

# dvipng

---

A DVI-to-PNG Translator  
Version 1.17

by Jan-Åke Larsson.

---

Copyright © 2002-2020 Jan-Åke Larsson

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided also that the section entitled “Copying” is included exactly as in the original, and provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the Free Software Foundation.

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>1</b>
<b>2</b>	<b>Installation .....</b>	<b>2</b>
2.1	Prerequisites .....	2
2.2	Configure.....	2
2.3	Build/install.....	3
2.4	Installation outside the texmf tree .....	3
2.5	Installation for non-privileged users .....	3
<b>3</b>	<b>Basic usage of dvipng .....</b>	<b>5</b>
<b>4</b>	<b>Command-line options .....</b>	<b>6</b>
4.1	Option summary .....	6
4.2	Option details .....	7
<b>5</b>	<b>Graphics .....</b>	<b>13</b>
5.1	Encapsulated PostScript.....	13
5.2	Bitmapped graphics .....	13
5.3	Raw PostScript.....	14
<b>6</b>	<b>Color .....</b>	<b>15</b>
6.1	Color specifications.....	15
6.2	Color specials .....	15
<b>7</b>	<b>Diagnosing problems.....</b>	<b>16</b>
7.1	Contact information.....	16
7.2	Debug options.....	16
<b>8</b>	<b>Credits .....</b>	<b>17</b>
<b>9</b>	<b>Copying.....</b>	<b>18</b>
	<b>Index.....</b>	<b>19</b>

# 1 Introduction

This program makes PNG and/or GIF graphics from DVI files as obtained from  $\text{\TeX}$  and its relatives.

If GIF support is enabled, GIF output is chosen by using the ‘`dvigif`’ binary or with the ‘`--gif`’ option.

It is intended to produce anti-aliased screen-resolution images as fast as is possible. The target audience is people who need to generate and regenerate many images again and again. The primary target is the Preview- $\text{\LaTeX}$  (X)Emacs package, a package to preview formulas from within (X)Emacs. Yes, you get to see your formulas in the (X)Emacs buffer, see <http://www.gnu.org/software/auctex/preview-latex.html>.

Another example is WeBWorK, an internet-based method for delivering homework problems to students over the internet, giving students instant feedback as to whether or not their answers are correct, see <http://webwork.math.rochester.edu>.

A more recent addition to the dvipng-using applications out there is MediaWiki, the software behind Wikipedia and many other wikis out there. Dvipng is used to render mathematical formulae from version 1.8.0 of MediaWiki, see <http://www.mediawiki.org>.

Other applications may also benefit, like web applications as latex2html and WYSIWYG editors like LyX.

The benefits of ‘`dvipng`’/‘`dvigif`’ include

- Speed. It is a very fast bitmap-rendering code for DVI files, which makes it suitable for generating large amounts of images on-the-fly, as needed in Preview- $\text{\LaTeX}$ , WeBWorK and others.
- It does not read the postamble, so it can be started before  $\text{\TeX}$  finishes. There is a ‘`--follow`’ switch that makes dvipng wait at end-of-file for further output, unless it finds the POST marker that indicates the end of the DVI.
- Interactive query of options. dvipng can read options interactively through stdin, and all options are usable. It is even possible to change the input file through this interface.
- Supports PK, VF, PostScript Type1, and TrueType fonts, subfonts (i.e., as used in CJK- $\text{\LaTeX}$ ), color specials, and inclusion of PostScript, PNG, JPEG or GIF images.
- and more...

## 2 Installation

Installing dvipng should be simple: merely `./configure`, `make`, and `make install`.

### 2.1 Prerequisites

- The GD Graphics Draw library, `libgd`  
 The drawing library ‘`libgd`’ is necessary, and is downloadable at <https://bitbucket.org/libgd/gd-libgd/downloads>, and there are binary packages for most operating systems from their respective distributors. In any case, the library installs using ‘`autoconf`’ so it should not be difficult for you to install it from source, and then proceed with installing dvipng.
- The path-searching library `kpathsea`  
`Kpathsea` is most likely included in your  $\text{\LaTeX}$  installation, but it may happen that `./configure` does not find it; see below. If you do not have it, download it from <http://www.ctan.org> and compile it. I have no experience with this, so I cannot help much here.
- The font-rendering library FreeType 2  
 While not strictly necessary, a recent FreeType 2 is recommended since dvipng currently will produce better-quality images when this library is available. To take advantage of this, you should have at least FreeType 2.1.9.  
`FreeType 2` will enable direct support for PostScript and TrueType fonts, so that dvipng will not need to generate bitmapped variants on disk of the  $\text{\TeX}$  fonts since modern  $\text{\TeX}$  distributions include PostScript versions of them. Then, you can render images at different (and unusual) resolutions without cluttering the disk with lots of bitmapped fonts.  
 Finally, it will enable subfont support in dvipng. That is, if you want to render CJK- $\text{\LaTeX}$  characters, you must have FreeType 2 installed.
- `libpng` and `libz`  
 To be able to compress and write PNG files to disk, dvipng (or really `libgd`) uses `libpng` which in turn uses `libz`. These should be available on any modern system, if not, download them and install them.
- The `texinfo` package  
 This is needed for building the documentation.

### 2.2 Configure

The first step is to configure the source code, telling it where various files will be. To do so, run

```
./configure options
```

(Note: if you have fetched dvipng from CVS rather than a regular release, you will have to first generate `./configure` by running `autoconf 2.53` or later.)

On many machines, you will not need to specify any options, but if `configure` cannot determine something on its own, you’ll need to help it out. For a list of the options type

```
./configure --help
```

On some machines, the libraries will be installed in directories that are not in the linker's search path. This will generate an error when running `./configure`, indicating that it cannot find `libgd` or `libkpathsea` (most likely). You then need to specify the path to the respective library's object files. They are typically called e.g., `libgd.a` or `libgd.so`. If they are located in e.g., `/sw/local/lib`, do

```
./configure LDFLAGS=-L/sw/local/lib
```

If the library is available as a shared object file (`.so`), the runtime linker may also need to be told where to find the library, then use

```
./configure LDFLAGS='-L/sw/local/lib -R/sw/local/lib'
```

When either of these is necessary, it is likely that the C header files are also installed in directories that are not in the C preprocessor's search path. This will also generate an error when running `./configure`, indicating that it cannot find e.g., `gd.h` or `kpathsea.h` (most likely). You then need to specify the path to the respective library's C header files. If they are located in e.g., `/sw/local/include`, do

```
./configure CPPFLAGS=-I/sw/local/include
```

On my SUN Solaris workstation, I had to combine this into

```
./configure CPPFLAGS='-I/sw/local/include -I/sw/tex/teTeX/1.0/include'\
LDFLAGS='-L/sw/local/lib -R/sw/local/lib -L/sw/tex/teTeX/1.0/lib/'
```

where the backslash denotes a continuation of the line.

## 2.3 Build/install

Once `configure` has been run, simply enter

```
make
```

at the prompt to compile the C code, and build the documentation files. To install the files into the locations chosen earlier, type

```
make install
```

You may need special privileges to install, e.g., if you are installing into system directories.

## 2.4 Installation outside the texmf tree

In some cases, a `dvipng` binary installed outside the `texmf` tree will not be able to find virtual fonts, or the PostScript font maps (normally used by `dvips`). This may be because *only* `$SELFAUTOLOC`, `$SELFAUTODIR`, and `$SELFAUTOPARENT` are used in the `texmf` tree configuration file '`texmf.cnf`'. If so, give the switch '`--enable-selfauto-set`' to '`./configure`'. This will make `dvipng` adjust these three internally so that `kpathsea` thinks that `dvipng` *is* installed in the `texmf` tree.

## 2.5 Installation for non-privileged users

Often people without system administration privileges want to install software for their private use. In that case you need to specify more options to the `configure` script, usually this is done by using the '`--prefix`' option to the `configure` script, and let it point to the personal home directory. In that way, resulting binaries will be installed under the `bin` subdirectory of your home directory, manual pages under `man` and so on. That way, it is

reasonably easy to maintain a bunch of additional packages, since the `prefix` argument is supported by most `configure` scripts.

You'll have to add something like `/home/myself/bin` to your `PATH` shell variable, if it isn't there already, and similarly set the `INFOPATH` and `MANPATH` variables to be able to access the documentation.

### 3 Basic usage of dvipng

To use dvipng at its simplest, simply type

```
dvipng foo
```

where `foo.dvi` is the output of TeX that you want to convert to PNG format. If there are four pages in `foo.dvi`, those pages will be output as `foo1.png`, `foo2.png`, `foo3.png`, and `foo4.png`, respectively.

If you have enabled the PostScript font support (via FreeType), fonts will be rendered as they are needed. Otherwise, dvipng will use bitmapped (PK) fonts, and if you use PK fonts that have not been used on your system before, they may be automatically generated; this process can take a few minutes, so progress reports appear by default. The next time the same font is used, it will have been saved on disk, so rendering will go much faster. (If dvipng tries to endlessly generate the same fonts over and over again, something is wrong. See Section “Unable to generate fonts” in *Kpathsea*.)

Many options are available (see the next section). For a brief summary of available options, just type

```
dvipng --help
```



## 4 Command-line options

dvipng has a plethora of command line options. Reading through this section will give a good idea of the capabilities of the driver.

### 4.1 Option summary

Here is a handy summary of the options; it is printed out when you run dvipng with no arguments or with the standard ‘--help’ option.

This is ./dvipng 1.17 Copyright 2002-2015, 2019 Jan-Ake Larsson

Usage: ./dvipng [OPTION]... FILENAME[.dvi]

Options are chosen to be similar to dvips’ options where possible:

-d #	Debug (# is the debug bitmap, 1 if not given)
-D #	Output resolution
-l #	Last page to be output
-o f	Output file, '%d' is pagenumber
-O c	Image offset
-p #	First page to be output
-pp #,#..	Page list to be output
-q*	Quiet operation
-T c	Image size (also accepts '-T bbox' and '-T tight')
-v*	Verbose operation
-	Interactive query of options

These do not correspond to dvips options:

-bd #	Transparent border width in dots
-bd s	Transparent border fallback color (TeX-style color)
-bg s	Background color (TeX-style color or 'Transparent')
--depth*	Output the image depth on stdout
--dvinum*	Use TeX page numbers in output filenames
-fg s	Foreground color (TeX-style color)
--follow*	Wait for data at end-of-file
--freetype*	FreeType font rendering (preferred, default on)
--gamma #	Control color interpolation
--gif	Output GIF images (dvigif default)
--height*	Output the image height on stdout
--nogs*	Don't use ghostscript for PostScript specials
--nogssafer*	Don't use -dSAFER in ghostscript calls
--norawps*	Don't convert raw PostScript specials
--palette*	Force palette output
--picky	When a warning occurs, don't output image
--png	Output PNG images (dvipng default)
--strict	When a warning occurs, exit
--truecolor*	Truecolor output
-Q #	Quality (PK subsampling)
--width*	Output the image width on stdout

`-z #`                    PNG compression level

# = number    f = file    s = string    \* = suffix, '0' to turn off  
c = comma-separated dimension pair (e.g., 3.2in,-32.1cm)

## 4.2 Option details

Many of the parameterless options listed here can be turned off by suffixing the option with a zero ('0'); for instance, to turn off page reversal, use `-r0`. Such options are marked with a trailing '\*'.

`-`                    Read additional options from standard input after processing the command line.

`--help`            Print a usage message and exit.

`--version`                    Print the version number and exit.

`-bd num`

`-bd color_spec`

`-bd 'num color_spec'`

Set the pixel width of the transparent border (default 0). Using this option will make the image edges transparent, but it only affects pixels with the background color. Giving a *color\_spec* will set the fallback color, to be used in viewers that cannot handle transparency (the default is the background color). The color spec should be in T<sub>E</sub>X color \special syntax, e.g., 'rgb 1.0 0.0 0.0'. Setting the fallback color makes the default border width 1 px. See Chapter 6 [Color], page 15.

`--bdpi num`

This option only has an effect when using bitmapped (PK) fonts. The option sets the base (Metafont) resolution, both horizontal and vertical, to *num* dpi (dots per inch). This option is necessary when manually selecting Metafont mode with the `-mode` option (see below).

`-bg color_spec`

Choose background color for the images. This option will be ignored if there is a background color \special in the DVI. The color spec should be in T<sub>E</sub>X color \special syntax, e.g., 'rgb 1.0 0.0 0.0'. You can also specify 'Transparent' or 'transparent' which will give you a transparent background with the normal background as a fallback color. A capitalized 'Transparent' will give a full-alpha transparency, while an all-lowercase 'transparent' will give a simple fully transparent background with non-transparent antialiased pixels. The latter would be suitable for viewers who cannot cope with a true alpha channel. GIF images do not support full alpha transparency, so in case of GIF output, both variants will use the latter behaviour. See Chapter 6 [Color], page 15.

`-d num`            Set the debug flags, showing what dvipng (thinks it) is doing. This will work unless dvipng has been compiled without the `DEBUG` option (not recommended).

Set the flags as you need them, use ‘-d -1’ as the first option for maximum output. See Section 7.2 [Debug options], page 16.

‘-D *num*’ Set the output resolution, both horizontal and vertical, to *num* dpi (dots per inch).

One may want to adjust this to fit a certain text font size (e.g., on a web page), and for a text font height of *font\_px* pixels (in Mozilla) the correct formula is

$$dpi = font\_px * 72.27 / 10 \text{ [px} * \text{T\_{E}Xpt/in} / \text{T\_{E}Xpt]}$$

The last division by ten is due to the standard font height 10pt in your document, if you use 12pt, divide by 12. Unfortunately, some proprietary browsers have font height in pt (points), not pixels. You have to rescale that to pixels, using the screen resolution (default is usually 96 dpi) which means the formula is

$$font\_px = font\_pt * 96 / 72 \text{ [pt} * \text{px/in} / \text{(pt/in)}]$$

On some high-res screens, the value is instead 120 dpi. Good luck!

‘--depth\*’ Report the depth of the image. This only works reliably when the L<sup>A</sup>T<sub>E</sub>X style `preview.sty` from Preview-L<sup>A</sup>T<sub>E</sub>X is used with the ‘active’ option. It reports the number of pixels from the bottom of the image to the baseline of the image. This can be used for vertical positioning of the image in, e.g., web documents, where one would use (Cascading StyleSheets 1)

```
<IMG SRC="filename.png" STYLE="vertical-align: -depthpx">
```

The depth is a negative offset in this case, so the minus sign is necessary, and the unit is pixels (px).

‘--dvinum\*’ Set this option to make the output page number be the T<sub>E</sub>X page numbers rather than the physical page number. See the ‘-o’ switch.

‘-fg *color\_spec*’ Choose foreground color for the images. This option will be ignored if there is a foreground color \special in the DVI. The color spec should be in T<sub>E</sub>X color \special syntax, e.g., ‘rgb 1.0 0.0 0.0’. See Chapter 6 [Color], page 15.

‘--follow\*’ Wait for data at end-of-file. One of the benefits of dvipng is that it does not read the postamble, so it can be started before T<sub>E</sub>X finishes. This switch makes dvipng wait at end-of-file for further output, unless it finds the POST marker that indicates the end of the DVI. This is similar to ‘tail -f’ but for DVI-to-PNG conversion.

‘--freetype\*’ Enable/disable FreeType font rendering (default on). This option is available if the FreeType2 font library was present at compilation time. If this is the case, dvipng will have direct support for PostScript Type1 and TrueType fonts internally, rather than using ‘gsftopk’ for rendering the fonts. If you have PostScript versions of Computer Modern installed, there will be no need to generate bitmapped (PK) variants on disk of these. Then, you can render

images at different (and unusual) resolutions without cluttering the disk with lots of bitmapped fonts. One reason to disable FreeType font rendering would be to generate identical output on different platforms, since FreeType uses the native renderer and therefore can give slightly different output on each platform.

`--gamma num`

Control the interpolation of colors in the greyscale anti-aliasing color palette. Default value is 1.0. For  $0 < num < 1$ , the fonts will be lighter (more like the background), and for  $num > 1$ , the fonts will be darker (more like the foreground).

`--gif*`

The images are output in the GIF format, if GIF support is enabled. This is the default for the `dvigif` binary, which only will be available when GIF support is enabled. GIF images are palette images (see the `--palette` option) and does not support true alpha channels (see the `--bg` option). See also the `--png` option.

`--height*`

Report the height of the image. This only works reliably when the `LATEX` style `preview.sty` from Preview-`LATEX` is used with the `active` option. It reports the number of pixels from the top of the image to the baseline of the image. The total height of the image is obtained as the sum of the values reported from `--height` and `--depth`.

`-l [=] num`

The last page printed will be the first one numbered *num*. Default is the last page in the document. If *num* is prefixed by an equals sign, then it (and the argument to the `-p` option, if specified) is treated as a physical (absolute) page number, rather than a value to compare with the `TEX` `\count0` values stored in the DVI file. Thus, using `-l =9` will end with the ninth page of the document, no matter what the pages are actually numbered.

`--mode mode`

This option only has an effect when using bitmapped (PK) fonts. Use *mode* as the Metafont device name for the PK fonts (both for path searching and font generation). This needs to be augmented with the base device resolution, given with the `--bdpi` option. See the file `ftp://ftp.tug.org/tex/modes.mf` for a list of resolutions and mode names for most devices. See Section “Unable to generate fonts” in *Kpathsea*.

`-M*`

This option only has an effect when using bitmapped (PK) fonts. It turns off automatic PK font generation (`mktexpk`).

`--nogs*`

This switch prohibits the internal call to GhostScript for displaying PostScript specials. `--nogs0` turns the call back on.

`--nogssafer*`

Normally, if GhostScript is used to render PostScript specials, the GhostScript interpreter is run with the option `-dSAFER`. The `--nogssafer` option runs GhostScript without `-dSAFER`. The `-dSAFER` option in Ghostscript disables PostScript operators such as `deletefile`, to prevent possibly malicious PostScript programs from having any effect.

`--norawps*`

Some packages generate raw PostScript specials, even non-rendering such specials. This switch turns off the internal call to GhostScript intended to display these raw PostScript specials. `--norawps0` turns the call back on.

`-o name`

Send output to the file *name*. A single occurrence of `%d` or `%01d`, ..., `%09d` will be exchanged for the physical page number (this can be changed, see the `--dvinum` switch). The default output filename is `file%d.png` where the input DVI file was `file.dvi`.

`-O x-offset,y-offset`

Move the origin by *x-offset,y-offset*, a comma-separated pair of dimensions such as `.1in,-.3cm`. The origin of the page is shifted from the default position (of one inch down, one inch to the right from the upper left corner of the paper) by this amount.

`-p [=]num`

The first page printed will be the first one numbered *num*. Default is the first page in the document. If *num* is prefixed by an equals sign, then it (and the argument to the `-l` option, if specified) is treated as a physical (absolute) page number, rather than a value to compare with the T<sub>E</sub>X `\count0` values stored in the DVI file. Thus, using `-p =3` will start with the third page of the document, no matter what the pages are actually numbered.

`--palette*`

When an external image is included, `dvipng` will automatically switch to truecolor mode, to avoid unnecessary delay and quality reduction, and enable the EPS translator to draw on a transparent background and outside of the boundingbox. This switch will force palette (256-color) output and make `dvipng` revert to opaque clipped image inclusion. This will also override the `--truecolor` switch if present.

`--picky*`

No images are output when a warning occurs. Normally, `dvipng` will output an image in spite of a warning, but there may be something missing in this image. One reason to use this option would be if you have a more complete but slower fallback converter. Mainly, this is useful for failed figure inclusion and unknown `\special` occurrences, but warnings will also occur for missing or unknown color specs and missing PK fonts.

`--png*`

The images are output in the PNG format. This is the default for the `dvipng` binary. See also the `--gif` option.

`-pp firstpage-lastpage`

Print pages *firstpage* through *lastpage*; but not quite equivalent to `-p firstpage -l lastpage`. For example, when rendering a book, there may be several instances of a page in the DVI file (one in `\frontmatter`, one in `\mainmatter`, and one in `\backmatter`). In case of several pages matching, `-pp firstpage-lastpage` will render *all* pages that matches the specified range, while `-p firstpage -l lastpage` will render the pages from the *first*

occurrence of *firstpage* to the *first* occurrence of *lastpage*. This is the (undocumented) behaviour of dvips. In dvipng you can give both kinds of options, in which case you get all pages that matches the range in ‘-pp’ between the pages from ‘-p’ to ‘-l’. Also multiple ‘-pp’ options accumulate, unlike ‘-p’ and ‘-l’. The ‘-’ separator can also be ‘:’. Note that ‘-pp -1’ will be interpreted as “all pages up to and including 1”, if you want a page numbered -1 (only the table of contents, say) put ‘-pp -1--1’, or more readable, ‘-pp -1:-1’.

- ‘-q\*’      Run quietly. Don’t chatter about pages converted, etc. to standard output; report no warnings (only errors) to standard error.
- ‘-Q *num*’      Set the quality to *num*. That is, choose the number of antialiasing levels for bitmapped fonts (PK), to be  $num * num + 1$ . The default value is 4 which gives 17 levels of antialiasing for antialiased fonts from these two. If FreeType is available, its rendering is unaffected by this option.
- ‘-r\*’      Toggle output of pages in reverse/forward order. By default, the first page in the DVI is output first.
- ‘--strict\*’  
             The program exits when a warning occurs. Normally, dvipng will output an image in spite of a warning, but there may be something missing in this image. One reason to use this option would be if you have a more complete but slower fallback converter. See the ‘--picky’ option above for a list of when warnings occur.
- ‘-T *image\_size*’  
             Set the image size to *image\_size* which can be either of ‘bbox’, ‘tight’, or a comma-separated pair of dimensions *hsize,vsize* such as ‘.1in,.3cm’. The default is ‘bbox’ which produces a PNG that includes all ink put on the page and in addition the DVI origin, located 1in from the top and 1in from the left edge of the paper. This usually gives whitespace above and to the left in the produced image. The value ‘tight’ will make dvipng only include all ink put on the page, producing neat images.
- ‘--truecolor\*’  
             This will make ‘dvipng’ generate truecolor output. Note that truecolor output is automatic if you include an external image in your DVI, e.g., via a PostScript special (i.e., the ‘graphics’ or ‘graphicx’ package). This switch is overridden by the ‘--palette’ switch.
- ‘-v\*’      Enable verbose operation. This will currently indicate what fonts is used, in addition to the usual output.
- ‘--width\*’  
             Report the width of the image. See also ‘--height’ and ‘--depth’.
- ‘-x *num*’      This option is deprecated; it should not be used. It is much better to select the output resolution directly with the ‘-D’ option. This option sets the magnification ratio to  $num/1000$  and overrides the magnification specified in the DVI file. Must be between 10 and 100000. It is recommended that you use standard magstep values (1095, 1200, 1440, 1728, 2074, 2488, 2986, and so on) to help

reduce the total number of PK files generated. *num* may be a real number, not an integer, for increased precision.

**‘-z *num*’** Set the PNG compression level to *num*. This option is enabled if your ‘libgd’ is new enough. The default compression level is 1, which selects maximum speed at the price of slightly larger PNGs. For an older ‘libgd’, the hard-soldered value 5 is used. The include file ‘png.h’ says

Currently, valid values range from 0 - 9, corresponding directly to the zlib compression levels 0 - 9 (0 - no compression, 9 - "maximal" compression). Note that tests have shown that zlib compression levels 3-6 usually perform as well as level 9 for PNG images, and do considerably fewer calculations. In the future, these values may not correspond directly to the zlib compression levels.

## 5 Graphics

`dvipng` attempts to handle graphics as included by the `graphicx` and `graphics` packages, without the need of specifying a driver to these packages. This means that it recognizes the encapsulated postscript inclusion meant for `dvips`, but is also able (from version 1.8) to include bitmapped graphics. It also tries to handle some of the raw PostScript that is output from various packages. Some of the possibilities and problems are mentioned below.

### 5.1 Encapsulated PostScript

When an EPS file is included, a call to GhostScript is performed to produce a bitmapped image that can be included. The default is to produce an image with transparent background, at the same size as the DVI page currently being converted to PNG, and include that as foreground on the PNG. Of course, if the image is to be cropped, that is done. The included image will be a truecolor image, so for maximum performance the output PNG will be in truecolor mode as well.

This conversion needs the `pngalpha` output device to be present in your copy of GhostScript. If that device is not present, or you use the `--palette` switch or request GIF output, the fallback is to use the `png16m` device to produce a cropped opaque image for inclusion. Other relevant switches are `--noghostscript` and `--nogssafer`. See Section 4.2 [Option details], page 7.

The most common problem with including graphics is an incorrect bounding box. Complain to whoever wrote the software that generated the file if the bounding box is indeed incorrect. An adjusted boundingbox can be specified in the `\includegraphics` call, as in this example (using `graphicx`):

```
\includegraphics[bb=10 20 100 200]{imagename.eps}
```

### 5.2 Bitmapped graphics

`dvipng` can include PNG, JPEG and GIF graphics. When including such images via `\includegraphics` you need to specify the bounding box since TeX itself cannot read them from the files in question. The bounding box size should be given as `'0 0 w h'` in pixels, e.g., if the file `imagename.png` is 300x400 pixels, the inclusion would read

```
\includegraphics[bb=0 0 300 400]{imagename.png}
```

The default size is the image size in bp (“big points” in TeX nomenclature or PostScript points as other people have it, 72 per inch). That is, default resolution will be 72 dpi for included bitmaps, which is the default size in the few other bitmap-capable drivers that are known to me (`dvipdfm` and `PDFLaTeX`).

If you want 100 dpi you need to specify the width accordingly. You just divide your image width by 100: a 135 pixel wide image at 100 dpi will take up 1.35 inches. If you want 200 dpi you divide by 200, and so on. Simple, eh? The example above at 200 dpi would be 1.5 inches wide:

```
\includegraphics[bb=0 0 300 400,width=1.5in]{imagename.png}
```



### 5.3 Raw PostScript

dvipng attempts to handle raw PostScript. Rendering raw PostScript specials is done on top of the page by including a transparent image generated by the ‘`pngalpha`’ device in GhostScript (automatically selecting ‘`truecolor`’ mode in dvipng).

Included PostScript headers are respected, and if the header ‘`tex.pro`’ is included, dvipng also throws in ‘`color.pro`’ and ‘`special.pro`’. The package ‘`xcolor`’ includes its own headers with color names, and this is not only kept as a PostScript header, but is also read and interpreted by dvipng itself. An attempt is also made to respect the PGF header. The non-rendering specials from ‘`hyperref`’ are handled via some heuristics and do not give an error.

Really rendering and moving things with raw PostScript specials is more troublesome. The `\rotatebox` macro serves as a good example. The dvips driver of the `graphicx` package surrounds DVI glyphs with PostScript code so that after conversion by dvips, the glyphs (now themselves in PostScript) will be rotated in the desired way. dvipng does not handle this, at present. An attempt has been made to handle the rendering specials output by PGF (`tikz`), and also PSTricks. Some things work, but others do not. This is especially clear when mixing PostScript and DVI rendering commands such as `glyphs`. dvipng cannot at present detect if PostScript code moves ‘`currentpoint`’ or rotates the frame since GhostScript does not return such information. A recommendation would be to produce images from these packages as EPS files and include them into your document in the standard manner.

Another way to handle this would be to use a slower fallback (with dvips and `gs`, for example). If you want to disable raw PostScript handling in dvipng, use the switch ‘`--norawps`’. This switch turns off the internal call to GhostScript intended to display these raw PostScript specials. Further, when dvipng encounters raw PostScript and the `gs` call is turned off, it gives a warning. It is now possible to use the switch ‘`--picky`’ to disable page rendering of pages with warnings, and use the slower fallback for these pages.

## 6 Color

To support color, dvipng recognizes a certain set of specials as generated by the ‘color’ and ‘xcolor’ style files. These specials start with the keyword ‘color’ or the keyword ‘background’, followed by a color specification.

### 6.1 Color specifications

The color specification supported by dvipng is by-value or by-name. The by-value spec starts with the name of a color model (one of ‘rgb’, ‘hsb’, ‘cmy’, ‘cmyk’, or ‘gray’) followed by the appropriate number of parameters. Thus, the color specification ‘rgb 0.3 0.4 0.5’ would correspond to the color that is ‘0.3 0.4 0.5’ in its red, blue and green values. The color model used internally in dvipng is ‘RGB’ (discretized to 256 levels), for details on the formulas used in conversion, see the ‘xcolor’ documentation.

By-name color specifications are single (case-dependent) words and are compared with color names defined in ‘dvipsnam.def’ (from the ‘graphics’ bundle), ‘svgnam.def’ and ‘xcolor.sty’ (from the ‘xcolor’ bundle). See the ‘xcolor’ documentation for a list of names and the corresponding colors.

On the command-line, the name ‘Transparent’ can also be used as an argument to ‘--bg’ to choose transparent background. See Section 4.2 [Option details], page 7.

### 6.2 Color specials

We will describe ‘background’ first, since it is the simplest. The ‘background’ keyword must be followed by a color specification. That color specification is used as a fill color for the background. The last ‘background’ special on a page is the one that gets used, and is used for the whole of the page image. (This is possible because the prescan phase of dvipng notices all of the color specials so that the appropriate information can be written out during the second phase.)

The ‘color’ special itself has three forms. The first is just ‘color’ followed by a color specification. In this case, the current global color is set to that color; the color stack must be empty when such a command is executed.

The second form is ‘color push’ followed by a color specification. This saves the current color on the color stack and sets the color to be that given by the color specification. This is the most common way to set a color.

The final form of the ‘color’ special is just ‘color pop’, with no color specification; this says to pop the color last pushed on the color stack from the color stack and set the current color to be that color.

dvipng correctly handles these color specials across pages, even when the pages are rendered repeatedly or in reverse order.

## 7 Diagnosing problems

You’ve gone through all the trouble of installing dvipng, carefully read all the instructions in this manual, and still can’t get something to work. The following sections provide some helpful hints if you find yourself in such a situation.

### 7.1 Contact information

Bug reports should be sent to `dvipng@nongnu.org`.

Questions, suggestions for new features, pleas for help, and/or praise should go to `dvipng@nongnu.org`. For more information on this mailing list, send a message with just the word ‘help’ as subject or body to `dvipng-request@nongnu.org` or look at <http://lists.nongnu.org/mailman/listinfo/dvipng>.

Offers to support further development will be appreciated. For developer access, ask on `dvipng@nongnu.org`.

For details on the T<sub>E</sub>X path-searching library, and `mktexpk` problems, see Section “Common problems” in *Kpathsea*.

### 7.2 Debug options

The ‘-d’ flag to dvipng helps in tracking down certain errors. The parameter to this flag is an integer that tells what errors are currently being tracked. To track a certain class of debug messages, simply provide the appropriate number given below; if you wish to track multiple classes, sum the numbers of the classes you wish to track. To track all classes, you can use -1.

Some of these debugging options are actually provided by Kpathsea (see Section “Debugging” in *Kpathsea*).

The classes are:

1	Normal dvi op-codes
2	Virtual fonts
4	PK fonts
8	TFM files
16	Glyph rendering
32	FreeType calls
64	Encoding loads
128	Color specials
256	GhostScript specials
512	Kpathsea <code>stat</code> calls
1024	Kpathsea hash table lookups
2048	Kpathsea path element expansion
4096	Kpathsea path searches

## 8 Credits

A number of persons have contributed, if I forget to mention someone, I apologize. First and foremost we have David Kastrup whose Preview-L<sup>A</sup>T<sub>E</sub>X project provided the incentive to write this program. There is also a number of people who have contributed by reporting bugs and suggesting improvements as the thing has evolved. These include but is perhaps not limited to (in semi-random order): Thomas Esser (t<sub>E</sub>X), Christian Schenk (M<sub>I</sub>K<sub>T</sub><sub>E</sub>X), Brian R Furry (debian package), Angus Leeming (L<sub>y</sub>X), Thomas Boutell (libgd), John Jones (first user report), Uwe Kern (xcolor), Karl Berry and Peter Breitenlohner (T<sub>E</sub>X Live), David Harvey (hinting in Freetype), Neal Harmon, Alan Shutko, Reiner Stieb, Nick Alcock, Adam Buchbinder, Svend Tollak Munkejord, James Longstreet, Bernhard Simon, Bob McElrath, Georg Schwarz, Jason Farmer, Brian V. Smith, Samuel Hathaway, Thomas R. Shemanske, Stephen Gibson, Christian Ridderström, Ezra Peisach, William H Wheeler, Thomas Klausner, Harald Koenig, Adrian Bunk, Kevin Smith, Jason Riedy, Wolfram Krause, Reinhard Kotucha, Takeshi Abe, Waldeck Schutzer, Ahzo, and Andy Nguyen.

## 9 Copying

This program is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Copyright © 2002-2015, 2019 Jan-Åke Larsson

# Index

—

-dSAFER ..... 9

## A

absolute page number, and ‘-l’ ..... 9

absolute page number, and ‘-p’ ..... 10

antialiasing levels, number of ..... 11

## B

background color (option) ..... 7

base resolution, setting ..... 7

baseline reporting ..... 8, 9

## C

color specifications ..... 15

command-line options ..... 6

compilation ..... 2

compression ..... 12

configuration, of dvipng ..... 2

## D

dark fonts ..... 9

debugging ..... 7, 16

depth reporting ..... 8

## E

exit on erroneous images ..... 11

## F

first page printed ..... 10

follow mode ..... 8

font generation, avoiding ..... 9

forcing palette output ..... 10

foreground color (option) ..... 8

FreeType font rendering ..... 8

fuzzy images ..... 9

## G

gamma ..... 9

GhostScript and -dSAFER ..... 9

GhostScript, turning off ..... 9

GIF image format ..... 9

## H

height reporting ..... 9

## I

installation, of dvipng ..... 2

invoking dvipng ..... 5

## L

last page printed ..... 9

light fonts ..... 9

## M

magnification, overriding DVI ..... 11

Metafont mode, specifying ..... 9

mktexpk, avoiding ..... 9

mode name, specifying ..... 9

## N

no erroneous images ..... 10

## O

offset pages ..... 10

option, details of ..... 7

options, dvipng ..... 6

options, reading from standard input ..... 7

options, summary ..... 6

output resolution, setting ..... 8

output, redirecting ..... 10

## P

page range ..... 10

page, first printed ..... 10

page, last printed ..... 9

physical page number, and ‘-l’ ..... 9

physical page number, and ‘-p’ ..... 10

PNG image format ..... 10

PostScript inclusion problems ..... 13

PostScript, turning off raw PostScript specials .. 10

problems ..... 16

## Q

quality ..... 11

quiet operation ..... 11

## R

reverse pagination ..... 11

**S**

silent operation ..... 11

standard input, reading options from ..... 7

standard output, output to ..... 10

**T**

transparent border fallback color ..... 7

transparent border width ..... 7

trouble ..... 16

truecolor output ..... 11

**W**

warnings, suppressing ..... 11

width reporting ..... 11