

**NAME**

hpgl2exp, hpgl2mcd, xhpgl, hpgl2psc, hpgl2ras, hpgl2xpm, hpgl2svg, replot, dashpens – emulate HPGL for output to other devices

**SYNOPSIS**

**hpgl2exp** [-t] [-f[vfont]] [-A | -B | -A3 | -A4] [file] ...  
**hpgl2mcd** [-t] [-f[vfont]] [-A | -B | -A3 | -A4] [-o] [file] ...  
**xhpgl** [-t] [-f[vfont]] [-A | -B | -A3 | -A4] [file] ...  
**hpgl2psc** [-t] [-f[vfont]] [-A | -B | -A3 | -A4] [-r] [-p] [-m] [file] ...  
**hpgl2ras** [-t] [-f[vfont]] [-A | -B | -A3 | -A4] [-xln] [-yln] [-xhn] [-yhn] [file] ...  
**hpgl2xpm** [-t] [-f[vfont]] [-A | -B | -A3 | -A4] [-xln] [-yln] [-xhn] [-yhn] [-bgcolour] [-pcolour,colour,...] [file] ...  
**hpgl2svg** [-t] [-f[vfont]] [-A | -B | -A3 | -A4] [-s] [-lseglimit] [-h height] [-w width] [file] ...  
**replot** [-t] [-f[vfont]] [-A | -B | -A3 | -A4] [-s] [-lseglimit] [file] ...  
**dashpens** [-t] [-f[vfont]] [-A | -B | -A3 | -A4] [-s] [-lseglimit]

**DESCRIPTION**

Each of these programs interprets the HPGL instructions stored in each *file* argument, or the standard input if no files are specified, and generates output as appropriate for a particular device. The programs emulate only a small subset of the full HPGL, which is adequate for handling output from programs like *analysis*(1) and *layout*(1).

The output graphics are of a size and aspect ratio equivalent to what would be produced by an HP plotter on A-size (8.5×11") paper. The page size options, **-A**, **-B**, **-A3**, and **-A4**, adjust the coordinate system and aspect ratio to correspond to the plotter's defaults for that page size. The default page size can also be changed by setting the environment variable **HPGLPAGE** to one of the above page sizes.

The **-t** option causes text output to be suppressed. The **-f** option causes text to be rendered as a series of line segments, using the specified *vfont* to select a variable Hershey font, or by default, a font approximating the default font on HP plotters. The default font used can be changed by setting the environment variable **VFONT** to the name of any of the variable size fonts in the directory **/usr/neuro/lib/font** (the default is "roman").

Pen selections are interpreted differently by the different programs; they may result in colour selections, or line thickness selections. In all cases, selecting pen number **0** suppresses the generation of the following text or graphics. Input pen number selections can be mapped to alternate selections by setting the environment variable **HPGLPENS** to a series of single-digit pen numbers, corresponding to the desired pen selection for each input pen number. For example, setting **HPGLPENS** to **01122222** would cause everything drawn with pen 1 selected to be suppressed, pens 2 and 3 would be mapped to pen 1, and pens 4 to 8 would be mapped to pen 2.

*Hpgl2exp* generates Express line drawing commands on its standard output, to be sent to a Unisys Model 37 laser printer, using letter size paper. Line types (LT commands) are ignored; solid lines are always generated. Pen selections affect line thickness: an "SP*n*;" instruction results in a line thickness of roughly *n* tenths of a millimeter.

*Hpgl2mcd* generates its output on a graphics terminal, to allow previewing. A subset of line types is supported. Pen selections affect drawing colour, according to the environment variable **SCRPENS**, a comma-separated list of CNS names corresponding to each pen's colour. The **-o** option causes lines drawn to be blended in with the existing screen colours, using a logical OR operation, rather than drawn over the existing colours. The page commands (PG, AF, and AH) and not-ready command (NR) cause the program to wait for a key to be pressed. You have the option of having the screen cleared before continuing.

*Hpgl2psc* generates PostScript drawing commands on its standard output. A subset of line types is supported. Pen selections affect the shade of gray used for line drawing, with pen 1 being black, and progressing to lighter shades as the pen number increases. With the **-p** option, pen changes are suppressed, and all lines are drawn in black. Line thickness is fixed at 0.3 mm. Normally, output is in landscape orientation on a letter sized page. If a page size of **B** or **A3** is specified, the image is reduced to 65% of its linear size, to make the large page image fit on a letter sized page. With the **-r** option, the image is reduced to 72% of its linear size, and rotated so that it appears in portrait orientation on the upper part of a letter sized page. Reductions are cumulative, if both apply. The output is encapsulated PostScript (EPS), with the bounding box adjusted to fit the hard-clip limits of an HPGL page, and reduced accordingly if the **-r** option is used. Text in the PostScript file will not line up exactly as it did in the HPGL file, but it should come close. Use the **-f** option if you need a more exact match, which will render the text as vectors using a Hershey font. Many HPGL text label operations are not supported in PostScript text, such as chord, rotation and slant angles, and extra character spacing. The **-f** can be used to provide an alternate Hershey *vfont* name if the given name is not capitalized, or an alternate PostScript font name for true text labels if the name is capitalized (as standard PostScript font names are). Note that this is a new feature, as of 2009, as older versions always vectorized text using *vfonts*. The **-m** option merges together multiple unconnected line segments into a single path, rather than stroking each one separately. This can help when importing EPS output into *Corel Draw*, with the interpreted PostScript input filter, as it reduces the number of objects it must deal with.

*Hpgl2ras* generates a one-bit-per-pixel raster on its standard output, in the *gpsras*(5) format. This raster can then be converted for any device that accepts bit-mapped graphics. The coordinate bounds for the raster can be set with the options **-xln**, **-yln**, **-xhn**, and **-yhn**, which set the low and high bounds for x and y, respectively. These bounds determine the maximum output raster size; the size may be reduced automatically along one axis to adjust to the aspect ratio of an A-size page, or the selected page size. The default bounds of (0,0) and (1015,783) result in a raster that will be 1/10th the resolution, and roughly the size of a graph produced by an HP plotter on A-size paper, if the raster is printed at 100 dots per inch. A subset of line types is supported. Pen selections affect line thickness: an "SP*n*;" instruction results in a line thickness of *n* dots (or pixels). Page commands are all ignored.

*Hpgl2xpm* generates an X Window colour pixel map (XPM) on its standard output. The XPM format is usable by other X Window applications, and can easily be converted to a number of common image file formats using an application such as the *xv* image viewer, or *xpmtoppm*(1) and other utilities in the **netpbm** package. The coordinate bounds for the raster can be set with the options **-xln**, **-yln**, **-xhn**, and **-yhn**, which set the low and high bounds for x and y, respectively. These bounds determine the maximum output raster size; the size may be reduced automatically along one axis to adjust to the aspect ratio of an A-size page, or the selected page size. However, unlike *hpgl2ras* if you specify both X and Y bounds, no automatic adjustment is done, so you can alter the aspect ratio. The default bounds of (0,0) and (1015,783) result in a raster that will be 1/10th the resolution, and roughly the size of a graph produced by an HP plotter on A-size paper, if the raster is printed at 100 dots per inch. A subset of line types is supported, and lines will be 1 pixel thick. Pen selections affect line colour, with the colours being taken from the **SCRPENS** environment variable by default. Pen colours can be overridden with the **-pcolour,colour,...** option, where you can specify up to eight colours using standard X11 colour names or hex codes. The background colour, white by default, can be overridden with the **-bgcolour** option. Page commands are all ignored.

*Hpgl2svg* generates an SVG (Scalable Vector Graphics) drawing file on its standard output. SVG is an emerging standard for vector graphics which can be resized, and can be imported into drawing programs such as *Corel Draw*. A subset of line types is supported. Pen selections affect line colour, with the colours being taken from the **SCRPENS** environment variable, and are used as-is in the SVG file, so these should be names that are in SVG's recognized list (itemized here: <http://www.w3.org/TR/SVG/types.html#ColorKeywords> ). Page commands are all ignored. You can use **-h height** to set the rendered SVG drawing height in pixels (default is 784), and **-w width** to set the width in pixels (default is 1008). If only one of **-h** or **-w** is given, the other is set proportionally. HPGL is preprocessed by the *replot* program to simplify it before converting, and all arguments to

*hpgl2svg*, except `-h` and `-w`, are passed directly to *replot*. Text in the SVG file will not line up exactly as it did in the HPGL file, but it should come close. Use the `-f` option if you need a more exact match, which will render the text as vectors using a Hershey font. Many HPGL text label operations are not supported in SVG text, such as chord, rotation and slant angles, and extra character spacing.

*Replot* regenerates simplified HPGL instructions on its standard output. The resulting output will not contain any absolute positioning, rotation, or sizing commands, and can thus be easily resized or rotated, by using *layout*(1) for example. By default, text labels will be regenerated using relative size and direction commands, with direction adjusted to compensate for any rotation in effect. If the `-f` or `-t` options are specified, the output will contain no text generating commands, as these will have either been converted to line drawing commands to render the Hershey font, or suppressed altogether.

*Replot* is useful as an import/export filter for HPGL files. If an HPGL file from some program cannot be used in a *layout* panel, run it through *replot* to produce a simplified, resizable version of the graph, which should then be usable. Similarly, you can process HPGL files output by *layout*, to then use them in panels of another layout file. By using the `-s` option to *replot*, you can convert HPGL files to a form that can be imported into *Corel Draw*. This option causes the scale (SC) and label size (SR) commands to be stripped out, and breaks up runs of over 2000 connected line segments. To change the maximum number of connected segments from the limit of 2000 which is chosen by default, use the `-lseglimit` option to specify the desired segment limit. With or without these options, *replot* also keeps output lines at reasonable lengths, since this matters for some programs.

*Dashpens* is a simple front-end to the *replot* program, and is used to prepare multi-colour (i.e. multi-pen) HPGL files for single-colour publication. It converts pen selections (SP) into line-type selections (LT) according to your specifications. It begins by showing you a menu of line types, numbered from 0 to 6, and asks you, for each of the 8 pen numbers, which line type to use (the default is **0**, for solid lines). It then asks for the input and output file names, and finally runs *replot* on the input file, modifies the output, and stores it in the output file you specified. Any option arguments given are passed on to *replot*.

## X WINDOW SUPPORT

When the X Window version of *hpgl2mcd* (also known as *xhpgl*) is run on an X Window terminal, a new window will be shown for displaying the output. Unless the input was redirected from a file, it be taken from the keyboard when this window is the "input focus", i.e. the active window. As for most other X Window programs in this package, the usual X command line options are accepted. The `-o` option is not implemented in the X Window version of *hpgl2mcd*.

## FILES

/usr/neuro/lib/font/\*.vfb   binary Hershey fonts  
/dev/tty                   controlling tty, used by *hpgl2mcd*

## SEE ALSO

*layout*(1), *analysis*(1), *hpgl2gif*(1), *gpsras*(5), *xpmtoppm*(1)