Epspdf and epspdftk User Manual

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

Siep Kroonenberg (`siepo at bitmuis dot nl`)

This manual is for epspdf and epspdftk, version 0.6.5.2.

Table of Contents

1 Usage

1.1 Overview

Epspdftk is a GUI program that converts files between eps, pdf and general PostScript in any direction.

It lets you interactively select a file, set conversion options, and save in the same of another format. A configuration screen offers some additional conversion options.
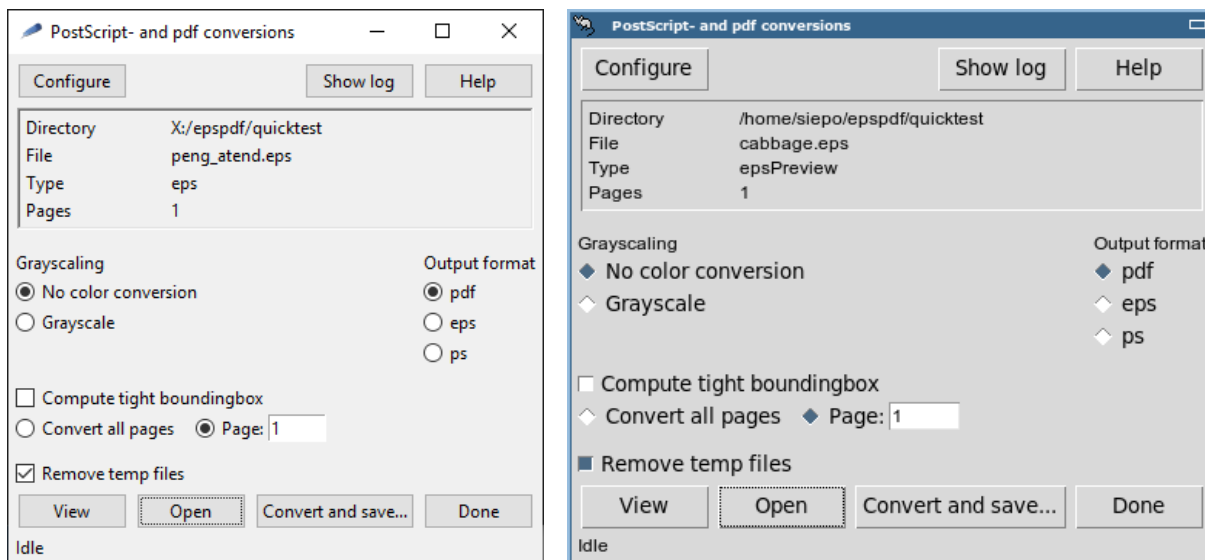
1.2 Viewing

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

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

Under Linux the PostScript- and pdf viewers are configurable; see Section 1.5 [The configuration screen], page 2.

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

1.3 Conversion options

- Grayscaling
- 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. Converting to eps implies selecting a single page.

Note. 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 anyway.

Specifying options such as grayscaling or page selection may require a multistep conversion.

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 – if epspdftk thinks that there is a suitable viewer.

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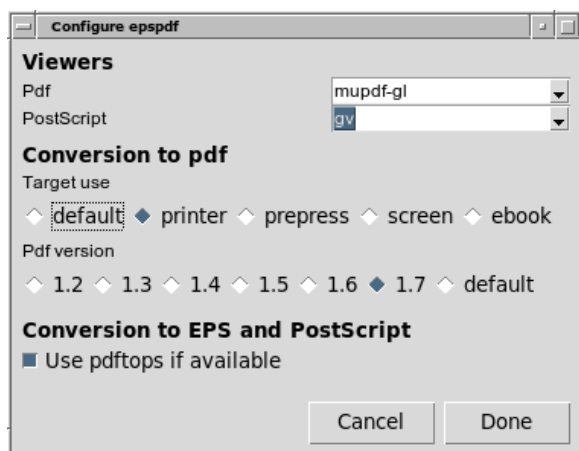
Epspdftk also has a configuration screen. The settings here are preserved between sessions.

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.

Some viewers require a plugin for eps and PostScript. The GUI does not check whether such a plugin is actually installed.

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

1.5.2 Options for converting to pdf

Double-check the setting "Target use" under "Conversion to pdf". "prepress" is for pdfs which are going to be printed commercially. The options prepress, printer and default will try to embed all fonts. Often, printshops insist on this.

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

Specifying anything other than "default" may cause an additional conversion step and possibly also loss of high-level structure.

As to pdf versions: this is a trade-off between more features for higher versions and better compatibility for lower versions. Versions below 1.4 do not support transparency. Converting a page or graphic with transparencies to version 1.3 will probably result in the whole page or graphic getting rasterized. For prepress use, consult your printshop or publisher. Otherwise leave the pdf version at "default".

1.5.3 Options for converting to PostScript

For conversion to plain or Encapsulated PostScript, pdftops sometimes does a better job of preserving fonts than Ghostscript. Under Windows, current versions of both MiKTeX and TeX Live include pdftops. Still, you may opt *not* to use pdftops even if it is available, in which case Ghostscript will be used instead.

2 Command-line usage

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

The first parameter of the epspdftk GUI program is interpreted as startup directory for the file browser. Epspdf itself has a more elaborate command-line interface.

Below, we assume that there is a suitable wrapper or symlink for epspdf on your searchpath. This is the case if you installed epspdf as a TeX Live- or MiKTeX package.

Basic usage of epspdf itself:

    epspdf [*options*] *infile* [*outfile*]

2.1 Option summary

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

```
$ epspdf --help
Epspdf version 0.6.5.1
Copyright (c) 2006-2023 Siep Kroonenberg

Convert between [e]ps and pdf formats
Usage: epspdf[.tlu] [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

-p, --page, --pagenumber PNUM
            Page number; must be a positive integer
-g, --grey, --gray, -G, --GREY, --GRAY
            Convert to grayscale
-b, --bbox, --BoundingBox
            Compute tight boundingbox
-T, --target TARGET
            One of screen, ebook, printer, prepress or default
-N, --pdfversion VERSION
            One of 1.2, 1.3, 1.4, 1.5, 1.6, 1.7 or default
-U          Use pdftops if available
-I          Reverses the above
-s, --save  Save some settings to configuration file
-i, --info  Info: display detected filetype and exit
-d          Debug: do not remove temp files
-v, --version
            Display version info and exit
-h, --help  Display this help message and exit
```

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 usually 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. According to the Ghostscript documentation, it embeds all fonts without exception for all targets except "screen", for which standard fonts such as Times may be omitted.

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 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.

Conversion to pdf usually translates the bottom-left corner to the (0,0) origin and sets the . . . box to the size of the graphic. Anything outside the . . . box should be cut off.

3.5 Orientation

Ghostscript may not be able to determine the right orientation of a PostScript- or pdf file. This may result in part of a graphic being cut off after conversion or even everything falling outside the page / mediabox / boundingbox.

3.6 Exporting PostScript or pdf from Windows programs

Microsoft Office 2010 can now export to pdf, either the entire document or a selection. This removes a major headache for Windows users. Windows 10 also includes a pdf printer.

As a last resort on earlier Windows versions, you can "print" to a PostScript file. From some programs, you can print a selection. A suitable driver which comes with Windows is Generic / MS Publisher Color Printer. Pay attention to the printer properties: choose "Outline" for font downloading and avoid the "Optimize for speed" setting for PostScript Output Option. In my tests, "Encapsulated PostScript" did not look very promising either. Try e.g. "Archive" instead. These options can be found under the Advanced button.

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- and pdf format.

4.2 Mac OS

Mac OS's built-in Preview application can read most bitmapped formats and save them as pdf or PostScript. In fact, in many cases it is an excellent alternative to epspdf.

4.3 Windows

I know of no built-in facility for converting bitmaps to eps or pdf, apart from "printing" to PostScript or pdf. However, many image editors can convert to eps or pdf. A couple of command-line utilities:

```
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.

With TeX Live, you can convert to eps by right-clicking an image in Windows Explorer and "open" with *bitmap2eps*, which uses sam2p or bmeps in the background.

5 Troubleshooting

5.1 No options for output format visible

Widen the window, to make the output format radio buttons visible.

5.2 Ghostscript stackunderflow error

Some Ghostscript versions, *e.g.* 9.10, have an error in color handling when converting to pdf: selecting "printer" as target may lead to an error `stackunderflow in .setdistillerparams`. If you run into this, use the "default" or "prepress" target instead.

5.3 Error message mentioning failure of `pdf_crop`

Problably, the luatex format has not been generated. Fix this with TeX Live Manager or with the MiKTeX Console.

5.4 Wrong boundingbox (.eps or .ps output)

This may be a bug in pdftops. Do not use pdftops (epspdftk: see configuration screen; command-line epspdf: use the "-I" option)

5.5 Part of the graphic gets cut off

If the PostScript file was generated with the old Windows PostScript driver, experiment with the Post-Script Output option. Don't choose Optimize for Speed.

5.6 Fonts look ugly

If Ghostscript has to do the conversion from pdf to ps then text may not remain text, but may be replaced by bitmaps. Newer Ghostscript versions tend to do better in this respect than older ones, but as to preserving fonts, the pdftops utility may still handle more cases. It is part of the xpdf suite and of the Poppler utilities.

5.7 The page has been converted to a bitmap

The usual cause is that the page contains features such as transparency which are not supported by the target format.

Otherwise, set pdf target use and target version both to "default" to avoid unnecessary conversions: `-T default -N default`

5.8 Temporary files are not removed

- Command-line: make sure that you did not specify the option `-d`.
- Epspdftk GUI: make sure that 'Remove temp files' is checked.
- Windows: this platform is notorious for excessive file locking. I already inserted a Windows-specific delay before attempting to delete the temporary files, but this is not always enough. However, you should have no trouble removing temporary files manually.

5.9 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 Mac OS 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:\Users*your user name*\AppData\Roaming. Type

```
echo %APPDATA%
```

in a Command Prompt window to find out which.

The logfile lists all epspdf calls and all Ghostscript- and pdftops calls plus error information.

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 VectorDevices.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. TeX Live includes the man page in pdf format: pdftops.pdf.

Appendix A Requirements

Epspdftk consists of a GUI front end and a command-line back end.

The command-line component, which can be used separately from the front end, requires ghostscript and a not too old TeX installation with luatex. It may further benefit from the presence of pdftops. MiKTeX and TeX Live for Windows meet these requirements out of the box.

The front end requires a Tcl/Tk installation, or at least a tclkit runtime. TeX Live for Windows contains either a tclkit runtime or a minimal Tcl/Tk installation since late 2018, and epspdftk as a TeX Live package will make use of it. The minimum version is 8.5.

At the time of writing, this front end does not work under MiKTeX; I do not know why not. The alternative epspdf-setup package for Windows with an embedded Tcl/Tk runtime and an installer has been withdrawn, since it was only useful on MiKTeX.

Appendix B Change history

B.1 Version 0.6.5.2

Epspdftk: compatibility fix for Tcl/Tk 9.

B.2 Version 0.6.5.1

Epspdf: An erroneous and unnecessary check for writability, with possible security implications, has been removed. For MiKTeX, os.execute replaces os.spawn invocations, since os.spawn gave trouble in my tests on MiKTeX.

B.3 Version 0.6.5

Epspdf: Compatibility change for ghostscript 9.50 and later.

Epspdftk: improvements to viewer selection; does not impact Windows or MacOS.

A few updates to the manual, mostly for the troubleshooting section.

B.4 Version 0.6.4

Compatibility changes for luatex 1.9 and later.

The location of pdftops is no longer configurable; it is only searched for on the searchpath.

Some corner cases should be handled more successfully.

The GUI now requires Tcl/Tk version 8.5 or higher. On Windows, epspdf.tlu is now invoked via a batchfile. This should prevent black console windows popping up.

B.5 Version 0.6

The command-line backend component has been rewritten in texlua and therefore no longer needs an external scripting language.

Grayscaling is now done by Ghostscript's color options for pdf output. This also works for bitmaps.

Croppping of pdfs is now accomplished by running luatex on a suitable wrapper file (same method as Heiko Oberdiek's pdfcrop). Such a conversion preserves advanced features which might otherwise get lost.

I no longer try to provide an AppleScript wrapper.

The current version has no provisions for custom Ghostscript- or pdftops parameters. The corresponding command-line options are accepted but have no effect.

B.6 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 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.

B.7 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.