Epspdf and epspdftk User Manual

A multiplatform GUI- and command-line converter for [e]ps and pdf

Siep Kroonenberg (n dot s dot kroonenberg at rug dot nl)

Table of Contents

1 Usage

1.1 Overview

Epspdftk converts files between eps, pdf and general PostScript. The main screen lets you open a file, select some options and convert the file.

When opening a file, epspdf tries to find out file type and, in the case of a pdf file, the number of pages. This information is displayed in the box in the upper part of the screen.
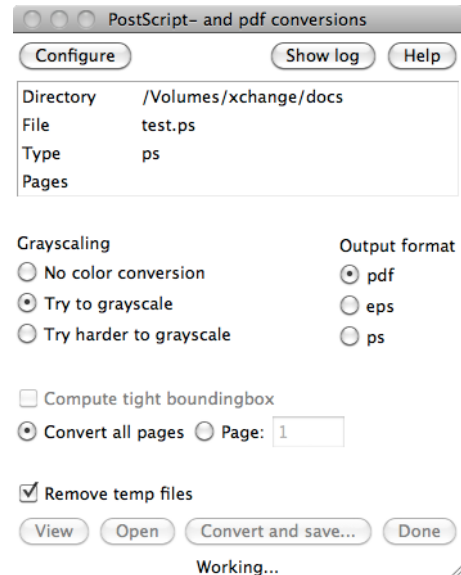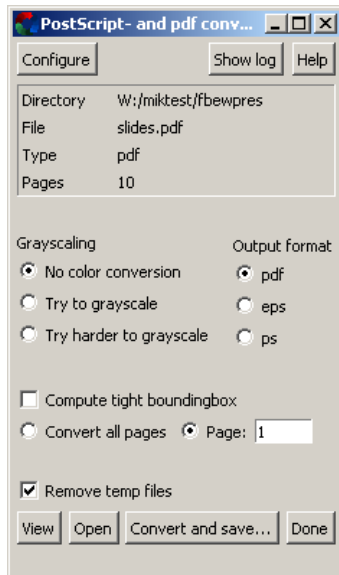
1.2 Viewing

The View button at the bottom of the screen calls an appropriate external viewer.

Windows and OS X: Epspdftk simply tries to use the default Open command.

Under Linux the PostScript- and pdf viewers are configurable; see .

The View button is grayed if epspdf thinks that there is no previewer for the current file.

1.3 Conversion options

Grayscaling: Simple color figures can be converted to grayscale. Bitmapped data won't be converted. If grayscaling doesn't work even on non-bitmapped data, select the "Try harder to grayscale" option. This may cause an extra pdf-PostScript roundtrip, but grayscaling may still fail.

Compute tight boundingbox: This option is only available if a single page is converted.

Page selection: The only possibilities are selecting a single page or selecting all pages. When converting to eps you are restricted to a single page.

For general PostScript files, there is no quick way to determine the number of pages, so the program may not check beforehand whether you picked an existing page. If you don't like that, convert the entire document to pdf first – which will be done behind the scenes anyhow.

Specifying options such as grayscaling or page selection may require several roundtrips between Post-Script or eps and pdf.

1.4 Converting

The Convert and save... button calls up a file save dialog. After a successful conversion, the result becomes the new current file, so you can judge the result by pressing the View button again.

1.5 The configuration screen
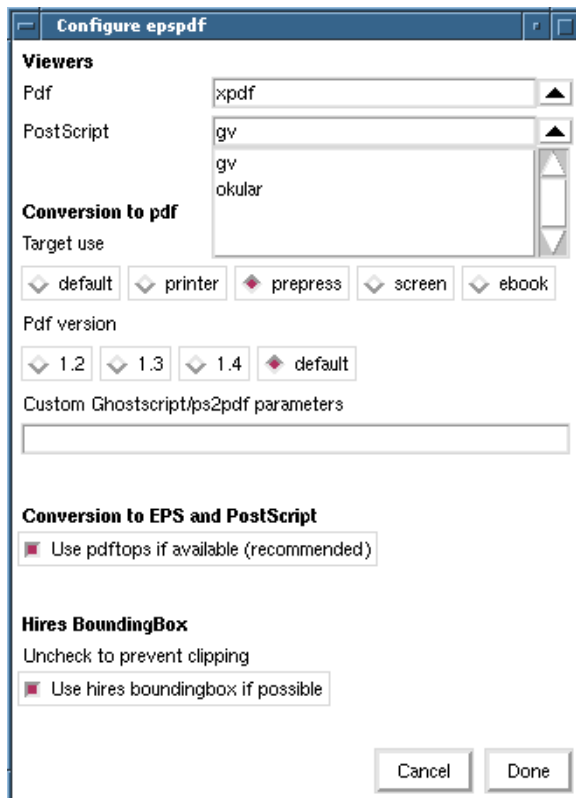
Epspdftk also has a configuration screen. The settings here are preserved between sessions.

1.5.1 Configuring viewers

Under Unix, the preferred PostScript- and pdf viewers can be configured in this screen. Epspdf looks for a number of PostScript- and pdf viewers, from which you can select one, but you can also enter one manually.

For Windows and Mac OS X, there is no such configuration option. Epspdf will use the program associated with the file type, which can be configured outside epspdf.

1.5.2 Options for converting to pdf

Double-check the setting "Target use" under "Conversion to pdf". Choose "prepress" for pdfs which are target for commercial printing. This option will try to embed all fonts, among other things. Often, printshops insist on this. With luck, Ghostscript fonts and system fonts will be used for fonts which were not originally embedded.

On the other hand, you may prefer "screen" if file size is a concern. See also the Ghostscript documentation, in particular Use.htm and Ps2pdf.htm.

1.5.3 Options for converting to PostScript

For conversion the other way you may opt *not* to use pdftops even if it is available. In this case, the conversion will be done by Ghostscript. Characters may be converted to drawn shapes or little bitmaps so this is not recommended if the file uses fonts.

Under Windows, this is also the place to point the program to the location of pdftops.exe.

1.5.4 Hires BoundingBox

Uncheck the option "Use hires boundingbox if possible" if your conversions come out slightly clipped. It has two effects:

- When converting to pdf, it uses the standard rather than the hires boundingbox for page dimensions
- When computing a boundingbox, it adds a small safety margin of 1pt to the standard boundingbox, except where it would make any boundingbox coordinate negative.

2 Command-line usage

epspdf.rb is the backend of epspdftk, but it can also be used standalone. It shares configuration settings with epspdftk.tcl.

It is strongly recommended to invoke epspdf.rb via a wrapper script epspdf or epspdf.bat on the search path, e.g.

```
#!/bin/sh
path_to_epspdf/epspdf.rb "£@"
```

for Unix/Linux/OS X, or

```
"path_to_ruby.exe" "path_to_epspdf.rb" %*
```

for Windows. It is the responsibility of the caller to provide double quotes.

The TeX Live package for epspdf already includes a wrapper.

Basic usage:

```
epspdf [options] infile outfile
```

2.1 Option summary

Typing *epspdf --help* gives you the following summary:

```
$ epspdf --help
Epspdf version 0.5.1
Copyright (C) 2006, 2008, 2009, 2010 Siep Kroonenberg
Convert between [e]ps and pdf formats
Usage: epspdf.rb [options] infile [outfile]

Default for outfile is file.pdf if infile is file.eps or file.ps
Default for outfile is file.eps if infile is file.pdf


    -g, --gray, --grey              Convert to grayscale;
                                    success not guaranteed
    -G, --GRAY, --GREY              Try harder to convert to grayscale
    -p, --pagenumber=PAGENUMBER     Page to be converted or selected
    -b, --bbox, --BoundingBox       Compute tight boundingbox
    -n, --no-hires                  Don't use hires boundingbox
    -r, --hires                     Use hires boundingbox
    -T, --target=TARGET             Target use of pdf; one of
                                    default, printer, prepress, screen, ebook
    -N, --pdfversion=PDFVERSION     Pdf version to be generated
    -I                              Ignore pdftops even if available
                                    (default: use if available)
    -U                              Use pdftops if available
                                    (overrides previous -I setting)
    -C, --custom=CUSTOMOPTIONS      Custom options for conversion to pdf,
                                    view Use.htm and ps2pdf.htm from
```

```
                                        the Ghostscript documentation set
        -P, --psoptions=PSOPTIONS       Options for pdftops; default -level3,
                                        don't include -eps or page number options;
                                        these will be generated by the program
        -i, --info                      Info: display detected filetype
        -s                              Save (some) settings
        -d                              Debug: don't remove temp files
           --gui[=ACTION]               Do not use; reserved for GUI

        -v, --version                   Prints version info
        -h, --help                      Show this message
```

3 Notes on PostScript and pdf

3.1 Bitmapped and vector

Pictures can be described either in terms of pixels, or more abstractly, in terms of geometric shapes, fonts and text.

Bitmapped or pixel-based graphics are appropriate for photographs and screenshots, but less so for diagrams and spreadsheet-generated graphics.

A file in PostScript- or pdf format can contain both types of graphic data.

Vector graphics can be freely scaled without losing sharpness or becoming pixellated. If bitmapped graphics are enlarged too much, individual pixels become apparent. With low-resolution bitmaps this happens sooner than with high-resolution bitmaps, but high-resolution bitmaps have (much) larger file sizes, and take longer to process.

So avoid converting vector to bitmap. However, converting from bitmap to vector is also best avoided, since it is very hard to do well.

Epspdf normally avoids conversion from vector to bitmap and never converts the other way. With the screen- and ebook "Target use" option, included bitmaps tend to be downsampled, i.e. reduced to a lower resolution.

3.2 Font embedding

When converting to pdf, Ghostscript handles font embedding differently depending on the "Target use" option. For prepress, it embeds even the standard fonts, including Times etc. Ghostscript has copies of these fonts in case the PostScript file doesn't have them. For screen, it doesn't embed standard fonts.

You can embed fonts after the fact by converting pdf to eps or ps, and back to pdf, with target use set to "prepress". However, don't try this unless you have pdftops.

If target use is set to "screen", then standard fonts will be removed instead of added. This option may also cause downsampling of bitmaps.

3.3 Eps preview headers

Preview headers are quietly stripped from eps files. These preview headers are used by e.g. desktop-publishing software to represent eps files on screen without having to interpret the PostScript code itself. Epspdf currently has no option to preserve or add them.

3.4 Bounding- and other boxes

A PostScript file may have a page size and a boundingbox defined. A pdf file may have a mediabox, a trimbox and various other boxes. Ghostscript by itself normally converts the PostScript page – which is anchored at (0,0) – to the pdf mediabox.

When epspdf or a similar program converts eps to pdf, it adds code to move the bottom left corner of the graphic to (0,0) and to set the page to the size of the graphic. Anything outside the boundingbox should be cut off.

Both Ghostscript and pdftops have some parameters for setting or using these boxes; consult Ps2pdf.htm from the Ghostscript documentation and "pdftops –help" or "man pdftops".

3.5 Orientation

There appears to be no reliable way to set orientation for PostScript- and pdf files. This may result part of a graphic being cut off after conversion or even everything falling outside the page / mediabox / boundingbox. It may help to change the extension from ".eps" to ".ps" before converting to pdf. In this case, let epspdf generate a new boundingbox.

3.6 Exporting PostScript from Windows programs

Often, the only way to get EPS or PostScript from a Windows program is by "printing" to a PostScript file. From some programs, you can print a selection. A suitable driver which comes with Windows is Generic / MS Publisher Imagesetter. Pay attention to the printer properties: choose "Outline" for font downloading and avoid the "Optimize for speed" setting for PostScript Output Option. Try e.g. "Archive" instead. These options can be found under the Advanced button.
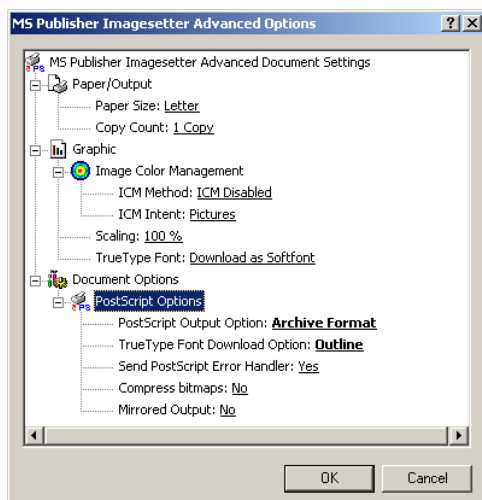
Other possibilities are the TpX and wmf2eps programs, which both have options to write clipboard contents to an emf file. Wmf2eps uses a printer driver in the background. For faithful conversion, pick wmf2eps; for subsequent editing, choose TpX. Both programs are available from http://www.tug.org/ctan.html.

4 Bitmapped graphics for LATEX and pdflatex

pdflatex can use graphics in .png format (best for screenshots) and .jpg format (best for photographs) directly. However, for LATEX you are stuck with .eps format. Tips for converting to .eps:

4.1 Linux

sam2p   This command-line bitmap-to-PostScript/pdf conversion utility is available from http://code.google.com/p/sam2p/ and may already be packaged for your distribution. It produces very small files: 'sam2p image.png image.eps'

ImageMagick/convert
       `convert` from the ImageMagick package is a command-line utility: 'convert image.png image.eps'

The GIMP   This is the premier open source image editing program. It is often pre-installed on Linux, and is also available for other platforms. The GIMP can save in eps format.

4.2 Mac OS X

Mac OS X's built-in Preview application can read most bitmapped formats and save them as .pdf, which epspdf can convert to .eps.

4.3 Windows

Windows is not a particularly friendly environment for PostScript and pdf. A couple of command-line utilities try to fill the gap, *e.g.*

```
sam2p image.png image.eps
```

or

```
bmeps -c image.png image.eps
```

sam2p is distributed with TeX Live (Windows only), bmeps both with TeX Live and with MikTeX. Without the `-c` option, bmeps produces a grayscale image. It produces larger files than sam2p.

4.3.1 GUI use of sam2p and bmeps

Create a batchfile 'bitmap2eps.bat' with contents

```
sam2p "%~1" "%~dpn1.eps"
```

(TeX Live) or

```
bmeps -c "%~1" "%~dpn1.eps"
```

(MikTeX) and place it *e.g.* on your desktop. Now if you drop a bitmapped graphic on this batchfile then you will get an eps file with the same name as the original, but with extension .eps.

You can also right-click on a bitmap, choose *Open With...* and browse to this file bitmap2eps.bat, resulting in conversion to .eps. The next time you do this with a file of the same type, bitmap2eps.bat will already be in the list of possible programs for opening the file.

5 Troubleshooting

5.1 Grayscaling fails

See above, under [Grayscaling], page 1. Epspdf is limited by what can be done by the backend tools, i.e. Ghostscript and pdftops. In particular, epspdf cannot grayscale bitmapped data.

5.2 Part of the graphic gets cut off

Things to try:

- If the PostScript file was generated with the Windows PostScript driver, experiment with the PostScript Output option. Don't choose Optimize for Speed.
- Replace the extension with ".ps" and let epspdf generate a boundingbox.
- Boundingbox calculations may be slightly inaccurate where text is involved. If just a sliver is cut off, let epspdf ignore the high-resolutuion boundingbox. Command-line option: "-n" or "–no-hires".

5.3 Fonts look ugly

If Ghostscript has to do the conversion from pdf to ps then text will not remain text, but will be replaced by drawn shapes or bitmaps. Try to get hold of the xpdf suite, which includes pdftops.

5.4 Epspdftk quit unexpectedly (Mac OS X applet)

This sometimes happens to me when I start epspdftk from the AppleScript applet. Unfortunately, I have no idea where this comes from.

Try the "Reopen" button in the popup dialog. If this repeatedly fails, install the regular Unix edition instead and run it from the command-line.

5.5 Resources for troubleshooting

Logfile. The GUI has a button for viewing log output. This same output is also written to a file 'epspdf.log'. For Linux/Unix/Mac OS X this is in a subdirectory .epspdf of your home directory; for Windows it is in a subdirectory epspdf of '%APPDATA%'. This APPDATA directory may be 'c:\Documents and Settings*your user name**Application Data*' or 'c:\Users*your user name**AppData**Roaming*'. Otherwise, open a command prompt and type *echo %APPDATA%*, which will produce this information.

The logfile lists all epspdf calls and all Ghostscript- and pdftops calls plus error information. "EPCallError" is probably a program error. 'EPSystemError" is more likely caused by a faulty file, but may be a program error as well.

Temporary files. The temporary files may give clues as well. Uncheck the button "Remove temp files", or for the command-line version, give a -d parameter to keep the temporary files. Check the log(file) as to which temporary files have been created.

Ghostscript- and pdftops documentation. For Ghostscript, the most important files are Ps2pdf.htm and Use.htm. For pdftops, type *pdftops -h*. For Unix, there is also a man page, and for Windows there is a file pdftops.txt in the distribution zip.

6 Change history

6.1 Version 0.5

The GUI has been rewritten in Tcl/Tk, removing the dependence on the Ruby/Tk interface library.

The Windows installer now installs a small Ruby subset and the standard epspdf distribution, but with epspdftk.tcl replaced with a starpack: a single executable containing 'epspdftk.tcl' and a Tcl/Tk runtime. See http://wiki.tcl.tk/52.

Epspdf now uses its own subdirectory for both the logfile and the configuration file. For Linux/Unix/Mac OS X this is '$HOME/.epspdf', for Windows it is '%APPDATA%\epspdf'. On all supported platforms, settings are stored in the file 'config' in this directory. Under Windows, the registry is no longer used for this.

A button has been added to view log output.

There is a second new button "Remove temp files", which is normally checked, causing temporary files to be deleted after each conversion. Unchecking this button may be useful for troubleshooting. In previous versions, temporary files were deleted at the end of the entire epspdftk session but that has become less practical now that the GUI and epspdf itself have become two separate programs.

The "Open with..." option for Windows has been dropped for technical reasons.

The "–version" option now prints the version string instead of setting the desired pdf version.

The "–info" option now also prints the number of pages for pdf files.

6.2 Version 0.4

Hi-res boundingboxes are now supported. By default, conversion from eps to pdf now uses the hires boundingbox as "page" to determine the page dimensions of the pdf file. Other conversions preserve or generate a hires boundingbox.

Under Windows, the new version looks for an installed TeX and will use its private Ghostscript if it cannot find a separately installed Ghostscript. TeX Live's pdftops, being on the searchpath, will be used unless epspdf finds another copy first.

There is now a "-v" (lowercase) option to print the version string.