File created: 4-Feb-87 19:48:57 **{ERIS}<IRIS>NEXT>IRISSTREAM.;10**

changes to: (RECORDS IRISDATA)
(FNS OPENIRISSTREAM CLEARIRIS DRAWBITMAP TRYGRAPHER \FONTCREATE.IRIS \OUTCHARFN.IRIS
\IRISSTREAMINIT \IRIS.ASSURE.COLOR \LOOKUPRGB BOXSCREEN IRIS.CONS.OBJNO IRISBITMAP
FILLPOLYGON INSTALL.OBJFONT \CLOSEF.IRIS R SPPINPUTSTREAM \BACKCOLOR.IRIS \BITBLT.IRIS
\BLTSHADE.IRIS \FONTSAVAILABLE.IRIS \LEFTMARGIN.IRIS \RESET.IRIS \PSPLINE.TO.BEZIER.GEOMETRY
\SCALE.IRIS \SCALE.SPLINE.BY.DERIVS \STRINGWIDTH.IRIS \TERPRI.IRIS \FONT.IRIS
\CREATECHARSET.IRIS \IRISSETFONTBASE \IRISFONTBASE \CHANGECHARSET.IRIS \CHARWIDTH.IRIS
\CLIPPINGREGION.IRIS \CLOSEFN.IRIS \COLOR.IRIS \DRAWCIRCLE.IRIS \DRAWCURVE.IRIS
\DRAWLINE.IRIS \CONVERTLINESTYLE.IRIS \MOVETO.IRIS \XPOSITION.IRIS \YPOSITION.IRIS
\FILLCIRCLE.IRIS \DRAWELLIPSE.IRIS \FILLPOLYGON.IRIS \IRIS.BITBLT \DRAWPOLYGON.IRIS ALIGN)
(VARS IRISSTREAMCOMS \BEZIERBASIS.IRIS)

previous date: 16-Jan-87 18:14:11 {ERIS}<IRIS>NEXT>IRISSTREAM.;5

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ IRISSTREAMCOMS
       ((FILES SFFONT COLOR IRISLIB IRISIO IRISNET)
        [ADDVARS (IMAGESTREAMTYPES (IRIS (OPENSTREAM OPENIRISSTREAM)
                                         (FONTCREATE \FONTCREATE.IRIS)
                                         (FONTSAVAILABLE \FONTSAVAILABLE.IRIS)
                                         (CREATECHARSET \CREATECHARSET.IRIS]
        (GLOBALVARS \IRIS.VERBOSE IRISNSHOSTNUMBER \IRIS.DEBUG \BEZIERBASIS.IRIS \IRISCOLORMAPCACHE
              \IRIS.VERSION \IRISSTREAM IRISCONN \IV.HIGHOBJNO \IRIS.BITPLANES)
        (INITVARS (\IRIS.VERBOSE T)
               (\IRISSTREAMS NIL)
               (\IRIS.VERSION 'GL2)
               (IRISFONTDIRECTORIES '{ERIS}<IRIS>SF>)
               (\CHARSEGMENTS.IRIS 10)
               (\IRIS.BITPLANES 4)
               (IRISFONTFAMILIES '(GACHA TIMESROMAN))
               (IRISFONTROTATIONS '(0))
               (IRISFONTSIZES '(8 10 12 14 18 24))
               (\IRIS.DEBUG NIL)
               (IRISNSHOSTNUMBER "0#4000.12000.41504#0")
               (\IV.HIGHOBJNO 100000))
        (VARS \BEZIERBASIS.IRIS \IRIS.VERBOSE \BEZIERBASIS.IRIS \IRISCOLORMAPCACHE \IRIS.VERSION)
        (CONSTANTS IRIS.YAXIS IRIS.ZAXIS)
        (FNS BOXSCREEN CLEARIRIS DRAWBITMAP IRIS.CONS.OBJNO IRISBITMAP INSTALL.OBJFONT OPENIRISSTREAM
             \CLOSEF.IRIS R SPPINPUTSTREAM TRYGRAPHER \BACKCOLOR.IRIS \BITBLT.IRIS \BLTSHADE.IRIS
             \FONTCREATE.IRIS \FONTSAVAILABLE.IRIS \LEFTMARGIN.IRIS \RESET.IRIS \LOOKUPRGB
             \PSPLINE.TO.BEZIER.GEOMETRY \SCALE.IRIS \SCALE.SPLINE.BY.DERIVS \TERPRI.IRIS \FONT.IRIS
             \CREATECHARSET.IRIS \IRISSETFONTBASE \IRISFONTBASE \CHANGECHARSET.IRIS \CHARWIDTH.IRIS
             \OUTCHARFN.IRIS \CLIPPINGREGION.IRIS \CLOSEFN.IRIS \COLOR.IRIS \IRIS.ASSURE.COLOR \DRAWCIRCLE.IRIS
             \DRAWCURVE.IRIS \DRAWLINE.IRIS \CONVERTLINESTYLE.IRIS \IRISSTREAMINIT \MOVETO.IRIS \XPOSITION.IRIS
             \YPOSITION.IRIS \FILLCIRCLE.IRIS \DRAWELLIPSE.IRIS \FILLPOLYGON.IRIS \IRIS.BITBLT \DRAWPOLYGON.IRIS
             ALIGN)
```

;;; test functions

```
        (RECORDS BEZIER IRISDATA IRISSTREAM SPLINE)
        (CONSTANTS (\ALTLINESTYLE.IRIS 1)
               (\IRIS.ITALICS.ROTATION -100)
               (\PRIMARYLINESTLE.IRIS 0)
               (\IRIS.BOLD.LINEWIDTH 2))
        [P (\IRISSTREAMINIT)
           (SETFONTCLASSCOMPONENT DEFAULTFONT 'IRIS '(GACHA 12]
        [ADDVARS (DEFAULTPRINTINGHOST (IRIS Iris))
               (PRINTERTYPES (IRIS (CANPRINT (IRIS))
                                   (BITMAPFILE (IRISBITMAP FILE BITMAP SCALEFACTOR REGION ROTATION TITLE]
        (P (PUTPROP 'Iris 'PRINTERTYPE 'IRIS))
        (PROP PRINTERTYPE Iris)
        (FUNCTIONS WITH.IRIS.ATTR)))

(FILESLOAD SFFONT COLOR IRISLIB IRISIO IRISNET)

(ADDTOVAR IMAGESTREAMTYPES (IRIS (OPENSTREAM OPENIRISSTREAM)
                                 (FONTCREATE \FONTCREATE.IRIS)
                                 (FONTSAVAILABLE \FONTSAVAILABLE.IRIS)
                                 (CREATECHARSET \CREATECHARSET.IRIS)))

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \IRIS.VERBOSE IRISNSHOSTNUMBER \IRIS.DEBUG \BEZIERBASIS.IRIS \IRISCOLORMAPCACHE \IRIS.VERSION
       \IRISSTREAM IRISCONN \IV.HIGHOBJNO \IRIS.BITPLANES)
)

(RPAQ? \IRIS.VERBOSE T)

(RPAQ? \IRISSTREAMS NIL)
```

```
(RPAQ? \IRIS.VERSION 'GL2)

(RPAQ? IRISFONTDIRECTORIES '{ERIS}<IRIS>SF>)

(RPAQ? \CHARSEGMENTS.IRIS 10)

(RPAQ? \IRIS.BITPLANES 4)

(RPAQ? IRISFONTFAMILIES '(GACHA TIMESROMAN))

(RPAQ? IRISFONTROTATIONS '(0))

(RPAQ? IRISFONTSIZES '(8 10 12 14 18 24))

(RPAQ? \IRIS.DEBUG NIL)

(RPAQ? IRISNSHOSTNUMBER "0#4000.12000.41504#0")

(RPAQ? \IV.HIGHOBJNO 100000)

(RPAQQ \BEZIERBASIS.IRIS ((-1.0 3.0 -3.0 1.0)
                          (3.0 -6.0 3.0 0.0)
                          (-3.0 3.0 0.0 0.0)
                          (1.0 0.0 0.0 0.0)))

(RPAQQ \IRIS.VERBOSE T)

(RPAQQ \BEZIERBASIS.IRIS ((-1.0 3.0 -3.0 1.0)
                          (3.0 -6.0 3.0 0.0)
                          (-3.0 3.0 0.0 0.0)
                          (1.0 0.0 0.0 0.0)))

(RPAQQ \IRISCOLORMAPCACHE
       (((0 0 0) . 0)
        ((255 255 255) . 7)
        ((0 255 0) . 2)
        ((0 0 255) . 4)
        ((255 0 0) . 1)
        ((255 255 0) . 3)
        ((255 0 255) . 5)
        ((0 255 255) . 6)))

(RPAQQ \IRIS.VERSION GL2)

(DECLARE%: EVAL@COMPILE

(RPAQQ IRIS.YAXIS 89)

(RPAQQ IRIS.ZAXIS 90)

(CONSTANTS IRIS.YAXIS IRIS.ZAXIS)
)

(DEFINEQ
```

(BOXSCREEN

```
  [LAMBDA NIL                                                    (* gbn " 8-Nov-85 16:56")

          (* * draw a box around the screen)

    (DRAWLINE 0 0 (SUB1 SCREENWIDTH)
           0 NIL NIL \IRISSTREAM)
    (DRAWTO (SUB1 SCREENWIDTH)
           (SUB1 SCREENHEIGHT)
           NIL NIL \IRISSTREAM)
    (DRAWTO 0 (SUB1 SCREENHEIGHT)
           NIL NIL \IRISSTREAM)
    (DRAWTO 0 0 NIL NIL \IRISSTREAM)
    (FLUSHOUTPUT IRISCONN])
```

(CLEARIRIS

```
  [LAMBDA (IRIS-CONNECTION IRISSTREAM)                           ; Edited  2-Feb-87 23:36 by gbn
    (OR IRISSTREAM (SETQ IRISSTREAM \IRISSTREAM))
    [OR IRIS-CONNECTION (SETQ IRIS-CONNECTION (fetch SPPOUTSTREAM of (fetch IRISDATA of \IRISSTREAM]
    (IRIS.SINGLEBUFFER IRIS-CONNECTION)
    (IRIS.GCONFIG IRIS-CONNECTION)
    (DSPCLIPPINGREGION WHOLESCREEN IRISSTREAM)
    (IRIS.ORTHO 0 SCREENWIDTH 0 SCREENHEIGHT -1000 1000 IRIS-CONNECTION)
    (DSPCOLOR 'WHITE IRISSTREAM)
    (IRIS.LINEWIDTH 1 IRIS-CONNECTION)                           ; IRIS.RESETLS 0 IRISCONN
                                                                 ; make the IRIS not reset the line style between curve segments
    (IRIS.CURSOFF IRIS-CONNECTION)
    (IRIS.CLEAR IRIS-CONNECTION)
    (DSPCOLOR 'BLUE IRISSTREAM)
    (SELECTQ \IRIS.VERSION
         (GL2 (IRIS.CURVEPRECISION \CHARSEGMENTS.IRIS IRIS-CONNECTION)
```

```
            (IRIS.DEFBASIS 1 \BEZIERBASIS.IRIS IRIS-CONNECTION)
            (IRIS.CURVEBASIS 1 IRIS-CONNECTION))
        (GL1)
        (ERROR "Unknown version of IRIS: " \IRIS.VERSION))
   (DSPRESET IRISSTREAM)
   (SPP.FORCEOUTPUT IRIS-CONNECTION])
```

(DRAWBITMAP

```
[LAMBDA (BITMAP SOURCELEFT SOURCEBOTTOM OUTPUTSTREAM DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT)
                                                                    ; Edited 2-Feb-87 23:37 by gbn
  (LET ((DESTBOTTOM (OR DESTINATIONBOTTOM (DSPYPOSITION NIL OUTPUTSTREAM)))
        (DESTLEFT (OR DESTINATIONLEFT (DSPXPOSITION NIL OUTPUTSTREAM)))
        (WIDTH (OR WIDTH (BITMAPWIDTH BITMAP)))
        (HEIGHT (OR HEIGHT (BITMAPHEIGHT BITMAP)))
        (SBOTTOM (OR SOURCEBOTTOM 0))
        (SLEFT (OR SOURCELEFT 0))
        ROW)
       [for Y from SOURCEBOTTOM to (IPLUS SOURCEBOTTOM HEIGHT) as YBASE from 0
          do (SETQ ROW (IPLUS DESTBOTTOM YBASE))

       (* if there is a pixel set on the row, it is better to set the row outside the loop)

            (bind (STATE _ 'SKIP0S)
                  START END for X from SLEFT to (IPLUS SLEFT WIDTH) as BASE from 0
               do (SELECTQ STATE
                       (SKIP0S (if (IEQP 0 (BITMAPBIT BITMAP X Y))
                                   then                          (* skipping zeros, found a zero, so do nothing)
                                        NIL
                                 else                            (* start a run.)
                                      (SETQ START BASE)
                                      (SETQ END BASE)
                                      (SETQ STATE 'COLLECT1S)))
                       (COLLECT1S (if (ILESSP END (add END (BITMAPBIT BITMAP X Y)))
                                      then                       (* collecting 1's, found one. The test already incremented END,
                                                                    so do nothing)
                                           NIL
                                    else (DRAWLINE (IPLUS DESTLEFT START)
                                                   ROW
                                                   (IPLUS DESTLEFT END)
                                                   ROW 1 NIL OUTPUTSTREAM)
                                         (SETQ STATE 'SKIP0S)))
                       (SHOULDNT "Unknown state: " STATE))
               finally (if (EQ STATE 'COLLECT1S)
                           then (DRAWLINE (IPLUS DESTLEFT START)
                                          ROW
                                          (IPLUS DESTLEFT END)
                                          ROW 1 NIL OUTPUTSTREAM]
       (MOVETO DESTLEFT DESTBOTTOM OUTPUTSTREAM])
```

(IRIS.CONS.OBJNO

```
[LAMBDA NIL                                                         (* gbn "15-Nov-85 15:41")
  (add \IV.HIGHOBJNO 1])
```

(IRISBITMAP

```
[LAMBDA (FILE BITMAP SCALEFACTOR REGION ROTATION TITLE)             (* gbn "24-Oct-85 16:51")
  (LET [(IRISSTREAM (OPENIMAGESTREAM '{LPT}Iris.IRIS 'IRIS]
       (BITBLT BITMAP (fetch (REGION LEFT) of REGION)
              (fetch (REGION BOTTOM) of REGION)
              IRISSTREAM
              (DSPXPOSITION NIL IRISSTREAM)
              (DSPYPOSITION NIL IRISSTREAM)
              (fetch (REGION WIDTH) of REGION)
              (fetch (REGION HEIGHT) of REGION])
```

(INSTALL.OBJFONT

```
[LAMBDA (FAMILY CHARSET LOWCHARCODE HIGHESTCHARCODE SCALE IRISSTREAM CSINFO)
                                                                    (* gbn "12-Nov-85 19:17")
```

(* * takes a font in SF format that is already in core, ie, part of the value of \SPLINEFONTSINCORE, and installs it on the iris connected to STREAM. Characters in the font which are nil are not downloaded)

(* * note that this fn is called by the fontcreate method for the iris, even when the font has already been downloaded. This fn looks up in the stream and just returns the old cached info from the original downloading.)

```
   (if \IRIS.DEBUG
       then (SETQ LOWCHARCODE 97)
            (SETQ HIGHESTCHARCODE 101)
     else (SETQ LOWCHARCODE (OR LOWCHARCODE 1))
          (SETQ HIGHESTCHARCODE (OR HIGHESTCHARCODE 255)))
   (PROG ((FONTARRAY (\LOOKUPSPLINEFONT FAMILY CHARSET))
          (IRISDATA (fetch IRISDATA of IRISSTREAM))
          (MAXHEIGHT 0)
          SPACEWIDTH OBJ# FONTBASE CHARDESC WIDTHARRAY STREAM)
```

```
        (SETQ STREAM (fetch SPPOUTSTREAM of IRISDATA))
        (if (NOT FONTARRAY)
            then (ERROR "Charset for spline font not in core:" (LIST FAMILY CHARSET)))
        (if (SETQ FONTBASE (\IRISFONTBASE FAMILY CHARSET IRISDATA))
            then                                          (* the font has already been downloaded, just return the cached
                 info)
                 (if \IRIS.DEBUG
                     then (SHOULDNT "font being redefined"))
                 (RETURN FONTBASE)
          else (SETQ FONTBASE (add (fetch HIFONT# of IRISDATA)
                                   256))
               (SETQ WIDTHARRAY (\CREATECSINFOELEMENT)))
        (if \IRIS.VERBOSE
            then (PROMPTPRINT "Installing font on IRIS: " FAMILY))
        (if (ZEROP (IRIS.ISOBJ 0 STREAM))
            then (IRIS.MAKEOBJ 0 STREAM)
                 (IRIS.CURSOFF STREAM)
                 (IRIS.CLOSEOBJ STREAM))

        (* character 0 of the font is always defined on the IRIS as the way of telling if this charset has been downloaded.)
                                                          (* NOT ANY MORE)
        [for I from LOWCHARCODE to HIGHESTCHARCODE
           do (SETQ CHARDESC (ELT FONTARRAY I))
              (if CHARDESC
                  then (IRIS.MAKEOBJ (SETQ OBJ# (IPLUS FONTBASE I))
                                     STREAM)
                       (SFDRAW CHARDESC NIL 0 0 SCALE IRISSTREAM)

        (* The scale is always one when called for the iris, because the printchar method makes the IRIS scale the character
        anyway)

                       (IRIS.CLOSEOBJ STREAM)
                       (ALIGN)                            (* CONSISTENCY CHECK)
                       (if (ZEROP (IRIS.ISOBJ OBJ# STREAM))
                           then (ERROR
                                      '"(OBJECT FONT CHARACTER IS UNDEFINED DIRECTLY AFTER DEFINING INSIDE
                                       INSTALL.OBJFONT)")
                         else (PRINTOUT PROMPTWINDOW (CHARACTER I]
        (IRIS.GFLUSH STREAM)
        (ALIGN)
        (\IRISSETFONTBASE FAMILY CHARSET IRISDATA FONTBASE)
        (SETQ OBJ# (IPLUS FONTBASE (CHARCODE SPACE)))
        (if (ZEROP (IRIS.ISOBJ OBJ# STREAM))
            then                                          (* install a fake space char if there isn't one)
                 (SETQ SPACEWIDTH (\FGETWIDTH WIDTHARRAY (OR HIGHESTCHARCODE 127)))
                 (IRIS.MAKEOBJ OBJ# STREAM)
                 (MOVETO SPACEWIDTH 0 IRISSTREAM)
                 (IRIS.CLOSEOBJ STREAM)
                 (\FSETWIDTH WIDTHARRAY (CHARCODE SPACE)
                         SPACEWIDTH))
        (RETURN (LIST FAMILY FONTBASE WIDTHARRAY MAXHEIGHT])
```

(OPENIRISSTREAM

```
  [LAMBDA (NSHOSTNUMBER OPTIONS)                          ; Edited  4-Feb-87 19:05 by gbn

        (* * opens a stream to an iris workstation)

    (DECLARE (GLOBALVARS \IRISIMAGEOPS))
    (PROG ((IRISDATA (create IRISDATA))
           (HOST (OR NSHOSTNUMBER IRISNSHOSTNUMBER))
           (IRISSTREAM (OPENSTREAM '{NODIRCORE}IRIS.SCRATCH 'OUTPUT 'NEW 8 'BINARY))
           TEMPCONN)
          (if (NOT HOST)
              then (ERROR "IRISNSHOSTNUMBER must be supplied"))
          (if [AND (NOT (EQMEMB 'RECONNECT OPTIONS))
                   (SPP.OPENP IRISCONN)
                   (OR (NULL HOST)
                       (SELECTQ (TYPENAME HOST)
                            (NSADDRESS (EQUAL (fetch (NSADDRESS NSHOSTNUMBER) of HOST)
                                              (fetch (NSADDRESS NSHOSTNUMBER) of (SPP.DESTADDRESS IRISCONN))))
                            (LISTP (EQUAL HOST (fetch (NSADDRESS NSHOSTNUMBER) of (SPP.DESTADDRESS IRISCONN))))
                            (LITATOM (EQ 'LPT (LISTGET (UNPACKFILENAME HOST)
                                                       'HOST)))
                            (NILL]
              then

        (* there is still a stream open to the iris. Just use that, since one can have at most a single stream open to the iris)

                   (RETURN \IRISSTREAM)
            else (if [AND (TYPENAMEP HOST 'LITATOM)
                          (EQ 'LPT (LISTGET (UNPACKFILENAME HOST)
                                            'HOST]
                     then

        (* * This is just a request to hardcopy when there is no open stream)
```

```
                    (SETQ HOST NIL))
                (if (NOT (SETQ TEMPCONN (OPEN.IRISCONN HOST)))
                    then (ERROR "Iris did not respond to connection attempt" HOST)))
                                                        (* replace (STREAM USERCLOSEABLE) of IRISSTREAM with
    NIL)
        (STREAMADDPROP IRISSTREAM 'BEFORECLOSE '\CLOSEF.IRIS)
        (replace (STREAM OUTCHARFN) of IRISSTREAM with (FUNCTION \OUTCHARFN.IRIS))
        (replace (IRISSTREAM IMAGEOPS) of IRISSTREAM with \IRISIMAGEOPS)
        (replace (IRISSTREAM IRISDATA) of IRISSTREAM with IRISDATA)
        (replace (IRISDATA SPPINSTREAM) of IRISDATA with TEMPCONN)
        (replace (IRISDATA SPPOUTSTREAM) of IRISDATA with (SPPOUTPUTSTREAM TEMPCONN))
        (replace (IRISDATA IRISCOLORMAPCACHE) of IRISDATA with \IRISCOLORMAPCACHE)
                                                        (* replace (IRISDATA IRISCOLORMAP) of IRISDATA with
                                                        (COLORMAPCREATE NIL \IRIS.BITPLANES))
        (replace (IRISDATA IRISCHARSET) of IRISDATA with -1)
        (replace (IRISDATA IRISPAGE) of IRISDATA with (COPY WHOLESCREEN))
        (SETQ IRISCONN (fetch SPPOUTSTREAM of IRISDATA))
        (IRIS.GINIT IRISCONN)
        (CLEARIRIS IRISCONN IRISSTREAM)
        (RETURN (SETQ \IRISSTREAM IRISSTREAM])
```

(\CLOSEF.IRIS

```
  [LAMBDA (IRISSTREAM)                                  (* gbn "25-Oct-85 17:18")
```

(* * this fn is installed on the stream as a streamprop. It flushs the output to the stream, but does not close it)

```
    (FORCEOUTPUT (fetch SPPOUTSTREAM of (fetch IRISDATA of IRISSTREAM)))
    (RETFROM 'CLOSEF NIL])
```

(R

```
  [LAMBDA NIL                                           (* gbn "21-Jun-85 03:57")
    (OPENIRISSTREAM NIL '(DONTCONNECT])
```

(SPPINPUTSTREAM

```
  [LAMBDA (OUTPUTSTREAM)                                (* gbn "17-Jun-85 17:40")
    (PROG ((CON (fetch (STREAM F1) of OUTPUTSTREAM)))
          (if CON
              then (RETURN (fetch (SPPCON SPPINPUTSTREAM) of CON])
```

(TRYGRAPHER

```
  [LAMBDA (DONTSETUP?)                                  ; Edited  2-Feb-87 23:43 by gbn
```

;;; just a hack to try to draw a grapher graph

;;; comment

```
    (PROG (G)
          [SETQ G (LAYOUTSEXPR '(stu (wxy xxx)
                                     (xyzzy))
                               NIL NIL (FONTCREATE 'GACHA 20 NIL NIL 'IRIS]
          (IF (NOT DONTSETUP?)
              THEN (IRIS.GRESET)
                   (IRIS.SINGLEBUFFER)                  ; (IRIS.DOUBLEBUFFER)
                   (IRIS.GCONFIG)
                   (IRIS.PERSPECTIVE 120 1 -1000 1000)
                   (IRIS.LOOKAT 0 0 30000 0 0 0 0)
                   (IRIS.LINEWIDTH 2)
                   (CLEAR 'BLUE)
                   (IRIS.COLOR 'WHITE))
          (DISPLAYGRAPH G \IRISSTREAM)
          (RETURN])
```

(\BACKCOLOR.IRIS

```
  [LAMBDA (STREAM COLOR)                                ; Edited 16-Jan-87 13:58 by gbn
    (IF COLOR
        THEN (REPLACE (IRISDATA BACKCOLOR) OF (FETCH IRISDATA OF STREAM) WITH COLOR)
      ELSE (FETCH (IRISDATA BACKCOLOR) OF (FETCH IRISDATA OF STREAM])
```

(\BITBLT.IRIS

```
  [LAMBDA (SOURCEBITMAP SOURCELEFT SOURCEBOTTOM DESTINATION DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT
                SOURCETYPE OPERATION TEXTURE CLIPPINGREGION CLIPPEDSOURCELEFT CLIPPEDSOURCEBOTTOM)
                                                        (* gbn "12-Nov-85 14:35")
```

(* * produces a 3-d bitmap composed of lines)

```
    (if (NOT (EQ (IMAGESTREAMTYPE DESTINATION)
                 'IRIS))
        then (ERROR "Destination not IRIS stream: " DESTINATION))
    (DRAWBITMAP SOURCEBITMAP SOURCELEFT SOURCEBOTTOM DESTINATION DESTINATIONLEFT DESTINATIONBOTTOM WIDTH
          HEIGHT)
    (FLUSHOUTPUT (fetch SPPOUTSTREAM of (fetch IRISDATA of \IRISSTREAM])
```

(\BLTSHADE.IRIS

```
  [LAMBDA (TEXTURE IRISSTREAM DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT OPERATION CLIPPINGREGION)
                                                                    ; Edited 16-Jan-87 15:00 by gbn
                                                                    ; should not affect anything, so do a with attr
    (LET* ((IRISDATA (fetch IRISDATA of IRISSTREAM))
           (SPPOUT (fetch (IRISDATA SPPOUTSTREAM) of IRISDATA)))
          (WITH.IRIS.ATTR (IRIS.POLF2 4 (LIST (CREATEPOSITION DESTINATIONLEFT DESTINATIONBOTTOM)
                                              (CREATEPOSITION (IPLUS DESTINATIONLEFT WIDTH)
                                                     DESTINATIONBOTTOM)
                                              (CREATEPOSITION (IPLUS DESTINATIONLEFT WIDTH)
                                                     (IPLUS DESTINATIONBOTTOM HEIGHT))
                                              (CREATEPOSITION DESTINATIONLEFT (IPLUS DESTINATIONBOTTOM HEIGHT)))
                              SPPOUT)
                IRISSTREAM SPPOUT :COLOR TEXTURE)
          (FORCEOUTPUT SPPOUT])
```

(\FONTCREATE.IRIS

```
  [LAMBDA (FAMILY SIZE FACE ROTATION DEVICE CHARSET)                ; Edited  2-Feb-87 23:44 by gbn
```

;;; This function reads in the spline definition for a font, but does not install it on the iris. The installation is done on a demand basis on the IRIS, charset
;;; by charset.

```
    (PROG (WIDTHS (SCALE 1)
                  FONTDESC CSINFO)
```

;; since a spline font can be any size, we must guarantee that relative sizes are guaranteed, i.e. a 10 point font is twice as big as a 5 point font

```
          (SETQ SCALE 1)
```

;;; the width arrays, the height, ascent, etc are all scaled

```
          (SETQ FONTDESC (create FONTDESCRIPTOR
                                 FONTDEVICE _ 'IRIS
                                 FONTFAMILY _ FAMILY
                                 FONTSIZE _ SIZE
                                 FONTFACE _ FACE
                                 ROTATION _ ROTATION))              ; CHECK WHAT FONTSCALE MEANS
          (SETQ CSINFO (\GETCHARSETINFO 0 FONTDESC T))
          (if (NOT CSINFO)
              then (RETURN NIL))                                    ; this will call the createcharset method for the IRIS
          [SETQ SCALE (replace OTHERDEVICEFONTPROPS of FONTDESC with (QUOTIENT (FLOAT SIZE)
                                                                               (IPLUS (fetch (CHARSETINFO
                                                                                                CHARSETASCENT)
                                                                                             of CSINFO)
                                                                                      (fetch (CHARSETINFO
                                                                                                CHARSETDESCENT)
                                                                                             of CSINFO]
          (SETQ WIDTHS (fetch (CHARSETINFO WIDTHS) of CSINFO))
          [for I from 0 to \MAXTHINCHAR DO (\FSETWIDTH WIDTHS I (FIX (TIMES (\FGETWIDTH WIDTHS I)
                                                                            SCALE]
          (replace \SFHeight of FONTDESC with SIZE)
          [replace \SFAscent of FONTDESC with (FIX (TIMES SCALE (fetch (CHARSETINFO CHARSETASCENT) of CSINFO]
          [replace \SFDescent of FONTDESC with (FIX (TIMES SCALE (fetch (CHARSETINFO CHARSETDESCENT) of CSINFO]
                                                                    ; OTHERDEVICEFONTPROPS is used to hide the scale of the
                                                                    ; font on the iris
          (RETURN FONTDESC])
```

(\FONTSAVAILABLE.IRIS

```
  [LAMBDA (FAMILY PSIZE FACE ROTATION DEVICE)                       (* gbn "13-Nov-85 12:06")

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

    (DECLARE (GLOBALVARS IRISFONTDIRECTORIES))                      (* Normalize face)
    (LET [(FAMILIES (if (MEMB FAMILY IRISFONTFAMILIES)
                        then FAMILY
                      else NIL))
          (SIZES (SELECTQ PSIZE
                      (* IRISFONTSIZES)
                      (PROG1 PSIZE)))
          [FACES (SELECTQ FACE
                      (* '((MEDIUM REGULAR REGULAR)
                           (MEDIUM ITALIC REGULAR)
                           (BOLD REGULAR REGULAR)
                           (BOLD ITALIC REGULAR)))
                      (PROG1 (LIST (\FONTFACE FACE]
          (ROTATIONS (SELECTQ ROTATION
                          (* IRISFONTROTATIONS)
                          (PROG1 ROTATION]
         (for FFAMILY inside FAMILIES
            join (for SSIZE inside SIZES join (for FFACE in FACES
                                                 join (for RROTATION inside ROTATIONS
                                                         collect (LIST FFAMILY SSIZE FFACE RROTATION 'IRIS])
```

(\LEFTMARGIN.IRIS

```
[LAMBDA (MARGIN STREAM)                                    (* gbn " 8-Nov-85 17:33")
  (OR 0 (if MARGIN
            then (replace LEFT of (fetch IRISPAGE OF (fetch IRISDATA of STREAM)) with MARGIN)
          else (fetch LEFT of (fetch IRISPAGE OF (fetch IRISDATA of STREAM])
```

(\RESET.IRIS

```
[LAMBDA (IRISSTREAM)                                       (* gbn "13-Nov-85 00:46")
  (MOVETO (DSPLEFTMARGIN NIL IRISSTREAM)
         (IDIFFERENCE (fetch TOP of (DSPCLIPPINGREGION NIL IRISSTREAM))
                (FONTPROP (DSPFONT NIL IRISSTREAM)
                          'HEIGHT))
         IRISSTREAM])
```

(\LOOKUPRGB

```
[LAMBDA (RGB IRISDATA)                                     ; Edited  2-Feb-87 20:37 by gbn
```

;;; returns the colormap index whose value is RGB. Returns the closest found and caches that value.

;;; Since the colormap code is so flakey, the iris now relies only on the cache in the stream. Not very efficient.

```
    (LET* ((CACHE (FETCH IRISCOLORMAPCACHE OF IRISDATA))
           (INDEX (SASSOC RGB CACHE)))
          (IF INDEX
              THEN (CDR INDEX)
            ELSE ;; didn't find exactly the right index.  Now look through the cache.  For closeness in the color space, we use cartesian difference
                 ;; of the rgb's.
                 [SETQ INDEX (CDR (FOR PAIR IN CACHE SMALLEST (CL:FLET [(ABSDIF (X Y)
                                                                          (IABS (IDIFFERENCE X Y]
                                                                 (APPLY #'+ (CL:MAPCAR #'ABSDIF RGB
                                                                                       (CAR PAIR]
                 (REPLACE IRISCOLORMAPCACHE OF IRISDATA WITH (CL:ACONS RGB INDEX CACHE))
                 INDEX])
```

(\PSPLINE.TO.BEZIER.GEOMETRY

```
[LAMBDA (SPLINE KNOT#)                                     (* gbn " 7-Jul-85 20:49")
```

(* * returns a bezier geometry matrix from the spline for knot KNOT#.
(compare with SF.DERIVS.TO.BEZIER which does the same thing for a SF spline description))

(* * the derivatives must already be scaled by the Factorials)

(* * should not create the BEZIER)

```
    (PROG [(BEZ (create BEZIER
                        B0X _ (ELT (fetch SPLINEX of SPLINE)
                                   KNOT#)
                        B0Y _ (ELT (fetch SPLINEY of SPLINE)
                                   KNOT#]
          (replace B1X of BEZ with (PLUS (ffetch B0X of BEZ)
                                         (QUOTIENT (ELT (ffetch SPLINEDX of SPLINE)
                                                        KNOT#)
                                                3)))
          (replace B1Y of BEZ with (PLUS (ffetch B0Y of BEZ)
                                         (QUOTIENT (ELT (ffetch SPLINEDY of SPLINE)
                                                        KNOT#)
                                                3)))
          (replace B2X of BEZ with (PLUS (ffetch B1X of BEZ)
                                         (QUOTIENT (PLUS (ELT (ffetch SPLINEDX of SPLINE)
                                                              KNOT#)
                                                         (ELT (ffetch SPLINEDDX of SPLINE)
                                                              KNOT#))
                                                3)))
          (replace B2Y of BEZ with (PLUS (ffetch B1Y of BEZ)
                                         (QUOTIENT (PLUS (ELT (ffetch SPLINEDY of SPLINE)
                                                              KNOT#)
                                                         (ELT (ffetch SPLINEDDY of SPLINE)
                                                              KNOT#))
                                                3)))
          (replace B3X of BEZ with (PLUS (ffetch B0X of BEZ)
                                         (ELT (ffetch SPLINEDX of SPLINE)
                                              KNOT#)
                                         (ELT (ffetch SPLINEDDX of SPLINE)
                                              KNOT#)
                                         (ELT (ffetch SPLINEDDDX of SPLINE)
                                              KNOT#)))
          (replace B3Y of BEZ with (PLUS (ffetch B0Y of BEZ)
                                         (ELT (fetch SPLINEDY of SPLINE)
                                              KNOT#)
                                         (ELT (ffetch SPLINEDDY of SPLINE)
                                              KNOT#)
                                         (ELT (ffetch SPLINEDDDY of SPLINE)
```

```
                                        KNOT#)))
        (RETURN BEZ])
```

(\SCALE.IRIS

```
[LAMBDA (STREAM SCALE)                                  (* gbn "24-Jun-85 18:50")
  (if (NOT SCALE)
      then 1
    else (ERROR])
```

(\SCALE.SPLINE.BY.DERIVS

```
[LAMBDA (SPLINE)                                        (* gbn " 8-Jul-85 17:20")
```

(* * For the form used by \PSPLINE.TO.BEZIER.GEOMETRY, the derivs can all be premultiplied by the factorial coefficients, rather than repeatedly multiplying them in)

```
  (bind (DDX _ (fetch SPLINEDDX of SPLINE))
        (DDY _ (fetch SPLINEDDY of SPLINE))
        (DDDX _ (fetch SPLINEDDDX of SPLINE))
        (DDDY _ (fetch SPLINEDDDY of SPLINE)) for I from 1 to (fetch %#KNOTS of SPLINE)
     do (SETA DDX I (FQUOTIENT (ELT DDX I)
                               2.0))
        (SETA DDY I (FQUOTIENT (ELT DDY I)
                               2.0))
        (SETA DDDX I (FQUOTIENT (ELT DDDX I)
                                6.0))
        (SETA DDDY I (FQUOTIENT (ELT DDDY I)
                                6.0])
```

(\TERPRI.IRIS

```
[LAMBDA (STREAM)                                        (* gbn "12-Nov-85 14:37")
  (MOVETO (DSPLEFTMARGIN NIL STREAM)
          (IDIFFERENCE (DSPYPOSITION NIL STREAM)
                 (FONTPROP (DSPFONT NIL STREAM)
                           'HEIGHT))
          STREAM)
  (if (ILESSP (DSPYPOSITION NIL STREAM)
              0)
      then (DSPRESET STREAM))
  (FLUSHOUTPUT (fetch SPPOUTSTREAM of (fetch IRISDATA of STREAM])
```

(\FONT.IRIS

```
[LAMBDA (IRISSTREAM FONTDESC)                           (* gbn "29-Oct-85 15:25")
  (if FONTDESC
      then (LET ((IRISDATA (fetch IRISDATA of IRISSTREAM)))
                [if [NOT (AND (type? FONTDESCRIPTOR FONTDESC)
                              (EQ 'IRIS (fetch FONTDEVICE of FONTDESC]
                    then (SETQ FONTDESC (\COERCEFONTDESC FONTDESC 'IRIS]
                                                        (* user supplied a font so install it)
                (replace (IRISDATA CURRENTFONTDESC) of IRISDATA with FONTDESC)
                (\CHANGECHARSET.IRIS IRISSTREAM 0)      (* this validates the caches for fontbase, current charset, etc.)
                FONTDESC)
    else (fetch CURRENTFONTDESC of (fetch IRISDATA of IRISSTREAM])
```

(\CREATECHARSET.IRIS

```
[LAMBDA (FAMILY SIZE FACE ROTATION DEVICE CHARSET FONTDESC NOSLUG?)
                                                        ; Edited 16-Jan-87 16:43 by gbn
```

;;; This function reads in the spline definition for a CHARSET but does not install it on the iris. The installation is done on a demand basis on the IRIS, ;;; charset by charset.

```
  (PROG ((FONTARRAY (\LOOKUPSPLINEFONT FAMILY CHARSET))
         (MAXHEIGHT 1)
         WIDTHS CSINFO FONTARRAY SCALE FILES)
        (if (NOT FONTARRAY)
            then                                        ; we haven't even read this into core.
                 (SETQ FILES (DIRECTORY (PACK* IRISFONTDIRECTORIES FAMILY '*.*SF)
                                        'COLLECT))
                 ;; THIS METHOD OF FINDING THE FILES WILL NEED TO BE UPGRADED IF WE GET SPLINE DEFINITIONS FOR NS
                 ;; CHARACTERS
                 (if (NOT FILES)
                     then ;; if you can't find the file then just return NIL to createcharset who will either report the error or build a slug charset
                          (RETURN (if NOSLUG?
                                      then              ; if you can't find the file then just return NIL to createcharset who
                                                        ; will report the error
                                           NIL
                                    else                ; this will guarantee that all the chars in the charset have 0 width
                                         (\BUILDSLUGCSINFO 0 0 0)))
                   else (if \IRIS.VERBOSE
                            then (PROMPTPRINT "Reading the following spline font files: " FILES))
                        (if \IRIS.DEBUG
```

```
                        then (READ.SPLINE.FONT (CAR FILES)
                                    FAMILY CHARSET)
                      else (READ.SPLINE.FONT FILES FAMILY CHARSET)))
                                                        ; now see if it really worked
                (if (NOT (SETQ FONTARRAY (\LOOKUPSPLINEFONT FAMILY CHARSET)))
                    then                                ; we just lost horribly, so die with an inconsistency
                     (SHOULDNT "Inside \FONTCREATE.IRIS, some SFFONTS were found, but reading them did not
                           produce an entry in \SPLINEFONTSINCORE")))
                                                        ; we have the FAMILY/CHARSET entries, now see if there is a
                                                        ; font descriptor ready made for this size
    ;; since a spline font can be any size, we must guarantee that relative sizes are guaranteed, i.e.  a 10 point font is twice as big as a 5 point font
      (SETQ CSINFO (create CHARSETINFO))
      (SETQ WIDTHS (fetch (CHARSETINFO WIDTHS) of CSINFO))
      (bind CHARDESC for I from 1 to \MAXTHINCHAR
         do (SETQ CHARDESC (ELT FONTARRAY I))
            ;; If there is no description for a character, set its width to zero, so that dspprintchar can recognize not to call this character.
            (if CHARDESC
                then (\FSETWIDTH WIDTHS I (fetch XWIDTH of (fetch SF.WIDTH of CHARDESC)))
                     [SETQ MAXHEIGHT (IMAX MAXHEIGHT (fetch YFIDUCIAL of (fetch FIDUCIAL of CHARDESC]
              else (\FSETWIDTH WIDTHS I 0)))
      (replace (CHARSETINFO CHARSETASCENT) of CSINFO with (FIX (TIMES 0.7 MAXHEIGHT)))
      (replace (CHARSETINFO CHARSETDESCENT) of CSINFO with (FIX (TIMES 0.3 MAXHEIGHT)))
    ;; It doesn't look like this scale junk is used anymore.  gbn Jan 17/87
      [if (SETQ SCALE (fetch OTHERDEVICEFONTPROPS of FONTDESC))
          then ;; this fontdescriptor has already build character sets, and has determined its scale.  So scale the widths in this character set.
               ;; For the first character set, this is done in \fontcreate.iris
               (for I from 0 to \MAXTHINCHAR do (\FSETWIDTH WIDTHS I (FIX (TIMES (\FGETWIDTH WIDTHS I)
                                                                                 SCALE]
      (RETURN CSINFO])
```

(\IRISSETFONTBASE

```
[LAMBDA (FAMILY CHARSET IRISDATA FONTBASE)                  (* gbn "18-Oct-85 16:15")
  (PUTASSOC (CONS FAMILY CHARSET)
         FONTBASE
         (fetch FONTSINIRIS of IRISDATA])
```

(\IRISFONTBASE

```
[LAMBDA (FAMILY CHARSET IRISDATA)                           (* gbn "18-Oct-85 16:15")
  (CDR (SASSOC (CONS FAMILY CHARSET)
               (fetch FONTSINIRIS of IRISDATA])
```

(\CHANGECHARSET.IRIS

```
[LAMBDA (IRISSTREAM CHARSET)                                (* gbn "18-Oct-85 16:16")
```

(* * called when a character is about to be printed which is in a different charset than the current one.)

```
  (LET ((IRISDATA (fetch IRISDATA of IRISSTREAM))
        CSINFO BASE FONTDESC)
       (SETQ FONTDESC (fetch CURRENTFONTDESC of IRISDATA))
       [replace (IRISDATA IRISWIDTHSCACHE) of IRISDATA with (fetch (CHARSETINFO WIDTHS) of (SETQ CSINFO
                                                                                                (\GETCHARSETINFO
                                                                                                      CHARSET
                                                                                                      FONTDESC]
       (SETQ BASE (\IRISFONTBASE (fetch FONTFAMILY of FONTDESC)
                          CHARSET IRISDATA))
       (if (NOT BASE)
           then                                             (* this stream has never seen this charset before so install it on
                the IRIS.)
                (INSTALL.OBJFONT (fetch FONTFAMILY of FONTDESC)
                       CHARSET NIL NIL NIL IRISSTREAM CSINFO)
                (SETQ BASE (\IRISFONTBASE (fetch FONTFAMILY of FONTDESC)
                                  CHARSET IRISDATA)))
       (replace (IRISDATA CURRENTFONTBASE) of IRISDATA with BASE)
       (replace (IRISDATA IRISCHARSET) of IRISDATA with CHARSET])
```

(\CHARWIDTH.IRIS

```
[LAMBDA (CHARCODE FONT)                                     (* gbn "18-Oct-85 19:11")
  (FIX (TIMES (fetch OTHERDEVICEFONTPROPS of FONT)
              (\FGETCHARWIDTH FONT CHARCODE])
```

(\OUTCHARFN.IRIS

```
[LAMBDA (IRISSTREAM CHARCODE)                               ; Edited  2-Feb-87 23:46 by gbn
  (LET* ((IRISDATA (fetch IRISDATA of IRISSTREAM))
         (SPPOUT (fetch SPPOUTSTREAM of IRISDATA))
         OBJNO
         (FONTDESC (fetch CURRENTFONTDESC of IRISDATA))
         PUSHEDATTRIBUTES SCALE)
        (if (NOT FONTDESC)
```

```
            then                                                    ; this is so that the stream can be opened without the expensive
                                                                    ; font create operation
                 (SETQ FONTDESC (DSPFONT (FONTCREATE 'GACHA 12 NIL NIL 'IRIS)
                                         IRISSTREAM)))
          (if (NEQ (fetch (IRISDATA IRISCHARSET) of IRISDATA)
                   (\CHARSET CHARCODE))
              then (\CHANGECHARSET.IRIS IRISSTREAM (\CHARSET CHARCODE)))
          (SETQ OBJNO (IPLUS (fetch CURRENTFONTBASE of IRISDATA)
                             CHARCODE))
          (COND
             ((EQ CHARCODE (CHARCODE EOL))
              (\TERPRI.IRIS IRISSTREAM))
             ((NILL)                                                ; ZEROP (IRIS.ISOBJ CURRENTFONTBASE SPPOUT)
              ;; this character set has not been installed on the IRIS.  character zero is defined for every charset that is installed.
              (SHOULDNT "\CHANGECHARSET.IRIS has not guaranteed that char 0 is defined.  Obj = " OBJNO))
             ((ZEROP (\FGETWIDTH (fetch (IRISDATA IRISWIDTHSCACHE) of IRISDATA)
                                 (\CHAR8CODE CHARCODE)))            ; the character is not defined.  don't call it
              NIL)
             (T (IRIS.PUSHMATRIX SPPOUT)
                (IRIS.TRANSLATE (DSPXPOSITION NIL IRISSTREAM)
                       (DSPYPOSITION NIL IRISSTREAM)
                       0 SPPOUT)
                (IRIS.SCALE (SETQ SCALE (fetch OTHERDEVICEFONTPROPS of FONTDESC))
                       SCALE SCALE SPPOUT)
                (if (EQ (CAR (fetch FONTFACE of FONTDESC))
                        'BOLD)
                    then (SETQ PUSHEDATTRIBUTES T)
                         (IRIS.PUSHATTRIBUTES SPPOUT)
                         (IRIS.LINEWIDTH \IRIS.BOLD.LINEWIDTH SPPOUT))
                (if (EQ (CADR (fetch FONTFACE of FONTDESC))
                        'ITALIC)
                    then                                            ; fake italics with a rotation
                         (IRIS.ROTATE \IRIS.ITALICS.ROTATION IRIS.ZAXIS SPPOUT))
                (if PUSHEDATTRIBUTES
                    then (IRIS.POPATTRIBUTES SPPOUT))
                (IRIS.CALLOBJ OBJNO SPPOUT)
                (IRIS.POPMATRIX SPPOUT)
                (RELMOVETO (FIX (\FGETWIDTH (fetch IRISWIDTHSCACHE of IRISDATA)
                                            CHARCODE))
                       0 IRISSTREAM)
                (FLUSHOUTPUT (fetch SPPOUTSTREAM of (fetch IRISDATA of IRISSTREAM])
```

(**CLIPPINGREGION.IRIS**

```
[LAMBDA (STREAM REGION)                                             (* gbn "30-Jun-85 21:21")
  (if REGION
      then (replace IRISCLIPPINGREGION of (fetch IRISDATA of STREAM) with REGION)
    else (fetch IRISCLIPPINGREGION of (fetch IRISDATA of STREAM])
```

(**CLOSEFN.IRIS**

```
[LAMBDA (STR)                                                       (* gbn "12-Nov-85 14:25")

        (* * (I DONT THINK THAT SGI IMPLEMENTS THE SPP CLOSE PROTOCOL, BUT WE SHOULD TRY TO CONVINCE
        THEM))

  (FORCEOUTPUT IRISCONN])
```

(**COLOR.IRIS**

```
[LAMBDA (STREAM COLOR)                                              (* gbn " 8-Nov-85 19:25")
  (if COLOR
      then (IRIS.COLOR (\IRIS.ASSURE.COLOR COLOR STREAM)
                  (fetch SPPOUTSTREAM of (fetch IRISDATA of STREAM)))
    else (IRIS.GETCOLOR (fetch SPPOUTSTREAM of (fetch IRISDATA of STREAM])
```

(**IRIS.ASSURE.COLOR**

```
[LAMBDA (COLOR# IRISSTREAM)                                         ; Edited 31-Jan-87 20:32 by gbn
  (PROG (LEVELS)
        (AND (COND
               ((NULL COLOR#)
                NIL)
               ((FIXP COLOR#)
                ;; since Sketch and others call fillpolygon with textures, just return a consistent color from a texture
                (RETURN (IMOD COLOR# 7)))
               [(LITATOM COLOR#)
                (RETURN (COND
                           ((SETQ LEVELS (\LOOKUPCOLORNAME COLOR#))
                                                                    ; recursively look up color number
                            (\IRIS.ASSURE.COLOR (CDR LEVELS)
                                  IRISSTREAM))
                           (T (ERROR "Unknown color name" COLOR#]
               ((HLSP COLOR#)                                       ; HLS form convert to RGB
                (SETQ LEVELS (HLSTORGB COLOR#)))
```

```
                ((RGBP COLOR#)                                  ; check for RGB or HLS
                 (SETQ LEVELS COLOR#))
                ((AND (LISTP COLOR#)
                      (RGBP (CADR COLOR#)))                     ; temporarily, handle the case of being given a texture and a
                                                                ; color, by using the color
                 (RETURN (\IRIS.ASSURE.COLOR (CADR COLOR#)
                               IRISSTREAM)))
                ((TYPENAMEP COLOR# 'BITMAP)                     ; just a hack to not blow up
                 (RETURN (IMOD (for I from 1 to (BITMAPWIDTH COLOR#)) sum (BITMAPBIT COLOR# I 1))
                         8)))
                (T (\ILLEGAL.ARG COLOR#)))
              (RETURN (COND
                        ((\LOOKUPRGB LEVELS (fetch IRISDATA of IRISSTREAM)))
                        (T (ERROR COLOR# "not available in color map"])
```

(\DRAWCIRCLE.IRIS

```
[LAMBDA (IRISSTREAM X Y RADIUS BRUSH DASHING)                   ; Edited 16-Jan-87 15:18 by gbn
  (LET [(SPPOUT (fetch (IRISDATA SPPOUTSTREAM) of (fetch IRISDATA of IRISSTREAM]
       (WITH.IRIS.ATTR (IRIS.CIRC X Y RADIUS SPPOUT)
              SPPOUT IRISSTREAM :COLOR (CADDR BRUSH)
              :WIDTH
              (if (NOT (EQP (CADR BRUSH)
                            1))
                  then (CADR BRUSH)
                else NIL)
              :DASHING DASHING])
```

(\DRAWCURVE.IRIS

```
[LAMBDA (IRISSTREAM KNOTS CLOSED BRUSH DASHING)                 ; Edited 16-Jan-87 16:00 by gbn
```

;;; takes a list of knots. It must build a set of bezier control points for each knot pair.

```
  (LET ((SPPOUT (fetch SPPOUTSTREAM of (fetch IRISDATA of IRISSTREAM)))
        GEOMETRY SPLINE)
       (WITH.IRIS.ATTR [PROGN [SETQ SPLINE (PARAMETRICSPLINE KNOTS CLOSED (fetch SCRATCHSPLINE
                                                                             of (fetch IRISDATA of IRISSTREAM]
                                                                ; convert the list of knots to a parametric spline description.
                              (\SCALE.SPLINE.BY.DERIVS SPLINE)  ; For each knot in the spline, use the knots and the derivatives to
                                                                ; compute bezier control points
                              (for KNOT# from 1 to (SUB1 (fetch %#KNOTS of SPLINE))
                                 do (SETQ GEOMETRY (\PSPLINE.TO.BEZIER.GEOMETRY SPLINE KNOT#))
                                    (SELECTQ \IRIS.VERSION
                                         (GL1 (IRIS.CURVE 10 \BEZIERBASIS.IRIS GEOMETRY SPPOUT))
                                         (GL2 (IRIS.CRV GEOMETRY SPPOUT))
                                         (ERROR "UNKNOWN IRIS VERSION" \IRIS.VERSION]
              SPPOUT IRISSTREAM :COLOR (CADDR BRUSH)
              :WIDTH
              (if (NOT (EQP (CADR BRUSH)
                            1))
                  then (CADR BRUSH)
                else NIL)
              :DASHING DASHING)
       (fetch %#KNOTS of SPLINE])
```

(\DRAWLINE.IRIS

```
[LAMBDA (IRISSTREAM X1 Y1 X2 Y2 WIDTH OPERATION COLOR DASHING) ; Edited 16-Jan-87 15:22 by gbn
```

;;; (check about color and operation) (sets irisx and irisy to x2 and y2 respectively)

```
  (PROG ((IRISDATA (fetch IRISDATA of IRISSTREAM))
         PUSHEDATTRIBUTES SPPOUT)
        (SETQ SPPOUT (fetch (IRISDATA SPPOUTSTREAM) of IRISDATA))
        (WITH.IRIS.ATTR (PROGN (IRIS.MOVE X1 Y1 (fetch IRISZ of IRISDATA)
                                   SPPOUT)
                               (IRIS.DRAW (replace IRISX of IRISDATA with X2)
                                      (replace IRISY of IRISDATA with Y2)
                                      (fetch IRISZ of IRISDATA)
                                      SPPOUT))
              IRISSTREAM SPPOUT :COLOR COLOR :WIDTH WIDTH :DASHING DASHING])
```

(\CONVERTLINESTYLE.IRIS

```
[LAMBDA (DASHING)                                               (* gbn "12-Nov-85 13:54")
```

(* * takes an Interlisp style dashing description (a list of on then off pixels) and turns it into a 16 bit dashing description, like the IRIS likes.)

```
  (bind (RESULT _ 0) for PIX in DASHING as (FLAG _ 1) by (IDIFFERENCE 1 FLAG)
     do [SETQ RESULT (LOGOR (LLSH RESULT PIX)
                            (ITIMES FLAG (SUB1 (EXPT 2 PIX]
     finally (RETURN (LOGAND (SUB1 (EXPT 2 16))
                             RESULT])
```

(\IRISSTREAMINIT

```
  [LAMBDA NIL                                                    ; Edited 31-Jan-87 19:57 by gbn
```

;;; installs the definition of the Iris ImageOps

```
    (DECLARE (GLOBALVARS \IRISIMAGEOPS \FACT.IRIS))
    (SETQ \IRISIMAGEOPS (create IMAGEOPS
                                IMAGETYPE _ 'IRIS
                                IMCLOSEFN _ (FUNCTION \CLOSEFN.IRIS)
                                IMMOVETO _ (FUNCTION \MOVETO.IRIS)
                                IMXPOSITION _ (FUNCTION \XPOSITION.IRIS)
                                IMYPOSITION _ (FUNCTION \YPOSITION.IRIS)
                                IMFONT _ (FUNCTION \FONT.IRIS)
                                IMFONTCREATE _ (FUNCTION IRIS)
                                IMDRAWLINE _ (FUNCTION \DRAWLINE.IRIS)
                                IMLEFTMARGIN _ (FUNCTION \LEFTMARGIN.IRIS)
                                IMRIGHTMARGIN _ (FUNCTION NILL)
                                IMLINEFEED _ (FUNCTION HELP)
                                IMDRAWCURVE _ (FUNCTION \DRAWCURVE.IRIS)
                                IMDRAWCIRCLE _ (FUNCTION \DRAWCIRCLE.IRIS)
                                IMDRAWELLIPSE _ (FUNCTION \DRAWELLIPSE.IRIS)
                                IMFILLCIRCLE _ (FUNCTION \FILLCIRCLE.IRIS)
                                IMSTRINGWIDTH _ (FUNCTION \STRINGWIDTH.IRIS)
                                IMCHARWIDTH _ (FUNCTION \CHARWIDTH.IRIS)
                                IMBLTSHADE _ (FUNCTION \BLTSHADE.IRIS)
                                IMBITBLT _ (FUNCTION \BITBLT.IRIS)
                                IMNEWPAGE _ (FUNCTION NILL)
                                IMSCALE _ (FUNCTION \SCALE.IRIS)
                                IMTERPRI _ (FUNCTION \TERPRI.IRIS)
                                IMTOPMARGIN _ (FUNCTION NILL)
                                IMBOTTOMMARGIN _ (FUNCTION NILL)
                                IMBACKCOLOR _ (FUNCTION \BACKCOLOR.IRIS)
                                IMCOLOR _ (FUNCTION \COLOR.IRIS)
                                IMCLIPPINGREGION _ (FUNCTION \CLIPPINGREGION.IRIS)
                                IMRESET _ (FUNCTION \RESET.IRIS)
                                IMDRAWPOLYGON _ (FUNCTION \DRAWPOLYGON.IRIS)
                                IMFILLPOLYGON _ (FUNCTION \FILLPOLYGON.IRIS)))
    (SETQ \FACT.IRIS (ARRAY 4 0 0.0 0))
    (SETA \FACT.IRIS 0 1.0)
    (SETA \FACT.IRIS 1 1.0)
    (SETA \FACT.IRIS 2 2.0)
    (SETA \FACT.IRIS 3 6.0)
    \IRISIMAGEOPS])
```

(\MOVETO.IRIS

```
  [LAMBDA (IRISSTREAM XPOS YPOS)                                 (* gbn "12-Nov-85 14:36")
    (LET ((IRISDATA (fetch IRISDATA of IRISSTREAM)))
         (IRIS.MOVE (replace IRISX of IRISDATA with XPOS)
                (replace IRISY of IRISDATA with YPOS)
                (fetch IRISZ of IRISDATA)
                (fetch SPPOUTSTREAM of IRISDATA])
```

(\XPOSITION.IRIS

```
  [LAMBDA (IRISSTREAM XPOS)                                      (* gbn "24-Jun-85 01:17")

          (* * adjust only the xpos)

    (PROG ((IRISDATA (fetch IRISDATA of IRISSTREAM)))
          (RETURN (if XPOS
                      then (IRIS.MOVE (replace IRISX of IRISDATA with XPOS)
                                      (fetch IRISY of IRISDATA)
                                      (fetch IRISZ of IRISDATA)
                                      (fetch SPPOUTSTREAM of IRISDATA))
                           XPOS
                      else (OR (fetch IRISX of IRISDATA)
                               (replace IRISX of IRISDATA with (CAR (IRIS.GETGPOS NIL NIL NIL NIL \IRISSTREAM])
```

(\YPOSITION.IRIS

```
  [LAMBDA (IRISSTREAM YPOS)                                      (* gbn "17-Jun-85 15:05")

          (* * adjust only the ypos)

    (PROG ((IRISDATA (fetch IRISDATA of IRISSTREAM)))
          (RETURN (if YPOS
                      then (IRIS.MOVE (fetch IRISX of IRISDATA)
                                      (replace IRISY of IRISDATA with YPOS)
                                      (fetch IRISZ of IRISDATA)
                                      (fetch SPPOUTSTREAM of IRISDATA))
                           YPOS
                      else (fetch IRISY of IRISDATA])
```

(\FILLCIRCLE.IRIS

```
  [LAMBDA (IRISSTREAM CENTERX CENTERY RADIUS TEXTURE)            ; Edited 16-Jan-87 15:24 by gbn
```

```
                                                                (* IRISSTREAM is guaranteed to be an IRIS stream)
    (COND
       ((OR (NOT (NUMBERP RADIUS))
            (ILESSP (SETQ RADIUS (FIXR RADIUS))
                    0))
        (\ILLEGAL.ARG RADIUS))
       (T (LET* ((IRISDATA (fetch IRISDATA of IRISSTREAM))
                 (SPPOUT (fetch (IRISDATA SPPOUTSTREAM) of IRISDATA)))
                (WITH.IRIS.ATTR (IRIS.CIRCF CENTERX CENTERY RADIUS SPPOUT)
                       IRISSTREAM SPPOUT :COLOR TEXTURE])
```

(\DRAWELLIPSE.IRIS

```
  [LAMBDA (IRISSTREAM CENTERX CENTERY SEMIMINORRADIUS SEMIMAJORRADIUS ORIENTATION BRUSH DASHING)
                                                                (* gbn "11-Nov-85 19:07")
    (PROG [(SINOR (COND
                     (ORIENTATION (SIN ORIENTATION))
                     (T 0.0)))
           (COSOR (COND
                     (ORIENTATION (COS ORIENTATION))
                     (T 1.0]
          (\DRAWCURVE.IRIS IRISSTREAM [LIST (CREATEPOSITION (PLUS CENTERX (FTIMES COSOR SEMIMAJORRADIUS))
                                                  (PLUS CENTERY (FTIMES SINOR SEMIMAJORRADIUS)))
                                            (CREATEPOSITION (DIFFERENCE CENTERX (FTIMES SINOR SEMIMINORRADIUS))
                                                  (PLUS CENTERY (FTIMES COSOR SEMIMINORRADIUS)))
                                            (CREATEPOSITION (DIFFERENCE CENTERX (FTIMES COSOR SEMIMAJORRADIUS))
                                                  (DIFFERENCE CENTERY (FTIMES SINOR SEMIMAJORRADIUS)))
                                            (CREATEPOSITION (PLUS CENTERX (FTIMES SINOR SEMIMINORRADIUS))
                                                  (DIFFERENCE CENTERY (FTIMES COSOR SEMIMINORRADIUS]
                 T BRUSH DASHING)
          (MOVETO CENTERX CENTERY IRISSTREAM])
```

(\FILLPOLYGON.IRIS

```
  [LAMBDA (IRISSTREAM POINTS TEXTURE CONVEX?)                   (* gbn "11-Nov-85 19:30")
    (LET ((IRISDATA (fetch IRISDATA of IRISSTREAM))
          COLOR PUSHEDATTRIBUTES SPPOUT)
         (SETQ COLOR (\IRIS.ASSURE.COLOR TEXTURE IRISSTREAM))
         (SETQ SPPOUT (fetch (IRISDATA SPPOUTSTREAM) of IRISDATA))
         (if COLOR
             then                                               (* save the current attributes since this fn is to have no side
                                                                effects)
                  (SETQ PUSHEDATTRIBUTES T)
                  (IRIS.PUSHATTRIBUTES SPPOUT)
                  (IRIS.COLOR COLOR SPPOUT))
         (if (NOT CONVEX?)
             then                                               (* break the polygon up into convex hunks, then fill each of
                                                                those.)
                  (for POLY in (TRAPLOOP POINTS) do (IRIS.POLF2 (LENGTH POLY)
                                                          (for P in POLY collect (LIST (CAR P)
                                                                                       (CDR P)))
                                                          SPPOUT))
           else (IRIS.POLF2 (LENGTH POINTS)
                       (for P in POINTS collect (LIST (CAR P)
                                                      (CDR P)))
                       SPPOUT))
         (if PUSHEDATTRIBUTES
             then (IRIS.POPATTRIBUTES SPPOUT])
```

(\IRIS.BITBLT

```
  [LAMBDA (SOURCE SOURCELEFT SOURCEBOTTOM DESTINATION DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT SOURCETYPE
            OPERATION TEXTURE CLIPPINGREGION)                   (* gbn " 7-Aug-85 23:36")
    (PROG ((COLOR (DSPCOLOR NIL DESTINATION))
           (SPPOUT (fetch SPPOUTSTREAM of (fetch IRISDATA of DESTINATION)))
           NLONGS)
          (for Y from DESTINATIONBOTTOM to (IPLUS DESTINATIONBOTTOM HEIGHT)
             do (IRIS.CMOV2I DESTINATIONLEFT (PLUS DESTINATIONBOTTOM Y)
                       SPPOUT)                                  (* IRIS.WRITEPIXELS WIDTH (for X from SOURCELEFT to
                                                                (IPLUS SOURCELEFT (SUB1 WIDTH)) collect
                                                                (ITIMES (BITMAPBIT SOURCE X Y) COLOR)) DESTINATION)
                                                                (* the current character position determines where a write pixels
                                                                op happens)
                       (PROGN

          (* * now do an inline IRIS.WRITEPIXELS)

                       (IRIS.GCMD 182 SPPOUT)
                       (IRIS.SENDS WIDTH SPPOUT)
                       (SETQ NLONGS (FOLDHI WIDTH 2))
                       (IRIS.SENDL (LLSH NLONGS 1)
                              SPPOUT)                           (* Send the number of bytes to be sent)
                       (bind ALONG for X from SOURCELEFT to (IPLUS SOURCELEFT (SUB1 WIDTH)) by 2
                          do (SETQ ALONG (LOGOR (LLSH (ITIMES (BITMAPBIT SOURCE X Y)
                                                              COLOR)
                                                      16)
                                                (ITIMES (BITMAPBIT SOURCE (ADD1 X)
```

```
                                                        Y)
                                                   COLOR)))
                                (COND
                                   ((IRIS.DOSYNC (IQUOTIENT X 2))
                                    (IRIS.PUTGCHAR IRIS\AESC SPPOUT)))
                                (IRIS.SENDL ALONG SPPOUT])
```

(**\DRAWPOLYGON.IRIS**

```
  [LAMBDA (IRISSTREAM POINTS TEXTURE)                    ; Edited 16-Jan-87 15:33 by gbn
    (PROG ((IRISDATA (fetch IRISDATA of IRISSTREAM))
           COLOR SPPOUT)
          (SETQ SPPOUT (fetch (IRISDATA SPPOUTSTREAM) of IRISDATA))
          (WITH.IRIS.ATTR (IRIS.POLY2 (LENGTH POINTS)
                                      POINTS SPPOUT)
                 IRISSTREAM SPPOUT :COLOR TEXTURE])
```

(**ALIGN**

```
  [LAMBDA (STREAM)                                       (* gbn "17-Jun-85 15:06")

          (* * this is a dummy to insure that the IRIS has caught up on the output side.
          When it returns a value, it has caught up)

    (IRIS.GETCOLOR (OR STREAM (fetch SPPINSTREAM of (fetch IRISDATA of \IRISSTREAM])

)


;;; test functions

(DECLARE%: EVAL@COMPILE

(RECORD BEZIER ((B0X B0Y B0Z)
                (B1X B1Y B1Z)
                (B2X B2Y B2Z)
                (B3X B3Y B3Z))
       B0Z _ 0 B1Z _ 0 B2Z _ 0 B3Z _ 0)

(DATATYPE IRISDATA (IRISX IRISY IRISZ SPPOUTSTREAM SPPINSTREAM SCRATCHSPLINE FONTSINIRIS CURRENTFONTDESC HIFONT#
                          CURRENTFONTBASE BACKCOLOR IRISCLIPPINGREGION OBSOLETE-FIELD IRISCOLORMAPCACHE
                          IRISCHARSET IRISWIDTHSCACHE IRISPAGE)
       FONTSINIRIS _ (LIST NIL)
       IRISX _ 0 IRISY _ 0 IRISZ _ 0 SCRATCHSPLINE _ (create SPLINE)
       HIFONT# _ -255)

[RECORD IRISSTREAM STREAM (SUBRECORD STREAM)
       [ACCESSFNS ((IRISDATA (FETCH (STREAM IMAGEDATA) OF DATUM)
                             (REPLACE (STREAM IMAGEDATA) OF DATUM WITH NEWVALUE]
       (TYPE? (TYPE? IRISDATA OF (FETCH (STREAM IMAGEDATA) OF DATUM]

(RECORD SPLINE (%#KNOTS SPLINEX SPLINEY SPLINEDX SPLINEDY SPLINEDDX SPLINEDDY SPLINEDDDX SPLINEDDDY))
)

(/DECLAREDATATYPE 'IRISDATA
       '(POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
              POINTER POINTER POINTER POINTER)
       ;; ---field descriptor list elided by lister---
       '34)

(DECLARE%: EVAL@COMPILE

(RPAQQ \ALTLINESTYLE.IRIS 1)

(RPAQQ \IRIS.ITALICS.ROTATION -100)

(RPAQQ \PRIMARYLINESTLE.IRIS 0)

(RPAQQ \IRIS.BOLD.LINEWIDTH 2)

(CONSTANTS (\ALTLINESTYLE.IRIS 1)
       (\IRIS.ITALICS.ROTATION -100)
       (\PRIMARYLINESTLE.IRIS 0)
       (\IRIS.BOLD.LINEWIDTH 2))
)

(\IRISSTREAMINIT)

(SETFONTCLASSCOMPONENT DEFAULTFONT 'IRIS '(GACHA 12))

(ADDTOVAR DEFAULTPRINTINGHOST (IRIS Iris))

(ADDTOVAR PRINTERTYPES (IRIS (CANPRINT (IRIS))
                             (BITMAPFILE (IRISBITMAP FILE BITMAP SCALEFACTOR REGION ROTATION TITLE))))

(PUTPROP 'Iris 'PRINTERTYPE 'IRIS)
```

```
(PUTPROPS Iris PRINTERTYPE IRIS)

(DEFMACRO WITH.IRIS.ATTR (FORM SPPOUT IRISSTREAM &KEY (COLOR NIL COLORSET)
                                (WIDTH NIL WIDTHSET)
                                (DASHING NIL DASHINGSET))
   `(LET [PUSHED ., [if COLORSET
                        then `((ECOLOR %, COLOR]
                    .,
                    [if WIDTHSET
                        then `((EWIDTH %, WIDTH]
                    .,
                    (if DASHINGSET
                        then `((EDASHING %, DASHING]
          .,
          [if COLORSET
              then `((SETQ ECOLOR (AND ECOLOR (\IRIS.ASSURE.COLOR ECOLOR IRISSTREAM]
          .,
          [if DASHINGSET
              then `((SETQ EDASHING (AND EDASHING (\CONVERTLINESTYLE.IRIS EDASHING]
          [IF [OR ., (if COLORSET
                         then '(ECOLOR))
                  .,
                  (if WIDTHSET
                      then '(EWIDTH))
                  .,
                  (if DASHINGSET
                      then '(EDASHING]
              THEN (SETQ PUSHED T)
                   (IRIS.PUSHATTRIBUTES %, SPPOUT)
                   .,
                   [IF COLORSET
                       THEN `((IF ECOLOR
                                  THEN (IRIS.COLOR ECOLOR %, SPPOUT]
                   .,
                   [IF WIDTHSET
                       THEN `((IF EWIDTH
                                  THEN (SELECTQ \IRIS.VERSION
                                                (GL2 (IRIS.LINEWIDTH EWIDTH %, SPPOUT))
                                                (GL1 (if (IGREATERP EWIDTH 2)
                                                         then NIL
                                                        else (IRIS.LINEWIDTH 2 %, SPPOUT)))
                                                (ERROR "UNKNOWN VERSION" \IRIS.VERSION]
                   .,
                   (IF DASHINGSET
                       THEN `((IF EDASHING
                                  THEN (IRIS.DEFLINESTYLE \ALTLINESTYLE.IRIS EDASHING %, SPPOUT)
                                       (IRIS.SETLINESTYLE \ALTLINESTYLE.IRIS SPPOUT)
                                       (IRIS.RESETLS 0 SPPOUT]
          %, FORM (IF PUSHED
                      THEN (IRIS.POPATTRIBUTES %, SPPOUT))))
```

FUNCTION INDEX

VARIABLE INDEX

CONSTANT INDEX

RECORD INDEX

MACRO INDEX

PROPERTY INDEX