
Implementing Device-Independent Graphics Through Image Streams

The Interlisp-D system does all image creation through a set of functions and data structures for device-independent graphics, known popularly as DIG. DIG is achieved through the use of a special flavor of stream, known as an *image stream*.

An image stream, by convention, is any stream that has its IMAGEOPS field (described in detail below) set to a vector of meaningful graphical operations. Using image streams, you can write programs that draw and print on an output stream without regard to the underlying device, be it a window, a disk, or a Dover, 8044, or Diablo printer. For example, the following devices are supported by image streams: windows, Press streams, Interpress streams, and Iris streams.

Image Stream Structure

As indicated above, image streams use a field that no other stream uses: IMAGEOPS. IMAGEOPS is an instance of the IMAGEOPS data type and contains a vector of the stream's graphical methods. The methods contained in the IMAGEOPS can make arbitrary use of the stream's IMAGEDATA field, which is provided for their use, and may contain any data needed.

The IMAGEOPS data type has the following fields:

IMAGETYPE is the name of an image type. Monochrome display streams have an IMAGETYPE of DISPLAY; color display streams are identified as (COLOR DISPLAY). The IMAGETYPE is informational and can be set to anything you choose.

IMFONTCREATE is the device name to pass to FONTCREATE when fonts are created for the stream.

The following fields are all stream methods, and are presented with their arguments in the manner of a function definition. With the exception of IMCLOSEFN, each method has a corresponding function that consists of the method's name with the "IM" prefix removed. All coordinates that refer to points in a display device's space are measured in the device's units. (The IMSCALE method provides access to a device's scale.)

(IMCLOSEFN *Stream*) [Stream method]

is used before a stream is CLOSEFed, e.g., to flush buffers, write header or trailer information, etc.

(IMDRAWLINE *Stream X1 Y1 X2 Y2*
Width Operation Color) [Stream method]

draws a line of width *Width* from (*X1*, *Y1*) to (*X2*, *Y2*). (Dashing is currently handled at a higher level, and thus is not an argument.)

(IMDRAWCURVE *Stream Knots Closed Brush Dashing*) [Stream method]

draws a curve through *Knots*.

(IMDRAWCIRCLE *Stream CenterX CenterY Radius Brush Dashing*) [Stream method]

draws a circle of radius *Radius* around (*CenterX*, *CenterY*).

(IMDRAWELLIPSE *Stream CenterX CenterY SemiMinorRadius SemiMajorRadius Orientation Brush Dashing*) [Stream method]

draws an ellipse around (*CenterX*, *CenterY*).

(IMFILLCIRCLE *Stream CenterX CenterY Radius Texture*) [Stream method]

draws a circle filled with texture *Texture* around (*CenterX*, *CenterY*).

(IMBLTSHADE *Texture Stream DestinationLeft DestinationBottom Width Height Operation ClippingRegion*) [Stream method]

is the texture-source case of BITBLT. *DestinationLeft*, *DestinationBottom*, *Width*, *Height*, and *ClippingRegion* are measured in *Stream*'s units. This method is invoked by the functions BITBLT and BLTSHADE.

(IMBITBLT *SourceBitMap SourceLeft SourceBottom Stream DestinationLeft DestinationBottom Width Height SourceType Operation Texture ClippingRegion ClippedSource Left ClippedSourceBottom Scale*) [Stream method]

contains the bit-map-source cases of BITBLT. *SourceLeft*, *SourceBottom*, *ClippedSourceLeft*, *ClippedSourceBottom*, *Width*, and *Height* are measured in pixels; *DestinationLeft*, *DestinationBottom*, and *ClippingRegion* are in the units of the destination stream.

(IMSCALEDBITBLT *SourceBitMap SourceLeft SourceBottom Stream DestinationLeft DestinationBottom Width Height SourceType Operation Texture ClippingRegion ClippedSourceLeft ClippedSourceBottom Scale*) [Stream method]

is a scaled version of IMBITBLT. Each pixel in *SourceBitMap* is replicated *Scale* times in the X and Y directions; currently, *Scale* must be an integer.

(IMMOVETO *Stream X Y*) [Stream method]

moves to (*X*,*Y*). This method is invoked by the functions MOVETO and RELMOVETO. If IMMOVETO is not supplied, a default method composed of calls to the IMXPOSITION and IMYPOSITION methods is used.

(IMSTRINGWIDTH *Stream Str RdtBl*) [Stream method]

returns the width of string *Str* in *Stream*'s units, using *Stream*'s current font. If IMSTRINGWIDTH is not supplied, it defaults to calling \STRINGWIDTH.GENERIC.

(IMCHARWIDTH *Stream CharCode*) [Stream method]

returns the width of character *CharCode* in *Stream*'s units. If IMCHARWIDTH is not supplied, it defaults to calling \STRINGWIDTH.GENERIC.

The following methods all have corresponding DSPxx functions (e.g., IMYPOSITION corresponds to DSPYPOSITION) that invoke them. They also have the property of returning their previous value; when called with NIL they return the old value without changing it.

(IMXPOSITION *Stream XPosition*) [Stream method]

sets a new X-position on *Stream*.

(IMYPOSITION *Stream YPosition*) [Stream method]

sets a new Y-position on *Stream*.

(IMFONT *Stream Font*) [Stream method]

sets *Stream*'s font to be *Font*.

(IMLEFTMARGIN *Stream LeftMargin*) [Stream method]

sets *Stream*'s left margin to be *LeftMargin*. The left margin is defined as the X-position set after the new line.

(IMRIGHTMARGIN *Stream RightMargin*) [Stream method]

sets *Stream*'s right margin to be *RightMargin*. The right margin is defined as the maximum X-position at which characters are printed; printing beyond it causes a new line.

(IMLINEFEED *Stream Delta*) [Stream method]

sets *Stream*'s line feed distance (distance to move vertically after a new line) to be *Delta*.

(IMNEWPAGE *Stream*) [Stream method]

causes a new page to be started; the position is set to (*DspLeftMargin*, *DspTopMargin*). If not supplied, defaults to (\OUTCHAR *Stream* (CHARCODE ^L)).

(IMSCALE *Stream Scale*) [Stream method]

returns the number of device points per screen point (a screen point being ~1/72 inch). *Scale* is ignored.

(IMTOPMARGIN *Stream YPosition*) [Stream method]

Sets *Stream*'s top margin (the Y-position of the tops of characters that is set after a new page) to be *YPosition*.

(IMBOTTOMMARGIN *Stream YPosition*) [Stream method]

sets *Stream*'s bottom margin (the Y-position beyond which any printing causes a new page) to be *YPosition*.

(IMSPACEFACTOR *Stream Factor*) [Stream method]

sets the amount by which to multiply the natural width of all following space characters on *Stream*; is used for justification of text. The default value is one. For example, if the natural width of a space in *Stream*'s current font is 12 units, and the space factor is set to two, spaces appear 24 units wide. The values returned by STRINGWIDTH and CHARWIDTH are also affected.

(IMOPERATION *Stream Operation*) [Stream method]

sets the default BITBLT *Operation* argument. See the DSOPERATION and BITBLT documentation for more information.

(IMBACKCOLOR *Stream Color*) [Stream method]

sets the background color of *Stream*.

(IMCOLOR *Stream Color*) [Stream method]

sets the default color of *Stream*.

In addition to the IMAGEOPS-borne methods described above, there are two other important methods, which are contained in the stream itself.

STRMBOUTFN [Stream method]

is a function called by BOUT. You can install a STRMBOUTFN in a stream *Stream* using the form (replace (STREAM STRMBOUTFN) of STREAM with (FUNCTION MYBOUTFN)).

OUTCHARFN [Stream method]

is the function that is called to output a single byte. This is like STRMBOUTFN, except for being one level higher: it is intended for text output. Hence, this function should convert (CHARCODE EOL) into the stream's actual end-of-line sequence and should adjust the stream's CHARPOSITION appropriately before invoking the stream's STRMBOUTFN (by calling BOUT) to actually put the character. Defaults to \FILEOUTCHARFN, which is definitely *not* what you want. OUTCHARFNs are installed using a form like (replace (STREAM OUTCHARFN) of STREAM with (FUNCTION MYOUTCHARFN)).

IMAGEDATA [Record field]

is used to hold data pertaining to this type of image stream; the content is completely up to you. For Interpress devices, this is an instance of the data type INTERPRESSDATA; for Press, PRESSDATA; for the display, \DISPLAYDATA.

Creating Image Streams

(OPENIMAGESTREAM *File ImageType Options*) [Function]

opens and returns an output stream of type *ImageType* (PRESS, INTERPRESS, DISPLAY, or other types) on a destination specified by *File*. *File* can name a file either on a normal storage device or on a printer device. In the latter case, the file is sent to

the printer when the stream is closed. Because of the way defaulted arguments are interpreted, OPENIMAGESTREAM provides a convenient and standard interface for interpreting user output-destination specifications.

If *ImageType* is NIL, the image type is inferred from the extension field of *File* and the EXTENSIONS properties in the list PRINTFILETYPES. Thus, a PRESS extension denotes a Press-format stream, while IP, IPR, and INTERPRESS indicate Interpress format. If *File* has no extension but is a file on the printer device {LPT}, then *ImageType* is the type that the indicated printer can print. If *File* has no extension but is not on the printer device, then *ImageType* defaults to the type accepted by the first printer on DEFAULTPRINTINGHOST.

File = NIL is equivalent to *FILE* = {LPT}. Names for printer files are of the form {LPT}PRINTERNAME.TYPE, where PRINTERNAME, TYPE, or both may be omitted. PRINTERNAME is the name of the particular printer to which the file is transmitted on closing; it defaults to the first printer on DEFAULTPRINTINGHOST that can print *ImageType* files. As just described, the TYPE extension supplies the *ImageType* when it is defaulted. OPENIMAGESTREAM generates an error if the specified printer does not accept the kind of file specified by *ImageType*.

Examples assuming IP: is an Interpress printer, P is a Press printer, and DEFAULTPRINTINGHOST is (IP: P):

(OPENIMAGESTREAM) returns an Interpress image stream on printer IP:

(OPENIMAGESTREAM NIL 'PRESS) returns a Press stream on P

(OPENIMAGESTREAM '{LPT}.INTERPRESS) returns an Interpress stream on IP:

(OPENIMAGESTREAM '{CORE}FOO.PRESS) returns a Press stream on the file {CORE}FOO.PRESS

For completeness and consistency, if *ImageType* is inferred to be DISPLAY, then you are prompted for a window to open. The file name in this case is used as the title of the window.

Options is a list in property list format that may be used to specify certain attributes of the image stream; not all attributes are meaningful or interpreted by all types of streams. Among the properties are:

REGION, whose value is the region on the page (in stream scale units, 0,0 being the lower-left corner of the page) that text will fill up. It establishes the initial values for DSPLEFTMARGIN, DSPRIGHTMARGIN, DSPBOTTOMMARGIN, and DSPTOPMARGIN.

FONTS, whose value is a list of fonts that are expected to be used in the stream. Some streams (e.g., Interpress) are more efficient if the expected fonts are called out in advance, but this is not necessary. The first font in this list is the initial font of the stream; otherwise the DEFAULTFONT for that image type is used.

HEADING is the heading to be placed automatically on each page; NIL means no heading.

IMAGESTREAMTYPES [Association list]

describes how to create a stream for a given image type. Contains OPENSTREAM, FONTCREATE, and FONTSAVAILABLE methods. The main a-list is indexed by the image-stream type name (e.g., DISPLAY, PRESS, or INTERPRESS) to get another a-list that associates device-dependent functions with generic operation names.

The format of a single a-list entry is:

```
(IMAGETYPE
  (OPENSTREAM FunctionToOpenTheStream)
  (FONTCREATE FunctionToCreateAFontDescriptor)
  (FONTSAVAILABLE
    FunctionToReturnAvailableFonts))
```

For example, for Interpress, the a-list entry is:

```
(INTERPRESS
  (OPENSTREAM OPENIPSTREAM)
  (FONTCREATE \CREATEINTERPRESSFONT)
  (FONTSAVAILABLE \SEARCHINTERPRESSFONTS))
```

The OPENSTREAM function is called with the arguments:

(OPENSTREAMFN *File Options*)

File is the file name as it was passed to OPENIMAGESTREAM, and *Options* is the OPTIONS property list passed to OPENIMAGESTREAM. The result must be a stream of the appropriate image type.

The FONTCREATE function is called with the arguments:

(FONTCREATEFN *Family Size Face Rotation Device*)

Family is the family name for the font, e.g., MODERN. *Size* is the body size of the font, in printer's points. *Face* is a three-element list describing the weight, slope, and expansion of the face desired, e.g., (MEDIUM ITALIC EXPANDED). *Rotation* is how much the font is to be rotated from the normal orientation, in minutes of arc. For example, to print a landscape page, fonts have the rotation 5,400 (90 degrees). The function's result must be a FONTDESCRIPTOR with the fields filled in appropriately.

The FONTSAVAILABLE function is called with the arguments:

(FONTSAVAILABLEFN *Family Size Face Rotation Device*)

This function returns a list of all fonts agreeing with the *Family*, *Size*, *Face*, and *Rotation* arguments; any of them may be wild-carded (i.e., equal to '*', which means "any"). Each element of the list should be a quintuple of the form (*Family Size Face Rotation Device*).

Where the function looks is an implementation decision: the FONTSAVAILABLEFN for the display device looks at DISPLAYFONTDIRECTORIES, the Interpress code looks on INTERPRESSFONTDIRECTORIES, and implementors of new devices should feel free to introduce new search path variables.

Image Stream Predicates

(IMAGESTREAMP <i>X ImageType</i>)	[Function]
returns <i>X</i> (possibly coerced to a stream) if it is an output image stream of type <i>ImageType</i> (or of any type if <i>ImageType</i> is NIL), otherwise NIL.	
(IMAGESTREAMTYPE <i>Stream</i>)	[Function]
returns the image type of <i>Stream</i> .	
(IMAGESTREAMTYPEP <i>Stream Type</i>)	[Function]
returns T if <i>Stream</i> is an image stream of type <i>Type</i> .	

Creating Your Own Flavor of Image Stream

In describing a task as complex as building a new flavor of image stream, no document can contain all of the answers, tricks, and shortcuts. There is no substitute for studying a working implementation in doing your own. Therefore, we recommend you look at the FX80STREAM package as an example of how to create a new imaging device. FX80STREAM is a DIG interface for the Epson FX-80 printer a device simple enough to drive that its details do not obscure the fundamentals of how its image stream works.