

File created: 5-Dec-2023 00:44:59 {WMEDLEY}<library>sketch>SKETCH-STREAM.;1

edit by: rmk

changes to: (FNS OPENSKECHSTREAM)

previous date: 12-Jun-90 15:21:13 {WMEDLEY}<library>sketch>SKETCHSTREAM.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

## (RPAQQ SKETCH-STREAMCOMS

;; contains the functions needed to support sketch streams. Sketch streams allow a user program to print, draw, etc. to a stream and builds a sketch of the result.

```
(FNS OPENSKECHSTREAM \SKETCHSTREAM.POSITION.CHANGED \SKETCHSTREAMINIT \SK.SET.FONT
\SKSTRM.WINDOW.FROM.STREAM ZOOM.SKETCH.STREAM)
```

; fns to support stream operations on sketches

```
(FNS \DSPFONT.SKETCH \DSPLEFTMARGIN.SKETCH \DSPRIGHTMARGIN.SKETCH \DSPLINEFEED.SKETCH
\DSPXPOSITION.SKETCH \DSPYPOSITION.SKETCH \DRAWCURVE.SKETCH \DRAWCIRCLE.SKETCH \FILLCIRCLE.SKETCH
\FILLPOLYGON.SKETCH \DRAWELLIPSE.SKETCH \DRAWARC.SKETCH \DRAWLINE.SKETCH \BOUT.SKETCH
\DSPCOLOR.SKETCH \DSPBACKCOLOR.SKETCH \DSPOPERATION.SKETCH \STRINGWIDTH.SKETCH \BLTSHADE.1BITSKETCH
\NEWPAGE.SKETCH \CHARWIDTH.SKETCH \BITBLT.1BITSKETCH \DSPCLIPPINGREGION.SKETCH \DSPRESET.SKETCH
\DSPSCALE.SKETCH \DRAWPOLYGON.SKETCH)
```

```
(ALISTS (IMAGESTREAMTYPES SKETCH))
```

```
(GLOBALVARS SketchFDEV)
```

```
(DECLARE%: DONTEVAL@LOAD DOCOPY (P (\SKETCHSTREAMINIT))
```

;; contains the functions needed to support sketch streams. Sketch streams allow a user program to print, draw, etc. to a stream and builds a sketch of the result.

(DEFINEQ

## (OPENSKECHSTREAM

```
[LAMBDA (TITLE OPTIONS)
```

(\* rrb "20-Dec-84 12:12")

(\* opens a stream onto a window that will keep a sketch of what is displayed there.)

(\* changes default alignment to left baseline and default font to the default font of display device.)

```
(PROG ((SKW (SKETCHW.CREATE NIL (LISTGET OPTIONS 'SKETCHREGION)
(LISTGET OPTIONS 'REGION)
TITLE)))
```

(\* changes default alignment to left baseline and default font to the default font of display device.)

```
(SK.SET.TEXT.HORIZ.ALIGN SKW 'LEFT)
[SK.SET.FONT SKW (FONTNAMELIST (DEFAULTFONT 'DISPLAY)
(RETURN (create STREAM
DEVICE _ SketchFDEV
ACCESS _ 'OUTPUT
USERCLOSEABLE _ NIL
OUTCHARFN _ (FUNCTION \BOUT.SKETCH)
IMAGEOPS _ \SKETCHIMAGEOPS
IMAGEDATA _ NIL
F1 _ SKW))
```

## (\SKETCHSTREAM.POSITION.CHANGED

```
[LAMBDA (SKW)
```

```
(RESET.LINE.BEING.INPUT SKW)
```

```
(SKED.CLEAR.SELECTION SKW)]
```

(\* called whenever the position of a sketch stream changes.)

## (\SKETCHSTREAMINIT

```
[LAMBDA NIL
```

(\* rrb " 4-Oct-85 17:35")

(\* Initializes global variables for the Sketch device)

(\* Sketch Streams are referred to only by themselves so they do not need directory operations. Most of the fields in the DisplayDevice are empty to avoid something bad happening.)

```
(DECLARE (GLOBALVARS SketchFDEV \SKETCHIMAGEOPS))
```

```
(SETQ \SKETCHIMAGEOPS (create IMAGEOPS
```

```
IMAGETYPE _ 'SKETCH
```

```
IMFONT _ (FUNCTION \DSPFONT.SKETCH)
```

```
IMLEFTMARGIN _ (FUNCTION \DSPLEFTMARGIN.SKETCH)
```

```
IMRIGHTMARGIN _ (FUNCTION \DSPRIGHTMARGIN.SKETCH)
```

```
IMLINEFEED _ (FUNCTION \DSPLINEFEED.SKETCH)
```

```
IMXPOSITION _ (FUNCTION \DSPXPOSITION.SKETCH)
```

```
IMYPOSITION _ (FUNCTION \DSPYPOSITION.SKETCH)
```

```
IMCLOSEFN _ (FUNCTION NIL)
```

```
IMDRAWCURVE _ (FUNCTION \DRAWCURVE.SKETCH)
```

```
IMFILLCIRCLE _ (FUNCTION \FILLCIRCLE.SKETCH)
```

```
IMFILLPOLYGON _ (FUNCTION \FILLPOLYGON.SKETCH)
```

```

IMDRAWPOLYGON _ (FUNCTION \DRAWPOLYGON.SKETCH)
IMDRAWLINE _ (FUNCTION \DRAWLINE.SKETCH)
IMDRAWELLIPSE _ (FUNCTION \DRAWELLIPSE.SKETCH)
IMDRAWCIRCLE _ (FUNCTION \DRAWCIRCLE.SKETCH)
IMBITBLT _ (FUNCTION \BITBLT.1BITSKETCH)
IMBLTSHADE _ (FUNCTION \BLTSHADE.1BITSKETCH)
IMNEWPAGE _ (FUNCTION \NEWPAGE.SKETCH)
IMSCALE _ (FUNCTION \DSPSCALE.SKETCH)
IMSPACEFACTOR _ (FUNCTION \DSPSPACEFACTOR.DISPLAY)
IMFONTCREATE _ 'DISPLAY
IMCOLOR _ (FUNCTION \DPCOLOR.SKETCH)
IMBACKCOLOR _ (FUNCTION \DSPBACKCOLOR.SKETCH)
IMOPERATION _ (FUNCTION \DSPOPERATION.SKETCH)
IMSTRINGWIDTH _ (FUNCTION \STRINGWIDTH.SKETCH)
IMCHARWIDTH _ (FUNCTION \CHARWIDTH.SKETCH)
IMCLIPPINGREGION _ (FUNCTION \DSPCLIPPINGREGION.SKETCH)
IMRESET _ (FUNCTION \DSPRESET.SKETCH)
IMDRAWARC _ (FUNCTION \DRAWARC.SKETCH))

```

```

(SETQ SketchFDEV (create FDEV
  DEVICENAME _ 'SKETCH
  RESETABLE _ NIL
  RANDOMACCESSP _ NIL
  PAGEMAPPED _ NIL
  CLOSEFILE _ (FUNCTION NIL)
  DELETEFILE _ (FUNCTION NIL)
  GETFILEINFO _ (FUNCTION NIL)
  OPENFILE _ [FUNCTION (LAMBDA (NAME ACCESS RECOG OTHERINFO FDEV)
    NAME)]
  READPAGES _ (FUNCTION \ILLEGAL.DEVICEOP)
  SETFILEINFO _ (FUNCTION NIL)
  GENERATEFILES _ (FUNCTION \GENERATENOFILES)
  TRUNCATEFILE _ (FUNCTION NIL)
  WRITEPAGES _ (FUNCTION \ILLEGAL.DEVICEOP)
  GETFILENAME _ [FUNCTION (LAMBDA (NAME RECOG FDEV)
    NAME)]
  REOPENFILE _ [FUNCTION (LAMBDA (NAME)
    NAME)]
  EVENTFN _ (FUNCTION NIL)
  DIRECTORYNAMEP _ (FUNCTION NIL)
  HOSTNAMEP _ (FUNCTION NIL)
  BIN _ (FUNCTION \ILLEGAL.DEVICEOP)
  BOUT _ (FUNCTION \BOUT.SKETCH)
  PEEKBIN _ (FUNCTION \ILLEGAL.DEVICEOP)
  BACKFILEPTR _ (FUNCTION \ILLEGAL.DEVICEOP)
  BLOCKIN _ (FUNCTION \ILLEGAL.DEVICEOP)
  BLOCKOUT _ (FUNCTION \ILLEGAL.DEVICEOP)))
(\DEFINEDEVICE NIL SketchFDEV])

```

(\SK.SET.FONT

[LAMBDA (FONTDESC SKW)

(\* rrb "12-Dec-84 08:48")

(\* sets the default font from a font descriptor.)

(replace (SKETCHCONTEXT SKETCHFONT) of (WINDOWPROP SKW 'SKETCHCONTEXT) with (FONTNAMELIST FONTDESC])

(\SKSTRM.WINDOW.FROM.STREAM

[LAMBDA (SKSTRM)

(\* rrb "12-Dec-84 08:53")

(\* returns the window that is associated with a sketch stream.)

(fetch (STREAM F1) of SKSTRM])

(ZOOM.SKETCH.STREAM

[LAMBDA (REGION SKSTREAM)

; Edited 9-Jan-87 16:13 by rrb

(\* changes the part of the sketch seen in a sketch window.)

(PROG1 (SKETCH.REGION.VIEWED (\SKSTRM.WINDOW.FROM.STREAM SKSTREAM))

[AND REGION (COND

((REGIONP REGION)

(\* move the sketch region to be the new clipping region.)

(SKETCH.GLOBAL.REGION.ZOOM (\SKSTRM.WINDOW.FROM.STREAM SKSTREAM)

REGION))

(T (\ILLEGAL.ARG REGION]))]

)

:: fns to support stream operations on sketches

(DEFINEQ

(\DSPFONT.SKETCH

[LAMBDA (SKETCHSTREAM FONT)

(\* rrb "2-Aug-85 10:12")

(\* sets the font that a display stream uses to print characters. SKETCHSTREAM is guaranteed to be a stream of type sketch)

(PROG ((SKETCHWINDOW (\SKSTRM.WINDOW.FROM.STREAM SKETCHSTREAM))

RESULT)

(SETQ RESULT (DSPFONT FONT SKETCHWINDOW))

(\* if the font was changed, update the current font.)

(COND

(FONT (\SKETCHSTREAM.POSITION.CHANGED SKETCHWINDOW)

```
(SK.SET.FONT (DSPFONT NIL SKETCHWINDOW)
             SKETCHWINDOW))
(RETURN RESULT)]
```

(\DSPLEFTMARGIN.SKETCH

```
[LAMBDA (SKSTRM LEFTMARGIN)
```

(\* rrb "21-Dec-84 08:55")  
 (\* version which passed the operation through without doing anything.)

```
(DSPLEFTMARGIN LEFTMARGIN (\SKSTRM.WINDOW.FROM.STREAM SKSTRM])
```

(\DSPRIGHTMARGIN.SKETCH

```
[LAMBDA (SKSTRM RIGHTMARGIN)
```

(\* rrb "21-Dec-84 08:55")  
 (\* version which passed the operation through without doing anything.)

```
(DSPRIGHTMARGIN RIGHTMARGIN (\SKSTRM.WINDOW.FROM.STREAM SKSTRM])
```

(\DSPLINEFEED.SKETCH

```
[LAMBDA (SKSTRM LINEFEED)
```

(\* rrb "21-Dec-84 08:55")  
 (\* version which passed the operation through without doing anything.)

```
(DSPLINEFEED LINEFEED (\SKSTRM.WINDOW.FROM.STREAM SKSTRM])
```

(\DSPXPOSITION.SKETCH

```
[LAMBDA (SKSTRM XPOSITION)
```

(\* rrb " 2-Aug-85 09:26")  
 (\* version which passed the operation through without doing anything.)

```
(PROG ((SKW (\SKSTRM.WINDOW.FROM.STREAM SKSTRM))
       RESULT)
      (SETQ RESULT (DSPXPOSITION XPOSITION SKW))
      (AND XPOSITION (\SKETCHSTREAM.POSITION.CHANGED SKW))
      (RETURN RESULT])
```

(\DSPYPOSITION.SKETCH

```
[LAMBDA (SKSTRM YPOSITION)
```

(\* rrb " 2-Aug-85 09:25")  
 (\* version which passed the operation through without doing anything.)

```
(PROG ((SKW (\SKSTRM.WINDOW.FROM.STREAM SKSTRM))
       RESULT)
      (SETQ RESULT (DSPYPOSITION YPOSITION SKW))
      (AND YPOSITION (\SKETCHSTREAM.POSITION.CHANGED SKW))
      (RETURN RESULT])
```

(\DRAWCURVE.SKETCH

```
[LAMBDA (SKSTRM KNOTS CLOSED BRUSH DASHING)
```

(\* rrb "30-Oct-85 14:25")  
 (\* draws a spline curve with a given brush.)

```
(PROG ((SKW (\SKSTRM.WINDOW.FROM.STREAM SKSTRM))
       (\SKETCHSTREAM.POSITION.CHANGED SKW))
      (RETURN (SK.ADD.ELEMENT (SK.CURVE.CREATE KNOTS CLOSED BRUSH DASHING (SK.INPUT.SCALE SKW))
                              SKW]))
```

(\DRAWCIRCLE.SKETCH

```
[LAMBDA (SKSTRM CENTERX CENTERY RADIUS BRUSH DASHING)
```

(\* rrb "30-Oct-85 14:25")  
 (\* draws a circle on a sketch stream)

```
(PROG ((SKW (\SKSTRM.WINDOW.FROM.STREAM SKSTRM))
       SKCONTEXT)
      (\SKETCHSTREAM.POSITION.CHANGED SKW)
      (SETQ SKCONTEXT (WINDOWPROP SKW 'SKETCHCONTEXT))
      (RETURN (SK.ADD.ELEMENT (SK.CIRCLE.CREATE (create POSITION
                                                    XCOORD _ CENTERX
                                                    YCOORD _ CENTERY)
                                              (create POSITION
                                                    XCOORD _ (PLUS CENTERX RADIUS)
                                                    YCOORD _ CENTERY)
                                              BRUSH DASHING (SK.INPUT.SCALE SKW)
                                              (fetch (SKETCHCONTEXT SKETCHFILLING) of SKCONTEXT))
                              SKW]))
```

(\FILLCIRCLE.SKETCH

```
[LAMBDA (SKSTRM CENTERX CENTERY RADIUS TEXTURE)
```

(\* rrb "27-Sep-85 09:25")  
 (\* implements fill circle on a sketch stream.)  
 (\* put the radius point on a horizontal line.)

```
(PROG ((SKW (\SKSTRM.WINDOW.FROM.STREAM SKSTRM))
       (\SKETCHSTREAM.POSITION.CHANGED SKW))
      (RETURN (SK.ADD.ELEMENT (SK.CIRCLE.CREATE (create POSITION
                                                    XCOORD _ CENTERX
                                                    YCOORD _ CENTERY)
                                              (create POSITION
                                                    XCOORD _ (PLUS CENTERX RADIUS)
                                                    YCOORD _ CENTERY)
                                              (create BRUSH
                                                    TEXTURE))
                              SKW]))
```

```

BRUSHSIZE _ 0)
(fetch (SKETCHCONTEXT SKETCHDASHING) of (WINDOWPROP SKW 'SKETCHCONTEXT))
(SK.INPUT.SCALE SKW)
(SK.INSURE.FILLING TEXTURE))
SKW])

```

(\FILLPOLYGON.SKETCH

```

[LAMBDA (SKSTRM KNOTS TEXTURE)
(* rrb "26-Sep-85 18:04")
(* implements fill polygon on a sketch stream.)
(PROG ((SKW (\SKSTRM.WINDOW.FROM.STREAM SKSTRM)))
(\SKETCHSTREAM.POSITION.CHANGED SKW)
(* add a closed wire element with a filling.)
(RETURN (SK.ADD.ELEMENT (SK.WIRE.CREATE KNOTS (create BRUSH
BRUSHSIZE _ 0)
(fetch (SKETCHCONTEXT SKETCHDASHING) of (WINDOWPROP SKW 'SKETCHCONTEXT))
T
(SK.INPUT.SCALE SKW)
NIL
(SK.INSURE.FILLING TEXTURE SKW))
SKW])

```

(\DRAWELLIPSE.SKETCH

```

[LAMBDA (SKSTRM CENTERX CENTERY SEMIMINORRADIUS SEMIMAJORRADIUS ORIENTATION BRUSH DASHING)
(* rrb "30-Oct-85 14:25")
(* draws an ellipse on a sketch stream)
(* put the radius point on a horizontal line.)
(PROG ((SKW (\SKSTRM.WINDOW.FROM.STREAM SKSTRM)))
(\SKETCHSTREAM.POSITION.CHANGED SKW)
(* have the major radius be the point on the circle, the minor one be perpendicular to it.)
(RETURN (SK.ADD.ELEMENT (ELLIPSE.CREATE (create POSITION
XCOORD _ CENTERX
YCOORD _ CENTERY)
[create POSITION
XCOORD _ [PLUS CENTERX (TIMES SEMIMINORRADIUS
(COS (PLUS ORIENTATION 90)
YCOORD _ (PLUS CENTERY (TIMES SEMIMINORRADIUS
(SIN (PLUS ORIENTATION 90)
[create POSITION
XCOORD _ (PLUS CENTERX (TIMES SEMIMAJORRADIUS (COS ORIENTATION))
)
YCOORD _ (PLUS CENTERY (TIMES SEMIMAJORRADIUS (SIN ORIENTATION)
BRUSH DASHING (SK.INPUT.SCALE SKW))
SKW])

```

(\DRAWARC.SKETCH

```

[LAMBDA (SKSTRM CENTERX CENTERY RADIUS STARTANGLE NDEGREES BRUSH DASHING)
(* rrb "30-Oct-85 14:26")
(* draws an ellipse on a sketch stream)
(* put the radius point on a horizontal line.)
(PROG ((SKW (\SKSTRM.WINDOW.FROM.STREAM SKSTRM)))
(\SKETCHSTREAM.POSITION.CHANGED SKW)
(* have the major radius be the point on the circle, the minor one be perpendicular to it.)
(RETURN (SK.ADD.ELEMENT (ARC.CREATE (create POSITION
XCOORD _ CENTERX
YCOORD _ CENTERY)
[create POSITION
XCOORD _ (PLUS CENTERX (TIMES RADIUS (COS STARTANGLE)))
YCOORD _ (PLUS CENTERY (TIMES RADIUS (SIN STARTANGLE]
[create POSITION
XCOORD _ [PLUS CENTERX (TIMES RADIUS (COS (PLUS STARTANGLE
NDEGREES]
YCOORD _ (PLUS CENTERY (TIMES RADIUS (SIN (PLUS STARTANGLE
NDEGREES]
BRUSH DASHING (SK.INPUT.SCALE SKW)
NIL
(LESSP NDEGREES 0))
SKW])

```

(\DRAWLINE.SKETCH

```

[LAMBDA (SKETCHSTREAM X1 Y1 X2 Y2 WIDTH OPERATION COLOR DASHING)
(* rrb " 4-Sep-85 16:34")
(* draws a line on a sketch stream)
(PROG ((SKW (\SKSTRM.WINDOW.FROM.STREAM SKETCHSTREAM)))
(\SKETCHSTREAM.POSITION.CHANGED SKW)
(RETURN (SK.ADD.ELEMENT (WIRE.INPUTFN SKW (LIST (create POSITION
XCOORD _ X1
YCOORD _ Y1)
(create POSITION
XCOORD _ X2
YCOORD _ Y2))
NIL
(OR WIDTH 1)

```

(SK.INPUT.SCALE SKW)
DASHING)
SKW])

(\BOUT.SKETCH

[LAMBDA (SKETCHSTREAM CHARCODE) (\* rrb " 4-Sep-85 16:34")
(\* bout function for the device that makes a sketch)

(\* It would be faster to keep the characters until a CR or reset line is done but it unclear what happens if the last operation
is printing.)

(PROG ((SKW (\SKSTRM.WINDOW.FROM.STREAM SKETCHSTREAM)))
(COND
(NULL (WINDOWPROP SKW 'SELECTION))
(SKED.SET.SELECTION (create POSITION
XCOORD \_ (DSPXPOSITION NIL SKW)
YCOORD \_ (DSPYPOSITION NIL SKW)
SKW)))
(SKED.INSERT (LIST CHARCODE)
SKW
(SK.INPUT.SCALE SKW))
(RETURN CHARCODE]))

(\DSPCOLOR.SKETCH

[LAMBDA (SKSTRM COLOR) (\* rrb "20-Dec-84 10:53")
(\* sketch stream function for changing the color.)
(DSPCOLOR COLOR (\SKSTRM.WINDOW.FROM.STREAM SKSTRM])

(\DSPBACKCOLOR.SKETCH

[LAMBDA (SKSTRM COLOR) (\* rrb "20-Dec-84 10:52")
(\* sketch stream function for changing the background color.)
(DSPBACKCOLOR COLOR (\SKSTRM.WINDOW.FROM.STREAM SKSTRM])

(\DSOPERATION.SKETCH

[LAMBDA (SKSTRM OPERATION) (\* rrb "20-Dec-84 10:53")
(\* sketch stream function for changing the operation.)
(DSOPERATION OPERATION (\SKSTRM.WINDOW.FROM.STREAM SKSTRM])

(\STRINGWIDTH.SKETCH

[LAMBDA (SKSTRM STR RDTBL) (\* rrb "21-Dec-84 08:56")
(\* computes the string width for a sketch stream.)
(\* calls the display stream function directly and probably
shouldn't.)
(\STRINGWIDTH.DISPLAY (WINDOWPROP (\SKSTRM.WINDOW.FROM.STREAM SKSTRM)
'DSP)
STR RDTBL))

(\BLTSHADE.1BITSKETCH

[LAMBDA (TEXTURE SKETCHSTREAM DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT OPERATION CLIPPINGREGION)
(\* rrb " 4-Sep-85 16:35")
(\* implements blt shade for a sketch stream.)
(PROG ((SKW (\SKSTRM.WINDOW.FROM.STREAM SKETCHSTREAM)))
(RETURN (SK.ADD.ELEMENT (SK.BOX.CREATE (CREATEREGION DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT)
(create BRUSH
BRUSHSIZE \_ 0)
NIL
(SK.INPUT.SCALE SKW)
TEXTURE)
SKW]))

(\NEWPAGE.SKETCH

[LAMBDA (SKSTRM) (\* rrb " 1-Aug-85 11:59")
(\* NEWPAGE function for sketch streams.)

(\* should probably save the current sketch before resetting it and if DSPRESET ever resets defaults this shouldn't.)

(\DSPRESET.SKETCH SKSTRM])

(\CHARWIDTH.SKETCH

[LAMBDA (SKSTRM CHARCODE) (\* rrb "21-Dec-84 08:54")
(\* computes the character width for a sketch stream.)
(\* calls the display stream function directly and probably
shouldn't.)

(\CHARWIDTH.DISPLAY (WINDOWPROP (\SKSTRM.WINDOW.FROM.STREAM SKSTRM)
'DSP)
CHARCODE])

(\BITBLT.1BITSKETCH

```
[LAMBDA (SOURCEBITMAP SOURCELEFT SOURCEBOTTOM DESTSTRM DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT
SOURCECTYPE OPERATION TEXTURE CLIPPINGREGION CLIPPEDSOURCELEFT CLIPPEDSOURCEBOTTOM)
; Edited 9-Jan-87 16:03 by rrb
(* handles bitblt to a sketch stream.
Does it by creating a bitmap imageobject.)

(COND
((BITMAPP SOURCEBITMAP) (* only handles simple cases.)
(PROG ((SKW (\SKSTRM.WINDOW.FROM.STREAM DESTSTRM))
(BMWIDTH (OR WIDTH (BITMAPWIDTH SOURCEBITMAP)))
(BMHEIGHT (OR HEIGHT (BITMAPHEIGHT SOURCEBITMAP)))
BM)
(SETQ BM (BITMAPCREATE BMWIDTH BMHEIGHT))
(BITBLT SOURCEBITMAP SOURCELEFT SOURCEBOTTOM BM 0 0 WIDTH HEIGHT NIL 'REPLACE NIL)
(SK.ADD.ELEMENT (SK.ELEMENT.FROM.IMAGEOBJ (BITMAPEDITOBJ BM 1 0)
SKW
(create POSITION
XCOORD _ DESTINATIONLEFT
YCOORD _ DESTINATIONBOTTOM)
SKW)))
(T (BITBLT SOURCEBITMAP SOURCELEFT SOURCEBOTTOM (WINDOWPROP (\SKSTRM.WINDOW.FROM.STREAM DESTSTRM)
'DSP)
DESTINATIONLEFT DESTINATIONBOTTOM WIDTH HEIGHT SOURCECTYPE OPERATION TEXTURE CLIPPINGREGION])
```

(\DSPCLIPPINGREGION.SKETCH

[LAMBDA (SKSTRM REGION) ; Edited 17-Aug-88 12:18 by jds

```
:: sets the clipping region in a sketch stream.
:: (DSPCLIPPINGREGION REGION (\SKSTRM.WINDOW.FROM.STREAM SKSTRM))
```

;;; JDS: Changed this to be a NO-OP, but to return the existing clipping region for the underlying windo. Changing the clipping region for the window  
;;; KILLS the screen.

```
(DSPCLIPPINGREGION NIL (\SKSTRM.WINDOW.FROM.STREAM SKSTRM))
```

(\DSPRESET.SKETCH

[LAMBDA (SKSTRM) (\* rrb "9-Jul-85 12:42")  
(\* reset the properties of a sketch stream.)

```
(PROG ((W (\SKSTRM.WINDOW.FROM.STREAM SKSTRM))
SKETCH OLDSKETCH)
(SKED.CLEAR.SELECTION W)
[WINDOWPROP W 'SKETCH (SETQ SKETCH (COND
((SETQ OLDSKETCH (WINDOWPROP W 'SKETCH))
(* copy properties and defaults from old sketch.)
(create SKETCH using OLDSKETCH SKETCHELTS _ NIL))
(T (SKETCH.CREATE NIL)
(* for now, don't reset the defaults other than position.))

(DSPRESET W)
(\DSPXPOSITION.SKETCH SKSTRM (DSPXPOSITION NIL W))
(\DSPYPOSITION.SKETCH SKSTRM (DSPYPOSITION NIL W))
(WINDOWPROP W 'SCALE INITIAL.SCALE)
(SK.UPDATE.REGION.VIEWED W)
(MAP.SKETCHSPEC.INTO.VIEWER SKETCH W)
(SK.CREATE.HOTSPOT.CACHE W)
(WINDOWPROP W 'GRIDFACTOR (SK.DEFAULT.GRIDFACTOR W))
(WINDOWPROP W 'USEGRID NIL)
(WINDOWPROP W 'SKETCHCHANGED NIL]))
```

(\DSPSCALE.SKETCH

[LAMBDA (SKSTRM SCALE) ; Edited 9-Jan-87 16:00 by rrb  
(\* returns the scale of a sketch stream.)

```
(PROG ((SKWINDOW (\SKSTRM.WINDOW.FROM.STREAM SKSTRM))
OLDSCALE)
(RETURN (PROG1 (SETQ OLDSCALE (VIEWER.SCALE SKWINDOW))
[AND SCALE (COND
[(NUMBERP SCALE) (* zoom the current sketch view around the center.)
(* don't redraw if scale is the same.)
(OR (EQP OLDSCALE SCALE)
(PROG [NEWWIDTH NEWHEIGHT (CENTERPT (REGION.CENTER (
SKETCH.REGION.VIEWED
SKWINDOW)
[SETQ NEWWIDTH (TIMES SCALE (WINDOWPROP SKWINDOW 'WIDTH)
[SETQ NEWHEIGHT (TIMES SCALE (WINDOWPROP SKWINDOW 'HEIGHT)
(SKETCH.GLOBAL.REGION.ZOOM SKWINDOW
(CREATEREGION (DIFFERENCE (fetch (POSITION XCOORD)
of CENTERPT)
(QUOTIENT NEWWIDTH 2))
(DIFFERENCE (fetch (POSITION YCOORD) of CENTERPT)
(QUOTIENT NEWHEIGHT 2))
NEWWIDTH NEWHEIGHT]
(T (\ILLEGAL.ARG SCALE]))))
```

(\DRAWPOLYGON.SKETCH

[LAMBDA (SKETCHSTREAM POINTS CLOSED BRUSH DASHING) (\* rrb "26-Sep-85 18:07")

(\* draws a polygon on a sketch stream)

```
(PROG ((SKW (\SKSTRM.WINDOW.FROM.STREAM SKETCHSTREAM))
(\SKETCHSTREAM.POSITION.CHANGED SKW)
(RETURN (SK.ADD.ELEMENT (SK.WIRE.CREATE POINTS BRUSH DASHING T (SK.INPUT.SCALE SKW)
NIL NIL)
SKW]))
```

)

```
(ADDTOVAR IMAGESTREAMTYPES (SKETCH (OPENSTREAM OPENSKECHSTREAM)
(FONTCREATE \CREATEDISPLAYFONT)))
```

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

```
(GLOBALVARS SketchFDEV)
)
```

```
(DECLARE%: DONTEVAL@LOAD DOCOPY
```

```
(\SKETCHSTREAMINIT)
)
```

---

**FUNCTION INDEX**

OPENSKECHSTREAM .....	1	\DRAWPOLYGON.SKETCH .....	6	\DSPXPOSITION.SKETCH .....	3
ZOOM.SKETCH.STREAM .....	2	\DSPBACKCOLOR.SKETCH .....	5	\DSPYPOSITION.SKETCH .....	3
\BITBLT.1BITSKETCH .....	5	\DSPCLIPPINGREGION.SKETCH .....	6	\FILLCIRCLE.SKETCH .....	3
\BLTSHADE.1BITSKETCH .....	5	\DSPCOLOR.SKETCH .....	5	\FILLPOLYGON.SKETCH .....	4
\BOUT.SKETCH .....	5	\DSPFONT.SKETCH .....	2	\NEWPAGE.SKETCH .....	5
\CHARWIDTH.SKETCH .....	5	\DSPLEFTMARGIN.SKETCH .....	3	\SK.SET.FONT .....	2
\DRAWARC.SKETCH .....	4	\DSPLINEFEED.SKETCH .....	3	\SKETCHSTREAM.POSITION.CHANGED ..	1
\DRAWCIRCLE.SKETCH .....	3	\DSPOPERATION.SKETCH .....	5	\SKETCHSTREAMINIT .....	1
\DRAWCURVE.SKETCH .....	3	\DSPRESET.SKETCH .....	6	\SKSTRM.WINDOW.FROM.STREAM .....	2
\DRAWELLIPSE.SKETCH .....	4	\DSPRIGHTMARGIN.SKETCH .....	3	\STRINGWIDTH.SKETCH .....	5
\DRAWLINE.SKETCH .....	4	\DSPSCALE.SKETCH .....	6		

---

**VARIABLE INDEX**

IMAGESTREAMTYPES .....	7
------------------------	---

---