The package epspdfconversion

Daniel Becker
daniel.becker@uni-rostock.de *

2. September 2009, version 0.5

Contents

1 What is it for?

This package enables the use of the epspdf tools (see http://tex.aanhet.net/epspdf/) from within (pdf)LaTeX "on the fly". It is similar to and based on the epstopdf package that uses the script `epstopdf` for the actual conversion while this packages uses `epspdf` (Note the `epsTOpdf` vs `epspdf`).[1] It is possible to pass several options to the `epspdf` conversion-command.

*Many thanks to Siep Kroonenberg and Heiko Oberdiek for their help.

[1] You might also want to read the documentation of epstopdf. See http://www.ctan.org/tex-archive/macros/latex/contrib/oberdiek/epstopdf.pdf.

While this package can be used for the conversion of eps-files to pdf, the epspdf-tools itself can do the conversion both ways.[2]

Use this package at your own risk.[3]

I am using this package for the inclusion of eps-figures that are produced en-masse by a software packages like Stata, Mathematica or Maple and that are often updated. The package makes sure that I can include the eps-figures easily and the updating of the corresponding pdf's is done "on-the-fly". Using the epspdf-tools (and not epstopdf) helps a lot to prepare a final pdf that is, for example, accepted by your print shop (really grayscale, prepress-ready, ...).

2 Installation

If you are using a recent version of TeXLive ($\geqslant$ 2008), you can skip steps 1-4.

1. Go to http://tex.aanhet.net/epspdf/ and follow the installation instructions there.

   Alternative URL's for epspdf are
   http://www.ctan.org/tex-archive/support/epspdf/ or
   http://www.ctan.org/tex-archive/help/Catalogue/entries/epspdf.html

   See also Bruno Voisin's notes at http://tug.org/pipermail/macostex-archives/2007-April/030318.html.

2. the package needs a recent version of the epstopf-package as a prerequisite, at least version 2.2.

3. Make sure that you can use epspdf from the command line. I am using Mac OS X. After the installation of epspdf (or with TeXLive / MacTeX $\geqslant$ 2008), the following command is working from the command line (assuming the file /Users/daniel/Desktop/testimage.eps exists):

   ```
   macbook-daniel:~ daniel$ which epstopdf
   /usr/texbin/epstopdf
   macbook-daniel:~ daniel$ epspdf /Users/daniel/Desktop/testimage.eps
   ```

   It results in a file /Users/daniel/Desktop/testimage.pdf.

   I did no testing for other Systems (Windows,...) – simply follow the instructions at the epspdf-website. It would be nice if you let me know whether this package works for you.[4]

4. Put the sty-file epspdfconversion.sty for this package where TeX can find it. For example in the same directory as your main .tex file. Or – again on Mac OS X – in ~/Library/texmf/tex/latex/ where the ~ stands for your home directory.

   If you are using MikTeX, you can use the package-manager to install epspdfconversion. If you are using a recent version of TeXLive, it is already

[2] It might be possible to add support for conversion like pdf->pdf or pdf->eps and so on. If you are interested, contact me.

[3] I am by no means an TEXpert. What I did is to take the epstopdf-package (http://www.ctan.org/tex-archive/help/Catalogue/entries/epstopdf.html) and adjusted it to my needs and then gave this the new name epspdfconversion. Heiko Oberdiek then improved the code a lot.

[4] The package seems to work as expected with a Linux/TeXLive2007 and with a Windows/MikTeX2.5 installation.

included in your distribution.

5. The package requires that shell escapes are enabled. You should get a warning in your .log-file if this is not the case.

   However, a test with MikTeX and the command-line option -shell-escape resulted in the warning "Package epstopdf Warning: Shell escape feature is not enabled." Try `--enable-write18` if you are using MikTeX. Look for "shell escape" and "write18" in the Help-Section of your preferred application for Typesetting on how to enable it (TeXShop, WinEdt, ...).

3 Usage

Put in the preamble of your .tex-file the line

```
\usepackage[OPTIONSHERE]{epspdfconversion}
```

where "OPTIONSHERE" can be either empty or be filled with the options described below.

If you typeset your document, and (pdf)LaTeX detects that you want to use an eps-figure, the `epspdfconversion`-package makes sure that it is converted to a pdf that is then included.

By default, if you include your eps-figure *with* the `.eps` extension, as in

```
\includegraphics[width=0.5\textwidth]{image.eps}
```

, there will be a conversion to pdf named `image-epspdf-to.pdf` that is then used in your document. The next run will only call a conversion if the original .eps-file is newer (has been updated in the meanwhile). This is to save typesetting time. You can change this behaviour with the option `update=false`, see below.

If you insist on using `\includegraphics` without the extension, as in

```
\includegraphics[width=0.5\textwidth]{image}
```

, the situation is more complicated. If you are using `\includegraphics` without the extension, pdfLaTeX when used with epstopdf or `epspdfconversion` looks for files that can be used in the following order:

```
Package grfext Info: Graphics extension search list:
(grfext)             [.png,.pdf,.jpg,.mps,.jpeg,.jbig2,.jb2,.PNG,.PDF,
.JPG,.JPEG,.JBIG2,.JB2,.eps]
```

If - for whatever reason, a file `image.png` exists, this one will be used, not the .eps or the converted pdf. However, you can use the option prepend the "Graphics extension search list" will look like this:

```
Package grfext Info: Graphics extension search list:
(grfext)             [.eps,.png,.pdf,.jpg,.mps,.jpeg,.jbig2,.jb2,.PNG,.PDF,
.JPG,.JPEG,.JBIG2,.JB2]
```

This implies that image.eps is found first and used with a conversion to pdf is necessary. Complicated? Consider to use `\includegraphics` with the extension, that avoids confusion which file actually makes it into you document. See options prepend, prefersuffix, update, suffix below and try to figure out how many different scenarios there are.

Below, I include a figure `testimage.eps` and another `testimage-static.eps` with the following lines.

```
\begin{center}
\includegraphics[width=0.5\textwidth]{testimage.eps}\\
\includegraphics[width=0.5\textwidth]{testimage-static}
\end{center}
```

4 Options

`epspdfconversion` accepts several options. Table 1 gives an overview. The explanations are more or less taken from the documentationof epspdf and epstopdf.

Table 1: Options for the package `epspdfconversion`

option	explanation
Options related to epspdf	
`help`	You will be shown the help of the epspdf command in your logfile. This option does not overrule all the others as previously.
`simple`	the epspdf-conversion will be done with no option at all. Don't use it together with any of the options below.
`gray | grey | GRAY | GREY`	gray | grey : convert eps-figures to grayscale (success not guaranteed); GRAY | GREY: Try harder to convert to grayscale (success still not guaranteed)
`nogray | nogrey`	nogray | nogrey : do not try to convert eps-figures to grayscale
`default | printer | prepress | screen | ebook | target=default | target=printer | target=prepess | target=screen | target=ebook`	target use of pdf
`pdfversion=default | pdfversion=1.2 | pdfversion=1.3 | pdfversion=1.4`	Pdf version to be generated. Setting another version than those on the left will result in an error. 'default' means whatever Ghostscript's default is.
`bbox | bbox=true | bbox=false`	If true: Compute tight boundingbox

... to be continued on next page ....

Table 1: Options for the package `epspdfconversion` – continued

option	explanation
`nopdftops`	Ignore pdftops even if available (default: use if available)
`pdftops`	use pdftops if available
`hires`	compute hires-Boundingbox
`no-hires`	don't compute hires-Boundingbox
`custom={-dPDFX}`	This option allows you to pass a string to the ghostscript-commandline. On the left it would be Here you can set custom options for conversion to pdf, view Use.htm and ps2pdf.htm from the Ghostscript documentation set. The example on the left adds `-dPDFX` to the ghostscript-call by epstopdf
`psoptions={-level2}`	This sets the options for pdftops; the default is -level2, don't include -eps or page number options; these will be generated by epstopdf itself
`pagenumber={1}`	Page (in the eps-file) to be converted
Options related to epstopdf.sty (the package)	These options are options that are passed over to epstopdf.sty that works in the background. You could also use `\epstopdfsetup{OPTIONSforEPSTOPDF.sty}`, but you can also control the behaviour of epstopdf.sty by means of setting option to `epspdfconversion`. The explanation is borrowed from the documentation of epstopdf.
`prepend` | `prepend=true` | `prepend=false`	Determines whether .eps is appended (if false) or prepended (if true) to the Graphics extension search list. (default: false). (Note that there is no option append. Use `prepend=false` instead.)
`update` | `update=true` | `update=false`	The conversion program is only called, if the target file does not exist or is older than the source image file.
`verbose` | `verbose=true` | `verbose=false`	prints some information about the image in the .log file (default: true).

... to be continued on next page ....

Table 1: Options for the package `epspdfconversion` – continued

option	explanation
`suffix={-mysuffix}`	defines a string that is put between the file name base and the extension of the output file. This avoids that existing pdf-files are overwritten. See the documentation of epstopdf for a more detailed explanation. (default: suffix=-epspdf-to)
`prefersuffix` | `prefersuffix=true` | `prefersuffix=false`	If a suffix is set by option su?x, then there can be two image file names that could be taken into account for inclusion: A image ?le name with the suffix string inside its name and a image file name without; e.g. for `foo.eps` the names could be: `foo-suffix.pdf`, `foo.pdf` If option prefersuffix is turned on, the file `foo-suffix.pdf` and its generation is preferred over using `foo.pdf`. Otherwise `foo.pdf` is included without generating `foo-suffix.pdf`. (default: true)
`outdir=./`	The converted file may put in an other output directory. The value of outdir must include the directory separator. Example for the current directory: `\epstopdfsetup{outdir=./}` For other directories ensure that they can be found. See `\graphicspath` or `TEXINPUTS`. You might need to set `suffix=` to use another directory than the current. (default: outdir=)

(end of table)

Important: When there are several options in the first column, divided by |, this means that you should *choose only one* of them. For example, it does not make sense have this in the preamble:

```
\usepackage[pdfversion=1.3,pdfversion=1.4]{epspdfconversion}
```

`\pdfminorversion`: When you choose the options pdfversion=1.2 or pdfversion=1.3, you need to set `\pdfminorversion` accordingly. The package checks if you have done that properly and shows a warning if not.

Changing the options somewhere in the middle of your .tex document is supported. Writing

```
\epspdfconversionsetup{target=prepress,bbox}
```

changes the options of `epspdfconversion` to `target=prepress,bbox`. Other options than `target=...` remain in effect.

5 \epspdfconversioncmdline

Many of the options described above change the command that is used to call epspdf for the conversion from .eps to .pdf.

Typing `\epspdfconversioncmdline` somewhere in your source-.tex file will output the call that you have defined in your preamble. For example, this file has in the preamble

```
\usepackage[pdfversion=1.3,GRAY]{epspdfconversion}
```

and the `\epspdfconversioncmdline` then is: `epspdf --GRAY --version=1.3`.

This means that you can use `\renewcommand` to define you own `\epspdfconversioncmdline`.

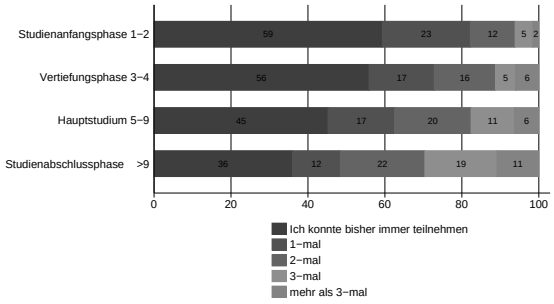
For example, to restore the behaviour of the epstopdf-package, you could write

```
\renewcommand{\epspdfconversioncmdline}%
{epstopdf }
```

This allows you to use whatever tool you want for your eps->pdf conversion.

6 A test

What follows is the output of the `\includegraphics`-command from page 4.

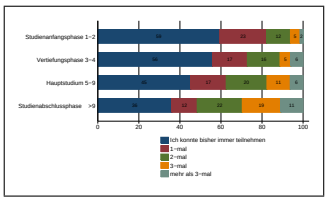


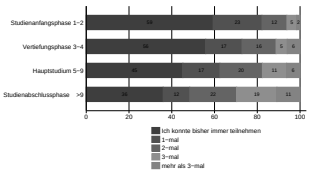
7 Switching options temporarily

It is possible to switch the options only temporarily using curly braces. Consider you have set the options `GRAY` such that all your figures appear in grayscale. Now you want color for a single figure. This can be done like this:

```
{% <= New group started
\epspdfconversionsetup{nogray,bbox=false}
\fbox{\includegraphics[width=4cm]{image2.eps}}
}% <= New group ended; grayscaling,bbox set to previous value.
\includegraphics[width=4cm]{image2.eps}
```

The figure `image2.eps` will be exceptionally in color, other figures in gray, according to the general rule for this document:





8 A note for users of latexmk

latexmk is a perl script for running LaTeX, BibTex, Makeindex etc the correct number of times. See http://www.phys.psu.edu/~collins/software/latexmk/. It can be configured to run pdflatex if an eps-image has been updated (since version V. 3.21i). In your local configuration file, you should have something like this:

```
# FOR USERS OF epstopdf v1.5 and later only:
add_cus_dep('eps', 'pdf', 0, 'cus_dep_require_primary_run');
```

Note that both epstopdf and latexmk are under active development. Please check the latexmk-documentation to if you have difficulties with setting up a latexmk-configuration that works together with epstopdf and `epspdfconversion`.

9 –help of epstopdf

The help of epspdf (version 0.4.1) reads:

```
macbook-daniel:~ daniel$ epspdf --help
Epspdf version 0.4.1
Copyright (C) 2006, 2008, 2009 Siep Kroonenberg
Epspdf 0.4.0
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
    -V, --version=PDFVERSION         Deprecated; use '-N' or '--pdfversion'.
    -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

    -v                              Prints version info
    -h, --help                      Show this message
macbook-daniel:~ daniel$
```

10 Version-history, ToDo's

ToDo's add support for ps->pdf, pdf->pdf / add support for tif and others in pdflatex via convert / add support for pdf-inclusion in latex (not pdf-latex) / add support for more file-types (tif, jpeg,...) in latex (not pdf-latex). Please report errors and missing features.

v.0.5, 2009-09-02: this update makes use of changes in the epstopdf-package v2.2

- new options update,verbose,prefersuffix,suffix,outdir (they are really epstopdf options, but can be set as options for this package)
- default is that converted files have a suffix
- info in logfile about the setup that is used for epstopdf
- new options hires, no-hires

v.0.4, 2007-11-24: the epstopdf-package is now loaded with options [update,prepend] (works only when epstopdf version 1.5 is used) An update of epstopd.sty (part of the oberdiek-bundle) is recommended. Added options nogrey,nogray

v.0.3, 2007-10-02:

- check whether `\pdfminorversion` has been set in accordance with option pdfversion=...
- Use the kvoptions-package for the implemention of options. It uses key value syntax that can be used both as package options and a separate setup macro.
- Almost all options of epstopdf are now available as an option of this package.
- The command `\epspdfconversionsetup` is new and allows a change of the options for this package anywhere in your document.
- The command `\epspdfconversioncmdline` has been renamed to `\epspdfconversioncmdline`.
- the documentation has been updated

v.0.2, 2007-09-21: the package is now simply based on epstopdf. It essentially defines `\@namedef{Gin@rule@.eps}#1{{pdf}{.pdf}{`\epspdfconversioncmdline #1}} differently than epstopdf. The code has been cleaned up. Improvements of documentation and additional warning about pdfminorversion....

v.0.1, 2007-09-21: first try