordinate by.
  (PROG ((IPDATA (ffetch IMAGEDATA of IPSTREAM)))
    (RETURN (PROG1 (ffetch IPLINEFEED of IPDATA)
      [AND DELTAY (COND
        ((NUMBERP DELTAY)
          (freplace IPLINEFEED of IPDATA with DELTAY))
        (T (\ILLEGAL.ARG DELTAY)))]))
```

(\DSPRIGHTMARGIN.IP

```
[LAMBDA (IPSTREAM XPOSITION) ; Edited 11-Aug-88 15:44 by rmk:
  (LET ((IPDATA (ffetch IPDATA of IPSTREAM)))
    (PROG1 (ffetch IPRIGHT of IPDATA)
      (COND
        (XPOSITION (freplace IPRIGHT of IPDATA with XPOSITION)
          (freplace IPMINCHARRIGHT of IPDATA with (MIN (ffetch IPVISRIGHT of IPDATA)
            (ffetch IPRIGHT of IPDATA)))
          (\FIXLINELENGTH.IP IPSTREAM))))]
```

(\DSPSPACEFACTOR.IP

```
[LAMBDA (STREAM FACTOR) ; Edited 2-May-2023 09:01 by Imm
  ; Edited 23-Mar-88 21:04 by jds
  (PROG ((IPDATA (ffetch IMAGEDATA of STREAM)))
    (RETURN (PROG1 (ffetch IPSPACEFACTOR of IPDATA)
      [COND
        (FACTOR [freplace IPSPACEWIDTH of IPDATA
          with (FIXR (TIMES FACTOR (CHARWIDTH (CHARCODE SPACE)
            (ffetch IPFONT of IPDATA))
          ; Doing the multiply first will insure that FACTOR is a number
          (freplace IPSPACEFACTOR of IPDATA with FACTOR)
          (SHOW.IP STREAM)
          (APPENDNUMBER.IP STREAM FACTOR)
          (SET.IP STREAM (\IPC AMPLIFYSPACE]))]
```

(\DSPTOPMARGIN.IP

```
[LAMBDA (IPSTREAM YPOSITION)
  (PROG1 (fetch IPTOP of (fetch IMAGEDATA of IPSTREAM))
    (COND
      (YPOSITION (replace IPTOP of (fetch IMAGEDATA of IPSTREAM) with YPOSITION))))])
```

(\* rmk%: "26-Jun-84 14:01")

(\DSPXPOSITION.IP

```
[LAMBDA (IPSTREAM XPOSITION)
  (PROG1 (fetch IPXPOS of (fetch IPDATA of IPSTREAM))
    (COND
      (XPOSITION (replace IPXPOS of (fetch IPDATA of IPSTREAM) with XPOSITION))))])
```

(\* jds "14-Feb-86 12:13")

;;; DSPXPOSITION method for interpress streams

```
(PROG1 (fetch IPXPOS of (fetch IPDATA of IPSTREAM))
  (COND
    ([AND XPOSITION (NOT (EQP XPOSITION (fetch IPXPOS of (fetch IPDATA of IPSTREAM)
      (SHOW.IP IPSTREAM T) (* (SETX.IP IPSTREAM XPOSITION))
      ;; Until our view of the printer's position is accurate, we can't rely on what we think the Xposition is, hence must be sure not to do a
      ;; SETXREL.
      (SETXY.IP IPSTREAM XPOSITION (fetch IPYPOS of (fetch IPDATA of IPSTREAM)))]))])
```

(\DSPROTATE.IP

```
[LAMBDA (IPSTREAM ROTATION)
  (ROTATE.IP IPSTREAM ROTATION)
  (CONCATT.IP IPSTREAM)]
```

(\* hdj "12-Nov-85 12:16")

(\PUSHSTATE.IP

```
[LAMBDA (IPSTREAM)
  (GETFRAMEVAR.IP IPSTREAM)
  (YVar# (GETFRAMEVAR.IP IPSTREAM))
  (State (IP-TOS IPSTREAM))
  (replace (IPSTATE XPOS) of State with XVar#)
  (replace (IPSTATE YPOS) of State with YVar#)
  (**)
  (GETCP.IP IPSTREAM)
  (FSET.IP IPSTREAM XVar#)
  (FSET.IP IPSTREAM YVar#)
  (**)
  (SHOW.IP IPSTREAM)
  (PUSH-IP-STACK IPSTREAM (create IPSTATE))
  (APPENDOP.IP IPSTREAM (\IPC DOSAVESIMPLEBODY))
  (APPENDOP.IP IPSTREAM (\IPC {}))
```

; Edited 2-May-2023 08:51 by Imm  
(\* hdj "3-Jan-86 11:10")

;;; push a new context onto the stack

(\POPSTATE.IP

```
[LAMBDA (IPSTREAM)
  (SHOW.IP IPSTREAM)
  (APPENDOP.IP IPSTREAM (\IPC {}))
  (POP-IP-STACK IPSTREAM)
  ;; restore X & Y pos
  (LET ((State (IP-TOS IPSTREAM))
    (FGET.IP IPSTREAM (fetch (IPSTATE XPOS) of State))
    (FGET.IP IPSTREAM (fetch (IPSTATE YPOS) of State))
    (APPENDOP.IP IPSTREAM (\IPC SETXY]))
```

; Edited 2-May-2023 08:52 by Imm  
(\* hdj "3-Jan-86 11:10")

;;; pop the current context

(\DEFAULTSTATE.IP

```
[LAMBDA (IPSTREAM)
  (SCALE.IP IPSTREAM 1)
  (ISET.IP IPSTREAM (\IPC CURRENTTRANS])
```

; Edited 2-May-2023 09:00 by Imm  
(\* hdj "30-Dec-85 17:18")

;;; establish meter coordinate system

(\DSPTRANSLATE.IP

```
[LAMBDA (IPSTREAM Tx Ty)
  (TRANSLATE.IP IPSTREAM Tx Ty)
  (CONCATT.IP IPSTREAM)]
```

(\* hdj "12-Nov-85 12:22")

(\DSPSCALE2.IP

```
[LAMBDA (IPSTREAM Sx Sy)
(SCALE2.IP IPSTREAM Sx Sy)
(CONCATT.IP IPSTREAM)]
```

(\* hdj "12-Nov-85 12:23")

(\DSPYPOSITION.IP

```
[LAMBDA (IPSTREAM YPOSITION)
(PROG1 (fetch IPYPOS of (fetch IPDATA of IPSTREAM))
(COND
(YPOSITION (SHOW.IP IPSTREAM)
(SETY.IP IPSTREAM YPOSITION))))])
```

(\* rmk%: "18-Jun-84 14:14")

(\FILLCIRCLE.IP

```
[LAMBDA (STREAM CENTERX CENTERY RADIUS TEXTURE OPERATION)
```

; Edited 1-Feb-89 17:12 by FS

;; Interpress2.1 doesn't support ARCTO, so must either approximate a circle (as here), or scan convert it (e.g. CIRC SHADE.IP)
;; This code does not generate as nicely "round" circles as circshade.ip (the difference is visible to the naked eye). However, this code should be
;; better for landscape printing, for code which uses pushstate/popstate, and for printers which scan in the X direction (e.g. Fuji Xerox XP-9),
;; because it generates a simpler master.
;; Wimp out and display regular N-gon. For smaller circles, can use fewer points? Could also render two half circles (thus allowing twice the
;; number of points since there are two trajectories), but what the heck.
;; Note also the clipping code isn't integrated with this (nor TRAJECTORY.IP, or others).
(FILLNGON.IP STREAM 90 RADIUS CENTERX CENTERY TEXTURE OPERATION])

(\FILLPOLYGON.IP

```
[LAMBDA (STREAM POINTS TEXTURE OPERATION WINDNUMBER)
```

; Edited 2-May-2023 08:51 by Imm
; Edited 2-Feb-89 17:39 by FS

;;; INTERPRESS 2.1 (OSD) subset allows convex polygons. This routine not used in DIG due to convexity requirement, but provided for true interpress
;;; printers

```
(LET (NUMPATHS)
(APPENDOP.IP STREAM (\IPC DOSAVESIMPLEBODY)) ; push state (because change color)
(APPENDOP.IP STREAM (\IPC {}))
(SETCOLOR.IP STREAM TEXTURE OPERATION)
(if (LISTP (CAAR POINTS))
then ;; Multiple trajectories, put them out.
(SETQ NUMPATHS (LENGTH POINTS))
(for TRAJECTORY in POINTS do (TRAJECTORY.IP STREAM TRAJECTORY))
else (SETQ NUMPATHS 1)
(TRAJECTORY.IP STREAM POINTS))
(APPENDINTEGER.IP STREAM NUMPATHS)
(if (EQ WINDNUMBER 0)
then (APPENDOP.IP STREAM (\IPC MAKEOUTLINE))
else (APPENDOP.IP STREAM (\IPC MAKEOUTLINEODD)))
(APPENDOP.IP STREAM (\IPC MASKFILL))
(APPENDOP.IP STREAM (\IPC {})))
```

(\DRAWPOLYGON.IP

```
[LAMBDA (IPSTREAM POINTS CLOSED BRUSH DASHING)
```

; Edited 2-May-2023 08:00 by Imm
; Edited 8-Aug-88 15:11 by rmk:

;; draws a polygon on a interpress stream.
(COND
(DASHING ; do dashing with the generic function until dashing is added to
; interpress standard.
(\DRAWPOLYGON.GENERIC IPSTREAM POINTS CLOSED BRUSH DASHING))
(T ;; NEEDS TO WATCH OUT FOR MAX#SEGMENTS AND CLIPPING (SEE \IPCURVE2)
(PROG ((HALFWIDTH (FQUOTIENT (\WIDTHFROMBRUSH BRUSH (\IPC MICASPERPOINT)
2))
(IPDATA (fetch IMAGEDATA of IPSTREAM))
(SEG# 0)
IPXPOS IPYPOS LASTPT LEFT RIGHT BOTTOM TOP VISIBLEP PREVX PREVY)
; Arguments are assumed to be in micas.
(OR POINTS (RETURN))
(AND CLOSED (NULL (CDDR POINTS))
(SETQ CLOSED NIL)) ; Don't bother closing a straight line
(SETQ LEFT (+ (fetch IPVISLEFT of IPDATA)
HALFWIDTH))
(SETQ RIGHT (- (fetch IPVISRIGHT of IPDATA)
HALFWIDTH))
(SETQ BOTTOM (+ (fetch IPVISBOTTOM of IPDATA)
HALFWIDTH))
(SETQ TOP (- (fetch IPVISTOP of IPDATA)
HALFWIDTH))
(SETQ IPXPOS (fetch (POSITION XCOORD) of (CAR POINTS)))
(SETQ IPYPOS (fetch (POSITION YCOORD) of (CAR POINTS)))
(SETQ VISIBLEP (\VISIBLE.IP IPXPOS IPYPOS LEFT RIGHT TOP BOTTOM))



```

(SHOW.IP IPSTREAM)
(APPENDOP.IP IPSTREAM (\IPC DOSAVESIMPLEBODY))
(APPENDOP.IP IPSTREAM (\IPC {}))
(if VISIBLEP
  then (MOVETO.IP IPSTREAM IPXPOS IPYPOS))
(for PTS on (CDR POINTS)
  do (COND
    ((IGREATERP (add SEG# 1)
      MAXSEGSPERTRAJECTORY)
      ;; Our printers limit the number of segments in a single TRAJECTORY; make sure we respect their limitations
      (\IMAGEPATH.IP BRUSH IPSTREAM)
      (SETQ SEG# 0)
      (MOVETO.IP IPSTREAM IPXPOS IPYPOS)))
    (SETQ PREVX IPXPOS)
    (SETQ PREVY IPYPOS)
    (SETQ IPXPOS (fetch (POSITION XCOORD) of (CAR PTS)))
    (SETQ IPYPOS (fetch (POSITION YCOORD) of (CAR PTS)))
    (if VISIBLEP
      then (if (SETQ VISIBLEP (\VISIBLE.IP IPXPOS IPYPOS LEFT RIGHT TOP BOTTOM))
        then
          ; Super-common case: both ends visible, draw the line
          (LINETO.IP IPSTREAM IPXPOS IPYPOS)
        else
          ; Starts visible, goes out
          (\CLIPCURVELINE.IP PREVX PREVY IPXPOS IPYPOS LEFT RIGHT TOP BOTTOM T
            IPSTREAM)
          (\IMAGEPATH.IP BRUSH IPSTREAM)
          ; Curve is now invisible
          (SETQ SEG# 0))
        else (if (SETQ VISIBLEP (\VISIBLE.IP IPXPOS IPYPOS LEFT RIGHT TOP BOTTOM))
          then
            ; Starts invisible, comes in. MOVETO is done in
            ; \CLIPCURVELINE.IP
            (\CLIPCURVELINE.IP PREVX PREVY IPXPOS IPYPOS LEFT RIGHT TOP BOTTOM NIL
              IPSTREAM)
          else
            ; Both ends invisible, could be visible in middle
            (if (\CLIPCURVELINE.IP PREVX PREVY IPXPOS IPYPOS LEFT RIGHT TOP BOTTOM NIL
              IPSTREAM)
              then
                ; Drew a segment disconnected from rest of curve
                (\IMAGEPATH.IP BRUSH IPSTREAM)
                (SETQ SEG# 0)
                ; SEG# goes to 0 whenever we end up outside
                ))
            (if (AND CLOSED (NULL (CDR PTS)))
              then
                ; fake a return to the beginning to close
                (SETQ PTS (LIST NIL (CAR POINTS)))
                (SETQ CLOSED NIL)))
            (if VISIBLEP
              then (\SETBRUSH.IP IPSTREAM BRUSH)
                ; Only need to clean up if we're now inside
                ; FS- Unfortunately no OPER.
                (\IMAGEPATH.IP BRUSH IPSTREAM))
            (APPENDOP.IP IPSTREAM (\IPC {}))
            (SETXY.IP IPSTREAM IPXPOS IPYPOS))
  )

```

(\FIXLINELENGTH.IP

```

[LAMBDA (IPSTREAM) (* hdj "18-Oct-85 15:47")
  ;; IPSTREAM is known to be a stream of type interpress. Called by RIGHTMARGIN LEFTMARGIN and \SFFIXFONT to update the LINELENGTH
  ;; field in the stream. also called when the stream is created.
  (PROG (LLEN (IPDATA (ffetch IMAGEDATA of IPSTREAM)))
    (freplace (STREAM LINELENGTH) of IPSTREAM with (COND
      ((IGREATERP [SETQ LLEN
        (FIXR (QUOTIENT (DIFFERENCE (ffetch IPRIGHT
          of IPDATA)
            (ffetch IPLEFT
              of IPDATA))
          (ffetch FONTAVGCHARWIDTH
            of (ffetch IPFONT
              of IPDATA])
          1)
        LLEN)
        (T 10)]

```

(\MOVETO.IP

```

[LAMBDA (IPSTREAM X Y) (* jds "11-Feb-86 14:47")

```

;;; Do MOVETO for interpress streams

```

(SHOW.IP IPSTREAM T) ; First, close out what we had been doing.
(SETXY.IP IPSTREAM X Y)

```

(\SETBRUSH.IP

```

[LAMBDA (IPSTREAM BRUSH OPERATION) ; Edited 2-May-2023 08:03 by Imm
  ; Edited 6-Aug-88 13:17 by rmk:

```

;; Sets the stroke shape parameters.

;; FS: I modified this function to simply call SETCOLOR.IP, since its probably the "right" thing to do. This function also should set the Operation, since e.g. \Drawline.ip never uses Operation and this is the place to do it.

```
(PROG (WIDTH SHAPE COLOR)
 [COND
  [(LISTP BRUSH)
   (SETQ SHAPE (CAR BRUSH))
   (SETQ WIDTH (OR (CAR (LISTP (CDR BRUSH)))
                   (\IPC MICASPERPOINT)
                   (T (SETQ SHAPE 'ROUND)
                      (SETQ WIDTH (OR BRUSH (\IPC MICASPERPOINT)
                                       (\APPENDNUMBER.IP IPSTREAM WIDTH)
                                       (\ISET.IP IPSTREAM (\IPC STROKEWIDTH))
                                       (\APPENDNUMBER.IP IPSTREAM (SELECTQ SHAPE
                                                             (ROUND (\IPC ROUND))
                                                             (SQUARE (\IPC SQUARE))
                                                             (BUTT (\IPC BUTT))
                                                             (\IPC ROUND))))
                      (ISET.IP IPSTREAM (\IPC STROKEEND))
                      ))
   ;; This was the old code here, new code is below.
   ;; (if (AND (SETQ COLOR (fetch (BRUSH BRUSHCOLOR) of BRUSH)) (STREAMPROP IPSTREAM 'COLOR)) then ; set the color (SETQ RGB
   ;; (ENSURE.RGB COLOR) (SETRGB.IP IPSTREAM (CAR RGB) (CADR RGB) (CADDR RGB)))
   (SETQ COLOR (fetch (BRUSH BRUSHCOLOR) of BRUSH))
   ;; If no color provided, presumably a previous routine has set the DSPCOLOR.
   (if COLOR
    then (if (AND (NUMBERP COLOR)
                  (<= 0 COLOR))
             then ;; Avoid the conflict between textures and color numbers, for positive integers
              NIL
             else (SETCOLOR.IP IPSTREAM COLOR OPERATION))
    ))
```

(\STRINGWIDTH.IP

```
[LAMBDA (STREAM STRING RDTBL) ; (* rmk%: "12-Apr-85 09:39")
 ;; Returns the width of STRING in the interpress STREAM, observing spacefactor
 (\STRINGWIDTH.GENERIC STRING (ffetch IPFONT of (ffetch IMAGEDATA of STREAM))
  RDTBL
  (ffetch IPSPACEWIDTH of (ffetch IMAGEDATA of STREAM]))
```

(\DSPCLIPPINGREGION.IP

```
[LAMBDA (STREAM REGION) ; Edited 21-Sep-88 21:20 by jds
 ;; Fetches and sets the clipping region field rather than the page region. Setting the clipping region also changes the visible region.
 (LET ((IPDATA (fetch (STREAM IMAGEDATA) of STREAM)))
  (PROG1 (create REGION using (fetch (INTERPRESSDATA IPClippingRegion) of IPDATA))
   (AND REGION (UNINTERRUPTABLY
              (replace (INTERPRESSDATA IPClippingRegion) of IPDATA with REGION)
              (\CHANGE-VISIBLE-REGION.IP IPDATA REGION)
              ; Changed to NOT intersect it with the notional page frame, since
              ; that's not yet well-defined (you can't yet tell if you're printing
              ; landscape, e.g.)
              ;; OLD CODE: (\CHANGE-VISIBLE-REGION.IP IPDATA (INTERSECTREGIONS REGION (fetch
              ;; (INTERPRESSDATA IPPAGEFRAME) of IPDATA)))
              )))
```

(\DSOPERATION.IP

```
[LAMBDA (IPSTREAM OPERATION) ; (* rrb " 6-Mar-86 16:16")
 ; sets the operation field of a interpress stream
 (PROG ((IPDATA (ffetch IMAGEDATA of IPSTREAM)))
  (RETURN (PROG1 (ffetch (INTERPRESSDATA IPOPERATION) of IPDATA)
                [AND OPERATION (COND
                  ((FMEMB OPERATION '(PAINT REPLACE INVERT ERASE))
                   (freplace (INTERPRESSDATA IPOPERATION) of IPDATA with OPERATION))
                  (T (\ILLEGAL.ARG OPERATION]))]))
```

;; Patch controller for the "Bonnet" printer bug that loses X,Y position when you do a DSPFONT

```
(RPAQ? *INTERPRESS-PRINTER-DSPFONT-PATCH* NIL)
```

;; image state

```
(DEFINEQ
```

(\IP-TOS

```
[LAMBDA (IPSTREAM) ; (* hdj "30-Dec-85 17:30")
 (LET [(STACK (STREAMPROP IPSTREAM 'STACK))
      (if STACK
       then (CAR STACK))
```

else (ERROR "Stack is empty" IPSTREAM)]

(POP-IP-STACK

[LAMBDA (IPSTREAM) (\* hdj "30-Dec-85 17:30")
(LET [(STACK (STREAMPROP IPSTREAM 'STACK)
(if STACK
then (STREAMPROP IPSTREAM 'STACK (CDR STACK))
else (ERROR "Stack is empty" IPSTREAM))

(PUSH-IP-STACK

[LAMBDA (IPSTREAM OBJECT) (\* hdj "30-Dec-85 17:31")
(STREAMPROP IPSTREAM 'STACK (CONS OBJECT (STREAMPROP IPSTREAM 'STACK))

(DECLARE%: EVAL@COMPILE
(RECORD IPSTATE (XPOS YPOS))
(DEFINEQ

(\CREATECHARSET.IP

[LAMBDA (FAMILY PSIZE FACE ROTATION DEVICE CHARSET FONTDESC NOSLUG?)
; Edited 8-Apr-88 09:54 by jds

::: Build the CHARSETINFO for an Interpress NS font. If we can't find widths info for that font, return NIL
::: 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 INTERPRESSFONTDIRECTORIES \ASCIITONS))
(RESETLST ; RESETLST to make sure the fontfiles get closed
(PROG (WFILE WSTRM FIXEDFLAGS RELFLAG FIRSTCHAR LASTCHAR TEM WIDTHS WIDTHSY FBBOX CHARSETHEIGHT
(NSMICASIZE (FIXR (FQUOTIENT (ITIMES PSIZE 2540)
72)))
(CSINFO (create CHARSETINFO)))
(SETQ WIDTHS (fetch (CHARSETINFO WIDTHS) of CSINFO))
[COND
((SETQ WFILE (\FINDFONTFILE FAMILY PSIZE FACE NIL NIL CHARSET INTERPRESSFONTDIRECTORIES
INTERPRESSFONTEXTENSIONS))

::: Look thru INTERPRESSFONTDIRECTORIES for a file that describes the font requested. Only continue if we can find one.

[RESETSAVE (SETQ WSTRM (OPENSTREAM WFILE 'INPUT 'OLD))
' (PROGN (CLOSEF? OLDVALUE)
[COND
((RANDACCESSP WSTRM)
(SETFILEPTR WSTRM 0))
(T (COPYBYTES WSTRM (SETQ WSTRM (OPENSTREAM '{NODIRCORE} 'BOTH 'NEW)
(SETQ RELFLAG (\POSITIONFONTFILE WSTRM NSMICASIZE FIRSTCHAR LASTCHAR NIL))
::: \POSITIONFONTFILE sets FIRSTCHAR LASTCHAR as well as positioning the font file at the beginning of the widths
::: Fill in the widths, and return a flag telling whether the widths are absolute, or are type-size relative. 0 => relative
)
(T ; Can't find a file to describe this font;
(RETURN (COND
(NOSLUG? ; the caller just wants NIL back to signal that nothing was found
NIL)
(T (\BUILDSLUGCSINFO (fetch (FONTDESCRIPTOR FONTAVGCHARWIDTH) of FONTDESC)
(FONTPROP FONTDESC 'ASCENT)
(FONTPROP FONTDESC 'DESCENT)
(FONTPROP FONTDESC 'DEVICE)
(SETQ RELFLAG (ZEROP RELFLAG)) ; Convert the flag to a logical value
(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 CHARSETHASCENT) of CSINFO with (IMINUS (SIGNED (\WIN WSTRM)
BITSPERWORD))) ; Descent is -FBBOY
; replace (FONTDESCRIPTOR FBBDX) of FD with (SIGNED
; (\WIN WSTRM) BITSPERWORD)
; And the standard kern value (?)
(\WIN WSTRM)
(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 FBDX) of FD with (IQUOTIENT (ITIMES (fetch (FONTDESCRIPTOR FBDX) of FD)
;; NSMICASIZE) 1000)
(SETQ CHARSETHEIGHT (IQUOTIENT (ITIMES CHARSETHEIGHT NSMICASIZE)
                                1000])
(replace (CHARSETINFO CHARSETHEIGHT) 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)
          (IMIN (UNFOLD (ADD1 (IDIFFERENCE LASTCHAR FIRSTCHAR))
                      BYTESPERWORD)
                (IDIFFERENCE (GETFILEINFO WSTRM 'LENGTH)
                              (GETFILEPTR WSTRM)) ; Read the X widths.
                (EQ noInfoCode (\FGETWIDTH WIDTHS I))
                ) ; 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
                (SETQ WIDTHSY 0)))
             ; STAR FONT FILES LIKE TO LEAVE OFF THE Y WIDTH.
             ; 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]
             ; 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))))))

```

;; Called when the character set information cached in a display stream doesn't correspond to CHARSET

```
(PROG* ((FONT (ffetch IPFONT of IPDATA))
        (CSINFO (\GETCHARSETINFO CHARSET FONT))))
```

;; since the call to \getcharsetinfo has NOSLUG? = NIL, we know that we will get a reasonable character set back

```
(UNINTERRUPTABLY
  (freplace IPWIDTHSCACHE of IPDATA with (ffetch (CHARSETINFO WIDTHS) of CSINFO))
  (freplace NSCHARSET of IPDATA with CHARSET)))
```

)

(DEFINEQ

(INTERPRESSINIT

[LAMBDA NIL

; Edited 2-May-2023 09:14 by Imm
; Edited 9-Dec-88 11:49 by jds

```
(DECLARE (GLOBALVARS \IPIMAGEOPS \ASCIITONS \ASCIITOSTAR)
(SETQ \IPIMAGEOPS (create IMAGEOPS
```

```
  IMAGETYPE _ 'INTERPRESS
  IMCLOSEFN _ (FUNCTION \CLOSEIPSTREAM)
  IMXPOSITION _ (FUNCTION \DSPXPOSITION.IP)
  IMYPOSITION _ (FUNCTION \DSPYPOSITION.IP)
  IMFONT _ (FUNCTION \DSPFONT.IP)
  IMLEFTMARGIN _ (FUNCTION \DSPLEFTMARGIN.IP)
  IMRIGHTMARGIN _ (FUNCTION \DSPRIGHTMARGIN.IP)
  IMLINEFEED _ (FUNCTION \DSPLINEFEED.IP)
  IMDRAWLINE _ (FUNCTION \DRAWLINE.IP)
  IMDRAWCURVE _ (FUNCTION \DRAWCURVE.IP)
  IMDRAWCIRCLE _ (FUNCTION \DRAWCIRCLE.GENERIC)
  IMDRAWELLIPSE _ (FUNCTION \DRAWELLIPSE.GENERIC)
  IMFILLCIRCLE _ (FUNCTION \CIRC SHADE.IP)
  IMBLTSHADE _ (FUNCTION \BLTSHADE.IP)
  IMBITBLT _ (FUNCTION \BITBLT.IP)
  IMNEWPAGE _ (FUNCTION \NEWPAGE.IP)
  IMMOVE _ (FUNCTION \MOVE.IP)
  IMSCALE _ [FUNCTION (LAMBDA NIL ; should this be a ratio instead of a float?
                    (\IPC (FQUOTIENT MICASPERINCH POINTSPERINCH)
                    IMTERPRI _ (FUNCTION \NEWLINE.IP)
  IMBOTTOMMARGIN _ (FUNCTION \DSPBOTTOMMARGIN.IP)
  IMTOPMARGIN _ (FUNCTION \DSPTOPMARGIN.IP)
  IMFONTCREATE _ 'INTERPRESS
  IMSPACEFACTOR _ (FUNCTION \DSPSPACEFACTOR.IP)
  IMCOLOR _ (FUNCTION \DSPCOLOR.IP)
  IMSTRINGWIDTH _ (FUNCTION \STRINGWIDTH.IP)
  IMCHARWIDTH _ (FUNCTION \CHARWIDTH.IP)
  IMSCALED BITBLT _ (FUNCTION \SCALED BITBLT.IP)
  IMCLIPPINGREGION _ (FUNCTION \DSPCLIPPINGREGION.IP)
  IMFILLPOLYGON _ (FUNCTION \POLY SHADE.IP)
  IMDRAWARC _ (FUNCTION \DRAWARC.IP)
  IMPUSHSTATE _ (FUNCTION \PUSHSTATE.IP)
  IMPOPSTATE _ (FUNCTION \POPSTATE.IP)
  IMROTATE _ (FUNCTION \DSPROTATE.IP)
  IMSCALE2 _ (FUNCTION \DSPSCALE2.IP)
  IMTRANSLATE _ (FUNCTION \DSPTRANSLATE.IP)
  IMDEFAULTSTATE _ (FUNCTION \DEFAULTSTATE.IP)
  IMOPERATION _ (FUNCTION \DSOPERATION.IP)
  IMBITMAPSIZE _ (FUNCTION \BITMAPSIZE.IP)
  IMDRAWPOLYGON _ (FUNCTION \DRAWPOLYGON.IP)
  IMDRAWPOINT _ (FUNCTION \DRAWPOINT.IP))])
```

;; FS: Removed left arrow mapping - ("\_" 0 172)

;; JDS: Removed old bullet mapping (183 239 102)

```
(LET [(MAPPINGS ' ("_" 33 62)
      ("^" 0 173)
      ("$" 0 164)
      ("^N" 0 197)
      ("^S" 239 37)
      ("^V" 239 36)
      ("^X" 0 45)
      ("^O" 239 45)
      ("^\ " 239 44)
      ("^Y" 239 46)
      ("^D" 0 200)
      ("^G" 0 169)
      ("^H" 0 161)
      ("^B" 0 191)
      (96 0 185)
      (155 239 36)
      (156 239 37)
      ("^^" 0 184])
```

;; Translation table for standard ascii to NS. Last 5 are backquote, en dash, em dash, bullet, and finally the "backward compatible" package
;; delimiter, rendered as the divide sign.

```
(SETQ \ASCIITONS (NSMAP NIL MAPPINGS))
```

; Map from ASCII to printer character code (XC1-1-1 NS
; Encoding standard)

```
(SETQ \ASCIITOSTAR (NSMAP NIL (CDR MAPPINGS)))
```

;; Map from ASCII to wedged OSD screen & .WD file character coding (alleged to be XC2-x-x, soon to come). The difference is that '-' maps  
 ;; to itself for width purposes.

```

)
NIL])
)
(DEFINEQ
(SCALEREGION
[LAMBDA (SCALE REGION)
(* rmk%: "21-JUL-82 13:06")
; Scales a region
(create REGION
LEFT _ (FIX (FTIMES SCALE (fetch (REGION LEFT) of REGION)))
BOTTOM _ (FIX (FTIMES SCALE (fetch (REGION BOTTOM) of REGION)))
WIDTH _ (FIX (FTIMES SCALE (fetch (REGION WIDTH) of REGION)))
HEIGHT _ (FIX (FTIMES SCALE (fetch (REGION HEIGHT) of REGION)))
)
)

```

```

(DECLARE%: DONTEVAL@LOAD DOCOPY
(RPAQ? \SPLINESTEP.IP 16.0)
)
(DECLARE%: DONTEVAL@LOAD DOCOPY
(RPAQ? IPPAGEREGION.ROT180 NIL)
(RPAQ? IPPAGEREGION.ROT270 NIL)
(RPAQ? DEFAULTPAGEREGION (SCALEREGION 2540 (CREATEREGION 1.1 0.75 (- 7.5 1.1)
(- 10.5 0.75))))
(RPAQ? DEFAULTLANDPAGEREGION (SCALEREGION 2540 (CREATEREGION 0.75 1.1 (- 10.5 0.75)
(- 7.5 1.1))))
)

```

;; Interpress encoding values

```

(DECLARE%: DONTCOPY
(DECLARE%: EVAL@COMPILE
(RPAQQ MAXSEGS PERTRAJECTORY 100)
(CONSTANTS MAXSEGS PERTRAJECTORY)
)
)
(DECLARE%: DONTCOPY
(DECLARE%: EVAL@COMPILE
(PUTPROPS APPENDBYTE.IP DMACRO (= . \BOUT))
(PUTPROPS APPENDOP.IP MACRO [OPENLAMBDA (STREAM OP)
(COND
((CONSTANT (OR (ILESSP OP 0)
(IGREATERP OP 8191)))
(ERROR "Invalid Interpress operator code:" OP)))
(COND
((CONSTANT (ILEQ OP 31))
(APPENDBYTE.IP STREAM (LOGOR (\IPC SHORTOP
OP)))
(T (APPENDBYTE.IP STREAM (LOGOR (\IPC LONGOP)
(FOLDLO OP 256)))
(APPENDBYTE.IP STREAM (MOD OP 256]))
)
)
(PUTPROPS .IPFONTNAME. DMACRO ((FAMILY)
(SELECTQ FAMILY
(TIMESROMAN 'CLASSIC)
(HELVETICA 'MODERN)
(LOGO 'LOGOTYPES)
(GACHA 'TERMINAL)
FAMILY)))
(PUTPROPS APPENDINT.IPMACRO MACRO [OPENLAMBDA (STREAM NUM LENGTH)
(for I from (SUB1 LENGTH) to 0 by -1
do (APPENDBYTE.IP STREAM (LOADBYTE NUM (UNFOLD I BITSPERBYTE)
BITSPERBYTE))]
)
(PUTPROPS APPENDINTEGER.IPMACRO MACRO [OPENLAMBDA (STREAM N)
(COND
((AND (ILEQ -4000 N)

```

```

      (ILEQ N 28767))
      (APPENDINT.IPMACRO STREAM (IPLUS N 4000)
      2))
      (T (PROG ((LEN (BYTESININT.IP N)))
      (APPENDSEQUENCEDESCRIPTOR.IP STREAM (\IPC SEQINTEGER)
      LEN)
      (APPENDINT.IP STREAM N LEN]))

```

```

(PUTPROPS \IMAGEPATH.IP MACRO ((BRUSH STREAM OPERATION)
(\SETBRUSH.IP IPSTREAM BRUSH OPERATION)
(MASKSTROKE.IP IPSTREAM)))

```

```

(PUTPROPS \WIDTHFROMBRUSH MACRO ((BRUSH DEFAULT) ; Extracts width from brush, defaulting to DEFAULT for
; unrecognized values
(COND
[(LISTP BRUSH)
(CAR (LISTP (CDR BRUSH)
(NUMBERP BRUSH)
BRUSH)
(T DEFAULT))]))

```

```

(PUTPROPS \VISIBLE.IP MACRO (OPENLAMBDA (X Y LEFT RIGHT TOP BOTTOM) ; T if the point X,Y is inside the specified region
(AND (IGEQ X LEFT)
(ILEQ X RIGHT)
(IGEQ Y BOTTOM)
(ILEQ Y TOP))))
)

```

```

(DECLARE%: EVAL@COMPILE

```

```

[RECORD IPSTREAM STREAM (SUBRECORD STREAM)
[ACCESSFNS ((IPDATA (fetch (STREAM IMAGEDATA) of DATUM)
(replace (STREAM IMAGEDATA) of DATUM with NEWVALUE))
(SHOWSTREAM (fetch (IPSTREAM IPDATA) of DATUM)
(replace (IPSTREAM IPDATA) of DATUM with NEWVALUE]
(TYPE? (type? INTERPRESSDATA of (fetch (STREAM IMAGEDATA) of DATUM]

```

```

(DATATYPE INTERPRESSDATA (IPHEADING IPHEADINGFONT (IPXPOS POINTER)
(IPYPOS POINTER)
IPFONT IPPREAMBLEFONTS IPPAGEFONTS IPWIDTHSCACHE IPCOLOR (IPLINEFEED POINTER)
IPPAGESTATE IPSHOWSTREAM IPPAGEREGION IPDOCNAME (IPLLEFT POINTER)
(IPBOTTOM POINTER)
(IPRIGHT POINTER)
(IPTOP POINTER)
(IPPAGENUM WORD)
(IPPREAMBLENEXTFRAMEVAR BYTE)
(IPNEXTFRAMEVAR BYTE)
(IPHEADINGOPVAR BYTE)
(NSCHARSET BYTE)
(NSTRANSTABLE POINTER)
(IPCORRECTSTARTX POINTER ; Used with IPXPOS to compute width for CORRECTing char
; strings during SHOW.
)
(IPSPACEFACTOR POINTER)
(IPSPACEWIDTH POINTER) ; cached width of space, taking space factor into account
(IPROTATION POINTER) ; Default rotation in which this document is to be printed: Set up
; with ROTATE and CONCATT at the start of each new page.
(IPXOFFSET POINTER) ; Default X offset, akin to the rotation. Used to do landscape
; printing
(IPYOFFSET POINTER) ; Default Y offset.
(IPClippingRegion POINTER) ; Clipping region, intersected with pageframe to determine the
; visible region
(IPCOLORMODEL WORD) ; preamble fvar in which we have stored the color model we are
; using (for post-IP 2.1 ONLY)
(IPOPERATION POINTER) ; used to keep the current operation mode PAINT, REPLACE,
; ERASE or INVERT.
(IPVISLEFT POINTER) ; Boundaries of stream's visible region, namely, the intersection
; of the clipping region and the page frame
(IPVISRIGHT POINTER)
(IPVISTOP POINTER)
(IPVISBOTTOM POINTER)
(IPPAGEFRAME POINTER) ; The physical page size as a mica region, can't be changed in
; midstream. Used to determine the visible region
(IPMAXVISIBLEBASELINE POINTER) ; The cached maximum character baseline for the current visible
; page region
(IPMINVISIBLEBASELINE POINTER) ; The cached minimum character baseline for the current visible
; page region
(IPVISIBLEREGION POINTER) ; Region corresponding to IPVISLEFT etc., to be passed to
; clipping functions
(IPCHARVISIBLEP POINTER) ; True if current pos is inside character clipping region, reset
; when X,Y is changed or font is changed
(IPMINCHARRIGHT POINTER) ; Min of right margin and clipping right, special tests needed only
; if new position is beyond this. Reset when margin or clipping
; region is changed
(IPCLIPINCLUSIVE POINTER) ; True if page should include characters that cross the right or
; bottom edges of the clipping region

```

```

)
IPXPOS _ 0 IPYPOS _ 0 IPNEXTFRAMEVAR _ 0 IPSPACEFACTOR _ 1 IPROTATION _ 0 IPXOFFSET _ 0 IPYOFFSET _ 0
IPCOLORMODEL _ 0 IPOPERATION _ 'PAINT IPCLIPINCLUSIVE _ NIL)
)
(/DECLAREDATATYPE 'INTERPRESSDATA
' (POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
  POINTER POINTER POINTER POINTER POINTER POINTER WORD BYTE BYTE BYTE BYTE POINTER POINTER POINTER POINTER
  POINTER POINTER POINTER POINTER WORD POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
  POINTER POINTER POINTER POINTER POINTER)
;; ---field descriptor list elided by lister---
' 80)
)
(/DECLAREDATATYPE 'INTERPRESSDATA
' (POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
  POINTER POINTER POINTER POINTER POINTER POINTER WORD BYTE BYTE BYTE BYTE POINTER POINTER POINTER POINTER
  POINTER POINTER POINTER POINTER WORD POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
  POINTER POINTER POINTER POINTER POINTER)
;; ---field descriptor list elided by lister---
' 80)
)

```

(DEFINEQ

**(INTERPRESSBITMAP**

```

[LAMBDA (OUTPUTFILE BITMAP SCALEFACTOR REGION ROTATION TITLE) ; Edited 2-May-2023 15:19 by Imm
; Edited 14-Jan-88 02:08 by FS
; Print a bitmap into an IP file

(PROG (IPSTREAM W H)
  (SETQ IPSTREAM (OPENIMAGESTREAM (OR OUTPUTFILE '{SCRATCH}IPBITMAP.SCRATCH)
    'INTERPRESS))
  [SETQ W (COND
    (REGION (fetch (REGION WIDTH) of REGION))
    (T (fetch (BITMAP BITMAPWIDTH) of BITMAP))
  [SETQ H (COND
    (REGION (fetch (REGION HEIGHT) of REGION))
    (T (fetch (BITMAP BITMAPHEIGHT) of BITMAP))
  (COND
    (TITLE (RELMOVETO (IDIFFERENCE (TIMES 4 MICASPERINCH)
      (STRINGWIDTH TITLE IPSTREAM))
      0 IPSTREAM)
      (PRIN1 TITLE IPSTREAM))) ; Try to center around within the pageframe margins
  [COND
    (SCALEFACTOR (SETQ W (TIMES W SCALEFACTOR))
      (SETQ H (TIMES H SCALEFACTOR))
  ;; These transformations are wrong!
  (SELECTQ (SETQ ROTATION (IMOD (OR ROTATION (IPC DEFAULT.INTERPRESS.BITMAP.ROTATION))
    360))
    (0 (SETQ W (- W))
      (SETQ H (- H)))
    (180)
    (90 (SETQ H (PROG1 (- W)
      (SETQ W H))))
    (270 (SETQ W (PROG1 (- H)
      (SETQ H W))))
    (ERROR ROTATION "rotation by other than multiples of 90 degrees not implemented"))
  [MOVETO.IP IPSTREAM [+ (TIMES MICASPERINCH 4.25)
    (TIMES W (CONSTANT (FQUOTIENT 635 36)
      (+ (TIMES MICASPERINCH 5.5)
        (TIMES H (CONSTANT (FQUOTIENT 635 36)
  ;; Position so that the bitmap's image is centered on the paper ((635 / 36) = half the micas in a point)
  (SHOWBITMAP.IP IPSTREAM BITMAP REGION SCALEFACTOR ROTATION)
  (RETURN (CLOSEF IPSTREAM))
)
)

```

```

(ADDTOVAR IMAGESTREAMTYPES (INTERPRESS (OPENSTREAM OPENIPSTREAM)
  (FONTCREATE \CREATEINTERPRESSFONT)
  (FONTSAVAILABLE \SEARCHINTERPRESSFONTS)
  (CREATECHARSET \CREATECHARSET.IP)))
)

```

;; HOSTNAMEP is NIL for DOCUPRINT instead of NSPRINTER.HOSTNAMEP, since that predicate merely tests for colon in the name. DOCUPRINT  
 ;; printers are only recognized from their PRINTERTYPE property, which must be on their CANONICAL.HOSTNAME. Preference is for INTERPRESS  
 ;; (CANPRNT ordering), for backward compatibility. But printer can be put on DEFAULTPRINTINGHOST twice, with the type CONSed on to the name,  
 ;; to give the user dynamic selection.

```

(ADDTOVAR PRINTERTYPES
  ((DOCUPRINT)
  (CANPRINT (INTERPRESS POSTSCRIPT))
  (HOSTNAMEP NIL)
  (STATUS NSPRINTER.STATUS)
  (PROPERTIES NSPRINTER.PROPERTIES)
)
)

```



```
(SEND NSPRINT)
(BITMAPSCALE INTERPRESS.BITMAPSCALE)
(BITMAPFILE (INTERPRESSBITMAP FILE BITMAP SCALEFACTOR REGION ROTATION TITLE)))
((INTERPRESS 8044)
(CANPRINT (INTERPRESS))
(HOSTNAMEP NSPRINTER.HOSTNAMEP)
(STATUS NSPRINTER.STATUS)
(PROPERTIES NSPRINTER.PROPERTIES)
(SEND NSPRINT)
(BITMAPSCALE INTERPRESS.BITMAPSCALE)
(BITMAPFILE (INTERPRESSBITMAP FILE BITMAP SCALEFACTOR REGION ROTATION TITLE))))
```

```
(ADDTOVAR PRINTFILETYPES (INTERPRESS (TEST INTERPRESSFILEP)
(EXTENSION (IP IPR INTERPRESS))
(CONVERSION (TEXT MAKEINTERPRESS TEDIT \TEDIT.HARDCOPY))))
```

```
(RPAQ? DEFAULT.INTERPRESS.BITMAP.ROTATION 90)
```

```
(ADDTOVAR SYSTEMINITVARS (INTERPRESSFONTDIRECTORIES {DSK}))
```

```
(RPAQ? INTERPRESSFONTEXTENSIONS ' (WD))
```

```
(RPAQ? INTERPRESSFONTDIRECTORIES ' ("{Erinyes}<Lyric>Fonts>"))
```

```
(RPAQ? INTERPRESSPRINTWHEELFAMILIES ' (BOLDPS ELITE LETTERGOTHIC MASTER PICA PSBOLD SCIENTIFIC SPOKESMAN TITAN
TREND TRENDPS TROJAN VINTAGE))
```

```
(RPAQ? INTERPRESSFAMILYALIASES ' (LOGO LOGOTYPES-XEROX))
```

:: NS Character Encoding

```
(DEFINEQ
```

(NSMAP

```
[LAMBDA (ZERODEFAULT MAP) (* bvm%: "23-Oct-86 12:52")
(LET ((TABLE (ARRAY 256 'WORD 0 0)))
(OR ZERODEFAULT (for I from 0 to 255 do (SETA TABLE I I)))
[for X in MAP do (SETA TABLE (OR (FIXP (CAR X))
(CHARCODE.DECODE (CAR X)))
(LOGOR (LLSH (CADR X)
8)
(CADDR X)
TABLE])])
```

(COERCEASCIIITONSFONT

```
[LAMBDA (ASCIIITONSMAPARRAY ASCIIITONSFIXARRAY ASCIIIFAMILY NSFAMILY SIZE FONTFACE ROTATION DEVICE)
(* gbn "12-Sep-85 15:10")
```

:: Produces an ascii font with the proper widths for the ns-character correspondences defined by ASCIIITONSMAPARRAY  
:: ASCIIITONSFIXARRAY is for temporary problems with font compatibility between printer and widths/screen. in OS5.0 fonts

```
(PROG (CHARSETDIR [ASCIIITONSMAP (fetch (ARRAYP BASE) of (\DTEST (OR ASCIIITONSFIXARRAY ASCIIITONSMAPARRAY)
'ARRAYP)
(FD (\CREATESTARFONT NSFAMILY SIZE FONTFACE ROTATION DEVICE)))
(OR FD (RETURN NIL))
[SETQ CHARSETDIR (CONS 0 (\GETCHARSETINFO 0 FD)
[bind NSCODE CS for I from 0 to 255 unless (OR (EQ I (SETQ NSCODE (\GETBASE ASCIIITONSMAP I)))
(ASSOC (SETQ CS (\CHARSET NSCODE))
CHARSETDIR))
do ; Run thru the translate table looking for non-0 charsets. Add
; their width info to the directory
(push CHARSETDIR (CONS CS (COND
((\GETCHARSETINFO CS FD))
(T ; There isn't any info for that character. Warn the guy, but
; continue.
(FRESHLINE PROMPTWINDOW)
(printout PROMPTWINDOW "Warning: Information about character
set ".I3.8 CS " missing from font " ASCIIIFAMILY %, SIZE
".")
NIL] ; Return if one of the fonts couldn't be found
[bind CHARSETINFO NSCODE (WIDTHS _ (fetch (CHARSETINFO WIDTHS) of (\GETCHARSETINFO 0 FD))) for I
from 0 to 255 unless (EQ I (SETQ NSCODE (\GETBASE ASCIIITONSMAP I)))
when (SETQ CHARSETINFO (CDR (ASSOC (\CHARSET NSCODE)
CHARSETDIR)))
do ; For each non-ASCII character, look for width info in the right NS
; place. If none, use zero width.
(\FSETWIDTH WIDTHS I (\FGETWIDTH (fetch (CHARSETINFO WIDTHS) of CHARSETINFO)
(\CHAR8CODE NSCODE)
[replace OTHERDEVICEFONTPROPS of FD with (fetch (ARRAYP BASE) of (\DTEST ASCIIITONSMAPARRAY 'ARRAYP)
[COND
((NEQ NSFAMILY ASCIIIFAMILY)
:: Update the font decriptor so it looks like it's really for the family the guy wanted. Also save the info we used to get here.
(replace FONTFAMILY of FD with ASCIIIFAMILY)
(replace FONTDEVICESPEC of FD with (LIST NSFAMILY SIZE FONTFACE ROTATION DEVICE)
(RETURN FD])])
```

(\CREATEINTERPRESSFONT

[LAMBDA (FAMILY SIZE FONTFACE ROTATION DEVICE)

; Edited 17-Feb-87 16:49 by FS

:: Creates a font descriptor for an NS font for hardcopy. Tries first on the assumption that he gave us the NS font name;

(\DECLARE (GLOBALVARS \ASCIITONS \ASCIITOSTAR \ASCIITONSTRANSLATIONS))

:: Test removal of \ASCIITOSTAR from \COERCEASCIITONSFONT, forces use of \ASCIITONS

(if (\COERCEASCIITONSFONT \ASCIITONS NIL FAMILY FAMILY SIZE FONTFACE ROTATION DEVICE)

elseif (for TRANSL in \ASCIITONSTRANSLATIONS bind NEWFONT when (AND (EQ FAMILY (CAR TRANSL))

(SETQ NEWFONT

(\COERCEASCIITONSFONT

(COND

((NULL (CADR TRANSL))

\ASCIITONS)

((LITATOM (CADR TRANSL))

(EVAL (CADR TRANSL)))

(T (CADR TRANSL)))

(COND

((NULL (CADR TRANSL))

\ASCIITOSTAR)

(T NIL))

FAMILY

(OR (CADDR TRANSL)

'MODERN)

SIZE FONTFACE ROTATION DEVICE)))

do (RETURN NEWFONT))

(\SEARCHINTERPRESSFONTS

[LAMBDA (FAMILY PSIZE FACE ROTATION)

; Edited 2-Jan-87 17:07 by FS

(\DECLARE (GLOBALVARS INTERPRESSFONTDIRECTORIES INTERPRESSFONTEXTENSIONS)

(\SEARCHFONTFILES FAMILY PSIZE FACE ROTATION 'INTERPRESS INTERPRESSFONTDIRECTORIES INTERPRESSFONTEXTENSIONS]

)

)

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(RPAQQ noInfoCode 32768)

(CONSTANTS (noInfoCode 32768))

)

(RPAQ? ASCIITONSTRANSLATIONS )

:: These are in priority order: if an early one doesn't find a font for a family, the later ones are tried (essentially going to MODERN as the default).

(ADDTOVAR ASCIITONSTRANSLATIONS

(TIMESROMAN NIL CLASSIC)

(GACHA NIL TERMINAL)

(HELVETICA NIL MODERN)

(CLASSIC NIL MODERN)

(GACHA NIL MODERN)

(TIMESROMAN NIL MODERN)

(LOGO NIL LOGOTYPES)

(HIPPO HIPPOTONSARRAY CLASSIC)

(CYRILLIC CYRILLICTONSARRAY CLASSIC)

(SYMBOL \SYMBOLTONSARRAY MODERN)

(MATH \MATHTONSARRAY CLASSIC))

(READVARS-FROM-STRINGS '(\SYMBOLTONSARRAY HIPPOTONSARRAY CYRILLICTONSARRAY \MATHTONSARRAY)

"({Y256 SMALLPOSP 0 0 0 180 8546 0 8574 177 61309 61282 61283 61284 61285 0 184 0 0 61296 61298 61273 61272 8549 8550 0 0 61054 61305 61275 61274 8546 61299 0 0 0 174 173 175 61266 61250 61251 61303 61261 61263 0 0 61262 {R4 0} 8551 61258 61259 61281 0 61292 172 61365 61364 61290 61351 0 0 0 47 0 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 61271 61270 0 61366 61367 61238 61239 61362 61363 61360 61361 123 125 61234 61235 61052 8514 61243 61242 8740 8742 61308 8546 0 61301 {R4 0} 167 61232 61233 182 64 211 163 36 {R128 0} } {Y256 SMALLPOSP 0 0 1 2 3 4 5 6 7 8 9 10 11 12 13 60973 61229 16 17 18 61221 20 21 61220 23 60973 61228 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 9793 9794 9809 9797 9798 9818 9796 9802 9804 74 9805 9806 9807 9808 9810 9811 9803 9813 9814 9816 9817 86 9821 9819 9820 9801 91 92 93 173 172 96 9825 9826 9841 9829 9830 9850 9828 9834 9836 106 9837 9838 9839 9840 9842 9843 9835 9845 9846 9848 9849 118 9853 9851 9852 9833 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 61220 61221 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 61286 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 } {Y256 SMALLPOSP 0 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 10023 37 38 39 40 41 10041 43 44 8510 46 47 48 49 10095 51 10071 53 10088 55 10089 57 58 59 171 61 187 63 10047 10017 10018 10046 10021 10022 10038 10020 10049 10026 10027 10028 10029 10030 10031 10032 10033 10039 10034 10035 10036 10037 10019 10024 10045 10048 10025 10090 9984 10091 10044 10092 9984 10065 10066 10110 10069 10070 10086 10068 10097 10074 10075 10076 10077 10078 10079 10080 10081 10087 10082 10083

```

10084 10085 10067 10072 10093 10096 10073 10042 9984 10043 10040 9984 128 129 130 131 132 133 134 135
136 137 138 139 140 141 142 10094 144 145 146 147 148 149 150 151 152 153 154 61220 61221 157 158 159
160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 61286 184
185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209
210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234
235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 } {Y256 SMALLPOSP
 0 0 61307 61234 61235 0 163 61301 61302 0 0 0 182 0 0 0 61286 0 0 0 61306 0 0 61295 {R9 0} 32 61232
61287 8551 162 184 61366 61299 194 61308 199 177 61260 61309 8552 61285 61287 8738 8740 8574 61282
61283 61284 61292 8570 199 167 0 8549 8546 8550 191 61248 61365 61258 61356 61369 61364 61233 61275
61279 61273 61274 61278 61272 61629 61259 61281 61297 61265 61358 61305 61296 61271 61367 61298 180
61626 61368 0 0 0 175 174 0 61351 61267 211 61370 61303 61266 61263 61288 61360 61361 61362 61363
61256 61290 61287 61238 61240 210 61246 61244 61247 61245 61250 61251 61270 61239 188 189 190 61264 {R
129 0} }}
")

```

(DECLARE%: DONTEVAL@LOAD DOCOPY

(\INTERPRESSINIT)  
)

(DECLARE%: EVAL@COMPILE DONTCOPY

(LOADDEF 'SYSTEMBRUSH 'RESOURCES 'IMAGEIO)

(LOADDEF 'BRUSH 'RECORDS 'IMAGEIO)  
)

FUNCTION INDEX

APPENDBYTE.IP .....5 LINETO.IP .....10 \CHARWIDTH.IP .....23
APPENDIDENTIFIER.IP .....5 MAKEINTERPRESS .....17 \CLIPCURVELINE.IP .....27
APPENDINT.IP .....5 MASKSTROKE.IP .....10 \CLIPLINE .....28
APPENDINTEGER.IP .....5 MOVETO.IP .....10 \CLOSEIPSTREAM .....23
APPENDLARGEVECTOR.IP .....6 NEWLINE.IP .....18 \COERCEASCIITONSFONT .....41
APPENDNUMBER.IP .....6 NEWPAGE.IP .....18 \CREATECHARSET.IP .....35
APPENDOP.IP .....6 NEWPAGE?.IP .....19 \CREATEINTERPRESSFONT .....42
APPENDRATIONAL.IP .....6 NSMAP .....41 \DEFAULTSTATE.IP .....31
APPENDSEQUENCEDESCRIPTOR.IP .....6 OPENIPSTREAM .....19 \DRAWARC.IP .....24
ARCTO.IP .....7 POP-IP-STACK .....35 \DRAWCURVE.IP .....24
BEGINMASTER.IP .....7 PUSH-IP-STACK .....35 \DRAWLINE.IP .....28
BEGINPAGE.IP .....7 ROTATE.IP .....10 \DRAWPOINT.IP .....24
BEGINPREAMBLE.IP .....7 SCALE.IP .....10 \DRAWPOLYGON.IP .....32
BYTESININT.IP .....6 SCALE2.IP .....10 \DSPBOTTOMMARGIN.IP .....29
CLIPRECTANGLE.IP .....7 SCALEREGION .....38 \DSPCLIPPINGREGION.IP .....34
CONCAT.IP .....7 SETCOLOR.IP .....10 \DSPCOLOR.IP .....24
CONCATT.IP .....7 SETCOLOR16.IP .....12 \DSPFONT.IP .....29
DEFINEFONT.IP .....15 SETCOLORLV.IP .....11 \DSPLEFTMARGIN.IP .....30
ENDMASTER.IP .....7 SETFONT.IP .....12 \DSPLINEFEED.IP .....30
ENDPAGE.IP .....7 SETRGB.IP .....11 \DSPOPERATION.IP .....34
ENDPREAMBLE.IP .....8 SETSPACE.IP .....12 \DSPRIGHTMARGIN.IP .....30
ENSURE.RGB .....24 SETUPFONTS.IP .....20 \DSPROTATE.IP .....31
FGET.IP .....8 SETX.IP .....13 \DSPSCALE2.IP .....32
FILLCIRCLE.IP .....32 SETXREL.IP .....12 \DSPSPACEFACTOR.IP .....30
FILLNGON.IP .....8 SETXY.IP .....13 \DSPTOPMARGIN.IP .....31
FILLRECTANGLE.IP .....8 SETXYREL.IP .....13 \DSPTRANSLATE.IP .....31
FILLTRAJECTORY.IP .....8 SETY.IP .....13 \DSPXPOSITION.IP .....31
FONTNAME.IP .....16 SETYREL.IP .....14 \DSPYPOSITION.IP .....32
FSET.IP .....9 SHOW.IP .....14 \FILLPOLYGON.IP .....32
GETCP.IP .....9 SHOWBITMAP.IP .....20 \FXLINELLENGTH.IP .....33
GETFRAMEVAR.IP .....9 SHOWBITMAP1.IP .....21 \INTERPRESSINIT .....37
HEADINGOP.IP .....15 SHOWSHADE.IP .....22 \IPCURVE2 .....25
INITIALIZECOLOR.IP .....9 TRAJECTORY.IP .....14 \MOVETO.IP .....33
INITIALIZEMASTER.IP .....9 TRANS.IP .....14 \PAPERSTATE.IP .....31
INTERPRESS.BITMAPSCALE .....16 TRANSLATE.IP .....14 \PUSHSTATE.IP .....31
INTERPRESS.OUTCHARFN .....16 \BITBLT.IP .....22 \SCALEDBITBLT.IP .....23
INTERPRESSBITMAP .....40 \BITMAPSIZE.IP .....21 \SEARCHINTERPRESSFONTS .....42
INTERPRESSFILEP .....17 \BLTSHADE.IP .....23 \SETBRUSH.IP .....33
IP-TOS .....34 \CHANGE-VISIBLE-REGION.IP .....15 \STRINGWIDTH.IP .....34
ISET.IP .....9 \CHANGECHARSET.IP .....36

VARIABLE INDEX

\*INTERPRESS-PRINTER-DSPFONT-PATCH\* .....34 IPPAGEREGION.ROT180 .....38
ASCIITONSTRANSLATIONS .....42 IPPAGEREGION.ROT270 .....38
CHARACTERCODEVERSION .....2 IPTYPES .....3
DEFAULT.INTERPRESS.BITMAP.ROTATION .....41 IPVALUES .....3
DEFAULTINTERPRESSMEDIUM .....2 KNOWN.MEDIA.SIZES .....2
DEFAULTLANDPAGEREGION .....38 NONPRIMS .....3
DEFAULTPAGEREGION .....38 OPERATORS .....3
IMAGERVARIABLES .....4 PRINTERTYPES .....40
IMAGESTREAMTYPES .....40 PRINTFILETYPES .....41
INTERPRESSFAMILYALIASES .....41 PRINTSERVICE .....2
INTERPRESSFONTDIRECTORIES .....41 RATIONALS .....3
INTERPRESSFONTEXTENSIONS .....41 SEQUENCETYPES .....3
INTERPRESSPRINTWHEELFAMILIES .....41 STROKEENDS .....5
INTERPRESSVERSION .....2 SYSTEMINITVARS .....41
IP82CONSTANTS .....5 TOKENFORMATS .....4
IPCONSTANTS .....3 \SPLINESTEP.IP .....38

MACRO INDEX

.IPFONTNAME .....38 APPENDINTEGER.IPMACRO .....38 \IPC .....5
APPENDBYTE.IP .....38 APPENDOP.IP .....38 \VISIBLE.IP .....39
APPENDINT.IPMACRO .....38 \IMAGEPATH.IP .....39 \WIDTHFROMBRUSH .....39

CONSTANT INDEX

MAXSEGSPERTRAJECTORY ...38 MICASPERINCH .....5 MICASPERMILLIMETER .....5 noInfoCode .....42

RECORD INDEX

INTERPRESSDATA .....39 IPSTATE .....35 IPSTREAM .....39

{MEDLEY}<sources>INTERPRESS.;1

---