```
File created:   4-Aug-89 16:46:48   {DSK}<LISPFILES>LIBRARY>PS-SKETCH-PATCH.;1

  changes to:   (VARS PS-SKETCH-PATCHCOMS)
                (PROPS (PS-SKETCH-PATCH MAKEFILE-ENVIRONMENT))
                (FNS FIX-SKETCH ADD.KNOWN.SKETCH.FONT NEW-SK-DECREASING-FONT-LIST NEW-SK-PICK-FONT
                     NEW-SKETCHW-HARDCOPYFN \BUILDSLUGCSINFO \CREATECHARSET)

 Read Table:    INTERLISP

    Package:    INTERLISP

      Format:    XCCS
```

```
(RPAQQ PS-SKETCH-PATCHCOMS
       ((FILES (SYSLOAD FROM LISPUSERS)
               SKETCH)
        (FNS FIX-SKETCH ADD.KNOWN.SKETCH.FONT NEW-SK-DECREASING-FONT-LIST NEW-SK-PICK-FONT
             NEW-SKETCHW-HARDCOPYFN)

        ;; NOTE: to compile the following 2 functions you need EXPORTS.ALL loaded.

        (FNS \BUILDSLUGCSINFO \CREATECHARSET)
        [VARS (SKETCH-PATCHES '((NEW-SK-PICK-FONT . SK.PICK.FONT)
                                (NEW-SK-DECREASING-FONT-LIST . SK.DECREASING.FONT.LIST)
                                (NEW-SKETCHW-HARDCOPYFN . SKETCHW.HARDCOPYFN]
        (ADDVARS (POSTSCRIPT.FONT.CONVERSIONS (HELVETICA . HELVETICA)
                        (TIMESROMAND . TIMESROMAN)
                        (COURIER . COURIER)
                        (GACHA . COURIER)
                        (CLASSIC . TIMESROMAN)
                        (MODERN . HELVETICA)
                        (CREAM . HELVETICA)
                        (TERMINAL . COURIER)
                        (LOGO . HELVETICA)
                        (MODERN . HELVETICA)))
        (VARS (\KNOWN.SKETCH.FONTSIZES))
        (GLOBALVARS (\KNOWN.SKETCH.FONTSIZES)
               POSTSCRIPT.FONT.CONVERSIONS)

        ;; finally actually do the patching of sketch.

        (P (FIX-SKETCH))
        (PROP (MAKEFILE-ENVIRONMENT FILETYPE)
              PS-SKETCH-PATCH)))

(FILESLOAD (SYSLOAD FROM LISPUSERS)
       SKETCH)

(DEFINEQ

(FIX-SKETCH
  [LAMBDA NIL                                                  ; Edited  7-Jul-89 19:40 by Matt Heffron
    (COND
       ((BOUNDP 'ALL.SKETCHES)

        ;; sketch is loaded

        (for X in SKETCH-PATCHES do (MOVD (CAR X)
                                          (CDR X)
                                          NIL T))
        (PROMPTPRINT "Sketch has been patched!")
        T)
       (T (PROMPTPRINT "Sketch doesn't seem to be loaded!")
          (PROMPTPRINT "When you load sketch, make sure to call the function FIX-SKETCH!")
          NIL])

(ADD.KNOWN.SKETCH.FONT
  [LAMBDA (FAMILY WID DEVICE FONT)                             ; Edited 21-Feb-89 15:06 by snow

    ;; add to the globally cached font list

    (DECLARE (GLOBALVARS \KNOWN.SKETCH.FONTSIZES))
    (LET ((CACHE (ASSOC FAMILY \KNOWN.SKETCH.FONTSIZES))
          (CACHED))
         (COND
            [(NULL CACHE)
             (if \KNOWN.SKETCH.FONTSIZES
                 then [NCONC1 \KNOWN.SKETCH.FONTSIZES (LIST FAMILY (LIST DEVICE (CONS WID FONT]
               else (SETQ \KNOWN.SKETCH.FONTSIZES (LIST (LIST FAMILY (LIST DEVICE (CONS WID FONT]
            (T (COND
                  ((SETQ CACHED (ASSOC DEVICE CACHE))
                   (NCONC1 CACHED (CONS WID FONT)))
                  (T (NCONC1 CACHE (CONS DEVICE (CONS WID FONT])

(NEW-SK-DECREASING-FONT-LIST
  [LAMBDA (FAMILY DEVICETYPE)                                    ; Edited 21-Feb-89 11:26 by snow

    ;; returns a list of fonts of family FAMILY which work on device DEVICETYPE

    [COND
       ((NULL FAMILY)
        (SETQ FAMILY 'MODERN]

    ;; convert to families that exist on the known devices.

;;; NOTE: this is a very bad way to convert the family.  It HARDCODES in the conversions for PRESS and INTERPRESS  and does nothing for new
;;; device types.  I have added the conversion for POSTSCRIPT that does things a little cleaner, but it should really look at a property of the device
;;; (fontconversions or some such animal.)  --was 2/19/89

    (LET ((CONVERSION))
         [COND
            [(EQ DEVICETYPE 'PRESS)
             (COND
                ((EQ FAMILY 'MODERN)
                 (SETQ FAMILY 'HELVETICA))
                ((EQ FAMILY 'CLASSIC)
                 (SETQ FAMILY 'TIMESROMAN))
                ((EQ FAMILY 'TERMINAL)
                 (SETQ FAMILY 'GACHA]
            [(EQ DEVICETYPE 'INTERPRESS)
             (COND
                ((EQ FAMILY 'HELVETICA)
                 (SETQ FAMILY 'MODERN))
                ((EQ FAMILY 'TIMESROMAN)
                 (SETQ FAMILY 'CLASSIC))
                ((EQ FAMILY 'GACHA)
                 (SETQ FAMILY 'TERMINAL]
            ((EQ DEVICETYPE 'POSTSCRIPT)
             (if (SETQ CONVERSION (ASSOC FAMILY POSTSCRIPT.FONT.CONVERSIONS))
                 then ;; convert the family here for postscript as well as the other well known devices.

                      (SETQ FAMILY (CDR CONVERSION]
         (for FONT in (SK.GUESS.FONTSAVAILABLE FAMILY DEVICETYPE) collect (FONTCOPY FONT 'DEVICE DEVICETYPE])
```

(NEW-SK-PICK-FONT

```
  [LAMBDA (WID STRING DEVICE FAMILY)                             ; Edited 22-Feb-89 07:53 by snow

    ;; returns the font in FAMILY that text should be printed in to have the text STRING fit into a region WID points wide

    (DECLARE (GLOBALVARS \KNOWN.SKETCH.FONTSIZES))
    (PROG (LASTFONT LASTSIZE DISPLAYFONT SCALE CACHEDFONT)
          (IF [SETQ CACHEDFONT (ASSOC WID (ASSOC DEVICE (ASSOC FAMILY \KNOWN.SKETCH.FONTSIZES]
              THEN (RETURN (CDR CACHEDFONT)))
          (RETURN (for FONT in (SK.DECREASING.FONT.LIST FAMILY DEVICE)
                     when (NOT (GREATERP [SETQ LASTSIZE (COND
                                                          ((SETQ SCALE (FONTPROP FONT 'SCALE))

                                                           ;; IF THERE IS A SCALE, YOU MUST SCALE THE FONT.

                                                           (QUOTIENT (STRINGWIDTH STRING FONT)
                                                                  SCALE))
                                                          ((SETQ DISPLAYFONT (FONTCOPY (SETQ LASTFONT FONT)
                                                                                    'DEVICE
                                                                                    'DISPLAY
                                                                                    'NOERROR T))
                                                                 ; use display if it exists.
                                                           (STRINGWIDTH STRING DISPLAYFONT))
                                                          (T     ; in some cases, font exists for devices other than display.
                                                             (QUOTIENT (STRINGWIDTH STRING FONT)
                                                                    (FONTPROP FONT 'SCALE]
                                        WID))
                     do                                          ; return a font for the proper device even though the display fonts
                                                                 ; are used to pick a size.
                        (ADD.KNOWN.SKETCH.FONT FAMILY WID DEVICE (FONTCOPY FONT 'DEVICE DEVICE))
                        (RETURN (FONTCOPY FONT 'DEVICE DEVICE))
                     finally (RETURN (COND
                                        ((OR (NULL LASTFONT)
                                             (GREATERP LASTSIZE (TIMES 1.5 WID)))
                                         'SHADE)
                                        (T                       ; use the smallest if it isn't too large.
                                           (FONTCOPY LASTFONT 'DEVICE DEVICE])
```

(NEW-SKETCHW-HARDCOPYFN

```
  [LAMBDA (SKETCHW OPENIMAGESTREAM)                              ; Edited 27-Jul-89 17:52 by Matt Heffron
                                                                 ; dumps the sketch onto OPENIMAGESTREAM.
                                                                 ; centers it within the DSPCLIPPINGREGION of
                                                                 ; OPENIMAGESTREAM
    (PROG ((SKETCH (INSURE.SKETCH (SKETCH.FROM.VIEWER SKETCHW)))
           (PAGEREGION (DSPCLIPPINGREGION NIL OPENIMAGESTREAM))
           (SKETCHREGION (SKETCH.REGION.VIEWED SKETCHW))
           (SCALE (VIEWER.SCALE SKETCHW))
           SKETCHREGIONINPAGECOORDS PAGELEFTSPACE PAGEBOTTOMSPACE PAGETOSKETCHFACTOR SKETCHX)
          (OR SKETCH (RETURN))
          (SPAWN.MOUSE)
```

```
        ;; move the margins out of the way

             (DSPLEFTMARGIN (MIN 0 (fetch (REGION LEFT) of PAGEREGION))
                    OPENIMAGESTREAM)
             (DSPBOTTOMMARGIN (MIN 0 (fetch (REGION BOTTOM) of PAGEREGION))
                    OPENIMAGESTREAM)
             (DSPTOPMARGIN (MAX (ITIMES MAX.SMALLP MAX.SMALLP)
                                (fetch (REGION TOP) of PAGEREGION))
                    OPENIMAGESTREAM)                                ; MAX.SMALLP^2 ought to be big enough...
             (DSPRIGHTMARGIN (MAX (ITIMES MAX.SMALLP MAX.SMALLP)
                                  (fetch (REGION RIGHT) of PAGEREGION))
                    OPENIMAGESTREAM)

        ;; PAGETOSKETCHFACTOR is the factor to multiply the page coordinates by to get into sketch coordinates.

             (STATUSPRINT SKETCHW "Hardcopying ...")
             [STREAMPROP OPENIMAGESTREAM 'PRINTOPTIONS (APPEND (LIST 'DOCUMENT.NAME (OR (SKETCH.TITLE SKETCHW)
                                                                                         "A Sketch"))
                                                              (STREAMPROP OPENIMAGESTREAM 'PRINTOPTIONS]
             (SETQ PAGETOSKETCHFACTOR (FQUOTIENT SCALE (DSPSCALE NIL OPENIMAGESTREAM)))
             (SETQ SKETCHREGIONINPAGECOORDS (SCALE.REGION.OUT SKETCHREGION PAGETOSKETCHFACTOR))
             (COND
                ((AND (NOT (EQ (IMAGESTREAMTYPE OPENIMAGESTREAM)
                               'PRESS))
                      (NOT (EQ (FETCH (IMAGEOPS IMROTATE) OF (FETCH (STREAM IMAGEOPS) OF OPENIMAGESTREAM))
                               'NILL))
                      (GREATERP (fetch WIDTH of SKETCHREGIONINPAGECOORDS)
                             (fetch WIDTH of PAGEREGION))
                      (GREATERP (fetch WIDTH of SKETCHREGIONINPAGECOORDS)
                             (fetch HEIGHT of SKETCHREGIONINPAGECOORDS)))

                    ;; we have a  stream that supports rotation, use it!

                    (DSPROTATE 90 OPENIMAGESTREAM)
                    (COND
                       ((NOT (EQ (IMAGESTREAMTYPE OPENIMAGESTREAM)
                                 'POSTSCRIPT))

                        ;; Since PostScript's DSPROTATE does the translate also..., dont't do it here. --HACK! HACK! HACK! --Matt.

                        (DSPTRANSLATE 0 (MINUS (FETCH (REGION HEIGHT) OF PAGEREGION))
                               OPENIMAGESTREAM)))
                    (DSPCLIPPINGREGION (SETQ PAGEREGION (SK.SWITCH.REGION.X.AND.Y PAGEREGION))
                           OPENIMAGESTREAM)

                    ;; (ROTATE.IP OPENIMAGESTREAM 90) (CONCATT.IP OPENIMAGESTREAM) (TRANSLATE.IP OPENIMAGESTREAM 0 -21590)
                    ;; (CONCATT.IP OPENIMAGESTREAM) (DSPCLIPPINGREGION (SETQ PAGEREGION (SK.SWITCH.REGION.X.AND.Y
                    ;; PAGEREGION)) OPENIMAGESTREAM)

                    ;; this was an incredibly bogus hack to make INTERPRESS only streams rotate the sketch image if they were too big.  Now it tries to
                    ;; do it on any stream that has a dsprotate function.

                    ))
             (SETQ PAGELEFTSPACE (QUOTIENT (DIFFERENCE (fetch (REGION WIDTH) of PAGEREGION)
                                                  (fetch (REGION WIDTH) of SKETCHREGIONINPAGECOORDS))
                                        2))
             (SETQ PAGEBOTTOMSPACE (QUOTIENT (DIFFERENCE (fetch (REGION HEIGHT) of PAGEREGION)
                                                    (fetch (REGION HEIGHT) of SKETCHREGIONINPAGECOORDS))
                                          2))

        ;; translate the sketch so that the lower left corner of the sketch region is at the lower left corner of the image on the page.

             [SETQ SKETCHX (TRANSLATE.SKETCH SKETCH (MINUS (TIMES (DIFFERENCE (SETQ PAGELEFTSPACE
                                                                                  (PLUS (fetch (REGION LEFT)
                                                                                           of PAGEREGION)
                                                                                        PAGELEFTSPACE))
                                                                            (fetch (REGION LEFT) of
                                                                                                    SKETCHREGIONINPAGECOORDS
                                                                                  ))
                                                                  PAGETOSKETCHFACTOR))
                                   (MINUS (TIMES (DIFFERENCE (SETQ PAGEBOTTOMSPACE (PLUS (fetch (REGION BOTTOM)
                                                                                           of PAGEREGION)
                                                                                        PAGEBOTTOMSPACE))
                                                        (fetch (REGION BOTTOM) of SKETCHREGIONINPAGECOORDS))
                                                 PAGETOSKETCHFACTOR] ; calculate the local parts for the interpress sketch.
             (SETQ SKETCHX (MAKE.LOCAL.SKETCH SKETCHX (CREATEREGION (TIMES PAGELEFTSPACE PAGETOSKETCHFACTOR)
                                                             (TIMES PAGEBOTTOMSPACE PAGETOSKETCHFACTOR)
                                                             (fetch (REGION WIDTH) of SKETCHREGION)
                                                             (fetch (REGION HEIGHT) of SKETCHREGION))
                                  PAGETOSKETCHFACTOR OPENIMAGESTREAM))
             (DRAW.LOCAL.SKETCH SKETCHX OPENIMAGESTREAM (CREATEREGION PAGELEFTSPACE PAGEBOTTOMSPACE
                                                               (fetch (REGION WIDTH) of SKETCHREGIONINPAGECOORDS)
                                                               (fetch (REGION HEIGHT) of SKETCHREGIONINPAGECOORDS)))
             (STATUSPRINT SKETCHW " done.")
             (RETURN OPENIMAGESTREAM])

)

;; NOTE: to compile the following 2 functions you need EXPORTS.ALL loaded.

(DEFINEQ
(\BUILDSLUGCSINFO
  [LAMBDA (WIDTH ASCENT DESCENT DEVICE SCALE)                    ; Edited 14-Feb-89 16:46 by snow

;;; builds a csinfo which contains only the slug (black rectangle) character

    (SETQ SCALE (OR SCALE 1))
    (PROG ((CSINFO (create CHARSETINFO
                          CHARSETASCENT _ ASCENT
                          CHARSETDESCENT _ DESCENT
                          IMAGEWIDTHS _ (\CREATECSINFOELEMENT)))
           WIDTHS OFFSETS BITMAP IMAGEWIDTHS)
             (SETQ WIDTHS (fetch (CHARSETINFO WIDTHS) of CSINFO))
             (for I from 0 to \MAXTHINCHAR do (\FSETWIDTH WIDTHS I WIDTH))
             (SETQ IMAGEWIDTHS (fetch (CHARSETINFO WIDTHS) of CSINFO))
             (for I from 0 to \MAXTHINCHAR do (\FSETWIDTH WIDTHS I WIDTH))
             [SELECTQ DEVICE
                 (INTERPRESS                                        ; don't need offsets in INTERPRESS fonts
                            NIL)
                 (PROGN (replace (CHARSETINFO OFFSETS) of CSINFO with (SETQ OFFSETS (\CREATECSINFOELEMENT)))
                        (for I from 0 to \MAXTHINCHAR do (\FSETOFFSET OFFSETS I 0))
                        [replace (CHARSETINFO CHARSETBITMAP) of CSINFO
                           with (SETQ BITMAP (BITMAPCREATE (ROUND (QUOTIENT WIDTH SCALE))
                                                    (ROUND (QUOTIENT (IPLUS ASCENT DESCENT)
                                                                  SCALE]
                        (BLTSHADE BLACKSHADE BITMAP 1 NIL (SUB1 (ROUND (QUOTIENT WIDTH SCALE]
             (RETURN CSINFO])


(\CREATECHARSET
  [LAMBDA (CHARSET FONT NOSLUG?)                                 ; Edited 14-Feb-89 16:29 by snow
    ;; Creates and returns the CHARSETINFO for charset CHARSET in fontdesc FONT, installing it in fonts FONTCHARSETVECTOR
                                                                 ; NOSLUG? means don't create an empty (slug) csinfo if the
                                                                 ; charset is not found, just return NIL
    (DECLARE (GLOBALVARS \DISPLAYSTREAMTYPES))
    (AND (IGREATERP CHARSET \MAXCHARSET)
         (\ILLEGAL.ARG CHARSET))
    (PROG (CSINFO CREATEFN)
        ;; For other charsets, create a font descriptor of info for that charset, and use it to fill things in.
             (if (OR (AND (IGEQ CHARSET 1)
                          (ILEQ CHARSET 32))
                     (AND (IGEQ CHARSET 127)
                          (ILEQ CHARSET 160)))
                    then ;; this is an illegal NS character set (reserved for control codes) so just return a slug (unless NOSLUG? is T)
                         [if NOSLUG?
                          then (RETURN NIL)
                        else (SETQ CSINFO (\BUILDSLUGCSINFO (fetch (FONTDESCRIPTOR FONTAVGCHARWIDTH) of FONT)
                                                 (FONTPROP FONT 'ASCENT)
                                                 (FONTPROP FONT 'DESCENT)
                                                 (FONTPROP FONT 'DEVICE)
                                                 (FONTPROP FONT 'SCALE]
                else [SETQ CREATEFN (COND
                                       ((FMEMB (FONTPROP FONT 'DEVICE)
                                               \DISPLAYSTREAMTYPES)
                                        (FUNCTION \CREATECHARSET.DISPLAY))
                                       (T (CADR (ASSOC 'CREATECHARSET (CDR (ASSOC (FONTPROP FONT 'DEVICE)
                                                                                     IMAGESTREAMTYPES]
                     [if [NOT (SETQ CSINFO (APPLY CREATEFN (APPEND (FONTPROP FONT 'DEVICESPEC)
                                                                  (LIST CHARSET FONT NOSLUG?]
                        then                                        ; the create method returned NIL. so if NOSLUG? return NIL
                                                                 ; else build a slug charsetinfo
                               (RETURN (if NOSLUG?
                                           then                       ; the caller just wants NIL back to signal that nothing was found
                                               NIL
                                        else (\BUILDSLUGCSINFO (fetch (FONTDESCRIPTOR FONTAVGCHARWIDTH) of FONT)
                                                    (FONTPROP FONT 'ASCENT)
                                                    (FONTPROP FONT 'HEIGHT)
                                                    (FONTPROP FONT 'DEVICE)
                                                    (FONTPROP FONT 'SCALE]
                     (replace \SFAscent of FONT with (IMAX (fetch \SFAscent of FONT)
                                                        (fetch CHARSETASCENT of CSINFO)))
                     (replace \SFDescent of FONT with (IMAX (fetch \SFDescent of FONT)
                                                            (ffetch CHARSETDESCENT of CSINFO)))
                     (replace \SFHeight of FONT with (IPLUS (fetch \SFAscent of FONT)
                                                            (ffetch \SFDescent of FONT)))
                                                                 ; jtm: height = ascent + descent, not (IMAX fontHeight
                                                                 ; charSetHeight)

                     )
             (RETURN (\SETCHARSETINFO (ffetch FONTCHARSETVECTOR of FONT)
                               CHARSET CSINFO])

)

(RPAQQ SKETCH-PATCHES ((NEW-SK-PICK-FONT . SK.PICK.FONT)
                               (NEW-SK-DECREASING-FONT-LIST . SK.DECREASING.FONT.LIST)
```

```
                         (NEW-SKETCHW-HARDCOPYFN . SKETCHW.HARDCOPYFN)))

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

(RPAQQ \KNOWN.SKETCH.FONTSIZES NIL)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS (\KNOWN.SKETCH.FONTSIZES)
       POSTSCRIPT.FONT.CONVERSIONS)
)
```

;; finally actually do the patching of sketch.

```
(FIX-SKETCH)

(PUTPROPS PS-SKETCH-PATCH MAKEFILE-ENVIRONMENT (:PACKAGE "INTERLISP" :READTABLE "INTERLISP"))

(PUTPROPS PS-SKETCH-PATCH FILETYPE :TCOMPL)

(PUTPROPS PS-SKETCH-PATCH COPYRIGHT ("ENVOS Corporation" 1989))
```

{MEDLEY}<lispusers>PS-SKETCH-PATCH.;1   28-Jun-2024 18:34:03

-- Listed on 30-Jun-2024 13:14:43 --

FUNCTION INDEX

VARIABLE INDEX

PROPERTY INDEX