
POSTSCRIPTSTREAM

By: Matt Heffron (mheffron@orion.cf.uci.edu)

INTRODUCTION

The PostScript package defines a set of imageops for printers which understand the PostScript page description language by Adobe. At Beckman we have successfully used TEdit, Sketch, LISTFILES, and HARDCOPYW to an Apple LaserWriter and an AST TurboLaser PS. The PostScript imagestream driver installs itself when it is loaded. All symbols in the PostScript driver are located in the INTERLISP: package.

VARIABLES

POSTSCRIPT.FONT.ALIST [InitVariable]

POSTSCRIPT.FONT.ALIST is an ALIST mapping Xerox Lisp font names into the root names of PostScript font files. It is also used for font family coercions. The default value should be acceptable for any of the fonts which are built into the Apple Laserwriter.

POSTSCRIPTFONTDIRECTORIES [InitVariable]

POSTSCRIPTFONTDIRECTORIES is the list of directories where the PostScript .PSCFONT font files can be found. The default value is: ("{DSK}/usr/local/lde/fonts/postscript/") on a Sun or IBM workstation and ("{DSK}<LISPFILES>FONTS>PSC>") for other cases .

POSTSCRIPT.DEFAULT.PAGEREGION [InitVariable]

POSTSCRIPT.DEFAULT.PAGEREGION indicates the area of the page to use for text file listings (i.e. LISTFILES). It is in units of 100'ths of points. The default value is: (4800 4800 52800 70800), which gives left and bottom margins of 0.75 inch and top and right margins of 0.5 inch on 8.5 x 11 paper.

POSTSCRIPT.PAGEREGIONS [InitVariable]

POSTSCRIPT.PAGEREGIONS is an ALIST mapping pagetypes into paper size and actual imageable area on the page. By default, it knows about LETTER, LEGAL, and NOTE pagetypes, and the corresponding sizes and imageable areas for the Apple Laserwriter. Others can be defined by the user by adding the appropriate entries onto this ALIST.

POSTSCRIPT.PAGETYPE [InitVariable]

POSTSCRIPT.PAGETYPE is used by OPENIMAGESTREAM to lookup the paper size and actual imageable area of the page in POSTSCRIPT.PAGEREGIONS to determine the initial margins. This value can be overridden with the PAGETYPE or PAPERTYPE options in the OPENIMAGESTREAM call. The name of the type of page selected is NOT passed through to the PostScript output.

\POSTSCRIPT.MAX.WILD.FONTSIZE [InitVariable]

\POSTSCRIPT.MAX.WILD.FONTSIZE indicates the maximum point size that should be returned from FONTSAVAILABLE when the SIZE argument is wild (i.e. *). All integer point sizes from 1 to \POSTSCRIPT.MAX.WILD.FONTSIZE will be indicated as available. The default value is: 72.

POSTSCRIPT.PREFER.LANDSCAPE [InitVariable]

POSTSCRIPT.PREFER.LANDSCAPE indicates if the OPENIMAGESTREAM method should default the orientation of output files to LANDSCAPE. It can have one of three values: NIL, T, or ASK. NIL means prefer portrait orientation output, T means prefer landscape, and ASK says to bring up a menu to ask the preferred orientation if it wasn't explicitly indicated in the OPENIMAGESTREAM call (with the ROTATION option). The default value is: NIL. An item (PS Orientation) is added to the Background Menu to let you change the value of this variable.

POSTSCRIPT.TEXTFILE.LANDSCAPE [InitVariable]

POSTSCRIPT.TEXTFILE.LANDSCAPE indicates if the printing of TEXT files (e.g. LISTFILES, ...) should force the orientation of output files to LANDSCAPE. When it is non-NIL the orientation of output files is forced to LANDSCAPE. (There is no ASK option here.) The default value is: NIL.

POSTSCRIPT.BITMAP.SCALE [InitVariable]

POSTSCRIPT.BITMAP.SCALE specifies an independent scale factor for display of bitmap images (e.g. window hardcopies). Values less than 1 will reduce the image size. (I.e. a value of 0.5 will give a half size bitmap image.) The position of the scaled bitmap will still have the SAME lower-left corner (i.e. the scaled bitmap is not centered in the region of the full size bitmap image). The default value is: 1.

HINT

Setting POSTSCRIPT.BITMAP.SCALE to 0.96, instead of 1, will give cleaner BITMAP images on a 300 dpi printer. (This corrects for the 72 ppi imagestream vs. the 75 dpi printer, using 4x4 device dots per bitmap pixel.) Also, values of 0.24, 0.48 and 0.72, instead of 0.25, 0.5 and 0.75, will also give cleaner images for reduced size output. In general, use integer multiples of 0.24 for a 300 dpi printer.

POSTSCRIPT.TEXTURE.SCALE [InitVariable]

POSTSCRIPT.TEXTURE.SCALE specifies an independent scale for the display of bitmap textures. The value represents the number of device space units per texture unit (bitmap bit). The default value is 4, which represents each bit of the texture as a 4x4 block, so that textures are approximately the same resolution as on the screen (for 300 dpi output devices, such as the Apple Laserwriter).

The PostScript package extends the allowed representations of a texture, beyond 16-bit FIXP and 16x16 bitmap, to ANY square bitmap. (If the bitmap is not square, its longer edge is truncated from the top or right to make it square.) Use this feature with caution, as large bitmap textures, or sizes other

than multiples of 16 bits square, require large amounts of storage in the PostScript interpreter (in the printer controller), and can cause limitcheck errors when actually printing.

Anywhere that a texture or color can be used on an imagestream or in the specification of a BRUSH, you can instead give a FLOATP between 0.0 and 1.0 (inclusive) to represent a PostScript halftone gray shade. (0.0 is black and 1.0 is white. Specifically, the value sets the brightness of the shade.) The value you specify will not be range checked, and will be passed directly through to the PostScript setgray operator. (E.g. you can pass 0.33 as the color to DRAWLINE to get a dark gray line with approximately 67% of the pixels in the line black.)

POSTSCRIPT.IMAGESIZEFACTOR

[InitVariable]

POSTSCRIPT.IMAGESIZEFACTOR specifies an independent factor to change the overall size of the printed image. This re-sizing affects the entire printed output (specifically, it superimposes its effects upon those of POSTSCRIPT.BITMAP.SCALE and POSTSCRIPT.TEXTURE.SCALE). Values greater than 1 enlarge the printed image, and values less than 1 reduce it. An invalid POSTSCRIPT.IMAGESIZEFACTOR (i.e. not a positive, non-zero number) will use a value of 1. The BITMAPSCALE function for the POSTSCRIPT printer type does NOT consider the POSTSCRIPT.IMAGESIZEFACTOR when determining the scale factor for a bitmap.

MISCELLANEOUS

The SCALE of a PostScript imagestream is 100. This is to allow enough resolution in the width information for fonts to enable TEdit to correctly fill and justify text.

The first time any PostScript imagestream is created (even if only to hardcopy a bitmap or window) the DEFAULTFONT is instantiated (unless a FONTS option was given to the OPENIMAGESTREAM, in which case the initial font for the imagestream will be set to that font, or to the CAR if a list).

The PostScript imagestream method for FILLPOLYGON uses the global variable FILL.WRULE as the default value for the WINDINGNUMBER argument. (This is the same variable which is used by the DISPLAY imagestream method for FILLPOLYGON.)

The PostScript imagestream method for OPENIMAGESTREAM (and, therefore, SEND.FILE.TO.PRINTER), supports an IMAGESIZEFACTOR option to change the size of the printed image. The IMAGESIZEFACTOR re-sizing is combined with the POSTSCRIPT.IMAGESIZEFACTOR to produce an overall re-sizing of the printed image. A HEADING option is also supported to give a running header on each page of output. The value of the HEADING option is printed at the top left of the page, followed by "Page " and the appropriate page number. They are printed in the DEFAULTFONT (unless a FONTS option was given to the OPENIMAGESTREAM, in which case it will be that font, or to the CAR if a list).

The PostScript package is contained in the files: POSTSCRIPTSTREAM.LCOM & PS-SEND.LCOM, with the source in the files: POSTSCRIPTSTREAM & PS-SEND. The module PS-SEND.LCOM is required and will be loaded automatically when POSTSCRIPTSTREAM.LCOM is loaded. It contains the function which is called by SEND.FILE.TO.PRINTER to actually transmit the file to the printer. It is, by its nature, quite site specific, so it is in a separate file to make modifying it for any site relatively

simple. System record declarations required to compile POSTSCRIPTSTREAM can be found in EXPORTS.ALL.

I'm pretty sure that the output generated by the PostScript imageops fully conforms to the Adobe Systems Document Structuring Conventions, Version 2.0, January 31, 1987.

Including Other PostScript Operations

If you wish to insert your own specific PostScript operations into a PostScript imagestream, you can do so with the following functions:

(POSTSCRIPT.OUTSTR *STREAM STRING*) [Function]

POSTSCRIPT.OUTSTR outputs a string or value to the imagestream. *STREAM* must be an open PostScript imagestream. *STRING* is the value to output (STRINGP and LITATOM are most efficient, but any value can be output (its PRIN1 pname is used)).

(POSTSCRIPT.PUTCOMMAND *STREAM STRING ... STRING*) [NoSpread Function]

POSTSCRIPT.PUTCOMMAND is more general for sequences of commands and values. It calls POSTSCRIPT.OUTSTR repeatedly to output each of the *STRING* arguments to *STREAM*.

(\POSTSCRIPT.OUTCHARFN *STREAM CHAR*) [Function]

\POSTSCRIPT.OUTCHARFN is used to output the characters forming the text of a PostScript string (e.g. the argument to a show or charpath operator). *STREAM* is the open PostScript imagestream to output to, and *CHAR* is the CHARCODE of the character to output. The / (slash), (and) (parenthesis) characters will be quoted with /, and characters with ASCII values less than 32 (space) or greater than 126 (tilde) will be output as *nnn* (in octal). \POSTSCRIPT.OUTCHARFN will output the (character to open the string, if necessary. Use POSTSCRIPT.CLOSESTRING (below) to close the string.

(POSTSCRIPT.CLOSESTRING *STREAM*) [Function]

POSTSCRIPT.CLOSESTRING closes a PostScript string (e.g. the argument to a show or charpath operator). *STREAM* is the open PostScript imagestream. It is important to use POSTSCRIPT.CLOSESTRING to output the) character to close the string, because it also clears the stream state flag that indicates that a string is in progress (otherwise, the next POSTSCRIPT.PUTCOMMAND would output the commands to close the string and show it).

Warning

Do not attempt to create a PostScript font larger than about 600 points, as much of Interlisp's font information is stored in SMALLP integers, and too large a font would overflow the font's height, or the width for any of the wider characters. (I know that 600 points is a ridiculously large limit (about 8.3 inches), but I thought I'd better mention it, or someone might try it!)

Changes from the Initial Medley Release

This second Medley release of the PostScript imagestream driver includes some performance enhancements when writing bitmaps to the output, some SUN-specific code (from Will Snow of envos), implementation of the SCALEDIBITBLT, DSPROTATE, and DSPTRANSLATE operations, and a lot of performance enhancements (many thanks to Tom Lipkis of Savoir).

Changes from the Lyric Release

The Medley release of this PostScript imagestream driver changed the default value of POSTSCRIPT.TEXTFILE.LANDSCAPE from T to NIL. It also added the support for the HEADING option.

Known Problems/Limitations

The output generated for a PostScript imagestream is rather brute force. It isn't particularly careful to generate the smallest output file for a given sequence of operations. Specifically, it often generates extra end-of-lines between PostScript operator sequences (this has no effect on the printed output, only on the file size).

Using BITMAPs or Functions as BRUSH arguments to the curve drawing functions is not supported, nor is using a non-ROUND BRUSH with DRAWCIRCLE or DRAWELLIPSE.

The implementation of DSPROTATE accepts ROTATION argument values of 0 and 90 (any non-NIL, non-zero value is converted to 90). A value of 0 converts the page orientation to Portrait, and 90 converts the page orientation to Landscape. These conversions perform the translations necessary to keep the clipping region on the page. (This may or may not be the right thing to do, but since DSPROTATE is undocumented in what it should do, this is what the PostScript driver does).

There is no support for NS character sets other than 0, and there is no translation of the character code values from NS encoding to PostScript encoding.

There is no support for color.

\POSTSCRIPT.OUTCHARFN is pretty wimpy in its handling of TAB characters. It just moves to the next multiple of (eight times the average character width of the current font) from the current left margin.

I haven't yet documented how to build the .PSCFONT files from .AFM files for new fonts that become available.