

File created: 5-Dec-2023 00:15:05 {WMEDLEY}<library>sketch>SKETCH-OBJ.;1

edit by: rmk

changes to: (RECORDS SKETCHIMAGEOBJ SKETCHDOCUMENTINFO)

previous date: 15-Jul-2022 14:33:25 {WMEDLEY}<library>sketch>SKETCHOBJ.;2

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

## (RPAQQ SKETCH-OBJCOMS

```
[[COMS (* the stuff to support sketch images in documents.)
(FNS MAKE.IMAGE.OBJECT.OF.SKETCH SK.ELEMENT.FROM.IMAGEOBJ SKETCHIMAGEOBJ.FROM.VIEWER
  SKETCH.IMAGEOBJ SKETCH.DISPLAYFN SKETCH.BITMAP.IMAGE SKIO.IMAGEBOXFN SKIO.GETFN.2
  SKIO.UPDATE.FROM.OLD.FORM SKIO.GETFN SKIO.PUTFN SKIO.COPYFN SKIO.BUTONEVENTINFN
  TRANSLATE.REGION UPDATE.IMAGE.IN.DOCUMENT SK.COPY.IMAGEOBJ COPY.IMAGE.OBJECT
  \CREATE.SKETCH.IMAGEFNS \SKIO.IN.TOO.SMALL.TEDITP SKETCH.VIEWER.GRID SKETCH.VIEWER.SCALE)
(DECLARE%: DONTCOPY DOEVAL@COMPILE (RECORDS SKETCHIMAGEOBJ SKETCHDOCUMENTINFO))
(P (\CREATE.SKETCH.IMAGEFNS))
(ADDVARS (IMAGEOBJGETFNS (SKIO.GETFN))
(COMS (* stuff to support image objects as elements in a sketch)
(FNS SKETCH.IMAGE.OBJECT.ELEMENT SKETCH.IMAGEOBJ.OF.ELEMENT SKETCH.SCALE.OF.ELEMENT
  SKETCH.POSITION.OF.ELEMENT CREATE.SKIMAGEOBJ.TYPE IMAGEBOXSIZE
  SK.UPDATE.IMAGEOBJECT.AFTER.CHANGE SKETCH.CREATE.IMAGE.OBJECT SKETCH.CREATE.IMAGE.OBJECT1)
(FNS SK.IMAGEOBJ.DRAWFN SK.IMAGEOBJ.REGIONFN SK.IMAGEOBJ.GLOBALREGIONFN SK.IMAGEOBJ.TRANSLATEFN
  SK.IMAGEOBJ.EXPANDFN SK.IMAGEOBJ.INSIDEFN SK.IMAGEOBJ.MOVEFN SK.IMAGEOBJ.CHANGEFN
  SK.IMAGEOBJ.READCHANGEFN SK.IMAGEOBJ.TRANSFORMFN)
(RECORDS LOCALSKIMAGEOBJ SKIMAGEOBJ ANNO)
(P (CREATE.SKIMAGEOBJ.TYPE]))
```

(\* the stuff to support sketch images in documents.)

(DEFINEQ

## (MAKE.IMAGE.OBJECT.OF.SKETCH

[LAMBDA (SKETCH REGION SCALE GRIDSIZE)

; Edited 18-Nov-87 17:50 by rrb

(\* Returns a sketch image object. REGION is the region in sketch coordinates that the image object will show.  
SCALE is the scale at which it will be shown. GRIDSIZE is the grid size of the sketch.  
If SKETCH is a viewer, any of the other arguments that are NIL will be filled in from the values in the viewer.  
If SKETCH is a sketch, REGION defaults to the extent of the sketch, SCALE defaults to 1.0 and GRIDSIZE defaults to 8.0.)

```
(SKETCH.IMAGEOBJ (INSURE.SKETCH SKETCH)
(COND
  ((REGIONP REGION))
  (REGION (ERROR REGION " illegal argument."))
  (T (SKETCH.REGION.VIEWED SKETCH)))
(COND
  ((NUMBERP SCALE))
  ((WINDOWP SKETCH)
   (VIEWER.SCALE SKETCH))
  (T 1.0))
(COND
  ((NUMBERP GRIDSIZE))
  ((WINDOWP SKETCH)
   (SK.GRIDFACTOR SKETCH))
  (T 8.0))
```

## (SK.ELEMENT.FROM.IMAGEOBJ

[LAMBDA (IMAGEOBJ SKETCHW ORGPOS)

(\* rrb "11-Jul-86 15:48")

(\* returns a sketch element for an image object.)

```
(SKETCH.IMAGE.OBJECT.ELEMENT IMAGEOBJ (VIEWER.SCALE SKETCHW)
  ORGPOS SKETCHW])
```

## (SKETCHIMAGEOBJ.FROM.VIEWER

[LAMBDA (SKETCHW)

(\* rrb "11-Jul-86 15:51")

(\* returns a SKETCH image object which describes the contents of a window.)

```
(SKETCH.IMAGEOBJ (INSURE.SKETCH (SKETCH.FROM.VIEWER SKETCHW))
  (SKETCH.REGION.VIEWED SKETCHW)
  (VIEWER.SCALE SKETCHW)
  (SK.GRIDFACTOR SKETCHW])
```

## (SKETCH.IMAGEOBJ

[LAMBDA (SKETCH REGION SCALE GRID)
(DECLARE (GLOBALVARS SKETCHIMAGEFNS))

(\* rrb "29-Jan-86 11:54")

(\* returns an image obj which gives the functional information for a sketch object in a tedit file.)

```
(IMAGEOBJCREATE (create SKETCHIMAGEOBJ
  SKIO.SKETCH _ SKETCH
  SKIO.REGION _ (COND
    ((REGIONP REGION))
    (T (SKETCH.REGION.OF.SKETCH SKETCH)))
  SKIO.SCALE _ (OR (NUMBERP SCALE)
    1.0)
  SKIO.GRID _ (OR (NUMBERP GRID)
    8.0))
  SKETCHIMAGEFNS))
```

## (SKETCH.DISPLAYFN

[LAMBDA (SKETCHIMAGEOBJ STREAM)

; Edited 15-Jul-2022 14:30 by larry  
; Edited 27-Feb-87 18:15 by rrb

(\* display function for a sketch image object)

```
(PROG ((SKIO (IMAGEOBJPROP SKETCHIMAGEOBJ 'OBJECTDATUM))
  REGION TYPE)
  (SETQ REGION (fetch (SKETCHIMAGEOBJ SKIO.REGION) of SKIO))
  (COND
    ((EQMEMB 'DISPLAY (SETQ TYPE (IMAGESTREAMTYPE STREAM))))
    (* This is being displayed on the screen)
    (BITBLT [COND
      ((fetch (SKETCHIMAGEOBJ SKIO.LOCALSPECS) of SKIO))
      (T
```

(\* SKIO.LOCALSPECS is used to cache the local bitmap of the sketch as it is being display now.)

```
(replace (SKETCHIMAGEOBJ SKIO.LOCALSPECS) of SKIO with (SKETCH.BITMAP.IMAGE
  (fetch (SKETCHIMAGEOBJ
    SKIO.SKETCH)
    of SKIO)
  REGION
  (fetch (SKETCHIMAGEOBJ
    SKIO.SCALE)
    of SKIO]
```

```
0 0 STREAM (DSPXPOSITION NIL STREAM)
(DSPYPOSITION NIL STREAM)))
(T (PROG ((SKSCALE (fetch (SKETCHIMAGEOBJ SKIO.SCALE) of SKIO))
  (STRMSCALE (DSPSCALE NIL STREAM))
  SKTOSTRMSCALE SKXOFFSET SKYOFFSET)
```

(\* the TRANSLATE.SKETCH is to move the sketch to the right place on the page.  
When all streams support tranlation, this should be taken out.)

```
(SETQ SKTOSTRMSCALE (QUOTIENT SKSCALE STRMSCALE))
(SETQ SKXOFFSET (DIFFERENCE (TIMES (DSPXPOSITION NIL STREAM)
  SKTOSTRMSCALE)
  (fetch (REGION LEFT) of REGION)))
(SETQ SKYOFFSET (DIFFERENCE (TIMES (DSPYPOSITION NIL STREAM)
  SKTOSTRMSCALE)
  (fetch (REGION BOTTOM) of REGION)))
```

(\* save and restore the font as Tedit assumes that it is preserved over the call.)

```
(RETURN (DSPFONT (PROG1 (DSPFONT NIL STREAM)
  (DRAW.LOCAL.SKETCH (MAKE.LOCAL.SKETCH
    (TRANSLATE.SKETCH (COPY (fetch (SKETCHIMAGEOBJ
      SKIO.SKETCH)
      of SKIO))
    (IMINUS SKXOFFSET)
    (IMINUS SKYOFFSET))
    (SETQ REGION (TRANSLATE.REGION REGION SKXOFFSET
      SKYOFFSET))
    SKTOSTRMSCALE STREAM T)
  STREAM
  (SK.SCALE.REGION REGION SKTOSTRMSCALE)))
  STREAM])
```

## (SKETCH.BITMAP.IMAGE

[LAMBDA (SKETCH REGION SCALE)

(\* rrb "21-Jan-86 15:56")  
(\* Returns a bitmap that has the sketch image in it.)

```
(SETQ SKETCH (INSURE.SKETCH SKETCH))
(OR (REGIONP REGION)
  (SETQ REGION (SKETCH.REGION.OF.SKETCH SKETCH)))
(OR (NUMBERP SCALE)
  (SETQ SCALE 1.0))
(PROG (BITMAP DSP)
```

(\* make the bitmap image 1 bit larger than might be absolutely necessary to allow a one bit slop for floating pt roundoff.)

```
(SETQ BITMAP (BITMAPCREATE (IPLUS (QUOTIENT (fetch (REGION WIDTH) of REGION)
```

```

                                SCALE)
                                1)
      (IPLUS (QUOTIENT (fetch (REGION HEIGHT) of REGION)
                                SCALE)
                                1)))
      (SETQ DSP (DSPCREATE BITMAP))

      (* adjust the offsets of the stream so that the sketch does not have to be translated.)

      (DSPXOFFSET (IMINUS (QUOTIENT (fetch (REGION LEFT) of REGION)
                                SCALE))
      DSP)
      (DSPYOFFSET (IMINUS (QUOTIENT (fetch (REGION BOTTOM) of REGION)
                                SCALE))
      DSP)
      (RESETFORM (CURSOR WAITINGCURSOR)
      (DRAW.LOCAL.SKETCH (MAKE.LOCAL.SKETCH SKETCH REGION SCALE DSP T)
      DSP
      (DSPCLIPPINGREGION NIL DSP)
      SCALE))
      (RETURN BITMAP])

```

**(SKIO.IMAGEBOXFN**

[LAMBDA (IMAGEOBJ STREAM)

; Edited 27-Feb-87 18:04 by rrb

(\* size function for a sketch image object.)

```

      (PROG ((SKOBJ (IMAGEOBJPROP IMAGEOBJ 'OBJECTDATUM))
      SKREG SKW SKH SCALEFACTOR)

```

(\* determine the scale between the sketch specs and the stream.)

```

      (SETQ SCALEFACTOR (QUOTIENT (fetch (SKETCHIMAGEOBJ SKIO.SCALE) of SKOBJ)
      (DSPSCALE NIL STREAM)))
      (SETQ SKW (FIXR (FQUOTIENT (fetch (REGION WIDTH) of (SETQ SKREG (fetch (SKETCHIMAGEOBJ SKIO.REGION)
      of SKOBJ)))
      SCALEFACTOR)))
      (SETQ SKH (FIXR (FQUOTIENT (fetch (REGION HEIGHT) of SKREG)
      SCALEFACTOR)))
      (RETURN (COND
      ((\SKIO.IN.TOO.SMALL.TEDITP STREAM SKH)

```

(\* special check for displaying in a Tedit window that is less than the height of the sketch.  
leave enough height for a few lines of text too.)

```

      (create IMAGEBOX
      XSIZE _ SKW
      YSIZE _ (IMAX 12 (DIFFERENCE (fetch (REGION HEIGHT) of (DSPCLIPPINGREGION NIL STREAM)
      24))
      YDESC _ 0
      XKERN _ 0))
      (T (create IMAGEBOX
      XSIZE _ SKW
      YSIZE _ SKH
      YDESC _ 0
      XKERN _ 0))

```

**(SKIO.GETFN.2**

[LAMBDA (STREAM)

(\* rrb "18-Oct-85 16:11")

(\* Get a description of a sketch object from the file.)

```

      (SKETCH.IMAGEOBJ [PROG ((READSKETCH (HREAD STREAM))

```

```

      (RETURN (SK.CHECK.SKETCH.VERSION (COND

```

```

      ((NLISTP (CAR READSKETCH))

```

(\* pre property list format, update it.)

```

      (SKIO.UPDATE.FROM.OLD.FORM READSKETCH))

```

```

      (T

```

(\* values of all properties and sketch elements were written out as a LIST.)

```

      (create SKETCH
      ALLSKETCHPROPS _ (CAR READSKETCH)
      SKETCHELTS _ (CDR READSKETCH])

```

```

      (READ STREAM)
      (READ STREAM)
      (READ STREAM])

```

**(SKIO.UPDATE.FROM.OLD.FORM**

[LAMBDA (OLDSKETCH)

(\* rrb "18-Jul-85 17:24")

(\* converts a sketch from old form to new form.)

(\* update the arrowhead format to the new form.)

```

      (MAPGLOBSKESCHELEMENTS (CDR OLDSKETCH)
      (FUNCTION SK.UPDATE.ARROWHEAD.FORMAT))
      (create SKETCH
      SKETCHNAME _ (CAR OLDSKETCH)
      SKETCHELTS _ (CDR OLDSKETCH])

```

**(SKIO.GETFN**

```
[LAMBDA (STREAM) (* rrb "7-May-85 11:21")

(* Get a description of a sketch object from the file. This is an old version left around in case old format object still exist.)

(printout T "This file contains sketch that is in an old format. " "To update it to the new format, "
"load this file into a Harmony sysout and do a 'Put' from there.")
(ERROR "old format Sketch object"])
```

**(SKIO.PUTFN**

```
[LAMBDA (IMAGEOBJ STREAM) (* rrb "12-May-85 18:34")
(* Put a description of a sketch object into the file.)

(PROG ((SKETCHIMAGEOBJ (IMAGEOBJPROP IMAGEOBJ 'OBJECTDATUM)
SKETCH)
(SETQ SKETCH (fetch (SKETCHIMAGEOBJ SKIO.SKETCH) of SKETCHIMAGEOBJ))

(* can't print sketch directly because it contains a TCONC cell which must be reconstructed on reading in.)

(HPRINT (CONS (fetch (SKETCH ALLSKETCHPROPS) of SKETCH)
(fetch (SKETCH SKETCHELTS) of SKETCH))
STREAM T)
(PRINT (fetch (SKETCHIMAGEOBJ SKIO.REGION) of SKETCHIMAGEOBJ)
STREAM)
(PRINT (fetch (SKETCHIMAGEOBJ SKIO.SCALE) of SKETCHIMAGEOBJ)
STREAM)
(PRINT (fetch (SKETCHIMAGEOBJ SKIO.GRID) of SKETCHIMAGEOBJ)
STREAM])
```

**(SKIO.COPYFN**

```
[LAMBDA (IMAGEOBJ) (* rrb "26-Oct-84 10:27")
(* makes a copy of a sketch image object.)

(PROG [(SKETCHOBJ (IMAGEOBJPROP IMAGEOBJ 'OBJECTDATUM)
(RETURN (SKETCH.IMAGEOBJ (COPY (fetch (SKETCHIMAGEOBJ SKIO.SKETCH) of SKETCHOBJ))
(COPY (fetch (SKETCHIMAGEOBJ SKIO.REGION) of SKETCHOBJ))
(fetch (SKETCHIMAGEOBJ SKIO.SCALE) of SKETCHOBJ)
(fetch (SKETCHIMAGEOBJ SKIO.GRID) of SKETCHOBJ))
```

**(SKIO.BUTTONEVENTINFN**

```
[LAMBDA (IMAGEOBJ WINDOW) (* rrb "31-Jul-86 10:24")

(* the user has pressed a button inside the sketch object IMAGEOBJ.
Offer a chance to edit it in a separate window.)

(PROG [(OBJ (IMAGEOBJPROP IMAGEOBJ 'OBJECTDATUM)
(SELECTQ [MENU (create MENU
ITEMS _ ' ((Edit% sketch 'EDIT "opens a window in which this sketch can be
changed.")
(EDIT (* user wants to edit it)
[PROG ((SKREG (fetch (SKETCHIMAGEOBJ SKIO.REGION) of OBJ))
(SCALE (fetch (SKETCHIMAGEOBJ SKIO.SCALE) of OBJ))
(SKETCH (fetch (SKETCHIMAGEOBJ SKIO.SKETCH) of OBJ))
SKW TITLE)

(* give the sketch a new name so that it doesn't get confused about the real sketch on the property list.
The whole idea of names should probably be scrapped)

(SETQ SKW (SKETCHW.CREATE (create SKETCH using SKETCH SKETCHNAME _
[SETQ TITLE (COND
((SETQ TITLE
(WINDOWPROP
WINDOW
'TEDIT.ICON.TITLE))
(CONCAT "figure from "
TITLE))
(T
,
|a figure from a document|
)
SKETCHPROPS _ (COPY (fetch (SKETCH
SKETCHPROPS
)
of SKETCH))
SKETCHELTS _ (COPY (fetch (SKETCH
SKETCHELTS
)
of SKETCH)))
SKREG
(GETBOXREGION (WIDTHIFWINDOW (FIXR (FQUOTIENT (fetch (REGION WIDTH)
of SKREG)
SCALE)))
(HEIGHTIFWINDOW (FIXR (FQUOTIENT (fetch (REGION HEIGHT)
of SKREG)
SCALE)))
T))
TITLE SCALE T (fetch (SKETCHIMAGEOBJ SKIO.GRID) of OBJ)))
```

(\* keep track of enough information to find this sketch in the document if the user closes the window.)

```
(WINDOWPROP SKW 'DOCUMENTINFO (create SKETCHDOCUMENTINFO
FROMIMAGEOBJ _ IMAGEOBJ
FROMTEDITWINDOW _ WINDOW))
(* give this process the tty so that it will stay on top.)
(TTY.PROCESS (WINDOWPROP SKW 'PROCESS))
```

(\* add a process to bring this window to the top after TEdit has cleared its selection which brings it up on top again.  
Yech!!)

```
(ADD.PROCESS (LIST 'TOTOPW (KWOTE SKW)))
NIL))
```

## (TRANSLATE.REGION

```
[LAMBDA (REGION NEWLEFT NEWBOTTOM)
```

(\* rrb "20-Sep-84 14:12")

(\* translates a region so that its new lower left corner is at  
NEWLEFT NEWBOTTOM)

```
(CREATEREGION (PLUS (fetch (REGION LEFT) of REGION)
NEWLEFT)
(PLUS (fetch (REGION BOTTOM) of REGION)
NEWBOTTOM)
(fetch (REGION WIDTH) of REGION)
(fetch (REGION HEIGHT) of REGION])
```

## (UPDATE.IMAGE.IN.DOCUMENT

```
[LAMBDA (SKW)
```

(\* rrb "26-Sep-86 10:16")

(\* \* this sketch window was the result of editing a sketch from a document.  
Ask if the user wants to put it back and if so, do it.)

```
(SELECTQ (\CURSOR.IN.MIDDLE.MENU (create MENU
TITLE _ "Put changes back into Document?"
ITEMS _ '( (Yes 'YES "this image used in the document instead of the
one that is there.")
(No 'NO "the changes made to this image will not be put
into the document."))
CENTERFLG _ T))
(YES (PROG ((DOCINFO (WINDOWPROP SKW 'DOCUMENTINFO))
TEXTOBJ OLDIMAGEOBJ POS)
(COND
([NOT (SETQ TEXTOBJ (TEXTOBJ (fetch (SKETCHDOCUMENTINFO FROMTEDITWINDOW) of DOCINFO]
(PROMPTPRINT "Can't find the edit window for the source document.")
(RETURN)))
(COND
([NOT (SETQ POS (TEDIT.FIND.OBJECT TEXTOBJ (SETQ OLDIMAGEOBJ (fetch (SKETCHDOCUMENTINFO
FROMIMAGEOBJ)
of DOCINFO]
(PROMPTPRINT "Can't find this sketch in the document it came from.")
(* later should allow the user to specify where.)
(RETURN)))
(TEDIT.DELETE (SETQ TEXTOBJ (TEXTSTREAM TEXTOBJ))
POS 1)
(TEDIT.INSERT.OBJECT (SKETCHIMAGEOBJ.FROM.VIEWER SKW)
TEXTOBJ POS)))
(NIL 'DON'T)
(NO NIL)
NIL))
```

(\* if the user clicks outside, stop the close.)

## (SK.COPY.IMAGEOBJ

```
[LAMBDA (GELT WINDOW CALLWHENCOPIEDFN)
```

(\* rrb "29-Jun-87 14:22")

(\* \* makes a copy of a image object sketch element. Has to call the image objects copyfn.  
Calls its its WHENCOPIEDFN if CALLWHENCOPIEDFN is not NIL This is normally NIL because the WHENINSERTEDFN  
is used instead.)

```
(PROG ((INDVGELT (fetch (GLOBALPART INDIVIDUALGLOBALPART) of GELT))
IMAGEOBJ FN NEWSKELT)
[SETQ NEWSKELT (create GLOBALPART
INDIVIDUALGLOBALPART _ [create SKIMAGEOBJ using INDVGELT SKIMAGEOBJ _
(COPY.IMAGE.OBJECT
(SETQ IMAGEOBJ
(fetch (SKIMAGEOBJ SKIMAGEOBJ)
of INDVGELT]
COMMONGLOBALPART _ (COPY (fetch (GLOBALPART COMMONGLOBALPART) of GELT]
(COND
((AND CALLWHENCOPIEDFN (SETQ FN (IMAGEOBJPROP IMAGEOBJ 'WHENCOPIEDFN))
(NEQ FN 'NIL))
(* documentation calls for passing text streams as well but there
aren't any.)
(APPLY* FN IMAGEOBJ WINDOW)))
(RETURN NEWSKELT])
```

**(COPY.IMAGE.OBJECT**

[LAMBDA (IMAGEOBJ)

(\* rrb "29-Jun-87 14:22")

(\* copies an image object calling its copyfn if possible.)

```

  (PROG (FN)
    (RETURN (COND
      ((AND (SETQ FN (IMAGEOBJPROP IMAGEOBJ 'COPYFN))
        (NEQ FN 'NILL))
        (APPLY* FN IMAGEOBJ))
      (T (COPY IMAGEOBJ))
    ))
  )

```

**(\CREATE.SKETCH.IMAGEFNS**

[LAMBDA NIL

(\* rrb "23-Oct-85 11:03")

(\* creates the IMAGEFNS vector for the sketch image object.)

```

  (DECLARE (GLOBALVARS SKETCHIMAGEFNS))
  (COND
    ((IMAGEFNSP SKETCHIMAGEFNS))
    (T (SETQ SKETCHIMAGEFNS (IMAGEFNSCREATE (FUNCTION SKETCH.DISPLAYFN)
      (FUNCTION SKIO.IMAGEBOXFN)
      (FUNCTION SKIO.PUTFN)
      (FUNCTION SKIO.GETFN.2)
      (FUNCTION SKIO.COPYFN)
      (FUNCTION SKIO.BUTTONEVENTINFN)
      (FUNCTION NILL)
      (FUNCTION NILL)
      (FUNCTION NILL)
      (FUNCTION NILL)
      (FUNCTION NILL)
      (FUNCTION NILL)
      (FUNCTION NILL)
      (FUNCTION NILL)
    ))
  )

```

**(SKIO.IN.TOO.SMALL.TEDITP**

[LAMBDA (STREAM HEIGHT)

; Edited 7-Oct-88 12:21 by rmk:

; Edited 27-Feb-87 18:19 by rrb

(\* is this stream a TEDIT window that is smaller than height?)

```

  (AND (DISPLAYSTREAMP STREAM)
    (EQ (DSPDESTINATION NIL STREAM)
      (SCREENBITMAP))
    (WINDOWPROP (WFROMDS STREAM)
      'TEXTOBJ)
    (GREATERP HEIGHT (fetch (REGION HEIGHT) of (DSPCLIPPINGREGION NIL STREAM))
  )

```

**(SKETCH.VIEWER.GRID**

[LAMBDA (VIEWER NEWGRID)

; Edited 18-Nov-87 17:51 by rrb

(\* returns and optionally sets the grid size of a sketch. VIEWER can be a viewer or a sketch image object.)

```

  (COND
    [(IMAGEOBJP VIEWER) (* it is a sketch image object)
      (PROG [(SKINFO (LISTP (IMAGEOBJPROP VIEWER 'OBJECTDATUM)
        (COND
          [(type? SKETCH (fetch (SKETCHIMAGEOBJ SKIO.SKETCH) of SKINFO))
            (RETURN (PROG1 (fetch (SKETCHIMAGEOBJ SKIO.GRID) of SKINFO)
              [COND
                (NEWGRID (COND
                  ((NUMBERP NEWGRID)
                    (replace (SKETCHIMAGEOBJ SKIO.GRID) of SKINFO with NEWGRID))
                  (T (\ILLEGAL.ARG NEWGRID))])
                (T (ERROR "not a sketch image object" VIEWER)
              ))
            ((WINDOWP VIEWER)
              (SK.GRIDFACTOR VIEWER NEWGRID))
            (T (\ILLEGAL.ARG VIEWER))
          ]
        )
      )

```

**(SKETCH.VIEWER.SCALE**

[LAMBDA (VIEWER NEWSCALE)

(\* rrb "21-Apr-87 12:25")

(\* returns and optionally sets the scale of a viewer.)

```

  (COND
    [(IMAGEOBJP VIEWER) (* it is a sketch image object)
      (PROG [(SKINFO (LISTP (IMAGEOBJPROP VIEWER 'OBJECTDATUM)
        (COND
          [(type? SKETCH (fetch (SKETCHIMAGEOBJ SKIO.SKETCH) of SKINFO))
            (RETURN (PROG1 (fetch (SKETCHIMAGEOBJ SKIO.SCALE) of SKINFO)
              [COND
                (NEWSCALE (COND
                  ((SK.INSURE.SCALE NEWSCALE)
                    (replace (SKETCHIMAGEOBJ SKIO.SCALE) of SKINFO with NEWSCALE))
                  (T (\ILLEGAL.ARG NEWSCALE))])
                (T (ERROR "not a sketch image object" VIEWER)
              ))
            ((WINDOWP VIEWER)
              (PROG1 (WINDOWPROP VIEWER 'SCALE)
                [COND
                  (NEWSCALE (COND
                    ((SK.INSURE.SCALE NEWSCALE)
                      (WINDOWPROP VIEWER 'SCALE NEWSCALE)
                    )
                  )
                )
              )
            )
          ]
        )
      )

```

```

    (SK.UPDATE.AFTER.SCALE.CHANGE VIEWER]])
  (T (\ILLEGAL.ARG VIEWER])
)

(DECLARE%: DONTCOPY DOEVAL@COMPILE

(DECLARE%: EVAL@COMPILE

(RECORD SKETCHIMAGEOBJ (SKIO.SKETCH SKIO.REGION SKIO.SCALE SKIO.LOCALSPECS SKIO.GRID))

(RECORD SKETCHDOCUMENTINFO (FROMIMAGEOBJ FROMTEDITWINDOW))
)
)

(\CREATE.SKETCH.IMAGEFNS)

(ADDTOTVAR IMAGEOBJGETFNS (SKIO.GETFN))

(* * stuff to support image objects as elements in a sketch)

(DEFINEQ

(SKETCH.IMAGE.OBJECT.ELEMENT
  [LAMBDA (IMAGEOBJ SCALE GLOBALPOS VIEWER)
    (* rrb "8-Jul-86 12:38")

    (* internal function for creating a global imageobj sketch element.
    Called during copy select insert and during editing.)

    (SK.UPDATE.IMAGEOBJECT.AFTER.CHANGE (SKETCH.CREATE.IMAGE.OBJECT1 IMAGEOBJ GLOBALPOS SCALE)
      VIEWER])

(SKETCH.IMAGEOBJ.OF.ELEMENT
  [LAMBDA (ELEMENT)
    (* rrb "14-Aug-85 16:38")
    (* returns the image object from an image object sketch
    element.)
    (fetch (SKIMAGEOBJ SKIMAGEOBJ) of (fetch (GLOBALPART INDIVIDUALGLOBALPART) of ELEMENT])

(SKETCH.SCALE.OF.ELEMENT
  [LAMBDA (ELEMENT)
    (* rrb "14-Aug-85 16:39")
    (* returns the scale from an image object sketch element.)
    (fetch (SKIMAGEOBJ SKIMOBJ.ORIGSCALE) of (fetch (GLOBALPART INDIVIDUALGLOBALPART) of ELEMENT])

(SKETCH.POSITION.OF.ELEMENT
  [LAMBDA (ELEMENT)
    (* rrb "14-Aug-85 16:42")
    (* returns the position from an image object sketch element.)
    (PROG [(REG (fetch (SKIMAGEOBJ SKIMOBJ.GLOBALREGION) of (fetch (GLOBALPART INDIVIDUALGLOBALPART) of ELEMENT)
      (RETURN (CREATEPOSITION (fetch (REGION LEFT) of REG)
        (fetch (REGION BOTTOM) of REG))

(CREATE.SKIMAGEOBJ.TYPE
  [LAMBDA NIL
    (* rrb "18-Oct-85 10:33")
    (* create a sketch type that allows image objects to appear in
    sketches.)

    (COND
      ((NOT (SKETCH.ELEMENT.TYPER 'SKIMAGEOBJ))
        (CREATE.SKETCH.ELEMENT.TYPE 'SKIMAGEOBJ NIL "functions for using image objects in sketches"
          (FUNCTION SK.IMAGEOBJ.DRAWFN)
          (FUNCTION SK.IMAGEOBJ.EXPANDFN)
          'OBSOLETE
          (FUNCTION SK.IMAGEOBJ.CHANGFN)
          (FUNCTION NIL)
          (FUNCTION SK.IMAGEOBJ.INSIDFN)
          (FUNCTION SK.IMAGEOBJ.REGIONFN)
          (FUNCTION SK.IMAGEOBJ.TRANSLATEFN)
          (FUNCTION NIL)
          (FUNCTION SK.IMAGEOBJ.READCHANGFN)
          (FUNCTION SK.IMAGEOBJ.TRANSFORMFN)
          NIL
          (FUNCTION SK.IMAGEOBJ.GLOBALREGIONFN])

(IMAGEBOXSIZE
  [LAMBDA (IMAGEOBJ IMAGESTREAM)
    (* rrb "4-Feb-86 14:41")
    (* returns the size of an imageobj)
    (APPLY* (fetch (IMAGEFNS IMAGEBOXFN) of (fetch (IMAGEOBJ IMAGEOBJFNS) of IMAGEOBJ))
      IMAGEOBJ IMAGESTREAM])

(SK.UPDATE.IMAGEOBJECT.AFTER.CHANGE
  [LAMBDA (IMOBJELT VIEWER)
    (* rrb "4-Feb-86 15:04")

```

(\* updates the dependent fields of a sketch image object element after the image object changes.)

```
(PROG (IMOBJSIZE IMAGEOBJ SCALE)
  (SETQ IMOBJSIZE (IMAGEBOXSIZE (fetch (SKIMAGEOBJ SKIMAGEOBJ) of (SETQ IMAGEOBJ (fetch (GLOBALPART
    INDIVIDUALGLOBALPART
    )
    of IMOBJELT)))
    VIEWER))
  (SETQ SCALE (fetch (SKIMAGEOBJ SKIMOBJ.ORIGSCALE) of IMAGEOBJ))
  (replace (SKIMAGEOBJ SKIMOBJ.GLOBALREGION) of IMAGEOBJ with (create REGION
    using (fetch (SKIMAGEOBJ
      SKIMOBJ.GLOBALREGION)
    of IMAGEOBJ)
    WIDTH _ (TIMES (fetch (IMAGEBOX XSIZE)
      of IMOBJSIZE)
    SCALE)
    HEIGHT _ (TIMES (fetch (IMAGEBOX YSIZE)
      of IMOBJSIZE)
    SCALE)))
  (replace (SKIMAGEOBJ SKIMOBJ.OFFSETPOS) of IMAGEOBJ with (create POSITION
    XCOORD _ (fetch (IMAGEBOX XKERN)
      of IMOBJSIZE)
    YCOORD _ (fetch (IMAGEBOX YDESC)
      of IMOBJSIZE)))
  (RETURN IMOBJELT])
```

### (SKETCH.CREATE.IMAGE.OBJECT

[LAMBDA (IMAGEOBJ POSITION SCALE)

(\* rrb "8-Jul-86 12:38")

(\* creates a sketch element from an image object.)

(\* calls update object with NIL viewer because no viewer is known.  
The image object must get called to calculate the size which should be in DISPLAY coordinates.  
Maybe could create a dummy display stream and pass that down.)

(SK.UPDATE.IMAGEOBJECT.AFTER.CHANGE (SKETCH.CREATE.IMAGE.OBJECT1 IMAGEOBJ POSITION SCALE)  
NIL])

### (SKETCH.CREATE.IMAGE.OBJECT1

[LAMBDA (IMAGEOBJ POSITION SCALE)

(\* rrb "8-Jul-86 12:12")

(\* creates a sketch element from an image object.)

```
(COND
  ((NUMBERP SCALE))
  ((NULL SCALE)
   (SETQ SCALE 1.0))
  (T (\ILLEGAL.ARG SCALE)))
(COND
  ((NULL POSITION))
  ((POSITIONP POSITION)
   (T (\ILLEGAL.ARG POSITION)))
  (create GLOBALPART
    INDIVIDUALGLOBALPART _ (create SKIMAGEOBJ
      SKIMAGEOBJ _ IMAGEOBJ
      SKIMOBJ.GLOBALREGION _ (CREATEREGION (COND
        (POSITION (fetch (POSITION
          XCOORD)
          of POSITION))
        (T 0))
        (COND
          (POSITION (fetch (POSITION YCOORD)
            of POSITION))
          (T 0))
        1 1)
      SKIMOBJ.ORIGSCALE _ SCALE)
    COMMONGLOBALPART _ (create COMMONGLOBALPART
      MAXSCALE _ (TIMES SCALE MINIMUM.VISIBLE.SCALE.FACTOR)
      MINSCALE _ (QUOTIENT SCALE DEFAULT.VISIBLE.SCALE.FACTOR])
  )
```

(DEFINEQ

### (SK.IMAGEOBJ.DRAWFN

[LAMBDA (IMAGEOBJELT WINDOW REGION)

(\* rrb "25-Oct-84 10:27")

(\* shows an image object element)

```
(PROG ((IMAGEOBJ (fetch (SKIMAGEOBJ SKIMAGEOBJ) of (fetch (SCREENELT INDIVIDUALGLOBALPART) of IMAGEOBJELT)))
  (LOCALIMOBJ (fetch (SCREENELT LOCALPART) of IMAGEOBJELT))
  LOCALPOS LOCALOFFSET)
  (SETQ LOCALPOS (fetch (LOCALSKIMAGEOBJ SKIMOBJLOCALPOS) of LOCALIMOBJ))
  (SETQ LOCALOFFSET (fetch (LOCALSKIMAGEOBJ SKIMOBJLOCALOFFSETPOS) of LOCALIMOBJ))
  (* move stream to correct position.)
  (MOVETO (PLUS (fetch (POSITION XCOORD) of LOCALPOS)
    (fetch (POSITION XCOORD) of LOCALOFFSET))
    (PLUS (fetch (POSITION YCOORD) of LOCALPOS)
```



```

        (fetch (POSITION YCOORD) of LOCALOFFSET))
      WINDOW)
    (COND
      ((type? ANNO (IMAGEOBJPROP IMAGEOBJ 'OBJECTDATUM)) (* handle annotations specially so they get the scale.)
        (ANNO.DISPLAYFN IMAGEOBJ WINDOW (IMAGESTREAMTYPE WINDOW)
          NIL
          (fetch (LOCALSKIMAGEOBJ SKIMOBJLOCALSCALE) of LOCALIMOBJ)))
      (T (APPLY* (IMAGEOBJPROP IMAGEOBJ 'DISPLAYFN)
        IMAGEOBJ WINDOW]))

```

**(SK.IMAGEOBJ.REGIONFN**

```

[LAMBDA (IMAGEOBJELT)
  (* rrb "4-Oct-84 13:34")
  (* determines the local region covered by image object elt.)
  (fetch (LOCALSKIMAGEOBJ SKIMOBJLOCALREGION) of (fetch (SCREENELT LOCALPART) of IMAGEOBJELT))

```

**(SK.IMAGEOBJ.GLOBALREGIONFN**

```

[LAMBDA (GIMOBJELT)
  (* rrb "18-Oct-85 10:33")
  (* returns the global region occupied by a global image object
  element.)
  (fetch (SKIMAGEOBJ SKIMOBJ.GLOBALREGION) of (fetch (GLOBALPART INDIVIDUALGLOBALPART) of GIMOBJELT))

```

**(SK.IMAGEOBJ.TRANSLATEFN**

```

[LAMBDA (GIMAGEOBJ DELTAPOS WINDOW)
  (* rrb "15-Dec-86 15:34")
  (* moves a imageobj figure element to a new position.)
  (PROG ((INDIMAGEOBJELT (fetch (GLOBALPART INDIVIDUALGLOBALPART) of GIMAGEOBJ))
    IMAGEOBJ FN)
    (COND
      ((AND (SETQ FN (IMAGEOBJPROP (SETQ IMAGEOBJ (fetch (SKIMAGEOBJ SKIMAGEOBJ) of INDIMAGEOBJELT))
        'WHENMOVEDFN))
        (NEQ FN 'NIL))
        (* documentation calls for passing text streams as well but there
        aren't any.)
        (* update the region positions.)
        (APPLY* FN IMAGEOBJ WINDOW WINDOW)))
      (RETURN (create GLOBALPART
        COMMONGLOBALPART _ (APPEND (fetch (GLOBALPART COMMONGLOBALPART) of GIMAGEOBJ))
        INDIVIDUALGLOBALPART _ (create SKIMAGEOBJ using INDIMAGEOBJELT SKIMOBJ.GLOBALREGION _
          (TRANSLATE.REGION (fetch (SKIMAGEOBJ
            SKIMOBJ.GLOBALREGION
            )
            of INDIMAGEOBJELT)
            (fetch (POSITION XCOORD)
              of DELTAPOS)
            (fetch (POSITION YCOORD)
              of DELTAPOS]))

```

**(SK.IMAGEOBJ.EXPANDFN**

```

[LAMBDA (GIMAGEOBJPART SCALE)
  (* rrb "11-Jul-86 15:55")
  (* creates a local imageobject screen element from a global
  imageobject element.)
  (PROG ((GIMAGEOBJ (fetch (GLOBALPART INDIVIDUALGLOBALPART) of GIMAGEOBJPART))
    LOCALREG LOCALPOS IMAGESIZE)
    (SETQ LOCALREG (SK.SCALE.REGION (fetch (SKIMAGEOBJ SKIMOBJ.GLOBALREGION) of GIMAGEOBJ)
      SCALE))
    (RETURN (create SCREENELT
      LOCALPART _ (create LOCALSKIMAGEOBJ
        SKIMOBJLOCALPOS _ (create POSITION
          XCOORD _ (fetch (REGION LEFT) of LOCALREG)
          YCOORD _ (fetch (REGION BOTTOM) of LOCALREG)
          SKIMOBJLOCALSCALE _ (QUOTIENT SCALE (fetch (SKIMAGEOBJ
            SKIMOBJ.ORIGSCALE)
            of GIMAGEOBJ))
          SKIMOBJLOCALREGION _ LOCALREG
          SKIMOBJLOCALOFFSETPOS _ (SK.SCALE.POSITION.INTO.VIEWER
            (fetch (SKIMAGEOBJ SKIMOBJ.OFFSETPOS)
              of GIMAGEOBJ)
            SCALE))
        GLOBALPART _ GIMAGEOBJPART]))

```

**(SK.IMAGEOBJ.INSIDFN**

```

[LAMBDA (GIMAGEOBJ WREG)
  (* rrb "31-Mar-84 09:15")
  (* determines if the global annotation element is inside of
  WREG.)
  (REGIONINTERSECTP (fetch (SKIMAGEOBJ SKIMOBJ.GLOBALREGION) of (fetch (GLOBALPART INDIVIDUALGLOBALPART)
    of GIMAGEOBJ))
    WREG])

```

**(SK.IMAGEOBJ.MOVEFN**

```

[LAMBDA (IMAGEOBJELT SELPOS NEWINPUTPT WINDOW)
  (* rrb "11-Jul-86 15:51")
  (* moves a annotation element to a new position.)
  (PROG ((GIMOBJ (fetch (SCREENELT GLOBALPART) of IMAGEOBJELT))
    (SCALEDNEWPOS (SK.MAP.INPUT.PT.TO.GLOBAL NEWINPUTPT (VIEWER.SCALE WINDOW)))

```

```

      GREG GINDV FN IMAGEOBJ) (* update the position)
[SETQ GREG (fetch (SKIMAGEOBJ SKIMOBJ.GLOBALREGION) of (SETQ GINDV (fetch (GLOBALPART
                                                                INDIVIDUALGLOBALPART)
                                                                of GIMOBJ]
(replace (SKIMAGEOBJ SKIMOBJ.GLOBALREGION) of GINDV with (CREATEREGION (fetch (POSITION XCOORD)
                                                                of SCALEDNEWPOS)
                                                                (fetch (POSITION YCOORD) of SCALEDNEWPOS)
                                                                (fetch (REGION WIDTH) of GREG)
                                                                (fetch (REGION HEIGHT) of GREG)))
(COND
  ((AND (SETQ FN (IMAGEOBJPROP (SETQ IMAGEOBJ (fetch (SKIMAGEOBJ SKIMAGEOBJ) of GINDV))
                                'WHENMOVEDFN))
    (NEQ FN 'NIL))
    aren't any.) (* documentation calls for passing text streams as well but there
    (APPLY* FN IMAGEOBJ WINDOW)))
(RETURN GIMOBJ])

```

**(SK.IMAGEOBJ.CHANGEFN**

```

[LAMBDA (IMAGEOBJELTS WINDOW HOW) (* rrb "4-Feb-86 14:58")

(* * user has indicated that they want to change the image object in IMAGEOBJELT)
always returns T) (* HOW is always T because SK.IMAGEOBJ.READCHANGEFN
                  (* for now only work on the first one.)
(PROG (FN (IMAGEOBJELT (CAR IMAGEOBJELTS))
      SKIMOBJELT NEWIMAGEOBJ IMAGEOBJ OLDREG)
  (SETQ SKIMOBJELT (fetch (SCREENELT INDIVIDUALGLOBALPART) of IMAGEOBJELT))
  (SETQ IMAGEOBJ (fetch (SKIMAGEOBJ SKIMAGEOBJ) of SKIMOBJELT))
  (* call the BUTTONEVENTINFN even though this may not work
  because much information is unavailable.)
  (COND
    ((AND (SETQ FN (IMAGEOBJPROP IMAGEOBJ 'BUTTONEVENTINFN))
      (NEQ FN 'NIL))
      (AND (SETQ NEWIMAGEOBJ (APPLY* (IMAGEOBJPROP IMAGEOBJ 'BUTTONEVENTINFN)
        IMAGEOBJ WINDOW))
        (RETURN (LIST (create SKHISTORYCHANGESPEC
          NEWELT _ (SKETCH.IMAGE.OBJECT.ELEMENT
            (COND
              ((type? IMAGEOBJ NEWIMAGEOBJ)
                NEWIMAGEOBJ)
              (T IMAGEOBJ))
            (fetch (SKIMAGEOBJ SKIMOBJ.ORIGSCALE) of SKIMOBJELT)
            (create POSITION
              XCOORD _ (fetch (REGION LEFT)
                of (SETQ OLDREG (fetch (SKIMAGEOBJ
                  SKIMOBJ.GLOBALREGION
                    of SKIMOBJELT)))
              YCOORD _ (fetch (REGION BOTTOM) of OLDREG))
            WINDOW)
            OLDELT _ (fetch (SCREENELT GLOBALPART) of IMAGEOBJELT)
            PROPERTY _ 'DATA
            NEWVALUE _ NEWIMAGEOBJ
            OLDVALUE _ IMAGEOBJ])
          )
    )
  )

```

**(SK.IMAGEOBJ.READCHANGEFN**

```

[LAMBDA (SKW) (* return T so SK.IMAGE.OBJ.CHANGEFN will always be
              called.)

T])

```

**(SK.IMAGEOBJ.TRANSFORMFN**

```

[LAMBDA (GELT TRANSFORMFN TRANSFORMDATA SCALEFACTOR) (* rrb "15-Dec-86 15:35")

(* * returns a copy of the global SKIMAGEOBJ element that has its region transformed by transformfn.
TRANSFORMDATA is arbitrary data that is passed to tranformfn.)

(PROG ((INDVPART (fetch (GLOBALPART INDIVIDUALGLOBALPART) of GELT)))
  (RETURN (create GLOBALPART
    COMMONGLOBALPART _ (APPEND (fetch (GLOBALPART COMMONGLOBALPART) of GELT))
    INDIVIDUALGLOBALPART _ (create SKIMAGEOBJ using INDVPART SKIMOBJ.GLOBALREGION _
      (SK.TRANSFORM.REGION (fetch (SKIMAGEOBJ
        SKIMOBJ.GLOBALREGION
          of INDVPART)
        TRANSFORMFN TRANSFORMDATA]))
  )

(DECLARE%: EVAL@COMPILE

(RECORD LOCALSKIMAGEOBJ ((SKIMOBJLOCALPOS)
  LOCALHOTREGION SKIMOBJLOCALSCALE SKIMOBJLOCALREGION SKIMOBJLOCALOFFSETPOS))

(TYPERECORD SKIMAGEOBJ (SKIMAGEOBJ SKIMOBJ.GLOBALREGION SKIMOBJ.ORIGSCALE SKIMOBJ.OFFSETPOS))

```

```
(DATATYPE ANNO (ANNO\SUBSTANCE ANNO\ID ANNO\WINDOW ANNO\DATE ANNO\PARENTS ANNO\NEXTSUBID ANNO\TYPE
                ANNO\SUMMARIZED\IN ANNO\UPDATE\DATE ANNO\CREATE\BY ANNO\UPDATE\BY ANNO\FONT))
)
```

```
(/DECLAREDATATYPE 'ANNO ' (POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
                           POINTER POINTER)
```

```
;; ---field descriptor list elided by lister---
```

```
' 24)
```

(CREATE.SKIMAGEOBJ.TYPE)

---

FUNCTION INDEX

COPY.IMAGE.OBJECT .....	6	SKETCH.DISPLAYFN .....	2
CREATE.SKIMAGEOBJ.TYPE .....	7	SKETCH.IMAGE.OBJECT.ELEMENT .....	7
IMAGEBOXSIZE .....	7	SKETCH.IMAGEOBJ .....	1
MAKE.IMAGE.OBJECT.OF.SKETCH .....	1	SKETCH.IMAGEOBJ.OF.ELEMENT .....	7
SK.COPY.IMAGEOBJ .....	5	SKETCH.POSITION.OF.ELEMENT .....	7
SK.ELEMENT.FROM.IMAGEOBJ .....	1	SKETCH.SCALE.OF.ELEMENT .....	7
SK.IMAGEOBJ.CHANGEFN .....	10	SKETCH.VIEWER.GRID .....	6
SK.IMAGEOBJ.DRAWFN .....	8	SKETCH.VIEWER.SCALE .....	6
SK.IMAGEOBJ.EXPANDFN .....	9	SKETCHIMAGEOBJ.FROM.VIEWER .....	1
SK.IMAGEOBJ.GLOBALREGIONFN .....	9	SKIO.BUTTONEVENTINFN .....	4
SK.IMAGEOBJ.INSIDEFN .....	9	SKIO.COPYFN .....	4
SK.IMAGEOBJ.MOVEFN .....	9	SKIO.GETFN .....	3
SK.IMAGEOBJ.READCHANGEFN .....	10	SKIO.GETFN.2 .....	3
SK.IMAGEOBJ.REGIONFN .....	9	SKIO.IMAGEBOXFN .....	3
SK.IMAGEOBJ.TRANSFORMFN .....	10	SKIO.PUTFN .....	4
SK.IMAGEOBJ.TRANSLATEFN .....	9	SKIO.UPDATE.FROM.OLD.FORM .....	3
SK.UPDATE.IMAGEOBJECT.AFTER.CHANGE .....	7	TRANSLATE.REGION .....	5
SKETCH.BITMAP.IMAGE .....	2	UPDATE.IMAGE.IN.DOCUMENT .....	5
SKETCH.CREATE.IMAGE.OBJECT .....	8	\CREATE.SKETCH.IMAGEFNS .....	6
SKETCH.CREATE.IMAGE.OBJECT1 .....	8	\SKIO.IN.TOO.SMALL.TEDITP .....	6

---

RECORD INDEX

ANNO .....	11	SKETCHDOCUMENTINFO .....	7	SKIMAGEOBJ .....	10
LOCALSKIMAGEOBJ .....	10	SKETCHIMAGEOBJ .....	7		

---

VARIABLE INDEX

IMAGEOBJGETFNS .....	7
----------------------	---

---