Pnmtops User Manual

pnmtops

Updated: 10 February 2006
Table Of Contents

NAME

pnmtops - convert PNM image to PostScript

SYNOPSIS

pnmtops [**-scale=***s*] [**-dpi=***N*[**x***N*]] [**-imagewidth=***n*] [**-imageheight=***n*] [**-width=***N*] [**-height=***N*] [**-equalpixels**] [**-turn**|**-noturn**] [**-rle**|**-runlength**] [**-flate**] [**-ascii85**] [**-nocenter**] [**-nosetpage**] [**-level=***N*] [**-psfilter**] [**-noshowpage**] [*pnmfile*]

All options can be abbreviated to their shortest unique prefix. You may use two hyphens instead of one. You may separate an option name and its value with white space instead of an equals sign.

DESCRIPTION

This program is part of Netpbm.

pnmtops reads a Netpbm image stream as input and produces Encapsulated PostScript (EPSF) as output.

If the input file is in color (PPM), **pnmtops** generates a color PostScript file. Some PostScript interpreters can't handle color PostScript. If you have one of these you will need to run your image through **ppmtopgm** first.

If you specify no output dimensioning options, the output image is dimensioned as if you had specified **-scale=1.0**, which means aproximately 72 pixels of the input image generate one inch of output (if that fits the page).

Use **-imagewidth**, **-imageheight**, **-equalpixels**, **-width**, **-height**, and **-scale** to adjust that.

Each image in the input stream becomes one complete one-page Postscript program in the output. (This may not be the best way to create a multi-page Postscript stream; someone who knows Postscript should work on this).

The line at the top of the file produced by **pnmtops** is either "%!PS-Adobe-3.0 EPSF-3.0" or just "%!PS-Adobe-3.0". The numbers do not reflect the Postscript language level, but the version of the DSC comment specification and EPS specification implmented. The Postscript language level is in the "%%LanguageLevel:" comment. **pnmtops** omits "EPSF-3.0" if you specify **-setpage**, because it is incorrect to claim EPS compliance if the file uses **setpagedevice**.

What is Encapsulated Postscript?

Encapsulated Postscript (EPSF) is a subset of Postscript (i.e. the set of streams that conform to EPSF is a subset of those that conform to Postscript). It is designed so that an EPSF stream can be embedded in another Postscript stream. A typical reason to do that is where an EPSF stream describes a picture you want in a larger document.

An Encapsulated Postscript document conforms to the DSC (Document Structuring Convention). The DSC defines some Postscript comments (they're comments from a Postscript point of view, but have semantic value from a DSC point of view).

More information about Encapsulated Postscript is at http://www.cs.indiana.edu/docproject/programming/postscript/eps.html.

Many of the ideas in **pnmtops** come from Dirk Krause's **bmeps**. See SEE ALSO

OPTIONS

-imagewidth, **-imageheight**Tells how wide and high you want the image on the page, in inches. The aspect ratio of the image is preserved, so if you specify both of these, the image on the page will be the largest image that will fit within the box of those dimensions.

If these dimensions are greater than the page size, you get Postscript output that runs off the page.

You cannot use **imagewidth** or **imageheight** with **-scale** or **-equalpixels**.

-equalpixels This option causes the output image to have the same number of pixels as the input image. So if the output device is 600 dpi and your image is 3000 pixels wide, the output image would be 5 inches wide.

You cannot use **-equalpixels** with **-imagewidth**, **-imageheight**, or **-scale**.

-scale tells how big you want the image on the page. The value is the number of inches of output image that you want 72 pixels of the input to generate.

But **pnmtops** rounds the number to something that is an integral number of output device pixels. E.g. if the output device is 300 dpi and you specify **-scale=1.0**, then 75 (not 72) pixels of input becomes one inch of output (4 output pixels for each input pixel). Note that the **-dpi** option tells **pnmtops** how many pixels per inch the output device generates.

If the size so specified does not fit on the page (as measured either by the **-width** and **-height** options or the default page size of 8.5 inches by 11 inches), **pnmtops** ignores the **-scale** option, issues a warning, and scales the image to fit on the page.

-dpi=*N*[**x***N*]

This option specifies the dots per inch resolution of your output device. The default is 300 dpi. In theory PostScript is device-independent and you don't have to worry about this, but in practice its raster rendering can have unsightly bands if the device pixels and the image pixels aren't in sync.

Also this option is crucial to the working of the **equalpixels** option.

If you specify *N***x**N, the first number is the horizontal resolution and the second number is the vertical resolution. If you specify just a single number *N*, that is the resolution in both directions.

-width, **-height** These options specify the dimensions, in inches, of the page on which the output is to be printed. This can affect the size of the output image.

The page size has no effect, however, when you specify the **-imagewidth**, **-imageheight**, or **-equalpixels** options.

These options may also affect positioning of the image on the page and even the paper selected (or cut) by the printer/plotter when the output is printed. See the **-nosetpage** option.

The default is 8.5 inches by 11 inches.

-turn

-noturn These options control whether the image gets turned 90 degrees. Normally, if an image fits the page better when turned (e.g. the image is wider than it is tall, but the page is taller than it is wide), it gets turned automatically to better fit the page. If you specify the **-turn** option, **pnmtops** turns the image no matter what its shape; If you specify **-noturn**, **pnmtops** does not turn it no matter what its shape.

-rle

-runlength These identical options tell **pnmtops** to use run length compression in encoding the image in the Postscript program. This may save time if the host-to-printer link is slow; but normally the printer's processing time dominates, so **-rle** has no effect (and in the absence of buffering, may make things slower.

This may, however, make the Postscript program considerable smaller.

This usually doesn't help at all with a color image and **-psfilter**, because in that case, the Postscript program **pnmtops** creates has the red, green, and blue values for each pixel together, which means you would see long runs of identical bytes only in the unlikely event that the red, green, and blue values for a bunch of adjacent pixels are all the same. But without **-psfilter**, the Postscript program has all the red values, then all the green values, then all the blue values, so long runs appear wherever there are long stretches of the same color.

-flate This option tells **pnmtops** to use "flate" compression (i.e. compression via the "Z" library -- the same as PNG).

See the **-rle** option for information about compression in general.

You must specify **-psfilter** if you specify **-flate**.

This option was new in Netbpm 10.27 (March 2005).

Before Netpbm 10.32 (February 2005), you could not specify **-rle** and **-flate** together.

-ascii85 By default, **pnmtops** uses "asciihex" encoding of the image raster. The image raster is a stream of bits, while a Postscript program is text, so there has to be an encoding from bits to text. Asciihex encoding is just the common hexadecimal representation of bits. E.g. 8 1 bits would be encoded as the two characters "FF".

With the **-ascii85** option, **pnmtops** uses "ascii85" encoding instead. This is an encoding in which 32 bits are encoded into five characters of text. Thus, it produces less text for the same raster than asciihex. But ascii85 is not available in Postscript Level 1, whereas asciihex is.

This option was new in Netbpm 10.27 (March 2005).

-psfilterpnmtops can generate two different kinds of Encapsulated Postscript programs to represent an image. By default, it generates a program that redefines **readstring** in a custom manner and doesn't rely on any built-in Postscript filters. But with the **-psfilter** option, **pnmtops** leaves **readstring** alone and uses the built-in Postscript filters **/ASCII85Decode**, **/ASCIIHexDecode**, **/RunLengthDecode**, and **/FlateDecode**.

This option was new in Netbpm 10.27 (March 2005). Before that, **pnmtops** always used the custom **readstring**.

The custom code can't do flate or ascii85 encoding, so you must use **-psfilter** if you want those (see **-flate**, **-ascii85**).

-level This option determines the level (version number) of Postscript that **pnmtops** uses. By default, **pnmtops** uses Level 2. Some features of **pnmtops** are available only in higher Postscript levels, so if you specify too low a level for your image and your options, **pnmtops** fails. For example, **pnmtops** cannot do a color image in Level 1.

This option was new in Netpbm 10.27 (March 2005). Before that, **pnmtops** always used Level 2.

-dict This causes the Postscript program create a separated dictionary for its local variables and remove it from the stack as it exits.

This option was new in Netbpm 10.27 (March 2005).

-vmreclaim This option causes the Postscript program to force a memory garbage collection as it exits.

This option was new in Netbpm 10.27 (March 2005).

-nocenter By default, **pnmtops** centers the image on the output page. You can cause **pnmtops** to instead put the image against the lower left corner of the page with the **-nocenter** option. This is useful for programs which can include PostScript files, but can't cope with pictures which are not positioned in the lower left corner.

For backward compatibility, **pnmtops** accepts the option **-center**, but it has no effect.

-setpage This causes **pnmtops** to include a "setpagedevice" directive in the output. This causes the output to violate specifications of EPSF encapsulated Postscript, but if you're not using it in an

encapsulated way, may be what you need. The directive tells the printer/plotter what size paper to use (or cut). The dimensions it specifies on this directive are those selected by the **-width** and **-height** options or defaulted.

From January through May 2002, the default was to include "setpagedevice" and this option did not exist. Before January 2002, there was no way to include "setpagedevice" and neither the **-setpage** nor **-nosetpage** option existed.

-nosetpage This tells **pnmtops** not to include a "setpagedevice" directive in the output. This is the default, so the option has no effect.

See the **-setpage** option for the history of this option.

-noshowpage This tells **pnmtops** not to include a "showpage" directive in the output. By default, **pnmtops** includes a "showpage" at the end of the EPSF program According to EPSF specs, this is OK, and the program that includes the EPSF is supposed to redefine showpage so this doesn't cause undesirable behavior. But it's often easier just not to have the showpage.

This options was new in Netpbm 10.27 (March 2005). Earlier versions of **pnmtops** always include the showpage.

-showpage This tells **pnmtops** to include a "showpage" directive at the end of the EPSF output. This is the default, so the option has no effect.

This option was new in Netpbm 10.27 (March 2005).

LIMITATIONS

If the PNM image has a maxval greater than 255, **pnmtops** will probably produce incorrect output unless you use the **-psfilter** option or use the **-level** option to demand Postscript Level 1.

SEE ALSO

Postscript is described in the Postscript Language Reference Manual.

bmeps converts from Netpbm and other formats to Encapsulated Postscript. It is suitable for hooking up to **dvips** so you can include an image in a Latex document just with an includegraphics directive. **bmeps** has a few functions **pnmtops** does not, such as the ability to include a transparency mask in the Postscript program (but not from PAM input -- only from PNG input).

pnm, **gs**, **psidtopgm**, **pstopnm**, **pbmtolps**, **pbmtoepsi**, **pbmtopsg3**, **ppmtopgm**,

AUTHOR

Copyright (C) 1989, 1991 by Jef Poskanzer.

Modified November 1993 by Wolfgang Stuerzlinger, wrzl@gup.uni-linz.ac.at

---

Table Of Contents

NAME
SYNOPSIS
DESCRIPTION
OPTIONS
LIMITATIONS
SEE ALSO
AUTHOR