```
(RPAQQ HARDCOPYCOMS
    [(COMS                                                        ; exported functionality
        (FNS HARDCOPY.SOMEHOW HARDCOPYIMAGEW HARDCOPYIMAGEW.TOFILE HARDCOPYIMAGEW.TOPRINTER
            HARDCOPYREGION.TOFILE HARDCOPYREGION.TOPRINTER COPY.WINDOW.TO.BITMAP)
                                                                  ; user interface jazz
        (INITVARS (ChangeDefaultPrinter))
        (FNS MakeMenuOfPrinters PRINTERS.WHENSELECTEDFN MakeMenuOfImageTypes GetNewPrinterFromUser
            PopUpWindowAndGetAtom PopUpWindowAndGetList NewPrinter GetPrinterName GetImageFile
            FetchDefaultPrinter)
                                                                  ; filename diddlers
        (FNS ExtensionForPrintFileType PRINTFILETYPE.FROM.EXTENSION))
    (COMS                                          ; Interface for PRINTERS and IMAGEFILES
        (FNS DEFAULTPRINTER CAN.PRINT.DIRECTLY CONVERT.FILE.TO.TYPE.FOR.PRINTER EMPRESS HARDCOPYW
            LISTFILES1 PRINTER.BITMAPFILE PRINTER.BITMAPSCALE PRINTER.SCRATCH.FILE PRINTERPROP
            PRINTERSTATUS PRINTERTYPE PRINTERNAME PRINTFILEPROP PRINTFILETYPE \EXPECTED.FILE.TYPE
            SEND.FILE.TO.PRINTER)
        (FNS PRINTERDEVICE)
        [DECLARE%: DONTEVAL@LOAD DOCOPY (P (PRINTERDEVICE 'LPT]
        (P                                                        ; for backward compatibility
            (MOVD? 'NILL 'PRINTERMODE))
        (INITVARS (DEFAULTPRINTINGHOST)
                  (DEFAULTPRINTERTYPE 'INTERPRESS)
                  (EMPRESS.SCRATCH)
                  (EMPRESS#SIDES T)
                  (PRINTFILETYPES NIL))
        (GLOBALVARS DEFAULTPRINTINGHOST DEFAULTPRINTERTYPE EMPRESS#SIDES PRINTERTYPES PRINTFILETYPES))
    (COMS                                          ; Converting text files to imagestreams
        (INITVARS (TEXTDEFAULTTABS (LIST 20320))
                  (TEXTDEFAULTPAGEREGION (CREATEREGION 2794 1905 18415 24765)))
                                          ; TEXTDEFAULTTABS Hack, mica equivalent of 8 inches.  NOT
                                          ; USED ANYWHERE
        (GLOBALVARS TEXTDEFAULTTABS TEXTDEFAULTPAGEREGION)
        (FNS TEXTTOIMAGEFILE COPY.TEXT.TO.IMAGE))
    (COMS (FNS \BLTSHADE.GENERICPRINTER)
                                                   ; hack for printers that can't really BLTSHADE
        )
    [COMS                                          ; stuff to support hardcopy streams on the display.
        (FNS MAKEHARDCOPYSTREAM UNMAKEHARDCOPYSTREAM HARDCOPYSTREAMTYPE \CHARWIDTH.HDCPYDISPLAY
            \DSPFONT.HDCPYDISPLAY \DSPRIGHTMARGIN.HDCPYDISPLAY \DSPXPOSITION.HDCPYDISPLAY
            \DSPYPOSITION.HDCPYDISPLAY \STRINGWIDTH.HDCPYDISPLAY \STRINGWIDTH.HCPYDISPLAYAUX
            \HDCPYBLTCHAR \HDCPYDISPLAY.FIX.XPOS \HDCPYDISPLAY.FIX.YPOS \HDCPYDISPLAYINIT
            \HDCPYDSPPRINTCHAR \SLOWHDCPYBLTCHAR \CHANGECHARSET.HDCPYDISPLAY)
        [DECLARE%: DONTCOPY DOEVAL@COMPILE (EXPORT (CONSTANTS (MICASPERPT (FQUOTIENT 2540 72))
                                                             (IHALFMICASPERPT 17)
                                                             (IMICASPERPT 35)
                                                             (DEFAULTTAB 36]
                                                   ; screen-points: 1/2 inch
        (DECLARE%: DONTCOPY DOEVAL@COMPILE (EXPORT (FUNCTIONS \MICASTOPTS)))
        (DECLARE%: DONTEVAL@LOAD DOCOPY (P (\HDCPYDISPLAYINIT)
    [COMS                                          ; Stuff to support MICA-unit hardcopy streams on the display
        (FNS MAKEHARDCOPYMODESTREAM UNMAKEHARDCOPYMODESTREAM \BLTSHADE.HCPYMODE \BITBLT.HCPYMODE
            \BRUSHCONVERT.HCPYMODE \CHANGECHARSET.HCPYMODE \DASHINGCONVERT.HCPYMODE \CHARWIDTH.HCPYMODE
            \DRAWLINE.HCPYMODE \DRAWCURVE.HCPYMODE \DRAWCIRCLE.HCPYMODE \DRAWELLIPSE.HCPYMODE
            \DSPFONT.HCPYMODE \DSPLEFTMARGIN.HCPYMODE \DSPLINEFEED.HCPYMODE \DSPRIGHTMARGIN.HCPYMODE
            \DSPSPACEFACTOR.HCPYMODE \DSPXPOSITION.HCPYMODE \DSPYPOSITION.HCPYMODE \MOVETO.HCPYMODE
            \FONTCREATE.HCPYMODE.PRESS \CREATECHARSET.HCPYMODE.PRESS \FONTCREATE.HCPYMODE.INTERPRESS
            \CREATECHARSET.HCPYMODE.INTERPRESS \STRINGWIDTH.HCPYMODE \HCPYMODEBLTCHAR
            \HCPYMODEDISPLAYINIT \HCPYMODEDSPPRINTCHAR \SLOWHCPYMODEBLTCHAR \SFFixY.HCPYMODE)
        [ADDVARS (IMAGESTREAMTYPES (PRESSDISPLAY (FONTCREATE \FONTCREATE.HCPYMODE.PRESS)
                                                 (CREATECHARSET \CREATECHARSET.HCPYMODE.PRESS))
                                   (INTERPRESSDISPLAY (FONTCREATE \FONTCREATE.HCPYMODE.INTERPRESS)
                                                      (CREATECHARSET \CREATECHARSET.HCPYMODE.INTERPRESS]
        (DECLARE%: DONTEVAL@LOAD DOCOPY (P (\HCPYMODEDISPLAYINIT]
    (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)
                                                                          (NLAML)
                                                                          (LAMA])
```

;; exported functionality

```
(DEFINEQ

(HARDCOPY.SOMEHOW
  [LAMBDA (WINDOW FILE PRINTERTYPE IMAGETYPE)                    ; Edited 26-Nov-96 15:59 by rmk:
                                                                ; Edited 13-Nov-87 14:16 by Snow

  ;; Either run window's HARDCOPYFN or run HARDCOPYW.  The HARDCOPYFN can be a list of the form (fn heading) where heading=TITLE
```

```
        ;; means use the window's title, otherwise using the non-nil heading.
      (LET ((HARDCOPYFN (WINDOWPROP WINDOW 'HARDCOPYFN))
             HEADING)
            (ALLOW.BUTTON.EVENTS)
            (COND
               ((NULL HARDCOPYFN)                                              ; knows how to default
                (HARDCOPYW WINDOW FILE NIL NIL NIL PRINTERTYPE))
               (T (CL:WHEN (AND (LISTP HARDCOPYFN)
                                (FNTYP (CAR HARDCOPYFN)))
                      (SETQ HEADING (CADR HARDCOPYFN))
                      (CL:WHEN (EQ HEADING 'TITLE)
                          (SETQ HEADING (WINDOWPROP WINDOW 'TITLE)))
                      (SETQ HARDCOPYFN (CAR HARDCOPYFN)))
                  (CL:WITH-OPEN-STREAM [IMAGESTREAM (OPENIMAGESTREAM FILE (OR IMAGETYPE PRINTERTYPE)
                                                                     (CL:WHEN HEADING
                                                                         '(HEADING ,HEADING))]
                      (APPLY* HARDCOPYFN WINDOW IMAGESTREAM])
```

₍HARDCOPYIMAGEW
```
  [LAMBDA (W)                                                              ; Edited 26-Aug-87 14:08 by Snow
```
;;; hardcopy this window to the DEFAULTPRINTINGHOST
```
    (HARDCOPY.SOMEHOW W])
```

₍HARDCOPYIMAGEW.TOFILE
```
  [LAMBDA (W)                                                              ; Edited 17-Jan-96 10:33 by rmk
    (LET ((FILE&TYPE (GetImageFile W)))
         (if FILE&TYPE
             then (HARDCOPY.SOMEHOW W (CAR FILE&TYPE)
                        (CDR FILE&TYPE])
```

₍HARDCOPYIMAGEW.TOPRINTER
```
  [LAMBDA (W)                                                              ; Edited 18-Oct-2022 18:45 by lmm
                                                                           ; Edited 22-Apr-98 16:19 by rmk:
                                                                           ; Edited 11-Jul-90 13:55 by jds
    (LET ((PRINTERCHOICE (GetPrinterName))
           PRINTERTYPE IMAGETYPE)
          [COND
             ((LISTP PRINTERCHOICE)                                        ; Got back a list, which is (TYPE NAME).  Break it apart.
              (SETQ PRINTERTYPE (CAR PRINTERCHOICE))
              (SETQ PRINTERCHOICE (CADR PRINTERCHOICE)))
             (PRINTERCHOICE                                                ; Got back just a name.
              (SETQ PRINTERTYPE (PRINTERTYPE PRINTERCHOICE]
          (SETQ PRINTERTYPE (PRINTERTYPE PRINTERCHOICE))
          (COND
             (PRINTERCHOICE (HARDCOPY.SOMEHOW W (CONCAT "{LPT}" PRINTERCHOICE)
                                   PRINTERTYPE
                                   (OR IMAGETYPE (CAR (PRINTERPROP PRINTERTYPE 'CANPRINT))
                                       PRINTERTYPE]))
```

₍HARDCOPYREGION.TOFILE
```
  [LAMBDA NIL                                                              ; Edited 26-Aug-87 14:08 by Snow
    (LET ((FILE&TYPE (GetImageFile)))
         (if FILE&TYPE
             then (PROG (REGION)
                        (SPAWN.MOUSE)
                        (PROMPTPRINT "Select a region")
                        (SETQ REGION (GETREGION))
                        (CLRPROMPT)
                        (HARDCOPYW REGION (CAR FILE&TYPE)
                              NIL NIL NIL (CDR FILE&TYPE])
```

₍HARDCOPYREGION.TOPRINTER
```
  [LAMBDA NIL                                                              ; Edited 13-Jul-90 01:57 by jds
    (LET ((PRINTERCHOICE (GetPrinterName))
           PRINTERTYPE)
          [COND
             ((LISTP PRINTERCHOICE)                                        ; Got back a list, which is (TYPE NAME).  Break it apart.
              (SETQ PRINTERTYPE (CAR PRINTERCHOICE))
              (SETQ PRINTERCHOICE (CADR PRINTERCHOICE)))
             (PRINTERCHOICE                                                ; Got back just a name.
              (SETQ PRINTERTYPE (PRINTERTYPE PRINTERCHOICE]
          (COND
             (PRINTERCHOICE (PROG (REGION)
                                  (SPAWN.MOUSE)
                                  (PROMPTPRINT "Select a region")
                                  (SETQ REGION (GETREGION))
                                  (CLRPROMPT)
                                  (HARDCOPYW REGION (PACK* '{LPT} PRINTERCHOICE)
                                        NIL NIL NIL (PRINTERTYPE PRINTERCHOICE])
```

(**COPY.WINDOW.TO.BITMAP**
  [LAMBDA (WINDOW)   ; Edited 26-Aug-87 14:09 by Snow

;;; copies contents of window (including title and border) into a bitmap

    (COND
        ((OPENWP WINDOW)
         (PROG (REGION SCREEN LEFT BOTTOM WIDTH HEIGHT BITMAP)
               (SETQ REGION (WINDOWPROP WINDOW 'REGION))
               (SETQ SCREEN (WINDOWPROP WINDOW 'SCREEN))
               (SETQ LEFT (**fetch** (REGION LEFT) **of** REGION))
               (SETQ BOTTOM (**fetch** (REGION BOTTOM) **of** REGION))
               (SETQ WIDTH (**fetch** (REGION WIDTH) **of** REGION))
               (SETQ HEIGHT (**fetch** (REGION HEIGHT) **of** REGION))
               (SETQ BITMAP (BITMAPCREATE WIDTH HEIGHT (BITSPERPIXEL WINDOW)))
               (.WHILE.TOP.DS. WINDOW (BITBLT (SCREENBITMAP SCREEN)
                                              LEFT BOTTOM BITMAP 0 0 WIDTH HEIGHT))
               (RETURN BITMAP)))
        (T (BITMAPCOPY (WINDOWPROP WINDOW 'IMAGECOVERED]
)

;; user interface jazz

(RPAQ? **ChangeDefaultPrinter**)

(DEFINEQ

(**MakeMenuOfPrinters**
  [LAMBDA (MENUTITLE)   ; Edited 22-Jun-2023 17:30 by rmk
                                                                               ; Edited 29-May-93 14:18 by rmk:
                                                                               ; Edited 11-Jul-90 13:35 by jds
    (**DECLARE** (GLOBALVARS DEFAULTPRINTINGHOST))
    (**CREATE** MENU
           ITEMS _ (APPEND (**FOR** P **INSIDE** DEFAULTPRINTINGHOST
                              COLLECT (LIST (COND
                                                   ((LISTP P)
                                                    (**IF** (CADDR P)
                                                        THEN (CONCAT (CADR P)
                                                                     " "
                                                                     (CADDR P))
                                                      ELSE (CADR P)))
                                                   (T (CL:IF (OR (NULL P)
                                                                 (ZEROP (NCHARS P)))
                                                          "(Default printer)"
                                                          P)))
                                            (KWOTE P)))
                         (LIST (LIST "Other..." (KWOTE 'OTHER)
                                     "You will be prompted for a printer")))
           TITLE _ MENUTITLE
           WHENSELECTEDFN _ (FUNCTION PRINTERS.WHENSELECTEDFN])

(**PRINTERS.WHENSELECTEDFN**
  [LAMBDA (ITEM MENU BUTTON)   ; Edited 16-Apr-2018 22:14 by rmk:
    (**DECLARE** (GLOBALVARS ChangeDefaultPrinter))

    ;; Fix Menu so that it doesn't ask about changing the default unless you click with middle

    (LET ((PRINTERCHOICE (CADR (CADR ITEM)))
          DEFAULTPRINTER)
         [COND
            ((EQ PRINTERCHOICE 'OTHER)
             (SETQ PRINTERCHOICE (**GetNewPrinterFromUser**]
         (CL:WHEN [AND PRINTERCHOICE (NEQ PRINTERCHOICE (SETQ DEFAULTPRINTER (**FetchDefaultPrinter**]
               [**NewPrinter** PRINTERCHOICE (AND DEFAULTPRINTER (EQ BUTTON 'MIDDLE)
                                              (MENU (OR ChangeDefaultPrinter (SETQ ChangeDefaultPrinter
                                                                                 (**create** MENU
                                                                                       TITLE _ "Make this the new
                                                                                       default?"
                                                                                       ITEMS _
                                                                                       '(("Yes" T "Yes, make this the
                                                                                            new default printer")
                                                                                         ("No" NIL "No, don't change
                                                                                            it"))
                                                                                       MENUROWS _ 1
                                                                                       CENTERFLG _ T])
         PRINTERCHOICE])

(**MakeMenuOfImageTypes**
  [LAMBDA (MENUTITLE)   ; Edited 26-Aug-87 14:10 by Snow

;;; type selection; elements of \DISPLAYSTREAMTYPES are temporarily disallowed

    (**DECLARE** (GLOBALVARS IMAGESTREAMTYPES))
    (**create** MENU

```
                    ITEMS _ [for IMAGETYPE in IMAGESTREAMTYPES bind IMAGETYPENAME
                                    collect (PROGN (SETQ IMAGETYPENAME (CAR IMAGETYPE))
                                                   (LIST (L-CASE IMAGETYPENAME T)
                                                         (KWOTE IMAGETYPENAME)))
                                    when (AND (ASSOC 'OPENSTREAM (CDR IMAGETYPE))
                                              (NOT (FMEMB (CAR IMAGETYPE)
                                                          \DISPLAYSTREAMTYPES]
                    TITLE _ MENUTITLE])
```

(**GetNewPrinterFromUser**
```
  [LAMBDA (PROMPTSTRING)                                          ; Edited  7-Jun-93 15:33 by rmk:
                                                                  ; Edited 26-Aug-87 14:10 by Snow

     ;; Changed from PopUpWindowAndGetAtom, so user can enter PRINTERTYPE PRINTERNAME PREFERREDIMAGETYPE.

     (PopUpWindowAndGetList (OR PROMPTSTRING "Printer (CR to abort): "])
```

(**PopUpWindowAndGetAtom**
```
  [LAMBDA (PROMPTSTRING CANDIDATE)                                ; Edited  6-Mar-2024 13:15 by rmk
                                                                  ; Edited 26-Aug-87 14:10 by Snow
     (RESETLST
        (RESETSAVE (TTY.PROCESS (THIS.PROCESS)))
        [LET* ((FONT (DEFAULTFONT))
               [WIDTH (WIDTHIFWINDOW (IPLUS (STRINGWIDTH PROMPTSTRING FONT)
                                            (CL:IF CANDIDATE
                                                (IPLUS (STRINGWIDTH CANDIDATE FONT)
                                                       (ITIMES 10 (CHARWIDTH (CHARCODE A)
                                                                             FONT)))
                                                (ITIMES 40 (CHARWIDTH (CHARCODE A)
                                                                      FONT)))]
               (PROMPTW (CREATEW [CREATEREGION (IMIN LASTMOUSEX (IDIFFERENCE SCREENWIDTH WIDTH))
                                               LASTMOUSEY WIDTH (HEIGHTIFWINDOW (FONTPROP FONT 'HEIGHT]
                                 NIL NIL T)))
              (RESETSAVE (OPENW PROMPTW)
                     (LIST (FUNCTION CLOSEW)
                           PROMPTW))
              (LET [(RESPONSE (TTYINPROMPTFORWORD PROMPTSTRING CANDIDATE NIL PROMPTW NIL NIL (CHARCODE (CR]
                   (AND RESPONSE (PACK* RESPONSE])
```

(**PopUpWindowAndGetList**
```
  [LAMBDA (PROMPTSTRING CANDIDATE)                                ; Edited 16-Apr-2018 22:13 by rmk:
                                                                  ; Edited 26-Aug-87 14:10 by Snow
     ;; Makes both image-type part of LISTP printers show up in menu, so you can see the imagetype in multiple-type printers

     (RESETLST
        (RESETSAVE (TTY.PROCESS (THIS.PROCESS)))
        [LET* ((FONT (DEFAULTFONT))
               [WIDTH (WIDTHIFWINDOW (IPLUS (STRINGWIDTH PROMPTSTRING FONT)
                                            (ITIMES 40 (CHARWIDTH (CHARCODE A)
                                                                  FONT]
               (PROMPTW (CREATEW [CREATEREGION (IMIN LASTMOUSEX (IDIFFERENCE SCREENWIDTH WIDTH))
                                               LASTMOUSEY WIDTH (HEIGHTIFWINDOW (TIMES 2 (FONTPROP FONT 'HEIGHT]
                                 NIL NIL T)))

              ;; Allow room for 2 lines so that TTYIN doesn't hang on page-full

              (RESETSAVE (TTYDISPLAYSTREAM PROMPTW))
              [RESETSAVE NIL '(CLOSEW ,PROMPTW]
              (LET ((RESPONSE (TTYIN PROMPTSTRING CANDIDATE NIL '(NORAISE READ)
                                     NIL NIL NIL TTYINWORDRDTBL)))
                   (CL:IF (CDR RESPONSE)
                       RESPONSE
                       (CAR RESPONSE))])))
```

(**NewPrinter**
```
  [LAMBDA (PRINTER NEW-DEFAULT?)                                  ; Edited 11-Jul-90 13:48 by jds

;;; If Printer is unknown it will be added to DEFAULTPRINTINGHOST. In addition, if NEW-DEFAULT? is true the printer will be pushed to the head of
;;; DEFAULTPRINTINGHOST, thus making it the default printer.

     (DECLARE (GLOBALVARS DEFAULTPRINTINGHOST))
     (CL:WHEN (NOT (LISTP DEFAULTPRINTINGHOST))                   ; If DEFAULTPRINTINGHOST Is an atom
        (SETQ DEFAULTPRINTINGHOST (LIST DEFAULTPRINTINGHOST)))
     (LET* ((PRINTER-NAME (COND
                             ((LISTP PRINTER)
                              (CADR PRINTER))
                             (T PRINTER)))
            [MEMBER? (CL:MEMBER PRINTER-NAME DEFAULTPRINTINGHOST :TEST '(LAMBDA (PRINTER ENTRY)
                                                                         (STRING-EQUAL PRINTER
                                                                                (CL:IF (LISTP ENTRY)
                                                                                    (CADR ENTRY)
                                                                                    ENTRY)]
            (ENTRY (CL:IF MEMBER?
                       (CAR MEMBER?)
                       PRINTER)))
```

```
            (CL:IF NEW-DEFAULT?
                (SETQ DEFAULTPRINTINGHOST (CONS ENTRY (REMOVE ENTRY DEFAULTPRINTINGHOST))))
                (CL:IF (NOT MEMBER?)
                    (RPLACD (LAST DEFAULTPRINTINGHOST)
                        (CONS ENTRY))))
            DEFAULTPRINTINGHOST])
```

(**GetPrinterName**
```
  [LAMBDA NIL                                            ; Edited 29-May-93 13:58 by rmk:
                                                         ; Edited 26-Aug-87 14:10 by Snow

    (MENU (MakeMenuOfPrinters "Which printer?"])
```

(**GetImageFile**
```
  [LAMBDA (W)                                            ; Edited 27-Apr-98 16:44 by rmk:
                                                         ; Edited 18-Jan-96 11:17 by
                                                         ; Edited 17-Jan-96 10:42 by rmk

    (PROG (FILE PRINTFILETYPE FILETYPEMENU)
```
    ;; Strip candidate version so overwrites must be explicitly indicated each time. Use previous file as candidate, and if no previous one, apply
    ;; function associated with the window to the window and the extension associated with the defaultprinting host. Such a function on a TEDIT
    ;; window, for example, could suggest the image-type file named after the underlying TEDIT file.
```
        [SETQ FILE
          (PopUpWindowAndGetAtom
            "File name (Clear to abort): "
            (OR [AND (WINDOWPROP W 'HARDCOPYFILE)
                     (PACKFILENAME 'VERSION NIL 'BODY (WINDOWPROP W 'HARDCOPYFILE]
                (AND (WINDOWPROP W 'HARDCOPYFILEFN)
                     (APPLY* (WINDOWPROP W 'HARDCOPYFILEFN)
                             W
                             (CAR (MKLIST (CADR (ASSOC 'EXTENSION (CDR (ASSOC (OR (CADDR (LISTP (DEFAULTPRINTER)
                                                                                             ))
                                                                             (PRINTERTYPE))
                                                                    PRINTFILETYPES]
            (CL:UNLESS (AND FILE (SETQ FILE (OUTFILEP FILE)))     ; Keep directory etc for reuse
                (RETURN))
            (WINDOWPROP W 'HARDCOPYFILE FILE)                     ; Save previous input for next candidate
            (SETQ FILETYPEMENU (MakeMenuOfImageTypes "File type?"))
            (COND
                ((SETQ PRINTFILETYPE (PRINTFILETYPE.FROM.EXTENSION FILE))
                 (RETURN (CONS FILE PRINTFILETYPE)))
                (T (SETQ PRINTFILETYPE (MENU FILETYPEMENU))
                   (COND
                       ((NULL PRINTFILETYPE)
                        (RETURN))
                       (T (RETURN (CONS FILE PRINTFILETYPE])
```

(**FetchDefaultPrinter**
```
  [LAMBDA NIL                                            ; Edited 26-Aug-87 14:11 by Snow
    (LET ((P (DEFAULTPRINTER)))
        (COND
           ((LISTP P)
            (CADR P))
           (T P])

)
```

;; filename diddlers

```
(DEFINEQ
```

(**ExtensionForPrintFileType**
```
  [LAMBDA (TYPE)                                         ; Edited 26-Aug-87 14:11 by Snow
    (DECLARE (GLOBALVARS PRINTFILETYPES))
    (CAADR (ASSOC 'EXTENSION (CDR (ASSOC TYPE PRINTFILETYPES]
```

(**PRINTFILETYPE.FROM.EXTENSION**
```
  [LAMBDA (FILE)                                         ; Edited 26-Aug-87 14:11 by Snow
                                                         ; return the imagestream type corresponding to the extension
    (bind [EXT _ (U-CASE (FILENAMEFIELD FILE 'EXTENSION] for TYPE in PRINTFILETYPES
        when [FMEMB EXT (CADR (ASSOC 'EXTENSION (CDR TYPE] do (RETURN (CAR TYPE))

)
```

;; Interface for PRINTERS and IMAGEFILES

```
(DEFINEQ
```

(**DEFAULTPRINTER**
```
  [LAMBDA NIL                                            ; Edited 26-Aug-87 14:11 by Snow
    (COND
       ((LISTP DEFAULTPRINTINGHOST)
        (CAR DEFAULTPRINTINGHOST))
```

```
        (T DEFAULTPRINTINGHOST])
```

⟨**CAN.PRINT.DIRECTLY**
```
  [LAMBDA (PRINTERTYPE FILETYPE)                                              ; Edited 26-Aug-87 14:11 by Snow
    (FMEMB FILETYPE (PRINTERPROP PRINTERTYPE 'CANPRINT])
```

⟨**CONVERT.FILE.TO.TYPE.FOR.PRINTER**
```
  [LAMBDA (FILE FILETYPE PRINTERTYPE HEADING PRINTOPTIONS)        ; Edited 24-Sep-2023 15:25 by rmk
                                                                 ; Edited 14-Sep-2023 22:58 by rmk
                                                                 ; Edited 29-Dec-88 15:39 by jds

    ;; Convert FILE to the kind of hardcopy file (Interpress, Press, 4045HQ, etc) appropriate to PRINTERTYPE.

    (SETQ FILETYPE (OR FILETYPE 'TEXT))
    (PROG [[(SCRATCH (CLOSEF (OPENSTREAM (PRINTER.SCRATCH.FILE FILE PRINTERTYPE)
                                    'OUTPUT
                                    'NEW]                        ; Doing the open & close gets us a guaranteed version number,
                                                                 ; so that all files are truly unique.
           (APPLY* (OR (LISTGET (PRINTERPROP PRINTERTYPE 'CONVERSION)
                            FILETYPE)
                       (for CANPRINT in (PRINTERPROP PRINTERTYPE 'CANPRINT) bind CONVERTER
                          when (SETQ CONVERTER (LISTGET (PRINTFILEPROP CANPRINT 'CONVERSION)
                                                    FILETYPE))
                          do (RETURN CONVERTER))
                       (ERROR (CONCAT "Can't convert a " FILETYPE " for a " PRINTERTYPE " printer")
                              (FULLNAME FILE)))
                   FILE SCRATCH (LISTGET PRINTOPTIONS 'FONTS)
                   HEADING NIL PRINTOPTIONS)
           (RESETSAVE NIL (LIST [FUNCTION (LAMBDA (SCRATCH)
                                         (CLOSEF? SCRATCH)
                                         (DELFILE SCRATCH]
                              SCRATCH))
           (RETURN SCRATCH])
```

⟨**EMPRESS**
```
  [LAMBDA (FILE %#COPIES HOST HEADING %#SIDES PRINTOPTIONS)         ; Edited 26-Aug-87 14:17 by Snow
    (SEND.FILE.TO.PRINTER FILE HOST (NCONC (COND
                                         (HEADING (LIST 'HEADING HEADING)))
                                       (COND
                                         (%#COPIES (LIST '%#COPIES %#COPIES)))
                                       (COND
                                         (%#SIDES (LIST '%#SIDES %#SIDES)))
                                       PRINTOPTIONS])
```

⟨**HARDCOPYW**
```
  [LAMBDA (WINDOW/BITMAP/REGION FILE HOST SCALEFACTOR ROTATION PRINTERTYPE HARDCOPYTITLE)
                                                                 ; Edited 31-Aug-89 10:05 by jds

    ;; Makes a hard copy of a window, bitmap, or region of the screen.
    ;;
    ;; WINDOW/BITMAP/REGION can be a WINDOW, a REGION, a BITMAP, or NIL = select region.  If FILE supplied, output goes there.  If HOST
    ;; supplied, it is printed.  If neither FILE nor HOST supplied, default is to print; if HARDCOPYTITLE is supplied it will be used as the document title
    ;; of the hardcopy file created.  If it isn't, 'Window Image' is used.

    (PROG (PRINTHOST BITMAP SCREENREGION REGION FULLFILE)
          (SETQ PRINTHOST HOST)
          [COND
             ((WINDOWP WINDOW/BITMAP/REGION)
              (SETQ BITMAP (COPY.WINDOW.TO.BITMAP WINDOW/BITMAP/REGION)))
             ((BITMAPP WINDOW/BITMAP/REGION)
              (SETQ BITMAP WINDOW/BITMAP/REGION))
             ((type? REGION WINDOW/BITMAP/REGION)
              (SETQ BITMAP (SCREENBITMAP))
              (SETQ REGION WINDOW/BITMAP/REGION))
             (T (SETQ SCREENREGION (GETSCREENREGION))
                (SETQ BITMAP (SCREENBITMAP (fetch (SCREENREGION SCREEN) of SCREENREGION)))
                (SETQ REGION (fetch (SCREENREGION REGION) of SCREENREGION]
      RETRY
          (COND
             [PRINTERTYPE (COND
                             [PRINTHOST (COND
                                          ((NOT (EQ PRINTERTYPE (PRINTERTYPE PRINTHOST)))
                                           (ERROR PRINTHOST (CONCAT "not of printer type " PRINTERTYPE))
                                           (GO RETRY]
                             (FILE                               ; don't need a PRINTHOST if you give a file
                                  )
                             [[(SETQ PRINTHOST (find HOST inside DEFAULTPRINTINGHOST
                                               suchthat (EQ PRINTERTYPE (PRINTERTYPE HOST]
                             (T (ERROR "Can't find a printing host in DEFAULTPRINTINGHOST that is of type "
                                       PRINTERTYPE)
                                (GO RETRY]
             (PRINTHOST (SETQ PRINTERTYPE (PRINTERTYPE PRINTHOST)))
             (DEFAULTPRINTINGHOST (SETQ PRINTHOST (DEFAULTPRINTER))
                   (SETQ PRINTERTYPE (PRINTERTYPE PRINTHOST)))
```

```
              [FILE (COND
                       ((NOT (SETQ PRINTERTYPE (PRINTFILETYPE FILE T)))
                        (ERROR FILE "Can't tell what kind of print file to produce -- PRINTERTYPE,
                              DEFAULTPRINTERTYPE, DEFAULTPRINTINGHOST all NIL")
                        (GO RETRY]
                  (T (ERROR "Can't tell where to send window image -- HOST, DEFAULTPRINTINGHOST are NIL")
                     (GO RETRY)))
              [COND
                 ((NOT SCALEFACTOR)
                  [SETQ SCALEFACTOR (COND
                                      (REGION (PRINTER.BITMAPSCALE (fetch (REGION WIDTH) of REGION)
                                                     (fetch (REGION HEIGHT) of REGION)
                                                     PRINTERTYPE PRINTHOST))
                                      (T (PRINTER.BITMAPSCALE (fetch (BITMAP BITMAPWIDTH) of BITMAP)
                                                     (fetch (BITMAP BITMAPHEIGHT) of BITMAP)
                                                     PRINTERTYPE PRINTHOST]
                  (COND
                     ((LISTP SCALEFACTOR)
                      (SETQ ROTATION (CDR SCALEFACTOR))
                      (SETQ SCALEFACTOR (CAR SCALEFACTOR]
              (SETQ FULLFILE (PRINTER.BITMAPFILE (OR FILE (PRINTER.SCRATCH.FILE))
                                     PRINTERTYPE BITMAP SCALEFACTOR REGION ROTATION (OR HARDCOPYTITLE "Window Image")
                                     ))
              [COND
                 ((OR HOST (NULL FILE))
                  (ADD.PROCESS `[PROGN [,(PRINTERPROP PRINTERTYPE 'SEND)
                                          ',(COND
                                               ((LISTP PRINTHOST)
                                                (CADR PRINTHOST))
                                               (T PRINTHOST))
                                          ',FULLFILE
                                          '(DELETE ,(NULL FILE)
                                                 DOCUMENT.NAME
                                                 ,(OR HARDCOPYTITLE "Window Image"]
                                       ,(AND (NULL FILE)
                                             `(DELFILE ',FULLFILE]
                        'NAME
                        'HARDCOPYW]
              (RETURN (AND FILE FULLFILE)])
```

(**LISTFILES1**
```
  [LAMBDA (FILE PRINTOPTIONS)                                      ; Edited 26-Aug-87 14:17 by Snow
     (SEND.FILE.TO.PRINTER FILE NIL PRINTOPTIONS])
```

(**PRINTER.BITMAPFILE**
```
  [LAMBDA (FILE PRINTERTYPE BITMAP SCALEFACTOR REGION ROTATION TITLE)
                                                                   ; Edited 26-Aug-87 14:19 by Snow
                                                                   ; convert a bitmap into a file
     (DECLARE (SPECVARS . T))
     (EVAL (PRINTERPROP PRINTERTYPE 'BITMAPFILE])
```

(**PRINTER.BITMAPSCALE**
```
  [LAMBDA (WIDTH HEIGHT PRINTERTYPE HOST)                          ; Edited 26-Aug-87 14:19 by Snow
                                                                   ; could ask the host what size paper it has
     (PROG NIL
           (RETURN (APPLY* (OR (PRINTERPROP PRINTERTYPE 'BITMAPSCALE)
                                (RETURN 1))
                           WIDTH HEIGHT HOST])
```

(**PRINTER.SCRATCH.FILE**
```
  [LAMBDA (FULLFILE)                                               ; Edited 26-Aug-87 14:20 by Snow
     '{SCRATCH}PRINTER-SCRATCH-FILE])
```

(**PRINTERPROP**
```
  [LAMBDA (PRINTERTYPE PROP)                                       ; Edited 26-Aug-87 14:20 by Snow
     (for X in PRINTERTYPES when (EQMEMB PRINTERTYPE (CAR X)) do (RETURN (CADR (ASSOC PROP (CDR X]
```

(**PRINTERSTATUS**
```
  [LAMBDA (PRINTER)                                                ; Edited 26-Aug-87 14:21 by Snow
     (LET [(STATUSFN (PRINTERPROP (PRINTERTYPE PRINTER)
                             'STATUS]
          (AND STATUSFN (APPLY* STATUSFN PRINTER])
```

(**PRINTERTYPE**
```
  [LAMBDA (HOST)                                                   ; Edited 27-Apr-98 16:16 by rmk:
                                                                   ; Edited 15-Feb-91 14:14 by gadener

     ;; Attempt to deduce the printer type of HOST.

     (SELECTQ HOST
           ((NIL LPT)
```

```
                (SETQ HOST (DEFAULTPRINTER)))
          NIL)
     (COND
        [(CAR (LISTP HOST))
```
;; Is a pair (type hostname) or maybe a triple of the form (printertype hostname preferred-imagetype). Check that type is one we know about.
```
         (LET ((TYPE (CAR HOST)))
              (COND
                 ((for X in PRINTERTYPES thereis (EQMEMB TYPE (CAR X)))
                  TYPE)
                 (T (ERROR "Undefined printer-type:" TYPE]
        ((NULL HOST)
         DEFAULTPRINTERTYPE)
        ((GETPROP (MKATOM HOST)
                  'PRINTERTYPE))
        ((GETPROP (SETQ HOST (OR (CANONICAL.HOSTNAME HOST)
                                 HOST))
                  'PRINTERTYPE))
        [(for TYPE in PRINTERTYPES bind FN when (AND (SETQ FN (CDR (ASSOC 'HOSTNAMEP TYPE)))
                                                     (APPLY* (CAR FN)
                                                             HOST))
```
                                                                   ; Try the predicates for each printer type for recognizing their
                                                                   ; own host names
```
           do
                (RETURN (CAAR TYPE]
        [(for PRINTER in (MKLIST DEFAULTPRINTINGHOST) do
```
                                                      ;; Try looking for literal match before doing canonical hostname, cause that
                                                      ;; may be expensive.
```
                                                      (COND
                                                         ((AND (LISTP PRINTER)
                                                               (STRING-EQUAL (CADR PRINTER)
                                                                      HOST))
                                                          (RETURN (CAR PRINTER]
        [(for PRINTER in (MKLIST DEFAULTPRINTINGHOST) do (COND
                                                            ((AND (LISTP PRINTER)
                                                                  (STRING-EQUAL (OR (CANONICAL.HOSTNAME
                                                                                       (CADR PRINTER))
                                                                                    (CADR PRINTER))
                                                                         HOST))
                                                             (RETURN (CAR PRINTER]
        (T DEFAULTPRINTERTYPE)])
```

(PRINTERNAME
```
 [LAMBDA (PRINTER-SPEC)                                            ; Edited 26-Nov-86 13:51 by hdj
```
    ;; takes a printer-spec (in form (type printer-name) or just printer-name) and returns printer-name. returns nil for null arg.
```
    (AND PRINTER-SPEC (if (LISTP PRINTER-SPEC)
                          then (CADR PRINTER-SPEC)
                        else PRINTER-SPEC])
```

(PRINTFILEPROP
```
 [LAMBDA (PRINTFILETYPE PROP)                                      ; Edited 26-Aug-87 14:22 by Snow
   (for X in PRINTFILETYPES when (EQMEMB PRINTFILETYPE (CAR X)) do (RETURN (CADR (ASSOC PROP (CDR X)))
```

(PRINTFILETYPE
```
 [LAMBDA (FILE DONTOPEN)                                           ; Edited  3-Mar-93 14:34 by rmk:
                                                                   ; Edited 22-Aug-92 14:27 by jds
                                                                   ; Edited 26-Aug-87 14:22 by Snow

    (COND
       ((IMAGESTREAMP FILE)
        (IMAGESTREAMTYPE FILE))
       (T (LET* [(HOST (FILENAMEFIELD FILE 'HOST))
                 (TYPE (GETFILEINFO FILE 'TYPE]
                (COND
                   ((AND TYPE (ASSOC TYPE PRINTFILETYPES))
```
                    ;; Type is in PRINTFILETYPES, so it's OK.
```
                    TYPE)
                   ((PRINTFILETYPE.FROM.EXTENSION FILE))
                   [(NOT DONTOPEN)
                    (RESETLST
                        [COND
                           ((STRINGP FILE)                         ; Yecch, OPENP of a string interprets string as a string stream!
                            (SETQ FILE (MKATOM FILE]
                        [COND
                           ((NOT (OPENP FILE 'INPUT))              ; Open file so testers don't have to repeatedly open and close it
                            (SETQ FILE (OPENSTREAM FILE 'INPUT))
                            (RESETSAVE NIL (LIST 'CLOSEF? FILE]
                        [COND
                           ((RANDACCESSP FILE)
                            (for TYPE in PRINTFILETYPES
                               when (CAR (NLSETQ (APPLY* (CADR (ASSOC 'TEST (CDR TYPE)))
                                                         FILE)))
                               do (RETURN (CAR TYPE]
                   ((EQ TYPE 'TEXT)
```

```
                         ;; This is AFTER the above clauses, so we catch PS files, which are type TEXT.  Other formats might be lost as well....
                         TYPE])
```

(**EXPECTED.FILE.TYPE**
```
  [LAMBDA (FILE)                                                              ; Edited 28-Jun-99 16:36 by rmk:
                                                                              ; Edited 27-Oct-90 18:14 by nm
```
```
    ;; rmk:  This is called by SEND.FILE.TO.PRINTER to somehow guess the TYPE parameter of the file in Maiko.  I don't see the point of this.
    ;; Eventually, the call to this function and even its definition should be removed, but nuking it is just as effective.

      (AND NIL (EQ \MACHINETYPE \MAIKO)
           FileTypeConfirmFlg
           (LET [(HOST (UNPACKFILENAME.STRING FILE 'HOST]
                (AND (OR (STRING-EQUAL HOST "DSK")
                         (STRING-EQUAL HOST "UNIX"))
                     '((TYPE ,(\UFSGetPrintFileType FILE]))
```

(**SEND.FILE.TO.PRINTER**
```
  [LAMBDA (FILE HOST PRINTOPTIONS)                                            ; Edited 21-Jan-93 11:34 by jds
```
```
    ;; Returns file name if successful, NIL if not.  The RESETLST makes sure the scratch file, if any, is deleted.

      (RESETLST
          [PROG (FULLFILE STRM FILETYPE PRINTERTYPE PFILE)
                [RESETSAVE NIL '(,(COND
                                     [(LISTGET PRINTOPTIONS 'DELETE)
                                      (FUNCTION (LAMBDA (STREAM)
                                                       (CLOSEF? STREAM)
                                                       (DELFILE (FULLNAME STREAM]
                                     (T (FUNCTION CLOSEF?)))
                                   ,(SETQ STRM (if (AND (STREAMP FILE)
                                                        (OPENP FILE 'INPUT))
                                               then
```
```
                                                   ;; Don't re-open it if it was previously open.  (Some gibberish here about "cause caller
                                                   ;; (PRINTERDEVICE) really wants us to use the same stream, which has the BEINGPRINTED
                                                   ;; property.")
                                                                        FILE
                                               else (OPENSTREAM FILE 'INPUT 'OLD (\EXPECTED.FILE.TYPE FILE]
                                                                        ; Do we need to convert the FILE ?
                (SETQ FULLFILE (FULLNAME (SETQ PFILE STRM)))              ; Do the FULLNAME on the open stream, as FULLNAME
                                                                         ; sometimes returns NIL on just a filename
                (SETQ FILETYPE (PRINTFILETYPE STRM))                      ; Find out what kind of file this is, so we can figure out how to
                                                                         ; print it.
            RETRY
                [COND
                   [[OR HOST (SETQ HOST (for X on PRINTOPTIONS by (CDDR X) when (MEMB (U-CASE (CAR X))
                                                                                       '(HOST SERVER))
                                         do (RETURN (CADR X]
                    (SETQ PRINTERTYPE (PRINTERTYPE HOST))
                    (COND
                       ((CAN.PRINT.DIRECTLY PRINTERTYPE FILETYPE)  ; IS OK, NO CONVERSION

                        )
                       (T (SETQ PFILE (CONVERT.FILE.TO.TYPE.FOR.PRINTER STRM FILETYPE PRINTERTYPE
                                            (LISTGET PRINTOPTIONS 'HEADING)
                                            PRINTOPTIONS]
                   ((NULL DEFAULTPRINTINGHOST)
                    (ERROR "DEFAULTPRINTINGHOST and HOST arg are NIL; don't know where to print " FULLFILE)
                    (GO RETRY))
                   ([AND FILETYPE (for X inside (OR DEFAULTPRINTINGHOST '(NIL)) when (CAN.PRINT.DIRECTLY
                                                                                        (SETQ PRINTERTYPE (
                                                                                                           PRINTERTYPE
                                                                                                           X))
                                                                                        FILETYPE)
                                      do (RETURN (SETQ HOST X]        ; no conversion necessary

                        )
                   (T (SETQ PFILE (CONVERT.FILE.TO.TYPE.FOR.PRINTER STRM FILETYPE [SETQ PRINTERTYPE
                                                                                     (PRINTERTYPE (SETQ HOST (
                                                                                                     DEFAULTPRINTER
                                                                                                              ]
                                            (LISTGET PRINTOPTIONS 'HEADING)
                                            PRINTOPTIONS]
                (COND
                   ([NLISTP (SETQ PFILE (CL:FUNCALL (OR (PRINTERPROP PRINTERTYPE 'SEND)
                                                        (ERROR (CONCAT "Don't know how to send to a " PRINTERTYPE)
                                                               HOST))
                                                    (COND
                                                       ((LISTP HOST)
                                                        (CADR HOST))
                                                       (T HOST))
                                                    PFILE
                                                    (APPEND PRINTOPTIONS '(%#COPIES 1)
                                                            (LIST 'DOCUMENT.NAME FULLFILE]
                (RETURN FULLFILE))
```

```
                    (T (LISPXPRIN1 (CDR PFILE)
                              T)
                       (LISPXTERPRI T)
                       (RETURN NIL)])])
)

(DEFINEQ
```

(**PRINTERDEVICE**
```
  [LAMBDA (NAME)                                                          ; Edited  5-Dec-96 11:23 by rmk:
                                                                          ; Edited  4-Dec-86 16:32 by hdj
```

  ;; This defines an LPT device. An LPT file is a file that gets sent to printer and deleted when it is closed. This must be defined on a CORE device
  ;; only because we have no way of inheriting the previous CLOSEFILE function that this function is replacing but needs to call internally. We have
  ;; \CORE.CLOSEFILE explicit in this code.

```
    (LET ((DEV (\CREATECOREDEVICE NAME)))
        [replace (FDEV OPENFILE) of DEV with (FUNCTION (LAMBDA (NAME ACCESS RECOG PARAMETERS FDEV OLDSTREAM)
                                                 (LET ((STRM (\CORE.OPENFILE NAME ACCESS RECOG PARAMETERS
                                                                   FDEV OLDSTREAM)))
```

                               ;; Mark the original name of the printer on the stream. Unless the user overrides this by changing the
                               ;; PRINTERNAME property, SEND.FILE.TO.PRINTER in the close function will get the user's original
                               ;; spelling, without any case conversions that might otherwise be done by \CORE.OPENFILE.

```
                                         (STREAMPROP STRM 'PRINTERNAME (FILENAMEFIELD
                                                                              NAME
                                                                              'NAME))

                                      STRM]
        [replace (FDEV CLOSEFILE) of DEV
            with (FUNCTION (LAMBDA (STREAM)
                       (LET [(SDEV (fetch (STREAM DEVICE) of STREAM))
                             (PRINTOPTIONS (STREAMPROP STREAM 'PRINTOPTIONS]
```

                        ;; Get PRINTOPTIONS property before closing the stream, in case the closing throws them away
                        ;;
                        ;; If we could save away and get at the previous CLOSEFILE method (e.g. by an FDEVPROP), this could
                        ;; be replaced by the generic (FDEVOP (QUOTE CLOSEFILE) SDEV STREAM)

```
                        (COND
                           [(AND (NOT RESETSTATE)
                                 (OPENP STREAM 'OUTPUT)
                                 (IGREATERP (GETEOFPTR STREAM)
                                      0))
```

                            ;; Close and send to printer only if open for output. If open for input, then we must already have
                            ;; started printing. Don't close until after getting EOF ptr.

```
                            (\CORE.CLOSEFILE STREAM)
                            (replace (STREAM ACCESS) of STREAM with NIL)
```
                                                    ; Hack, cause this is usually done later in the generic
                                                    ; \CLOSEFILE.

                            ;; The PRINTERNAME might be marked explicitly on the stream. Otherwise let
                            ;; SEND.FILE.TO.PRINTER choose the host if it is the generic printer LPT, or use the name in the
                            ;; devicename field.

```
                            (SEND.FILE.TO.PRINTER STREAM
                                [IF (STREAMPROP STREAM 'PRINTERNAME)
                                    ELSEIF (NEQ 'LPT (fetch (FDEV DEVICENAME) of SDEV))
                                      THEN (fetch (FDEV DEVICENAME) of SDEV)
                                      ELSE (LET ((NAME (fetch (STREAM FULLNAME) of STREAM))
                                                 POS POS2)
                                              (AND (SETQ POS (STRPOS "}" NAME))
                                                   (SETQ POS2 (STRPOS "." NAME (ADD1 POS)))
                                                   (SUBATOM NAME (ADD1 POS)
                                                          (SUB1 POS2]
                                (APPEND '(DELETE T)
                                        PRINTOPTIONS
                                        '(HEADING T]
                           (T
```
                            ;; Error while creating the file, if the user had wrapped a RESETLST/CLOSEF around his code.
                            ;; Presumably, he doesn't want the file printed

```
                            (\CORE.CLOSEFILE STREAM)
                            (FDEVOP 'DELETEFILE SDEV STREAM SDEV T]
        (\DEFINEDEVICE NAME DEV)
        NAME])
)

(DECLARE%: DONTEVAL@LOAD DOCOPY
```

(**PRINTERDEVICE** 'LPT)
```
)
```

;; for backward compatibility

```
(MOVD? 'NILL 'PRINTERMODE)
```

(RPAQ? **DEFAULTPRINTINGHOST**)

(RPAQ? **DEFAULTPRINTERTYPE** 'INTERPRESS)

(RPAQ? **EMPRESS.SCRATCH**)

(RPAQ? **EMPRESS#SIDES** T)

(RPAQ? **PRINTFILETYPES** NIL)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS DEFAULTPRINTINGHOST DEFAULTPRINTERTYPE EMPRESS#SIDES PRINTERTYPES PRINTFILETYPES)
)

;; Converting text files to imagestreams

(RPAQ? **TEXTDEFAULTTABS** (LIST 20320))

(RPAQ? **TEXTDEFAULTPAGEREGION** (CREATEREGION 2794 1905 18415 24765))

;; TEXTDEFAULTTABS Hack, mica equivalent of 8 inches. NOT USED ANYWHERE

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS TEXTDEFAULTTABS TEXTDEFAULTPAGEREGION)
)

(DEFINEQ

(**TEXTTOIMAGEFILE**
  [LAMBDA (FILE IMAGEFILE IMAGETYPE FONTS HEADING TABS OPTIONS)   ; Edited 26-Aug-87 14:23 by Snow

;;; Generic function for converting PSPOOL format text files into image files

    (RESETLST
        [PROG (IMAGESTREAM INPUT-STREAM INPUT-FILENAME)   ; FONTARRAY is an array of font-descriptors
              [RESETSAVE [SETQ INPUT-STREAM (OPENSTREAM FILE 'INPUT 'OLD 8 '((SEQUENTIAL T]
                         '(PROGN (CLOSEF? OLDVALUE]
              (SETQ INPUT-FILENAME (FULLNAME INPUT-STREAM))
          ;; Strip off the extension if we are generating the name from the INFILE, so that OPENIMAGESTREAM can pack on the appropriate
          ;; extension
              [RESETSAVE [SETQ IMAGESTREAM (OPENIMAGESTREAM
                                              (OR IMAGEFILE (PACKFILENAME 'EXTENSION NIL 'VERSION NIL 'BODY
                                                                  INPUT-FILENAME))
                                              IMAGETYPE
                                              (APPEND [AND (NEQ HEADING T)
                                                           (LIST 'HEADING (OR HEADING (CONCAT INPUT-FILENAME "
" (GETFILEINFO INPUT-STREAM 'CREATIONDATE]
                                                      (APPEND (LIST 'DOCUMENT.NAME INPUT-FILENAME 'TABS TABS
                                                                    'FONTS FONTS)
                                                              OPTIONS]
                         '(AND RESETSTATE (DELFILE (CLOSEF? OLDVALUE]
                                                                   ; Make \BIN return NIL on EOS
              (**COPY.TEXT.TO.IMAGE** INPUT-STREAM IMAGESTREAM FONTS TABS)
              (RETURN (LIST (CLOSEF INPUT-STREAM)
                            (CLOSEF IMAGESTREAM]])

(**COPY.TEXT.TO.IMAGE**
  [LAMBDA (INFILE IMAGESTREAM FONTS TABS)   ; Edited 3-Mar-2023 23:46 by rmk
                                                              ; Edited 20-Jul-2022 17:14 by rmk
                                                              ; Edited 8-Oct-2021 22:23 by rmk:
                                                              ; Edited 10-Apr-95 21:23 by rmk:

    ;; Copy text to an image stream, obeying PSPOOL control characters

    (LET*
     [(IMAGESTREAM (GETSTREAM IMAGESTREAM 'OUTPUT))
      (RIGHTMAR (DSPRIGHTMARGIN NIL IMAGESTREAM))
      (FONTARRAY (FONTMAPARRAY FONTS))
      (MAXFONT (ARRAYSIZE FONTARRAY))
      (INSTRM (GETSTREAM INFILE 'INPUT))
      DEFTAB C FC (EOSP (GETFILEINFO INSTRM 'ENDOFSTREAMOP]
     ;; RMK: EOS function changed to NILL from ZERO. 0 in low-order bits is OK in UNICODE, when we switch

     (SETFILEINFO INSTRM 'ENDOFSTREAMOP (FUNCTION NILL))
     [**while** (SETQ C (\INCCODE.EOLC INSTRM ANY.EOLC))
        do (COND
              ((AND RIGHTMAR (> (DSPXPOSITION NIL IMAGESTREAM)
                                RIGHTMAR))   ; Not to walk off the right edge of the paper
               (TERPRI IMAGESTREAM)))
            (COND
              ([> C (CONSTANT (APPLY (FUNCTION MAX)
                                     (CHARCODE (^F CR LF ^L EOL TAB]
               (\OUTCHAR IMAGESTREAM C))
              (T (SELCHARQ C

```
                    (^F                                                    ; Font shift
                 ;; For FX-XP-9 printer:SETXY interpress command to avoid printer's BUG(Take)

                    (DSPXPOSITION (IPLUS (DSPXPOSITION NIL IMAGESTREAM)
                                            1)
                         IMAGESTREAM)
                    [SELCHARQ (SETQ FC (\INCCODE.EOLC INSTRM ANY.EOLC))
                       (^T                                         ; tab to absolute pos.
                          (CL:UNLESS (SETQ FC (\INCCODE INSTRM))
                               (\OUTCHAR IMAGESTREAM (CHARCODE ^F))
                               (\OUTCHAR IMAGESTREAM (CHARCODE ^T))
                               (RETURN))
                 ;; DEFAULTTAB is now a constant defined here as 36 = 1/2 inch.  Maybe that should be scaled by the
                 ;; stream's scale factor vis a vis points, not related to the current font.  If you are tabbing for alignment,
                 ;; you wouldn't want it to be ragged based on what font one line is in compare to another.
                 ;; TEXTDEFAULTTAB is a hack that should be removed.

                          [SETQ FC
                            (IF TABS
                                THEN (OR (CAR (NTH TABS FC))
                                          (ERROR "Undefined absolute tab number" FC))
                                ELSE (TIMES FC (OR DEFTAB (SETQ DEFTAB
                                                            (TIMES 8 (CHARWIDTH (CHARCODE SPACE)
                                                                        (FONTCREATE (ELT FONTARRAY 1)
                                                                            NIL NIL NIL IMAGESTREAM]
                          (DSPXPOSITION FC IMAGESTREAM))
                       (NIL (\OUTCHAR IMAGESTREAM (CHARCODE ^F))
                                                                  ; EOS after ^F
                            (RETURN))
                    (COND
                       ((AND (>= MAXFONT FC)
                             (NEQ FC 0))
                        (DSPFONT (ELT FONTARRAY FC)
                             IMAGESTREAM))
                       (T (\OUTCHAR IMAGESTREAM (CHARCODE ^F))
                          (\OUTCHAR IMAGESTREAM FC])
                 (EOL  ;; Assumes that CR and possibly following LF denote a single EOL, independent of the EOL convention and
                       ;; independent of whether the file was opened binary or text.  Originally, this function tried to discriminate, treating
                       ;; an LF in a CR-mode file as a line-feed and a CR in an LF file as a carriage-return.  But these formatting effects
                       ;; cannot be guaranteed across text-file transfers (which is all that it makes sense to print), so it is silly to take them
                       ;; seriously.  Given that just this information can be lost in text-mode file transfers, we adopt here the 99% correct
                       ;; solution, which is to treat all instances of CR, CRLF, and LF as end-of-line (ANY.EOLC above)

                    (TERPRI IMAGESTREAM))
                 (LF                                              ; Isolatedx LF, see comment at CR
                    (TERPRI IMAGESTREAM))
                 (TAB (OR (LET* [(LEFTMARGIN (DSPLEFTMARGIN NIL IMAGESTREAM))
                                 (TAB.WIDTH (TIMES (CHARWIDTH (CHARCODE SPACE)
                                                        IMAGESTREAM)
                                                8))
                                 (CURRENT.X (- (DSPXPOSITION NIL IMAGESTREAM)
                                                LEFTMARGIN))
                                 (CURRENT.STOP (- CURRENT.X (REMAINDER CURRENT.X TAB.WIDTH]
                              (NLSETQ (RELMOVETO (- (+ CURRENT.STOP TAB.WIDTH)
                                                     CURRENT.X)
                                            0 IMAGESTREAM)))
                          (\OUTCHAR IMAGESTREAM C)))
                 (\OUTCHAR IMAGESTREAM C]
          (SETFILEINFO INSTRM 'ENDOFSTREAMOP EOSP])
)

(DEFINEQ
```

(**BLTSHADE.GENERICPRINTER**

```
  [LAMBDA (TEXTURE STREAM DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT OPERATION CLIPPINGREGION SHADESCALE)
                                                                  ; Edited 26-Aug-87 14:23 by Snow
    (PROG (FINALREGION SCRATCHBM BMWIDTH BMHEIGHT)
      ;; do the clipping to reduce the size of the scratch bitmap created.  This also keeps Press from doing the wrong thing.
                                                              ; don't do anything if clipped region is empty
          (OR (SETQ FINALREGION (INTERSECTREGIONS (CREATEREGION DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT)
                                     (DSPCLIPPINGREGION NIL STREAM)))
              (RETURN))
          (AND CLIPPINGREGION (OR (SETQ FINALREGION (INTERSECTREGIONS FINALREGION CLIPPINGREGION))
                              (RETURN)))
          (COND
             ([ZEROP (SETQ BMWIDTH (FIXR (FQUOTIENT (fetch (REGION WIDTH) of FINALREGION)
                                            SHADESCALE]
              (RETURN)))
          (COND
             ([ZEROP (SETQ BMHEIGHT (FIXR (FQUOTIENT (fetch (REGION HEIGHT) of FINALREGION)
                                            SHADESCALE]
              (RETURN)))
          (SETQ SCRATCHBM (BITMAPCREATE BMWIDTH BMHEIGHT))
          (\BLTSHADE.BITMAP TEXTURE SCRATCHBM 0 0 NIL NIL 'REPLACE)
          (BITBLT SCRATCHBM 0 0 STREAM (fetch (REGION LEFT) of FINALREGION)
                 (fetch (REGION BOTTOM) of FINALREGION)
```

```
                        NIL NIL 'INPUT OPERATION])
)
```

;; hack for printers that can't really BLTSHADE

;; stuff to support hardcopy streams on the display.

```
(DEFINEQ
```

(**MAKEHARDCOPYSTREAM**
```
  [LAMBDA (DISPLAYSTREAM IMAGETYPE)                                  ; Edited 26-Aug-87 14:23 by Snow
```

;;; creates a hardcopy stream from a display stream.

```
     (DECLARE (GLOBALVARS \HDCPYDISPLAYIMAGEOPS))
     (PROG [(DS (COND
                  ((DISPLAYSTREAMP DISPLAYSTREAM))
                  ((WINDOWP DISPLAYSTREAM)
                   (WINDOWPROP DISPLAYSTREAM 'DSP))
                  ((NULL DISPLAYSTREAM)
                   (DSPCREATE))
                  (T (\ILLEGAL.ARG DISPLAYSTREAM]
           (replace (STREAM IMAGEOPS) of DS with \HDCPYDISPLAYIMAGEOPS)
           [STREAMPROP DS 'HARDCOPYIMAGETYPE (OR IMAGETYPE (CAR (PRINTERPROP (PRINTERTYPE)
                                                                     'CANPRINT]
                                                     ; set the bout fn to one that updates the mica fields and sets the
                                                     ; position from them.
           (replace (STREAM STRMBOUTFN) of DS with (FUNCTION \HDCPYDSPPRINTCHAR))
           (replace (STREAM OUTCHARFN) of DS with (FUNCTION \HDCPYDSPPRINTCHAR))
                                                     ; set the parameters that are different to initialize the mica
                                                     ; defined fields.
           (DSPFONT (DSPFONT NIL DS)
                 DS)
           (DSPXPOSITION 0 DS)
           (DSPYPOSITION 0 DS)
           (DSPRIGHTMARGIN (DSPRIGHTMARGIN NIL DS)
                 DS)
           (RETURN DS])
```

(**UNMAKEHARDCOPYSTREAM**
```
  [LAMBDA (DISPLAYSTREAM)                                            ; Edited 26-Aug-87 14:23 by Snow
```

;;; returns a hardcopy stream to a display stream.

```
     (DECLARE (GLOBALVARS \DISPLAYIMAGEOPS))
     (PROG [(DS (COND
                  ((DISPLAYSTREAMP DISPLAYSTREAM))
                  ((WINDOWP DISPLAYSTREAM)
                   (WINDOWPROP DISPLAYSTREAM 'DSP))
                  (T (\ILLEGAL.ARG DISPLAYSTREAM]
           (COND
              ((FMEMB 'HARDCOPY (IMAGESTREAMTYPE DS)))
              (T (RETURN DS)))
           (replace (STREAM IMAGEOPS) of DS with \DISPLAYIMAGEOPS)
           (STREAMPROP DS 'HARDCOPYIMAGETYPE NIL)                    ; restore the bout fn
           (replace (STREAM STRMBOUTFN) of DS with (FUNCTION \DSPPRINTCHAR))
           (replace (STREAM OUTCHARFN) of DS with (FUNCTION \DSPPRINTCHAR))
           (RETURN DS])
```

(**HARDCOPYSTREAMTYPE**
```
  [LAMBDA (IMAGESTREAM)                                              ; Edited 26-Aug-87 14:24 by Snow
```

;;; returns the type of a hard copy stream which is either PRESS or INTERPRESS.

```
     (LET ((STREAM (\OUTSTREAMARG IMAGESTREAM T)))
          (AND STREAM (STREAMPROP STREAM 'HARDCOPYIMAGETYPE])
```

(**CHARWIDTH.HDCPYDISPLAY**
```
  [LAMBDA (STREAM CHARCODE)                                          ; Edited 26-Aug-87 14:24 by Snow
                                                                    ; gets the width of a character code in a hardcopy stream.
                                                                    ; Should be updated for spacefactor
     (IQUOTIENT (IPLUS (\FGETCHARIMAGEWIDTH (FONTCREATE (ffetch (\DISPLAYDATA DDFONT) of (ffetch IMAGEDATA
                                                                                          of STREAM))
                                                       NIL NIL NIL (STREAMPROP STREAM 'HARDCOPYIMAGETYPE))
                                           CHARCODE)
                       (CONSTANT IHALFMICASPERPT))
                (CONSTANT IMICASPERPT])
```

(**DSPFONT.HDCPYDISPLAY**
```
  [LAMBDA (HCPYDSTREAM FONT)                                         ; Edited 12-Jan-88 16:18 by jds
```

;; changes the font of a hardcopy display stream. Does what the display does then puts the hardcopy widths where they can be found {FOR NOW
;; USE THE DDCHARIMAGEWIDTHS FIELD}

```
(LET [(FD (AND FONT (FONTCREATE FONT NIL NIL NIL (STREAMPROP HDCPYDSTREAM 'HARDCOPYIMAGETYPE]
    (PROG1 (\DSPFONT.DISPLAY HDCPYDSTREAM FD)
        [AND FD (PROG ((DD (fetch IMAGEDATA of HDCPYDSTREAM)))
```
                                                                    ; For now, use a streamprop instead of a special field in the
                                                                    ; display data
                                                                    ; Scale widths to printer device units, so we don't have to fetch
                                                                    ; the constants to scale by for every char we print
```
                    (replace DDCHARIMAGEWIDTHS of DD
                        with (PROG (W OLDWIDTH (SCALE (FONTPROP FD 'SCALE))
                                        (CSINFO (\GETCHARSETINFO (fetch (STREAM CHARSET) of HDCPYDSTREAM)
                                                    FD)))
```
                        ;; set linefeed from scaled height. This may be off by almost half a pixel per line but it is better than not
                        ;; doing so.
```
                            [freplace DDLINEFEED of DD
                                with (IMINUS (FIXR (QUOTIENT (fetch \SFHeight of FD)
                                                            SCALE]
                            [COND
                                ((EQP SCALE (CONSTANT MICASPERPT))
                                 (RETURN (fetch (CHARSETINFO WIDTHS) of CSINFO]
                            (SETQ W (\CREATECSINFOELEMENT))
                            (SETQ OLDWIDTH (fetch (CHARSETINFO WIDTHS) of CSINFO))
                            (SETQ SCALE (FQUOTIENT (CONSTANT MICASPERPT)
                                            SCALE))
                            [for I from 0 to \MAXTHINCHAR
                                do (\FSETWIDTH W I (FIXR (FTIMES (\FGETWIDTH OLDWIDTH I)
                                                                SCALE]
                            (RETURN W])])
```

(**DSPRIGHTMARGIN.HDCPYDISPLAY**
```
  [LAMBDA (DISPLAYSTREAM XPOSITION)                                  ; Edited 26-Aug-87 14:25 by Snow
```
;;; Sets the right margin that determines when a cr is inserted by print for the hardcopy display stream.

  ;; mica right margin is kept accurately using 35.27778. Since the updating at each character is done with 35, this may lead to a small error.

```
    (PROG1 (\DSPRIGHTMARGIN.DISPLAY DISPLAYSTREAM XPOSITION)
        [AND XPOSITION (replace (\DISPLAYDATA DDMICARIGHTMARGIN) of (fetch IMAGEDATA of DISPLAYSTREAM)
                            with (FIX (FTIMES XPOSITION (CONSTANT MICASPERPT]])
```

(**DSPXPOSITION.HDCPYDISPLAY**
```
  [LAMBDA (HARDCOPYSTREAM XPOSITION)                                 ; Edited 26-Aug-87 14:25 by Snow
                                                                    ; updates the mica xposition too.
    (PROG1 (\DSPXPOSITION.DISPLAY HARDCOPYSTREAM XPOSITION)
        (AND XPOSITION (\HDCPYDISPLAY.FIX.XPOS HARDCOPYSTREAM)))])
```

(**DSPYPOSITION.HDCPYDISPLAY**
```
  [LAMBDA (HARDCOPYSTREAM YPOSITION)                                 ; Edited 26-Aug-87 14:25 by Snow
                                                                    ; updates the mica xposition too.
    (PROG1 (\DSPYPOSITION.DISPLAY HARDCOPYSTREAM YPOSITION)
        (AND YPOSITION (\HDCPYDISPLAY.FIX.YPOS HARDCOPYSTREAM)))])
```

(**STRINGWIDTH.HDCPYDISPLAY**
```
  [LAMBDA (STREAM STR RDTBL)                                         ; Edited 26-Aug-87 14:25 by Snow
                                                                    ; Returns the width of for the current font/spacefactor in hardcopy
                                                                    ; stream STREAM.
    (LET [(HARDCOPYFD (FONTCREATE (ffetch (\DISPLAYDATA DDFONT) of (ffetch IMAGEDATA of STREAM))
                            NIL NIL NIL (STREAMPROP STREAM 'HARDCOPYIMAGETYPE]
        (IQUOTIENT (IPLUS (\STRINGWIDTH.GENERIC STR HARDCOPYFD RDTBL (\FGETCHARIMAGEWIDTH HARDCOPYFD
                                                                        (CHARCODE SPACE)))
                        (CONSTANT IHALFMICASPERPT))
                (CONSTANT IMICASPERPT])
```

(**STRINGWIDTH.HCPYDISPLAYAUX**
```
  [LAMBDA (STR FONT RDTBL SPACEWIDTH)                                ; Edited  3-Apr-87 13:48 by jop
```
  ;; Returns the width of STR with SPACEWIDTH for the width of spaces. RDTBL has already been coerced, so no FLG is needed

  ;; This is cloned in \STRINGWIDTH.HCPYDISPLAYAUX by straight substitution -- (PUTDEF (QUOTE \STRINGWIDTH.HCPYDISPLAYAUX)
  ;; (QUOTE FNS) (SUBLIS (QUOTE ((IMAGEWIDTHS . IMAGEWIDTHS) (\FGETIMAGEWIDTH . \FGETIMAGEWIDTH)
  ;; (\FGETCHARIMAGEWIDTH . \FGETCHARIMAGEWIDTH))) (GETDEF (QUOTE \STRINGWIDTH.GENERIC))))

  ;; \MAPPNAME uses WIDTHSBASE CSET TOTALWIDTH FONT SPACEWIDTH free, so these become special in bytecompiler

```
    (PROG NIL
        [COND
            [(LITATOM STR)
             (if RDTBL
                 then (GO SLOW)
                 else (RETURN (for C WIDTHSBASE CSET inatom STR sum [COND
                                                                        ((NEQ CSET (\CHARSET C))
                                                                         (SETQ CSET (\CHARSET C))
                                                                         (SETQ WIDTHSBASE (ffetch (CHARSETINFO
                                                                                                        IMAGEWIDTHS)
```

```
                                                                             of (\GETCHARSETINFO
                                                                                    CSET FONT]
                                                       (COND
                                                          ((EQ C (CHARCODE SPACE))
                                                           SPACEWIDTH)
                                                          (T (\FGETIMAGEWIDTH WIDTHSBASE (\CHAR8CODE
                                                                                            C]
                    ((STRINGP STR)
                     (RETURN (LET ((TOTAL 0)
                                   ESC ESCWIDTH WIDTHSBASE CSET)
                                  [COND
                                     (RDTBL                                 ; Count delimiting quotes and internal escapes
                                          (SETQ TOTAL (UNFOLD (\FGETCHARIMAGEWIDTH FONT (CHARCODE %"))
                                                              2))
                                          (SETQ ESC (fetch (READTABLEP ESCAPECHAR) of RDTBL))
                                          (SETQ ESCWIDTH (\FGETCHARIMAGEWIDTH FONT ESC]
                                  [for C instring STR
                                     do [COND
                                           ((NEQ (\CHARSET C)
                                                 CSET)                      ; Get the widths vector for this character set
                                            (SETQ CSET (\CHARSET C))
                                            (SETQ WIDTHSBASE (ffetch (CHARSETINFO IMAGEWIDTHS) of (\GETCHARSETINFO CSET
                                                                                                     FONT]
                                        (add TOTAL (COND
                                                      ((EQ C (CHARCODE SPACE))
                                                       SPACEWIDTH)
                                                      (T (IPLUS (\FGETIMAGEWIDTH WIDTHSBASE (\CHAR8CODE C))
                                                                (COND
                                                                   ((AND RDTBL (OR (EQ C (CHARCODE %"))
                                                                                   (EQ C ESC)))
                                                                    ; String char must be escaped
                                                                    ESCWIDTH)
                                                                   (T 0]
                                  TOTAL]
             SLOW
                                                                           ; Do the general case here
                 (RETURN (LET ((TOTALWIDTH 0)
                               WIDTHSBASE CSET (FONT FONT)
                               (SPACEWIDTH SPACEWIDTH))
                              (DECLARE (SPECVARS TOTALWIDTH WIDTHSBASE CSET FONT SPACEWIDTH))
                              (\MAPPNAME [FUNCTION (LAMBDA (DUMMY CC)
                                                      (add TOTALWIDTH (COND
                                                                         ((EQ CC (CHARCODE SPACE))
                                                                          SPACEWIDTH)
                                                                         ((EQ CSET (\CHARSET CC))
                                                                          (\FGETIMAGEWIDTH WIDTHSBASE (\CHAR8CODE CC)))
                                                                         (T (SETQ CSET (\CHARSET CC))
                                                                            (SETQ WIDTHSBASE (ffetch (CHARSETINFO
                                                                                                        IMAGEWIDTHS)
                                                                                              of (\GETCHARSETINFO
                                                                                                    CSET FONT)))
                                                                            (\FGETIMAGEWIDTH WIDTHSBASE (\CHAR8CODE
                                                                                                           CC]
                                         STR RDTBL RDTBL *PRINT-LEVEL* *PRINT-LENGTH*)
                              TOTALWIDTH]))
```

(**HDCPYBLTCHAR**

```
   [LAMBDA (CHARCODE DISPLAYSTREAM DISPLAYDATA)                            ; Edited 26-Aug-87 14:26 by Snow

     ;; puts a character on a hardcopy display stream.  Much of the information needed by the BitBlt microcode is prestored by the routines that change
     ;; it.  This is kept in the BitBltTable.                              ; knows about the representation of a DisplayStream.
     (DECLARE (LOCALVARS . T))
     (PROG (LOCAL1 RIGHT LEFT CURX (CHAR8CODE (\CHAR8CODE CHARCODE))
                   MICARIGHT)
           (COND
              ((NEQ (ffetch DDCHARSET of DISPLAYDATA)
                    (\CHARSET CHARCODE))
               (\CHANGECHARSET.HCPYDISPLAY DISPLAYDATA (\CHARSET CHARCODE)
                      DISPLAYSTREAM)))
           [COND
              ((ffetch (\DISPLAYDATA DDSlowPrintingCase) of DISPLAYDATA)
               (RETURN (\SLOWHDCPYBLTCHAR CHARCODE DISPLAYSTREAM]
       CRLP
           (SETQ CURX (ffetch DDXPOSITION of DISPLAYDATA))
           [COND
              ((IGREATERP (SETQ MICARIGHT (IPLUS (ffetch (\DISPLAYDATA DDMICAXPOS) of DISPLAYDATA)
                                                 (\FGETWIDTH (ffetch (\DISPLAYDATA DDCHARIMAGEWIDTHS) of
                                                                       DISPLAYDATA
                                                                       )
                                                             CHAR8CODE)))
                          (ffetch (\DISPLAYDATA DDMICARIGHTMARGIN) of DISPLAYDATA))
                                                                           ; would go past right margin, force a cr
               (COND
                  ((IGREATERP CURX (ffetch DDLeftMargin of DISPLAYDATA))
                                                                           ; don't bother CR if position is at left margin anyway.  This also
                                                                           ; serves to break the loop.
                   (\DSPPRINTCR/LF (CHARCODE EOL)
```

```
                              DISPLAYSTREAM)                                      ; reuse the code in the test of this conditional rather than repeat it
                                                                                 ; here.
                         (GO CRLP]
                    (freplace (\DISPLAYDATA DDMICAXPOS) of DISPLAYDATA with MICARIGHT)
        ;; update the display stream x position.  Make sure that there is at least one point width for each character.
              [freplace DDXPOSITION of DISPLAYDATA with (IMAX (ADD1 CURX)
                                                              (IQUOTIENT (IPLUS MICARIGHT (CONSTANT IHALFMICASPERPT))
                                                                         (CONSTANT IMICASPERPT]
                                                              ; transforms an x coordinate into the destination coordinate.
              (SETQ CURX (IPLUS CURX (ffetch DDXOFFSET of DISPLAYDATA)))
              (SETQ RIGHT (IPLUS CURX (\DSPGETCHARWIDTH CHAR8CODE DISPLAYDATA)))
              (COND
                 ((IGREATERP RIGHT (SETQ LOCAL1 (ffetch DDClippingRight of DISPLAYDATA)))
                                                                                 ; character overlaps right edge of clipping region.
                  (SETQ RIGHT LOCAL1)))
              (SETQ LEFT (COND
                           ((IGREATERP CURX (SETQ LOCAL1 (ffetch DDClippingLeft of DISPLAYDATA)))
                            CURX)
                           (T LOCAL1)))
              (RETURN (COND
                         ((AND (ILESSP LEFT RIGHT)
                               (NEQ (fetch PBTHEIGHT of (SETQ LOCAL1 (ffetch DDPILOTBBT of DISPLAYDATA)))
                                    0))
                          (.WHILE.TOP.DS. DISPLAYSTREAM (freplace PBTDESTBIT of LOCAL1 with LEFT)
                               (freplace PBTWIDTH of LOCAL1 with (IDIFFERENCE RIGHT LEFT))
                               (freplace PBTSOURCEBIT of LOCAL1 with (IDIFFERENCE (IPLUS (\DSPGETCHAROFFSET CHAR8CODE
                                                                                               DISPLAYDATA)
                                                                                        LEFT)
                                                                                 CURX))
                               (\PILOTBITBLT LOCAL1 0))
                          T])
```

(\ **HDCPYDISPLAY.FIX.XPOS**
```
  [LAMBDA (HARDCOPYSTREAM)                                          ; Edited 26-Aug-87 14:26 by Snow
```

;;; updates the mica X position from the x position in the display stream. This is called whenever the X position changes in a hardcopy stream.

```
    (PROG ((DD (fetch IMAGEDATA of HARDCOPYSTREAM)))
          (replace (\DISPLAYDATA DDMICAXPOS) of DD with (FIX (FTIMES (fetch (\DISPLAYDATA DDXPOSITION) of DD)
                                                                     (CONSTANT MICASPERPT])
```

(\ **HDCPYDISPLAY.FIX.YPOS**
```
  [LAMBDA (HARDCOPYSTREAM)                                          ; Edited 26-Aug-87 14:26 by Snow
```

;;; updates the mica Y position from the Y position in the display stream. This is called whenever the Y position changes in a hardcopy stream.

```
    (PROG ((DD (fetch IMAGEDATA of HARDCOPYSTREAM)))
          (replace (\DISPLAYDATA DDMICAYPOS) of DD with (FIX (FTIMES (fetch (\DISPLAYDATA DDYPOSITION) of DD)
                                                                     (CONSTANT MICASPERPT])
```

(\ **HDCPYDISPLAYINIT**
```
  [LAMBDA NIL                                                       ; Edited 26-Aug-87 14:26 by Snow
```

;;; Initializes global variables for the hardcopy Display device. This device appears to the user as an INTERPRESS or PRESS device meaning units in
;;; micas but outputs to the screen. Much of this code was borrowed from the display case.

```
    (DECLARE (GLOBALVARS \HDCPYDISPLAYIMAGEOPS))
    (SETQ \HDCPYDISPLAYIMAGEOPS (create IMAGEOPS using \DISPLAYIMAGEOPS IMAGETYPE _ '(HARDCOPY DISPLAY)
                                        IMFONT _ (FUNCTION \DSPFONT.HDCPYDISPLAY)
                                        IMRIGHTMARGIN _ (FUNCTION \DSPRIGHTMARGIN.HDCPYDISPLAY)
                                        IMXPOSITION _ (FUNCTION \DSPXPOSITION.HDCPYDISPLAY)
                                        IMYPOSITION _ (FUNCTION \DSPYPOSITION.HDCPYDISPLAY)
                                        IMSTRINGWIDTH _ (FUNCTION \STRINGWIDTH.HDCPYDISPLAY)
                                        IMCHARWIDTH _ (FUNCTION \CHARWIDTH.HDCPYDISPLAY])
```

(\ **HDCPYDSPPRINTCHAR**
```
  [LAMBDA (STREAM CHARCODE)                                         ; Edited 26-Aug-87 14:27 by Snow
```

;;; displays a character on a hardcopy display stream. This uses a display font but updates the x position according to hardcopy widths.

```
    (PROG ((DD (fetch IMAGEDATA of STREAM)))
          (\CHECKCARET STREAM)
          (RETURN (SELECTC (fetch CCECHO of (\SYNCODE \PRIMTERMSA CHARCODE))
                     (INDICATE.CCE [PROG ((CC CHARCODE))
                                         (add (fetch CHARPOSITION of STREAM)
                                              (IPLUS (COND
                                                        ((IGREATERP CC 127)
                                                              ; META character
                                                         (\HDCPYBLTCHAR (CHARCODE %#)
                                                             STREAM DD)
                                                         (SETQ CC (LOGAND CC 127))
                                                         1)
```

```
                                                    (T 0))
                                                (COND
                                                    ((ILESSP CC 32)
                                                        ; CONTROL character
                                                     (\HDCPYBLTCHAR (CHARCODE ^)
                                                           STREAM DD)
                                                      (SETQ CC (LOGOR CC 64))
                                                     1)
                                                    (T 0))
                                                (PROGN (\HDCPYBLTCHAR CC STREAM DD)
                                                       1])
                        (SIMULATE.CCE (SELCHARQ CHARCODE
                                          ((EOL CR LF)
                                              (\DSPPRINTCR/LF CHARCODE STREAM)
                                              (replace CHARPOSITION of STREAM with 0))
                                          (ESCAPE (\HDCPYBLTCHAR (CHARCODE $)
                                                       STREAM DD)
                                              (add (fetch CHARPOSITION of STREAM)
                                                   1))
                                          (BELL                        ; make switching of bits uninterruptable but allow interrupts
                                                                       ; between flashes.
                                              (SELECTQ (MACHINETYPE)
                                                  (DANDELION [PLAYTUNE '((880 . 2500])
                                                  (FLASHWINDOW (WFROMDS STREAM))))
                                          (TAB (PROG (TABWIDTH (SPACEWIDTH (CHARWIDTH (CHARCODE SPACE)
                                                                                      STREAM)))
                                                   (SETQ TABWIDTH (UNFOLD SPACEWIDTH 8))
                                                   (COND
                                                       ((IGREATERP (\DISPLAYSTREAMINCRXPOSITION
                                                                        (SETQ TABWIDTH
                                                                           (IDIFFERENCE TABWIDTH
                                                                               (MOD (IDIFFERENCE (fetch DDXPOSITION
                                                                                                     of DD)
                                                                                          (ffetch DDLeftMargin
                                                                                             of DD))
                                                                                   TABWIDTH)))
                                                                        DD)
                                                                   (ffetch DDRightMargin of DD))
                                                                       ; tab was past rightmargin, force cr.
                                                           (\DSPPRINTCR/LF (CHARCODE EOL)
                                                               STREAM))
                                                              ; return the number of spaces taken.
                                                   (add (fetch CHARPOSITION of STREAM)
                                                        (IQUOTIENT TABWIDTH SPACEWIDTH))))
                                          (PROGN                        ; this case was copied from \DSCCOUT.
                                              (\HDCPYBLTCHAR CHARCODE STREAM DD)
                                              (add (fetch CHARPOSITION of STREAM)
                                                   1))))
                        (REAL.CCE (SELECTC CHARCODE
                                       ((CHARCODE (EOL CR LF))
                                           (\DSPPRINTCR/LF CHARCODE STREAM)
                                           (replace CHARPOSITION of STREAM with 0))
                                       (ERASECHARCODE
                                           (DSPBACKUP (CHARWIDTH (CHARCODE A)
                                                          STREAM)
                                               STREAM)           ; line buffering routines have already taken care of backing up
                                                                 ; the position
                                           0)
                                       (PROGN (\HDCPYBLTCHAR CHARCODE STREAM DD)
                                              (add (fetch CHARPOSITION of STREAM)
                                                   1))))
                        (IGNORE.CCE)
                        (SHOULDNT])
```

(**SLOWHDCPYBLTCHAR**

```
  [LAMBDA (CHARCODE DISPLAYSTREAM)                                        ; Edited  9-Nov-89 14:37 by gadener
```

;;; IS THIS CODE JUST GOING TO DUPLICATE AND GET OUT OF SYNC WITH \SLOWBLTCHAR? KBR 1-FEB-86. *

;;; THIS HAS BEEN SEPARATED OUT BUT HASN'T BEEN EDITTED TO DO CORRECT THING WRT UPDATING MICA FIELDS.

   ;; case of BLTCHAR where either font is rotated or destination is a color bitmap. DISPLAYSTREAM is known to be a hardcopy display stream.

```
  (PROG (ROTATION (CHAR8CODE (\CHAR8CODE CHARCODE))
              (DD (ffetch (STREAM IMAGEDATA) of DISPLAYSTREAM)))
        (SETQ ROTATION (ffetch (FONTDESCRIPTOR ROTATION) of (ffetch (\DISPLAYDATA DDFONT) of DD)))
        (COND
            [(EQ 0 ROTATION)
             (PROG (NEWX LEFT RIGHT (CURX (ffetch (\DISPLAYDATA DDXPOSITION) of DD))
                        PILOTBBT DESTBIT WIDTH SOURCEBIT)
                   (SETQ NEWX (IPLUS CURX (\DSPGETCHARWIDTH CHAR8CODE DD)))
                   [COND
                       ((IGREATERP NEWX (ffetch (\DISPLAYDATA DDRightMargin) of DD))
                                                       ; past RIGHT margin, force eol
                        (\DSPPRINTCR/LF (CHARCODE EOL)
                            DISPLAYSTREAM)
                        (SETQ CURX (ffetch (\DISPLAYDATA DDXPOSITION) of DD))
```

```
                              (SETQ NEWX (IPLUS CURX (\DSPGETCHARWIDTH CHAR8CODE DD]
                                                                          ; update the x position.
                        (freplace (\DISPLAYDATA DDXPOSITION) of DD with NEWX)
                        (SETQ CURX (\DSPTRANSFORMX CURX DD))
                        (SETQ LEFT (IMAX (ffetch (\DISPLAYDATA DDClippingLeft) of DD)
                                         CURX))
                        (SETQ RIGHT (IMIN (ffetch (\DISPLAYDATA DDClippingRight) of DD)
                                          (\DSPTRANSFORMX NEWX DD)))
                        (SETQ PILOTBBT (ffetch (\DISPLAYDATA DDPILOTBBT) of DD))
                        (COND
                           ((AND (ILESSP LEFT RIGHT)
                                 (NEQ (ffetch (PILOTBBT PBTHEIGHT) of PILOTBBT)
                                      0))
                            (SETQ DESTBIT LEFT)
                            (SETQ WIDTH (IDIFFERENCE RIGHT LEFT))
                            (SETQ SOURCEBIT (IDIFFERENCE (IPLUS (\DSPGETCHAROFFSET CHAR8CODE DD)
                                                               LEFT)
                                                        CURX))
                            (SELECTQ (ffetch (BITMAP BITMAPBITSPERPIXEL) of (ffetch (\DISPLAYDATA DDDestination)
                                                                                    of DD))
                               (1)
                               (4 (SETQ DESTBIT (LLSH DESTBIT 2))
                                  (SETQ WIDTH (LLSH WIDTH 2))
                                  (SETQ SOURCEBIT (LLSH SOURCEBIT 2)))
                               (8 (SETQ DESTBIT (LLSH DESTBIT 3))
                                  (SETQ WIDTH (LLSH WIDTH 3))
                                  (SETQ SOURCEBIT (LLSH SOURCEBIT 3)))
                               (SHOULDNT))
                            (.WHILE.TOP.DS. DISPLAYSTREAM (freplace (PILOTBBT PBTDESTBIT) of PILOTBBT with DESTBIT)
                                  (freplace (PILOTBBT PBTWIDTH) of PILOTBBT with WIDTH)
                                  (freplace (PILOTBBT PBTSOURCEBIT) of PILOTBBT with SOURCEBIT)
                                  (\PILOTBITBLT PILOTBBT 0))
                            T]
                 (T                                              ; handle rotated fonts
                    (PROG (YPOS HEIGHTMOVED CSINFO)
                          (SETQ YPOS (ffetch (\DISPLAYDATA DDYPOSITION) of DD))
                          (SETQ HEIGHTMOVED (\DSPGETCHARWIDTH CHAR8CODE DD))
                          (SETQ CSINFO (\GETCHARSETINFO (\CHARSET CHARCODE)
                                              (ffetch (\DISPLAYDATA DDFONT) of DD)))
                          (COND
                             ((EQ ROTATION 90)                         ; don't force CR for rotated fonts.
                              (\DSPYPOSITION.DISPLAY DISPLAYSTREAM (IPLUS YPOS HEIGHTMOVED))
                                                                       ; update the display stream x position.
                              (BITBLT (ffetch (CHARSETINFO CHARSETBITMAP) of CSINFO)
                                      0
                                      (\DSPGETCHAROFFSET CHAR8CODE DD)
                                      DISPLAYSTREAM
                                      (ADD1 (IDIFFERENCE (ffetch (\DISPLAYDATA DDXPOSITION) of DD)
                                                  (ffetch (CHARSETINFO CHARSETASCENT) of CSINFO)))
                                      YPOS
                                      (IPLUS (ffetch (CHARSETINFO CHARSETASCENT) of CSINFO)
                                             (ffetch (CHARSETINFO CHARSETDESCENT) of CSINFO))
                                      HEIGHTMOVED))
                             ((EQ ROTATION 270)
                              (\DSPYPOSITION.DISPLAY DISPLAYSTREAM (IDIFFERENCE YPOS HEIGHTMOVED))
                              (BITBLT (ffetch (CHARSETINFO CHARSETBITMAP) of CSINFO)
                                      0
                                      (\DSPGETCHAROFFSET CHAR8CODE DD)
                                      DISPLAYSTREAM
                                      (IDIFFERENCE (ffetch (\DISPLAYDATA DDXPOSITION) of DD)
                                             (ffetch (CHARSETINFO CHARSETDESCENT) of CSINFO))
                                      (ffetch (\DISPLAYDATA DDYPOSITION) of DD)
                                      (IPLUS (ffetch (CHARSETINFO CHARSETASCENT) of CSINFO)
                                             (ffetch (CHARSETINFO CHARSETDESCENT) of CSINFO))
                                      HEIGHTMOVED))
                             (T (ERROR "Not implemented to rotate by other than 0, 90 or 270"])
```

CHANGECHARSET.HDCPYDISPLAY

```
  [LAMBDA (DISPLAYDATA CHARSET HDCPYDSTREAM)                              ; Edited 26-Aug-87 14:27 by Snow

    ;; Called when the character set information cached in a display stream doesn't correspond to CHARSET Only sets those field that are different
    ;; from the regular DISPLAY case and uses the regular display case to get the rest.

    (\CHANGECHARSET.DISPLAY DISPLAYDATA CHARSET)
    (PROG [(FD (FONTCREATE (ffetch DDFONT of DISPLAYDATA)
                    NIL NIL NIL (STREAMPROP HDCPYDSTREAM 'HARDCOPYIMAGETYPE]
                                                                ; For now, use a streamprop instead of a special field in the
                                                                ; dispay data
                                                                ; Scale widths to micas, so we don't have to fetch the constants
                                                                ; to scale by for every char we print
          (replace DDCHARIMAGEWIDTHS of DISPLAYDATA
             with (PROG (W OLDWIDTH (SCALE (FONTPROP FD 'SCALE))
                         (CSINFO (\GETCHARSETINFO CHARSET FD)))
                        (SETQ OLDWIDTH (fetch (CHARSETINFO WIDTHS) of CSINFO))
                        (COND
                           ((EQP SCALE (CONSTANT MICASPERPT))
                            (RETURN OLDWIDTH)))
```

```
                            (SETQ W (\CREATECSINFOELEMENT))
                            (SETQ SCALE (FQUOTIENT (CONSTANT MICASPERPT)
                                                        SCALE))
                      [for I from 0 to \MAXTHINCHAR do (\FSETWIDTH W I (FIXR (FTIMES (\FGETWIDTH OLDWIDTH I)
                                                                                      SCALE]
                      (RETURN W])
)

(DECLARE%: DONTCOPY DOEVAL@COMPILE
```

;; FOLLOWING DEFINITIONS EXPORTED

```
(DECLARE%: EVAL@COMPILE

(RPAQ MICASPERPT (FQUOTIENT 2540 72))

(RPAQQ IHALFMICASPERPT 17)

(RPAQQ IMICASPERPT 35)

(RPAQQ DEFAULTTAB 36)

(CONSTANTS (MICASPERPT (FQUOTIENT 2540 72))
          (IHALFMICASPERPT 17)
          (IMICASPERPT 35)
          (DEFAULTTAB 36))
)
)
```

;; END EXPORTED DEFINITIONS
;; screen-points: 1/2 inch

```
(DECLARE%: DONTCOPY DOEVAL@COMPILE
```

;; FOLLOWING DEFINITIONS EXPORTED

```
(DEFMACRO \MICASTOPTS (MICAS)
    [COND
        ((NUMBERP MICAS)
         (QUOTIENT MICAS MICASPERPT))
        (T `(QUOTIENT ,MICAS MICASPERPT])
)
```

;; END EXPORTED DEFINITIONS

```
(DECLARE%: DONTEVAL@LOAD DOCOPY

(\HDCPYDISPLAYINIT)
)
```

;; Stuff to support MICA-unit hardcopy streams on the display

```
(DEFINEQ

(MAKEHARDCOPYMODESTREAM
  [LAMBDA (DISPLAYSTREAM IMAGETYPE)                                   ; Edited  1-Apr-88 11:25 by jds
```

;;; Creates a hardcopy-mode display stream from a normal one. That stream operates in units of micas, but displays on the screen as usual.

```
    (DECLARE (GLOBALVARS \HCPYMODEDISPLAYIMAGEOPS.PRESS \HCPYMODEDISPLAYIMAGEOPS.INTERPRESS))
    (PROG [(DS (COND
                ((DISPLAYSTREAMP DISPLAYSTREAM))
                ((WINDOWP DISPLAYSTREAM)
                 (WINDOWPROP DISPLAYSTREAM 'DSP))
                ((NULL DISPLAYSTREAM)
                 (DSPCREATE))
                (T (\ILLEGAL.ARG DISPLAYSTREAM]
        (SELECTQ [OR IMAGETYPE (SETQ IMAGETYPE (CAR (PRINTERPROP (PRINTERTYPE)
                                                           'CANPRINT]
            (PRESS                                                ; Give the stream PRESS-style imageops, so it will deal with
                                                                  ; press fonts right.
                    (replace (STREAM IMAGEOPS) of DS with \HCPYMODEDISPLAYIMAGEOPS.PRESS))
            (INTERPRESS                                           ; Give the stream INTERPRESS-style operations, so it will deal
                                                                  ; with Interpress fonts right.
                    (replace (STREAM IMAGEOPS) of DS with \HCPYMODEDISPLAYIMAGEOPS.INTERPRESS))
            NIL)
        (STREAMPROP DS 'HARDCOPYIMAGETYPE IMAGETYPE)              ; set the bout fn to one that updates the mica fields and sets the
                                                                  ; position from them.
        (replace (STREAM STRMBOUTFN) of DS with (FUNCTION \HCPYMODEDSPPRINTCHAR))
                                                                  ; Set the character-printing functions for the stream to the
                                                                  ; hardcopy-mode ones.
        (replace (STREAM OUTCHARFN) of DS with (FUNCTION \HCPYMODEDSPPRINTCHAR))
```

;;; set the parameters that are different to initialize the mica defined fields.

```
          (DSPFONT (DSPFONT NIL DS)
                  DS)                                               ; Hardcopy version of the current font...
          (DSPXPOSITION 0 DS)                                       ; Reset the X and Y positions to 0
          (DSPYPOSITION 0 DS)
          (STREAMPROP DS 'DSPRIGHTMARGIN (DSPRIGHTMARGIN NIL DS))
                                                                    ; Stash the right margin in points for later restoral
          (DSPRIGHTMARGIN (FIXR (FTIMES (OR (DSPRIGHTMARGIN NIL DS)
                                            (fetch WIDTH of (DSPCLIPPINGREGION NIL DS)))
                                        MICASPERPT))
                  DS)                                               ; And reuse the right margin
          (DSPSPACEFACTOR 1 DS)
          (RETURN DS])
```

(**UNMAKEHARDCOPYMODESTREAM**
```
  [LAMBDA (DISPLAYSTREAM)                                           ; Edited 26-Aug-87 14:28 by Snow
```

;;; returns a hardcopy stream to a display stream.

```
    (DECLARE (GLOBALVARS \DISPLAYIMAGEOPS))
    (PROG [(DS (COND
                 ((DISPLAYSTREAMP DISPLAYSTREAM))
                 ((WINDOWP DISPLAYSTREAM)
                  (WINDOWPROP DISPLAYSTREAM 'DSP))
                 (T (\ILLEGAL.ARG DISPLAYSTREAM]
          (COND
            ((FMEMB 'HARDCOPY (IMAGESTREAMTYPE DS))                 ; Make sure the stream really WAS a hardcopy-mode stream.

             )
            (T                                                      ; It wasn't a hardcopy-mode stream.  Don't make any changes
             (RETURN DS)))
          (replace (STREAM IMAGEOPS) of DS with \DISPLAYIMAGEOPS)   ; Give it back the usual operations
          (STREAMPROP DS 'HARDCOPYIMAGETYPE NIL)                    ; restore the bout fn
          (replace (STREAM STRMBOUTFN) of DS with (FUNCTION \DSPPRINTCHAR))
          (replace (STREAM OUTCHARFN) of DS with (FUNCTION \DSPPRINTCHAR))
          (DSPXPOSITION 0 DS)
          (DSPYPOSITION 0 DS)
          (DSPRIGHTMARGIN (OR (STREAMPROP DISPLAYSTREAM 'DSPRIGHTMARGIN)
                              (fetch (REGION WIDTH) of (DSPCLIPPINGREGION NIL DS)))
                  NIL DS)                                           ; Reset the right margin back to points
          (RETURN DS])
```

(**BLTSHADE.HCPYMODE**
```
  [LAMBDA (TEXTURE STREAM DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT OPERATION CLIPPINGREGION)
                                                                    ; Edited 26-Aug-87 14:28 by Snow
```

;;; BLTSHADE to a hardcopy-mode display stream
```
                                                                    ; Just convert the coordinates and do the normal display thing.
    (\BLTSHADE.DISPLAY TEXTURE STREAM (\MICASTOPTS DESTINATIONLEFT)
          (\MICASTOPTS DESTINATIONBOTTOM)
          WIDTH HEIGHT OPERATION (\DASHINGCONVERT.HCPYMODE CLIPPINGREGION])
```

(**BITBLT.HCPYMODE**
```
  [LAMBDA (SOURCEBITMAP SOURCELEFT SOURCEBOTTOM DESTSTRM DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT
                 SOURCETYPE OPERATION TEXTURE CLIPPINGREGION CLIPPEDSOURCELEFT CLIPPEDSOURCEBOTTOM)
                                                                    ; Edited 26-Aug-87 14:28 by Snow
```
    ;; BITBLT to a hardcopy-mode display stream. Convert the destination coordinates to micas and do the normal operation.
```
    (\BITBLT.DISPLAY SOURCEBITMAP SOURCELEFT SOURCEBOTTOM DESTSTRM (\MICASTOPTS DESTINATIONLEFT)
          (\MICASTOPTS DESTINATIONBOTTOM)
          WIDTH HEIGHT SOURCETYPE OPERATION TEXTURE (\DASHINGCONVERT.HCPYMODE CLIPPINGREGION)
          CLIPPEDSOURCELEFT CLIPPEDSOURCEBOTTOM])
```

(**BRUSHCONVERT.HCPYMODE**
```
  [LAMBDA (BRUSH)                                                  ; Edited 26-Aug-87 14:29 by Snow
                                                                   ; Convert a brush description from points to micas
    (COND
       ((LISTP BRUSH)
        (FOR BB IN BRUSH COLLECT (COND
                                   ((NUMBERP BB)
                                    (\MICASTOPTS BB))
                                   (T BB])
```

(**CHANGECHARSET.HCPYMODE**
```
  [LAMBDA (DISPLAYDATA CHARSET)                                    ; Edited 26-Aug-87 14:29 by Snow
                                                                   ; Called when the character set information cached in a display
                                                                   ; stream doesn't correspond to CHARSET
    (PROG [BM (PBT (ffetch DDPILOTBBT of DISPLAYDATA))
          (CSINFO (\GETCHARSETINFO CHARSET (ffetch DDFONT of DISPLAYDATA)))
          (CSDINFO (\GETCHARSETINFO CHARSET (FONTCOPY (ffetch DDFONT of DISPLAYDATA)
```

```
                                                        'DEVICE
                                                        'DISPLAY]
            (UNINTERRUPTABLY
                (freplace DDWIDTHSCACHE of DISPLAYDATA with (ffetch (CHARSETINFO WIDTHS) of CSINFO))
                (freplace DDOFFSETSCACHE of DISPLAYDATA with (ffetch (CHARSETINFO OFFSETS) of CSINFO))
                (freplace DDCHARIMAGEWIDTHS of DISPLAYDATA with (ffetch (CHARSETINFO IMAGEWIDTHS) of CSINFO))
                (freplace DDCHARSET of DISPLAYDATA with CHARSET)
                (SETQ BM (ffetch CHARSETBITMAP of CSINFO))
                (freplace PBTSOURCEBPL of PBT with (UNFOLD (ffetch BITMAPRASTERWIDTH of BM)
                                                          BITSPERWORD))
                [replace OTHERDEVICEFONTPROPS of (ffetch DDFONT of DISPLAYDATA)
                    with (LIST 'WIDTHS (fetch (CHARSETINFO WIDTHS) of CSDINFO)
                            'ASCENT
                            (fetch (CHARSETINFO CHARSETASCENT) of CSDINFO)
                            'DESCENT
                            (fetch (CHARSETINFO CHARSETDESCENT) of CSDINFO)
                            'HEIGHT
                            (IPLUS (fetch (CHARSETINFO CHARSETASCENT) of CSDINFO)
                                   (fetch (CHARSETINFO CHARSETDESCENT) of CSDINFO]
            ;; Cache the DISPLAY info, for the various X- and Y-position updating tasks that affect the display bitmap itself
            [COND
                ((OR (NEQ (ffetch DDCHARSETASCENT of DISPLAYDATA)
                          (ffetch CHARSETASCENT of CSINFO))
                     (NEQ (ffetch DDCHARSETDESCENT of DISPLAYDATA)
                          (ffetch CHARSETDESCENT of CSINFO)))
                 (\SFFixY.HCPYMODE DISPLAYDATA CSINFO))
                (T (freplace PBTSOURCE of PBT with (\ADDBASE (ffetch BITMAPBASE of BM)
                                                             (ITIMES (ffetch BITMAPRASTERWIDTH of BM)
                                                                     (ffetch DDCHARHEIGHTDELTA of DISPLAYDATA]))
```

(\DASHINGCONVERT.HCPYMODE
```
  [LAMBDA (DASHING)                                              ; Edited 26-Aug-87 14:29 by Snow

    ;; Convert a list of numbers from micas to points.  Usually this will be a dashing spec, but it might be a REGION as well.

    (for DD in DASHING collect (\MICASTOPTS DD])
```

(\CHARWIDTH.HCPYMODE
```
  [LAMBDA (STREAM CHARCODE)                                      ; Edited 26-Aug-87 14:29 by Snow
                                                                 ; gets the width of a character code in a hardcopy stream.
                                                                 ; Should be updated for spacefactor
    (\FGETWIDTH (ffetch (\DISPLAYDATA DDCHARIMAGEWIDTHS) of (fetch IMAGEDATA of STREAM))
           CHARCODE])
```

(\DRAWLINE.HCPYMODE
```
  [LAMBDA (STREAM X1 Y1 X2 Y2 WIDTH OPERATION COLOR)             ; Edited 26-Aug-87 14:29 by Snow
                                                                 ; Do DRAWLINE for a hardcopy-mode display stream.
    (\DRAWLINE.DISPLAY STREAM (MICASTOPTS X1)
           (\MICASTOPTS Y1)
           (\MICASTOPTS X2)
           (\MICASTOPTS Y2)
           (IMAX 1 (\MICASTOPTS WIDTH))
           OPERATION COLOR])
```

(\DRAWCURVE.HCPYMODE
```
  [LAMBDA (STREAM KNOTS CLOSED BRUSH DASHING)                    ; Edited 26-Aug-87 14:30 by Snow

    ;; Do DRAWCURVE for a hardcopy-mode displaystream.  Converts all the mica values to points and uses the usual display version.

    (\DRAWCURVE.DISPLAY STREAM [FOR KNOT IN KNOTS COLLECT (CONS (\MICASTOPTS (CAR KNOT))
                                                               (\MICASTOPTS (CDR KNOT]
           CLOSED
           (\BRUSHCONVERT.HCPYMODE BRUSH)
           (\DASHINGCONVERT.HCPYMODE DASHING])
```

(\DRAWCIRCLE.HCPYMODE
```
  [LAMBDA (STREAM CENTERX CENTERY RADIUS BRUSH DASHING)          ; Edited 26-Aug-87 14:30 by Snow

    ;; DRAWCIRCLE for a hardcopy-mode display stream.  Convert coordinates to points and use the display driver

    (\DRAWCIRCLE.DISPLAY STREAM (\MICASTOPTS CENTERX)
           (\MICASTOPTS CENTERY)
           (\MICASTOPTS RADIUS)
           (\BRUSHCONVERT.HCPYMODE BRUSH)
           (\DASHINGCONVERT.HCPYMODE DASHING])
```

(\DRAWELLIPSE.HCPYMODE
```
  [LAMBDA (STREAM CENTERX CENTERY SEMIMINORRADIUS SEMIMAJORRADIUS ORIENTATION BRUSH DASHING)
                                                                 ; Edited 26-Aug-87 14:30 by Snow

    ;; DRAWELLIPSE driver for hardcopy-mode displaystreams.  Convert all the values to points from micas, and use the display DRAWELLIPSE.

    (\DRAWELLIPSE.DISPLAY STREAM (\MICASTOPTS CENTERX)
           (\MICASTOPTS CENTERY)
```

```
                    (\MICASTOPTS SEMIMINORRADIUS)
                    (\MICASTOPTS SEMIMAJORRADIUS)
                    ORIENTATION
                    (\BRUSHCONVERT.HCPYMODE BRUSH)
                    (\DASHINGCONVERT.HCPYMODE DASHING])
```

(\DSPFONT.HCPYMODE

```
  [LAMBDA (HDCPYDSTREAM FONT)                                            ; Edited 20-Apr-88 11:53 by jds

    ;; changes the font of a hardcopy display stream.  Does what the display does then puts the hardcopy widths where they can be found {FOR NOW
    ;; USE THE DDCHARIMAGEWIDTHS FIELD}

    (PROG (XFONT OLDFONT (DD (fetch IMAGEDATA of HDCPYDSTREAM)))          ; save old value to return, smash new value and update the
                                                                         ; bitchar portion of the record.
          (RETURN (PROG1 (SETQ OLDFONT (fetch DDFONT of DD))
                         [COND
                             (FONT (SETQ XFONT (OR (\GETFONTDESC FONT (fetch IMFONTCREATE
                                                                               of (fetch IMAGEOPS of HDCPYDSTREAM))
                                                         T)
                                   (FONTCOPY (ffetch DDFONT of DD)
                                             FONT)))            ; updating font information is fairly expensive operation.  Don't
                                                                ; bother unless font has changed.
                         (OR (EQ XFONT OLDFONT)
                             (UNINTERRUPTABLY
                                 (freplace DDFONT of DD with XFONT)
                                 (freplace DDLINEFEED of DD with (IMINUS (fetch \SFHeight of XFONT)))
                                                                         ; Each line moves down by the font height, by default
                                 [freplace DDSPACEWIDTH of DD
                                    with (FIXR (FTIMES (OR (ffetch DDMICAXPOS of DD)
                                                           1)
                                                       (\FGETCHARWIDTH XFONT (CHARCODE SPACE]
                                 (\SFFixFont HDCPYDSTREAM DD)
                                                                         ; Fix up the font-dependent fields of the DISPLAYSTREAM

                         )])))])
```

(\DSPLEFTMARGIN.HCPYMODE

```
  [LAMBDA (DISPLAYSTREAM XPOSITION)                                      ; Edited 26-Aug-87 14:30 by Snow

;;; Sets the left margin that determines when a cr is inserted by print for the hardcopy display stream.

;;; Sets the left margin for a hardcopy-mode displaystream, to determine where CR returns you to.

    (PROG1 [\DSPRIGHTMARGIN.DISPLAY DISPLAYSTREAM (AND XPOSITION (FIXR (FQUOTIENT XPOSITION MICASPERPT]
           ;; LATER, WHEN DDLEFTMARGININMICA EXISTS... (AND XPOSITION (replace (\DISPLAYDATA DDMICARIGHTMARGIN) of (fetch
           ;; IMAGEDATA of DISPLAYSTREAM) with XPOSITION))

           ])
```

(\DSPLINEFEED.HCPYMODE

```
  [LAMBDA (DISPLAYSTREAM DELTAY)                                         ; Edited 26-Aug-87 14:33 by Snow
                                                                         ; For a hardcopy-mode displaystream, sets the amount that a
                                                                         ; line feed increases the y coordinate by.
    (PROG1 (ffetch DDLINEFEED of (fetch IMAGEDATA of DISPLAYSTREAM))
           [AND DELTAY (COND
                           ((NUMBERP DELTAY)
                            (freplace DDLINEFEED of (ffetch IMAGEDATA of DISPLAYSTREAM) with DELTAY))
                           (T (\ILLEGAL.ARG DELTAY])])
```

(\DSPRIGHTMARGIN.HCPYMODE

```
  [LAMBDA (DISPLAYSTREAM XPOSITION)                                      ; Edited 26-Aug-87 14:32 by Snow

;;; Sets the right margin that determines when a cr is inserted by print for the hardcopy display stream.

    (PROG1 (fetch (\DISPLAYDATA DDMICARIGHTMARGIN) of (fetch IMAGEDATA of DISPLAYSTREAM))
                                                                         ; Return the old mica value.
           [\DSPRIGHTMARGIN.DISPLAY DISPLAYSTREAM (AND XPOSITION (FIXR (FQUOTIENT XPOSITION MICASPERPT]
                                                                         ; Set the right margin in display units,
           (AND XPOSITION (replace (\DISPLAYDATA DDMICARIGHTMARGIN) of (fetch IMAGEDATA of DISPLAYSTREAM) with
                                                                                                            XPOSITION
                     ))                                                  ; And set the new mica value

           ])
```

(\DSPSPACEFACTOR.HCPYMODE

```
  [LAMBDA (DISPLAYSTREAM FACTOR)                                         ; Edited  1-Apr-88 11:28 by jds
    ;; Sets the space factor for a hardcopy-mode displaystream.

    (LET ((DDATA (fetch IMAGEDATA of DISPLAYSTREAM)))
         (PROG1 (fetch (\DISPLAYDATA DDMICAXPOS) of DDATA)
                (COND
                    [(NUMBERP FACTOR)
                     (replace (\DISPLAYDATA DDMICAXPOS) of DDATA with FACTOR)
```

```
                  (replace (\DISPLAYDATA DDSPACEWIDTH) of DDATA with (FIXR (FTIMES FACTOR
                                                                            (CHARWIDTH (CHARCODE SPACE)
                                                                              (fetch (\DISPLAYDATA DDFONT
                                                                                     )
                                                                                of DDATA]
            (T (\ILLEGAL.ARG FACTOR))))])
```

(\DSPXPOSITION.HCPYMODE

```
  [LAMBDA (HARDCOPYSTREAM XPOSITION)                           ; Edited 26-Aug-87 14:32 by Snow
                                                               ; Update the X position for a mica-unit hardcopy-mode
                                                               ; displaystream
     (PROG1 (fetch (\DISPLAYDATA DDXPOSITION) of (fetch IMAGEDATA of HARDCOPYSTREAM))
                                                               ; Return the old value...
          [\DSPXPOSITION.DISPLAY HARDCOPYSTREAM (AND XPOSITION (FIXR (FQUOTIENT XPOSITION MICASPERPT]
                                                               ; Set up the display right for this mica value
          (AND XPOSITION (replace (\DISPLAYDATA DDXPOSITION) of (fetch IMAGEDATA of HARDCOPYSTREAM) with XPOSITION))
                                                               ; And remember what it was.

          )])
```

(\DSPYPOSITION.HCPYMODE

```
  [LAMBDA (HARDCOPYSTREAM YPOSITION)                           ; Edited 26-Aug-87 14:35 by Snow
                                                               ; Move to a new mica Y position

     (LET* ((DD (fetch IMAGEDATA of HARDCOPYSTREAM))
            (OLD-POS (ffetch DDYPOSITION of DD)))
           (COND
              ((NULL YPOSITION))
              ((NUMBERP YPOSITION)
               (UNINTERRUPTABLY
                   (freplace DDYPOSITION of DD with YPOSITION))
               (\INVALIDATEDISPLAYCACHE DD))
              (T (\ILLEGAL.ARG YPOSITION)))
           OLD-POS])
```

(\MOVETO.HCPYMODE

```
  [LAMBDA (STREAM X Y)                                         ; Edited 26-Aug-87 14:36 by Snow
     (\DSPXPOSITION.HCPYMODE STREAM X)
     (\DSPYPOSITION.HCPYMODE STREAM Y])
```

(\FONTCREATE.HCPYMODE.PRESS

```
  [LAMBDA (FAMILY SIZE FACE ROTATION DEVICE)                   ; Edited 26-Aug-87 14:36 by Snow
                                                               ; Create a font descriptor for a display stream that is mimicing an
                                                               ; PRESS device
     (PROG* ((DFONT (FONTCREATE FAMILY SIZE FACE ROTATION 'DISPLAY))
             (HFONT (create FONTDESCRIPTOR using (FONTCREATE FAMILY SIZE FACE ROTATION 'PRESS)
                                       FONTCHARSETVECTOR _ (\CREATEFONTCHARSETVECTOR)))
             (CS0DINFO (\GETCHARSETINFO \DEFAULTCHARSET DFONT)))
            (replace FONTDEVICE of HFONT with 'PRESSDISPLAY)
            [replace OTHERDEVICEFONTPROPS of HFONT with (LIST 'WIDTHS (fetch (CHARSETINFO WIDTHS) of CS0DINFO)
                                                              'ASCENT
                                                              (fetch (CHARSETINFO CHARSETASCENT) of CS0DINFO)
                                                              'DESCENT
                                                              (fetch (CHARSETINFO CHARSETDESCENT) of CS0DINFO)
                                                              'HEIGHT
                                                              (IPLUS (fetch (CHARSETINFO CHARSETASCENT) of CS0DINFO)
                                                                     (fetch (CHARSETINFO CHARSETDESCENT)
                                                                        of CS0DINFO]
        ;; Cache the DISPLAY info, for the various X- and Y-position updating tasks that affect the display bitmap itself
            (RETURN HFONT])
```

(\CREATECHARSET.HCPYMODE.PRESS

```
  [LAMBDA (FAMILY SIZE FACE ROTATION DEVICE CHARSET FONTDESC)  ; Edited 26-Aug-87 14:36 by Snow
                                                               ; Build the CHARSETINFO for an PRESSDISPLAY font
     (PROG* ((DFONT (FONTCREATE FAMILY SIZE FACE ROTATION 'DISPLAY))
             (HFONT (FONTCREATE FAMILY SIZE FACE ROTATION 'PRESS))
             (CSDINFO (\GETCHARSETINFO CHARSET DFONT))
             (CSHINFO (\GETCHARSETINFO CHARSET HFONT))
             (CSINFO (CREATE CHARSETINFO USING CSHINFO)))
            (replace (CHARSETINFO OFFSETS) of CSINFO with (fetch (CHARSETINFO OFFSETS) of CSDINFO))
                                                               ; Fill in the right offsets from the display font--into the hcpy font,
                                                               ; and its Charset-0 info block
            (replace (CHARSETINFO CHARSETBITMAP) of CSINFO with (fetch (CHARSETINFO CHARSETBITMAP) of CSDINFO))
                                                               ; Likewise the character rasters
            (replace (CHARSETINFO IMAGEWIDTHS) of CSINFO with (fetch (CHARSETINFO IMAGEWIDTHS) of CSDINFO))
                                                               ; And the raster widths (as distinct from the nominal mica widths)
            (RETURN CSINFO])
```

(\FONTCREATE.HCPYMODE.INTERPRESS

```
  [LAMBDA (FAMILY SIZE FACE ROTATION DEVICE)                   ; Edited 26-Aug-87 14:36 by Snow
```

;;; Create a font descriptor for a display stream that is mimicing an INTERPRESS device

```
    (PROG* ((DFONT (FONTCREATE FAMILY SIZE FACE ROTATION 'DISPLAY))
            (HFONT (create FONTDESCRIPTOR using (FONTCREATE FAMILY SIZE FACE ROTATION 'INTERPRESS)
                                               FONTCHARSETVECTOR _ (\CREATEFONTCHARSETVECTOR)))
            (CS0DINFO (\GETCHARSETINFO \DEFAULTCHARSET DFONT)))
           (replace FONTDEVICE of HFONT with 'INTERPRESSDISPLAY)
           [replace OTHERDEVICEFONTPROPS of HFONT with (LIST 'WIDTHS (fetch (CHARSETINFO WIDTHS) of CS0DINFO)
                                                             'ASCENT
                                                             (fetch (CHARSETINFO CHARSETASCENT) of CS0DINFO)
                                                             'DESCENT
                                                             (fetch (CHARSETINFO CHARSETDESCENT) of CS0DINFO)
                                                             'HEIGHT
                                                             (IPLUS (fetch (CHARSETINFO CHARSETASCENT) of CS0DINFO)
                                                                    (fetch (CHARSETINFO CHARSETDESCENT)
                                                                         of CS0DINFO]
```

    ;; Cache the DISPLAY info, for the various X- and Y-position updating tasks that affect the display bitmap itself

```
            (RETURN HFONT])
```

(**CREATECHARSET.HCPYMODE.INTERPRESS**
```
  [LAMBDA (FAMILY SIZE FACE ROTATION DEVICE CHARSET FONTDESC)        ; Edited 26-Aug-87 14:37 by Snow
```

;;; Build the CHARSETINFO for an INTERPRESSDISPLAY font

```
    (PROG* ((DFONT (FONTCREATE FAMILY SIZE FACE ROTATION 'DISPLAY))
            (HFONT (FONTCREATE FAMILY SIZE FACE ROTATION 'INTERPRESS))
            (CSDINFO (\GETCHARSETINFO CHARSET DFONT))
            (CSHINFO (\GETCHARSETINFO CHARSET HFONT))
            (CSINFO (CREATE CHARSETINFO USING CSHINFO)))
           (replace (CHARSETINFO OFFSETS) of CSINFO with (fetch (CHARSETINFO OFFSETS) of CSDINFO))
                                                ; Fill in the right offsets from the display font--into the hcpy font,
                                                ; and its Charset-0 info block
           (replace (CHARSETINFO CHARSETBITMAP) of CSINFO with (fetch (CHARSETINFO CHARSETBITMAP) of CSDINFO))
                                                ; Likewise the character rasters
           (replace (CHARSETINFO IMAGEWIDTHS) of CSINFO with (fetch (CHARSETINFO IMAGEWIDTHS) of CSDINFO))
                                                ; And the raster widths (as distinct from the nominal mica widths)

           (RETURN CSINFO])
```

(**STRINGWIDTH.HCPYMODE**
```
  [LAMBDA (STREAM STR RDTBL)                                         ; Edited 26-Aug-87 14:38 by Snow
                                                                    ; Returns the width of for the current font/spacefactor in hardcopy
                                                                    ; stream STREAM.
    (LET [(WIDTHSBASE (ffetch (\DISPLAYDATA DDCHARIMAGEWIDTHS) of (ffetch IMAGEDATA of STREAM]
         (IQUOTIENT (IPLUS (\STRINGWIDTH.GENERIC STR WIDTHSBASE RDTBL (\FGETWIDTH WIDTHSBASE (CHARCODE SPACE)))
                           (CONSTANT IHALFMICASPERPT))
                    (CONSTANT IMICASPERPT])
```

(**HCPYMODEBLTCHAR**
```
  [LAMBDA (CHARCODE DISPLAYSTREAM DISPLAYDATA)                       ; Edited  1-Apr-88 11:35 by jds
```

    ;; puts a character on a hardcopy display stream. Much of the information needed by the BitBlt microcode is prestored by the routines that change
    ;; it. This is kept in the BitBltTable.   ; knows about the representation of a DisplayStream.
```
    (DECLARE (LOCALVARS . T))
    (PROG (LOCAL1 RIGHT LEFT CURX MICARIGHT (CHAR8CODE (\CHAR8CODE CHARCODE))
                  CHARWIDTH)
      CRLP
          [COND
             ((NEQ (ffetch DDCHARSET of DISPLAYDATA)
                   (\CHARSET CHARCODE))
              (\CHANGECHARSET.HCPYMODE DISPLAYDATA (\CHARSET CHARCODE]
          [COND
             ((ffetch (\DISPLAYDATA DDSlowPrintingCase) of DISPLAYDATA)
              (RETURN (\SLOWHCPYMODEBLTCHAR CHARCODE DISPLAYSTREAM]
          (SETQ CURX (FIXR (FQUOTIENT (ffetch DDXPOSITION of DISPLAYDATA)
                           MICASPERPT)))                            ; Convert the mica-position value to points only at the last
                                                                    ; minute.
          [SETQ CHARWIDTH (COND
                             ((IEQP CHARCODE (CHARCODE SPACE))
                              (FFETCH DDSPACEWIDTH OF DISPLAYDATA))
                             (T (\DSPGETCHARWIDTH CHAR8CODE DISPLAYDATA]
          [COND
             ((IGREATERP (SETQ MICARIGHT (IPLUS (ffetch (\DISPLAYDATA DDXPOSITION) of DISPLAYDATA)
                                                CHARWIDTH))
                         (ffetch (\DISPLAYDATA DDMICARIGHTMARGIN) of DISPLAYDATA))
                                                                    ; would go past right margin, force a cr
              (COND
                 ((IGREATERP CURX (ffetch DDLeftMargin of DISPLAYDATA))
                                                                    ; don't bother CR if position is at left margin anyway.  This also
                                                                    ; serves to break the loop.
                  (\DSPPRINTCR/LF (CHARCODE EOL)
                         DISPLAYSTREAM)                             ; reuse the code in the test of this conditional rather than repeat it
                                                                    ; here.
                  (GO CRLP]
          (freplace (\DISPLAYDATA DDXPOSITION) of DISPLAYDATA with MICARIGHT)
```

```
      ;; update the display stream x position.  Make sure that there is at least one point width for each character.
            [SETQ CURX (IPLUS CURX (SETQ LOCAL1 (ffetch DDXOFFSET of DISPLAYDATA]
                                                         ; Screen position of the window, generally.
            (SETQ RIGHT (IPLUS CURX (\FGETWIDTH (ffetch DDCHARIMAGEWIDTHS of DISPLAYDATA)
                                          CHAR8CODE)))            ; Right edge of the character's image.
            (COND
               ((IGREATERP RIGHT (SETQ LOCAL1 (ffetch DDClippingRight of DISPLAYDATA)))
                                                         ; character overlaps right edge of clipping region.
                (SETQ RIGHT LOCAL1)))
            (SETQ LEFT (COND
                        ((IGREATERP CURX (SETQ LOCAL1 (ffetch DDClippingLeft of DISPLAYDATA)))
                          CURX)
                        (T LOCAL1)))                    ; Left edge of the character, as displayed.
            (RETURN (COND
                        ((AND (ILESSP LEFT RIGHT)
                              (NEQ (fetch PBTHEIGHT of (SETQ LOCAL1 (ffetch DDPILOTBBT of DISPLAYDATA)))
                                   0))                   ; If the character will appear on screen at all, let's display it.
                          (.WHILE.TOP.DS. DISPLAYSTREAM (freplace PBTDESTBIT of LOCAL1 with LEFT)
                                                        ; Set up the destination bit with the screen-relative left edge
                             (freplace PBTWIDTH of LOCAL1 with (IDIFFERENCE RIGHT LEFT))
                                                        ; The display width from the clipped left and right edges
                                   (freplace PBTSOURCEBIT of LOCAL1 with (IDIFFERENCE (IPLUS (\DSPGETCHAROFFSET CHAR8CODE
                                                                                               DISPLAYDATA)
                                                                               LEFT)
                                                         CURX))
                                                        ; And the source bit-offset from the OFFSETS array
                             (\PILOTBITBLT LOCAL1 0)    ; Do the BITBLT
                             
                             )
                         T])
```

⟨**HCPYMODEDISPLAYINIT**

```
  [LAMBDA NIL                                                      ; Edited  1-Apr-88 11:36 by jds
```

;;; Initializes global variables for the hardcopy Display device. This device appears to the user as an INTERPRESS or PRESS device meaning units in
;;; micas but outputs to the screen. Much of this code was borrowed from the display case.

```
    (DECLARE (GLOBALVARS \HCPYMODEDISPLAYIMAGEOPS.PRESS \HCPYMODEDISPLAYIMAGEOPS.INTERPRESS))
    (SETQ \HCPYMODEDISPLAYIMAGEOPS.PRESS (create IMAGEOPS using \DISPLAYIMAGEOPS IMAGETYPE _ '(HARDCOPY DISPLAY)
                                                    IMFONT _ (FUNCTION \DSPFONT.HCPYMODE)
                                                    IMRIGHTMARGIN _ (FUNCTION \DSPRIGHTMARGIN.HCPYMODE
                                                                    )
                                                    IMLEFTMARGIN _ (FUNCTION \DSPLEFTMARGIN.HCPYMODE)
                                                    IMLINEFEED _ (FUNCTION \DSPLINEFEED.HCPYMODE)
                                                    IMDRAWLINE _ (FUNCTION \DRAWLINE.HCPYMODE)
                                                    IMDRAWCURVE _ (FUNCTION \DRAWCURVE.HCPYMODE)
                                                    IMDRAWCIRCLE _ (FUNCTION \DRAWCIRCLE.HCPYMODE)
                                                    IMDRAWELLIPSE _ (FUNCTION \DRAWELLIPSE.HCPYMODE)
                                                    IMFILLCIRCLE _ (FUNCTION \FILLCIRCLE.HCPYMODE)
                                                    IMBLTSHADE _ (FUNCTION \BLTSHADE.HCPYMODE)
                                                    IMBITBLT _ (FUNCTION \BITBLT.HCPYMODE)
                                                    IMXPOSITION _ (FUNCTION \DSPXPOSITION.HCPYMODE)
                                                    IMYPOSITION _ (FUNCTION \DSPYPOSITION.HCPYMODE)
                                                    IMMOVETO _ (FUNCTION \MOVETO.HCPYMODE)
                                                    IMSTRINGWIDTH _ (FUNCTION \STRINGWIDTH.HCPYMODE)
                                                    IMCHARWIDTH _ (FUNCTION \CHARWIDTH.HCPYMODE)
                                                    IMFONTCREATE _ (FUNCTION PRESSDISPLAY)
                                                    IMSCALE _ [FUNCTION (LAMBDA NIL
                                                                          (CONSTANT (FQUOTIENT
                                                                                       MICASPERINCH
                                                                                       72]
                                                    IMNEWPAGE _
                                                    [FUNCTION (LAMBDA (STREAM)
                                                              (LET ((WINDOW (AND \WINDOWWORLD
                                                                                (WFROMDS STREAM)))
                                                                    WINDOWFN)
                                                                 (COND
                                                                    ([AND WINDOW
                                                                          (SETQ WINDOWFN
                                                                           (WINDOWPROP
                                                                            WINDOW
                                                                            'PAGEFULLFN]
                                                                     (APPLY* WINDOWFN STREAM))
                                                                    (T (PAGEFULLFN STREAM)))
                                                                 (CLEARW STREAM]
                                                    IMSPACEFACTOR _ (FUNCTION \DSPSPACEFACTOR.HCPYMODE
                                                                    )))
    (SETQ \HCPYMODEDISPLAYIMAGEOPS.INTERPRESS (create IMAGEOPS using \DISPLAYIMAGEOPS IMAGETYPE _
                                                       '(HARDCOPY DISPLAY)
                                                       IMFONT _ (FUNCTION \DSPFONT.HCPYMODE)
                                                       IMRIGHTMARGIN _ (FUNCTION
                                                                        \DSPRIGHTMARGIN.HCPYMODE)
                                                       IMLEFTMARGIN _ (FUNCTION
                                                                        \DSPLEFTMARGIN.HCPYMODE)
                                                       IMLINEFEED _ (FUNCTION \DSPLINEFEED.HCPYMODE)
```

```
                                             IMDRAWLINE _ (FUNCTION \DRAWLINE.HCPYMODE)
                                             IMDRAWCURVE _ (FUNCTION \DRAWCURVE.HCPYMODE)
                                             IMDRAWCIRCLE _ (FUNCTION \DRAWCIRCLE.HCPYMODE
                                                                 )
                                             IMDRAWELLIPSE _ (FUNCTION
                                                                   \DRAWELLIPSE.HCPYMODE)
                                             IMFILLCIRCLE _ (FUNCTION \FILLCIRCLE.HCPYMODE
                                                                 )
                                             IMBLTSHADE _ (FUNCTION \BLTSHADE.HCPYMODE)
                                             IMBITBLT _ (FUNCTION \BITBLT.HCPYMODE)
                                             IMXPOSITION _ (FUNCTION
                                                                  \DSPXPOSITION.HCPYMODE)
                                             IMYPOSITION _ (FUNCTION
                                                                  \DSPYPOSITION.HCPYMODE)
                                             IMMOVETO _ (FUNCTION \MOVETO.HCPYMODE)
                                             IMSTRINGWIDTH _ (FUNCTION
                                                                  \STRINGWIDTH.HCPYMODE)
                                             IMCHARWIDTH _ (FUNCTION \CHARWIDTH.HCPYMODE)
                                             IMFONTCREATE _ (FUNCTION INTERPRESSDISPLAY)
                                             IMSCALE _ [FUNCTION (LAMBDA NIL
                                                                    (CONSTANT (FQUOTIENT
                                                                                 MICASPERINCH
                                                                                 72]
                                             IMNEWPAGE _
                                             [FUNCTION (LAMBDA (STREAM)
                                                           (LET
                                                             ((WINDOW (AND \WINDOWWORLD
                                                                           (WFROMDS STREAM)))
                                                              WINDOWFN)
                                                             (COND
                                                               ([AND WINDOW
                                                                     (SETQ WINDOWFN
                                                                       (WINDOWPROP
                                                                        WINDOW
                                                                        'PAGEFULLFN]
                                                                (APPLY* WINDOWFN STREAM))
                                                               (T (PAGEFULLFN STREAM)))
                                                             (CLEARW STREAM]
                                             IMSPACEFACTOR _ (FUNCTION
                                                                  \DSPSPACEFACTOR.HCPYMODE])
```

(\HCPYMODEDSPPRINTCHAR
```
  [LAMBDA (STREAM CHARCODE)                                        ; Edited 26-Aug-87 14:39 by Snow
```

;;; displays a character on a hardcopy display stream. This uses a display font but updates the x position according to hardcopy widths.

```
      (PROG ((DD (fetch IMAGEDATA of STREAM)))
            (\CHECKCARET STREAM)
            (RETURN (SELECTC (fetch CCECHO of (\SYNCODE \PRIMTERMSA CHARCODE))
                      (INDICATE.CCE [PROG ((CC CHARCODE))
                                          (add (fetch CHARPOSITION of STREAM)
                                               (IPLUS (COND
                                                        ((IGREATERP CC 127)
                                                             ; META character
                                                         (\HCPYMODEBLTCHAR (CHARCODE %#)
                                                               STREAM DD)
                                                         (SETQ CC (LOGAND CC 127))
                                                         1)
                                                        (T 0))
                                                      (COND
                                                        ((ILESSP CC 32)
                                                             ; CONTROL character
                                                         (\HCPYMODEBLTCHAR (CHARCODE ^)
                                                               STREAM DD)
                                                         (SETQ CC (LOGOR CC 64))
                                                         1)
                                                        (T 0))
                                                      (PROGN (\HCPYMODEBLTCHAR CC STREAM DD)
                                                             1])
                      (SIMULATE.CCE (SELCHARQ CHARCODE
                                      ((EOL CR LF)
                                           (\DSPPRINTCR/LF CHARCODE STREAM)
                                           (replace CHARPOSITION of STREAM with 0))
                                      (ESCAPE (\HCPYMODEBLTCHAR (CHARCODE $)
                                                     STREAM DD)
                                             (add (fetch CHARPOSITION of STREAM)
                                                  1))
                                      (BELL                      ; make switching of bits uninterruptable but allow interrupts
                                                                 ; between flashes.
                                             (SELECTQ (MACHINETYPE)
                                               (DANDELION [PLAYTUNE '((880 . 2500])
                                               (FLASHWINDOW (WFROMDS STREAM))))
                                      (TAB (PROG (TABWIDTH (SPACEWIDTH (CHARWIDTH (CHARCODE SPACE)
                                                                                  STREAM)))
                                                 (SETQ TABWIDTH (UNFOLD SPACEWIDTH 8))
                                                 (COND
```

```
                                                       ((IGREATERP (\DISPLAYSTREAMINCRXPOSITION
                                                                    (SETQ TABWIDTH
                                                                     (IDIFFERENCE TABWIDTH
                                                                                   (MOD (IDIFFERENCE (fetch DDXPOSITION
                                                                                                             of DD)
                                                                                              (ffetch DDLeftMargin
                                                                                                   of DD))
                                                                                        TABWIDTH)))
                                                                    DD)
                                                        (ffetch DDRightMargin of DD))
                                                                       ; tab was past rightmargin, force cr.
                                                        (\DSPPRINTCR/LF (CHARCODE EOL)
                                                                STREAM)))
                                                                      ; return the number of spaces taken.
                                                      (add (fetch CHARPOSITION of STREAM)
                                                           (IQUOTIENT TABWIDTH SPACEWIDTH))))
                                             (PROGN                 ; this case was copied from \DSCCOUT.
                                                    (\HCPYMODEBLTCHAR CHARCODE STREAM DD)
                                                    (add (fetch CHARPOSITION of STREAM)
                                                         1))))
                                    (REAL.CCE (SELECTC CHARCODE
                                                 ((CHARCODE (EOL CR LF))
                                                  (\DSPPRINTCR/LF CHARCODE STREAM)
                                                  (replace CHARPOSITION of STREAM with 0))
                                                 (ERASECHARCODE
                                                  (DSPBACKUP (CHARWIDTH (CHARCODE A)
                                                                        STREAM)
                                                             STREAM)         ; line buffering routines have already taken care of backing up
                                                                             ; the position
                                                  0)
                                                 (PROGN (\HCPYMODEBLTCHAR CHARCODE STREAM DD)
                                                        (add (fetch CHARPOSITION of STREAM)
                                                             1))))
                                    (IGNORE.CCE)
                                    (SHOULDNT])
```

(**SLOWHCPYMODEBLTCHAR**
```
  [LAMBDA (CHARCODE DISPLAYSTREAM)                                            ; Edited 26-Aug-87 14:39 by Snow
```

;;; IS THIS CODE JUST GOING TO DUPLICATE AND GET OUT OF SYNC WITH \SLOWBLTCHAR? KBR 1-FEB-86. *

;;; THIS HAS BEEN SEPARATED OUT BUT HASN'T BEEN EDITTED TO DO CORRECT THING WRT UPDATING MICA FIELDS.

   ;; case of BLTCHAR where either font is rotated or destination is a color bitmap. DISPLAYSTREAM is known to be a hardcopy display stream.

```
    (PROG (ROTATION (CHAR8CODE (\CHAR8CODE CHARCODE))
                 (DD (ffetch (STREAM IMAGEDATA) of DISPLAYSTREAM)))
          (SETQ ROTATION (ffetch (FONTDESCRIPTOR ROTATION) of (ffetch (\DISPLAYDATA DDFONT) of DD)))
          (COND
             [(EQ 0 ROTATION)
              (PROG (NEWX LEFT RIGHT (CURX (ffetch (\DISPLAYDATA DDXPOSITION) of DD))
                          PILOTBBT DESTBIT WIDTH SOURCEBIT)
                    (SETQ NEWX (IPLUS CURX (\DSPGETCHARWIDTH CHAR8CODE DD)))
                    [COND
                       ((IGREATERP NEWX (ffetch (\DISPLAYDATA DDRightMargin) of DD))
                                                                             ; past RIGHT margin, force eol
                        (\DSPPRINTCR/LF (CHARCODE EOL)
                                DISPLAYSTREAM)
                        (SETQ CURX (ffetch (\DISPLAYDATA DDXPOSITION) of DD))
                        (SETQ NEWX (IPLUS CURX (\DSPGETCHARWIDTH CHAR8CODE DD]
                                                                             ; update the x position.
                    (freplace (\DISPLAYDATA DDXPOSITION) of DD with NEWX)
                    (SETQ CURX (\DSPTRANSFORMX CURX DD))
                    (SETQ LEFT (IMAX (ffetch (\DISPLAYDATA DDClippingLeft) of DD)
                                     CURX))
                    (SETQ RIGHT (IMIN (ffetch (\DISPLAYDATA DDClippingRight) of DD)
                                      (\DSPTRANSFORMX NEWX DD)))
                    (SETQ PILOTBBT (ffetch (\DISPLAYDATA DDPILOTBBT) of DD))
                    (COND
                       ((AND (ILESSP LEFT RIGHT)
                             (NEQ (ffetch (PILOTBBT PBTHEIGHT) of PILOTBBT)
                                  0))
                        (SETQ DESTBIT LEFT)
                        (SETQ WIDTH (IDIFFERENCE RIGHT LEFT))
                        (SETQ SOURCEBIT (IDIFFERENCE (IPLUS (\DSPGETCHAROFFSET CHAR8CODE DD)
                                                            LEFT)
                                                     CURX))
                        (SELECTQ (ffetch (BITMAP BITMAPBITSPERPIXEL) of (ffetch (\DISPLAYDATA DDDestination)
                                                                                of DD))
                              (1)
                              (4 (SETQ DESTBIT (LLSH DESTBIT 2))
                                 (SETQ WIDTH (LLSH WIDTH 2))
                                 (SETQ SOURCEBIT (LLSH SOURCEBIT 2)))
                              (8 (SETQ DESTBIT (LLSH DESTBIT 3))
                                 (SETQ WIDTH (LLSH WIDTH 3))
                                 (SETQ SOURCEBIT (LLSH SOURCEBIT 3)))
                              (SHOULDNT))
```

```
                                (.WHILE.TOP.DS. DISPLAYSTREAM (freplace (PILOTBBT PBTDESTBIT) of PILOTBBT with DESTBIT)
                                                             (freplace (PILOTBBT PBTWIDTH) of PILOTBBT with WIDTH)
                                                             (freplace (PILOTBBT PBTSOURCEBIT) of PILOTBBT with SOURCEBIT)
                                                             (\PILOTBITBLT PILOTBBT 0))
                              T]
            (T                                                            ; handle rotated fonts
                (PROG (YPOS HEIGHTMOVED CSINFO)
                      (SETQ YPOS (ffetch (\DISPLAYDATA DDYPOSITION) of DD))
                      (SETQ HEIGHTMOVED (\DSPGETCHARWIDTH CHAR8CODE DD))
                      (SETQ CSINFO (\GETCHARSETINFO (\CHARSET CHARCODE)
                                                   (ffetch (\DISPLAYDATA DDFONT) of DD)))
                      (COND
                         ((EQ ROTATION 90)                               ; don't force CR for rotated fonts.
                          (\DSPYPOSITION.DISPLAY DISPLAYSTREAM (IPLUS YPOS HEIGHTMOVED))
                                                                         ; update the display stream x position.
                          (BITBLT (ffetch (CHARSETINFO CHARSETBITMAP) of CSINFO)
                                  0
                                  (\DSPGETCHAROFFSET CHAR8CODE DD)
                                  DISPLAYSTREAM
                                  (ADD1 (IDIFFERENCE (ffetch (\DISPLAYDATA DDXPOSITION) of DD)
                                                     (ffetch (CHARSETINFO CHARSETASCENT) of CSINFO)))
                                  YPOS
                                  (IPLUS (ffetch (CHARSETINFO CHARSETASCENT) of CSINFO)
                                         (ffetch (CHARSETINFO CHARSETDESCENT) of CSINFO))
                                  HEIGHTMOVED))
                         ((EQ ROTATION 270)
                          (\DSPYPOSITION.DISPLAY DISPLAYSTREAM (IDIFFERENCE YPOS HEIGHTMOVED))
                          (BITBLT (ffetch (CHARSETINFO CHARSETBITMAP) of CSINFO)
                                  0
                                  (\DSPGETCHAROFFSET CHAR8CODE DD)
                                  DISPLAYSTREAM
                                  (IDIFFERENCE (ffetch (\DISPLAYDATA DDXPOSITION) of DD)
                                               (ffetch (CHARSETINFO CHARSETDESCENT) of CSINFO))
                                  (ffetch (\DISPLAYDATA DDYPOSITION) of DISPLAYSTREAM)
                                  (IPLUS (ffetch (CHARSETINFO CHARSETASCENT) of CSINFO)
                                         (ffetch (CHARSETINFO CHARSETDESCENT) of CSINFO))
                                  HEIGHTMOVED))
                         (T (ERROR "Not implemented to rotate by other than 0, 90 or 270"])
```

(\SFFixY.HCPYMODE

```
  [LAMBDA (DISPLAYDATA CSINFO)                                           ; Edited 26-Aug-87 14:40 by Snow

    ;; makes that part of the bitblt table of a display stream which deals with the Y information consistent.  This is called whenever any of the
    ;; information which effects it changes by the DSPFn eg DSPPosition.  If the change affected the clipping region, \SFFixClippingRegion should be
    ;; called before \SFFixY.HCPYMODE                                    ; assumes DISPLAYDATA has already been type checked.
    (PROG ((PBT (ffetch DDPILOTBBT of DISPLAYDATA))
           (FONT (ffetch DDFONT of DISPLAYDATA))
           (Y (\DSPTRANSFORMY (MICASTOPTS (ffetch DDYPOSITION of DISPLAYDATA))
                             DISPLAYDATA))
           TOP CHARTOP BM)
          [SETQ CHARTOP (IPLUS Y (LISTGET (fetch OTHERDEVICEFONTPROPS of FONT)
                                          'ASCENT]
          [freplace PBTDEST of PBT with (\ADDBASE (fetch BITMAPBASE of (SETQ BM (ffetch DDDestination of DISPLAYDATA))
                                                         )
                                                 (ITIMES (ffetch BITMAPRASTERWIDTH of BM)
                                                         (\SFInvert BM (SETQ TOP (IMAX (IMIN (ffetch DDClippingTop
                                                                                                    of DISPLAYDATA)
                                                                                             CHARTOP)
                                                                                       0]
          [freplace PBTSOURCE of PBT with (\ADDBASE (fetch BITMAPBASE of (SETQ BM (ffetch (CHARSETINFO CHARSETBITMAP)
                                                                                          of CSINFO)))
                                                   (ITIMES (ffetch BITMAPRASTERWIDTH of BM)
                                                           (freplace DDCHARHEIGHTDELTA of DISPLAYDATA
                                                              with (IMIN (IMAX (IDIFFERENCE CHARTOP TOP)
                                                                               0)
                                                                         MAX.SMALL.INTEGER]
          (freplace PBTHEIGHT of PBT with (IMAX (IDIFFERENCE TOP (IMAX [IDIFFERENCE Y
                                                                                   (freplace DDCHARSETDESCENT
                                                                                      of DISPLAYDATA
                                                                                      with (LISTGET (fetch
                                                                                                     OTHERDEVICEFONTPROPS
                                                                                                     of FONT)
                                                                                                    'DESCENT]
                                                                        (ffetch DDClippingBottom of DISPLAYDATA)))
                                                0])
)

(ADDTOVAR IMAGESTREAMTYPES (PRESSDISPLAY (FONTCREATE \FONTCREATE.HCPYMODE.PRESS)
                                         (CREATECHARSET \CREATECHARSET.HCPYMODE.PRESS))
                           (INTERPRESSDISPLAY (FONTCREATE \FONTCREATE.HCPYMODE.INTERPRESS)
                                              (CREATECHARSET \CREATECHARSET.HCPYMODE.INTERPRESS)))

(DECLARE%: DONTEVAL@LOAD DOCOPY

(\HCPYMODEDISPLAYINIT)
)
```

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

(ADDTOVAR **NLAMA**)

(ADDTOVAR **NLAML**)

(ADDTOVAR **LAMA**)
)

FUNCTION INDEX

VARIABLE INDEX

CONSTANT INDEX

MACRO INDEX