The pdfpages Package*

Andreas MATTHIAS
amat@kabsi.at

2002/07/24

Abstract

This package makes it easy to insert pages of external PDF documents. It is based on pdfLaTeX and does *not* work with LaTeX.

Contents

1 Introduction

Creating PDF documents, it is sometimes useful to insert pages of other, external PDF documents. This can be done with the `\includegraphics` command from the `graphics` package. But a simple `\includegraphics{doc.pdf}` normally produces '`Overfull \hbox`' and '`Overfull \vbox`' warnings, because the size of the inserted pages does not match the print space.

The `pdfpages` package makes it easy to insert pages of external PDF documents without worrying about the print space. Here are some features of the `pdfpages` package: Several logical pages can be arranged onto each sheet of paper and the layout can be changed individually. A lot of hypertext operations are supported, like links to the inserted pages, links to the original PDF document, threads, etc.

*This file has version number v0.2i, last revised 2002/07/24.

2 Usage

2.1 Package Options

`\usepackage[`⟨*options*⟩`]{pdfpages}`

⟨*option*⟩ – `final`: Inserts pages. This is the default.

`draft`: Does not insert pages, but prints a box and the filename instead.

2.2 Commands

`\includepdf` Inserts pages of an external PDF document.

`\includepdf[`⟨*key=val*⟩`]{`⟨*filename*⟩`}`

⟨*key=val*⟩ – A comma separated list of options using the ⟨*key*⟩=⟨*value*⟩ syntax.

⟨*filename*⟩ – Filename of the PDF document. (The filename *must not* contain any blanks!)

The following list describes all possible options of `\includepdf`. All options are using the ⟨*key=value*⟩ syntax. If options are marked with '*changed*' on the left margin, their syntax or meaning has changed since pdfpages v0.1i. New options are marked with '*new*'.

- Main options:

pages Selects pages to insert. The argument is a comma separated list, containing page numbers (`pages={3,5,6,8}`), ranges of page numbers (`pages={4-9}`) or any combination. To insert empty pages use `{}`.

E.g.: `pages={3,{},8-11,15}` will insert page 3, an empty page, and pages 8, 9, 10, 11, and 15.

Page ranges are specified by the following syntax: ⟨*m*⟩`-`⟨*n*⟩. This selects all pages between ⟨*m*⟩ and ⟨*n*⟩. Omitting ⟨*m*⟩ defaults to the first page, omitting ⟨*n*⟩ defaults to the last page of the document.

E.g.: `pages=-` will insert *all* pages of the document.

(Default: `pages=1`)

changed **nup** Puts multiple logical pages onto each sheet of paper. The syntax of this option is: `nup=`⟨*xnup*⟩`x`⟨*ynup*⟩. Where ⟨*xnup*⟩ and ⟨*ynup*⟩ specify the number of logical pages in horizontal and vertical direction, which are arranged on each sheet of paper. (Default: `nup=1x1`)

changed **landscape** Specifies the format of the sheet of paper (and *not* of the logical pages). Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `landscape=false`)

- Layout options:

new **delta** Puts some horizontal and vertical space between the logical pages. The argument should be two dimensions, separated by space. See Chapter 2.3 and Figure 1. (Default: `delta=0 0`).

offset Displaces the origin of the inserted pages. The argument should be two dimensions, separated by space. See Chapter 2.3 and Figure 1. (Default: `offset=0 0`)

frame Puts a frame around each logical page. Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `frame=false`)

new **column** Pdfpages normally uses 'row-major' layout, where successive pages are placed in rows along the paper. The `column` option changes the output into a 'column-major' layout, where successive pages are arranged in columns down the paper. Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `column=false`)

new **columnstrict** By default the last page is not set in a strict 'column-major' layout, if the logical pages do not fill up the whole page. The `columnstrict` option forces a strict 'column-major' layout for the last page. Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `columnstrict=false`)

1	4	
2	5	
3		

`columnstrict=true`

1	3	5
2	4	

`columnstrict=false`

openright This option puts an empty page before the first logical page. In combination with `nup=2x1`, `nup=2x2`, etc., this means that the first page is on the right side. The same effect can be achieved with the `pages` option, if an empty page is inserted in front of the first page. Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `openright=false`)

pagecommand Declares LaTeX commands, which are executed on each sheet of paper. (Default: `pagecommand={\thispagestyle{empty}}`)

turn By default pages in landscape format are displayed in landscape orientation (if the PDF viewer supports this). With `turn=false` this can be prohibited. Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `turn=true`)

new **noautoscale** By default pages are scaled automatically. This can be suppressed with the `noautoscale` option. In combination with the `scale` option (from graphicx) you has full control over the scaling process. Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `noautoscale=false`)

new **fitpaper** Adjusts the paper size to the one of the inserted document. Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `fitpaper=false`)

new **doublepages** Inserts every page twice. This is useful for 2-up printing, if one wants to cut the stack of paper afterwards to get two copies. Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `doublepages=false`)

new **signature** Creates booklets by rearranging pages into signatures and setting `nup=1x2` or `nup=2x1`, respectively. This option takes one argument specifying the size of the signature, which should be a multiply of 4.

An example for documents in portrait orientation:

```
\includepdf[pages=-, signature=8,
            landscape]{portrait-doc.pdf}
```

An example for documents in landscape orientation:

```
\includepdf[pages=-, signature=8]{landscape-doc.pdf}
```

- Hypertext options:

changed **link** Inserted pages become a target of a hyperlink. The name of the link is '⟨*filename*⟩.⟨*page number*⟩'. The file extension of ⟨*filename*⟩ *must not* be stripped. Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `link=false`)

linkname Changes the default linkname created by the option `link`. Instead of ⟨*filename*⟩ the value of this option is used. E.g. `linkname=mylink` produces the linknames 'mylink.⟨*page number*⟩'.

thread Combines inserted pages to an article thread. Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `thread=false`)

threadname Several threads are distinguished by their threadnames. By default the threadname is equal to the filename, but it can be changed with this option. This is useful if the same file is inserted twice or more times and should not be combined to one single thread. Or the other way round if pages from different documents should be combined to one single thread. (Default: `threadname=`⟨*filename*⟩)

linktodoc Lets the inserted pages be hyperlinks to the document from which they were extracted. Note that the PDF-Viewer will not find the file, if ⟨*filename*⟩ has not filename extension (.pdf). Either 'true' or 'false' (or no value, which is equivalent to 'true'). (Default: `linktodoc=false`)

- Additional hypertext options:

linkfit Specifies the way the viewer displays a linked page. This option changes the default behavior of the option `link`. Possible values are: `fitb`, `fith`, `fitv`, `fitr`, `xyz zoom` ⟨*integer*⟩, etc. These are destinations (`\pdfdest`) like they are described in [1]. (Default: `linkfit=fitr`)

linktodocfit By default the option `linktodoc` opens the page in 'Fit in Window' view. Another view can be specified with this option. Possible values are the legal PDF tokens: `/FitH` ⟨*top*⟩, `/FitV` ⟨*left*⟩, etc. (See [2] for more details.) (Default: `linktodocfit=/Fit`)

linkfilename Sets the name (with path) of the file to be linked to by the option `linktodoc`. You will hardly ever need this option. (Default: `linkfilename=`⟨*filename*⟩)

- Experimental options: (Syntax may change in future versions!)

new **addtotoc** Adds an entry to the table of contents. This option requires five arguments, separated by commas:

`addtotoc={`⟨*page number*⟩`,`⟨*section*⟩`,`⟨*level*⟩`,`⟨*heading*⟩`,`⟨*label*⟩`}`

⟨*page number*⟩: Page number of the inserted page.

⟨*section*⟩: LaTeX sectioning name – e.g., section, subsection, ...

⟨*level*⟩: Number, denoting depth of section – e.g., section=1, subsection=2, ...

⟨*heading*⟩: Title inserted in the table of contents.

⟨*label*⟩: Name of the label. This label can be referred to with `\ref` and `\pageref`.

Note: The order of the five arguments must not be mixed. Otherwise you will get very strange error messages.

The `addtotoc` option accepts multiple sets of the above mentioned five arguments, all separated by commas. The sets must be sorted such that the ⟨*page number*⟩s are in ascending order. (Strictly speaking they must have the same order as the page numbers specified by the `pages` option.)

The proper recursive definition of the `addtotoc` option is:

`addtotoc=`{⟨*toc-list*⟩}

⟨*toc-list*⟩ → ⟨*page number*⟩,⟨*section*⟩,⟨*level*⟩,⟨*heading*⟩,⟨*label*⟩[,⟨*toc-list*⟩]

new **addtolist** Adds an entry to the list of figures, the list of tables, or any other list (e.g. from *float.sty*). This option requires four arguments, separated by commas:

`addtolist=`{⟨*page number*⟩,⟨*type*⟩,⟨*heading*⟩,⟨*label*⟩}

⟨*page number*⟩: Page number of the inserted page.

⟨*type*⟩: Name of a floating environment. (`figure`, `table`, etc.)

⟨*heading*⟩: Title inserted into LoF, LoT, etc.

⟨*label*⟩: Name of the label. This label can be referred to with `\ref` and `\pageref`.

Like `addtotoc`, `addtolist` accepts multiple sets of the above mentioned four arguments, all separated by commas. The proper recursive definition is:

`addtolist=`{⟨*lof-list*⟩}

⟨*lof-list*⟩ → ⟨*page number*⟩,⟨*type*⟩,⟨*heading*⟩,⟨*label*⟩[,⟨*lof-list*⟩]

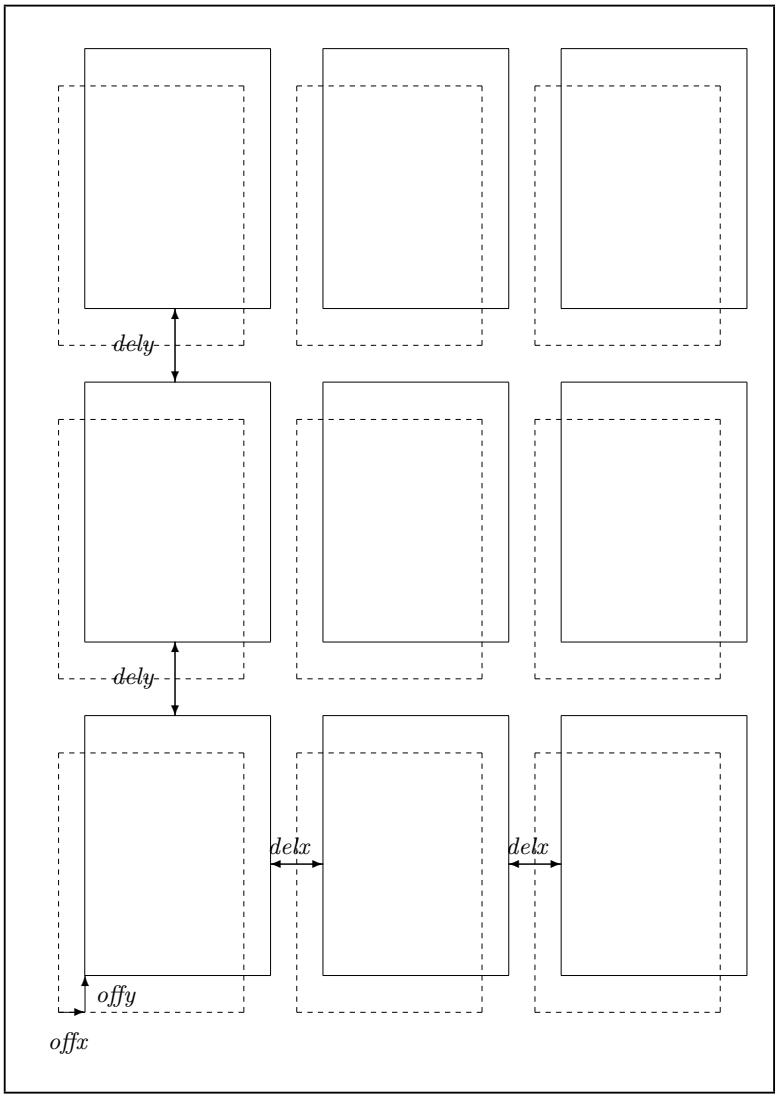
- Obsolete options:

deltax Puts some horizontal space between the logical pages. The argument should be one dimension, separated by space. (Default: `deltax=0`).

deltay Puts some vertical space between the logical pages. The argument should be one dimension, separated by space. (Default: `deltay=0`).

Internally the command `\includepdf` makes use of the `\includegraphics` command from the `graphicx` (actually `graphics`) package. Hence it is possible to use all the options of `\includegraphics`, too. Options which are not interpreted by `\includepdf` are passed directly to `\includegraphics`.

Especially the 'trim' and 'clip' options of `\includegraphics` are quite useful, if only parts of a page should be inserted. (Maybe to cut off the header and footer of the inserted pages.) Just use the 'trim' and 'clip' options as if they were options of `\inlcudepdf`. They will be passed to `\includegraphics` internally.



— physical sheet of paper
— logical pages
--- centered position

`delta=`⟨*delx*⟩ ⟨*dely*⟩
`offset=`⟨*offx*⟩ ⟨*offy*⟩

Figure 1: Layout

`\includepdfmerge` Inserts pages of several external PDF documents.

`\includepdfmerge[`⟨*key=val*⟩`]{`⟨*file-page-list*⟩`}`

⟨*key=val*⟩ – A comma separated list of options using the ⟨*key*⟩=⟨*value*⟩ syntax.

⟨*file-page-list*⟩ – ⟨*filename*⟩[,⟨*page spec*⟩][,⟨*file-page-list*⟩]
A comma separated list of filenames and optional ⟨*page spec*⟩ specifiers. A ⟨*page spec*⟩ can be everything the option `pages` accepts. Leading and trailing spaces of items in the list is stripped.

The `\includepdfmerge` command uses the same options as `\includepdf` with one exception. The option `pages` has no meaning for `\includepdfmerge`. Instead the ⟨*page spec*⟩ specifier is used to specify which pages should be inserted. The ⟨*page spec*⟩ specifier accepts the same values as the `pages` option. If no ⟨*page spec*⟩ specifier is given, only the first page will be inserted.

Examples: To create a kind of summary of three PDF documents, it might be nice to insert just the first page of each document and to provide links to the original documents:

```
\includepdfmerge[nup=1x3, landscape, linktodoc]
                {doc1.pdf, doc2.pdf, doc3.pdf}
```

But sometimes the title page of a document is not the first page. So it would be more pleasant to insert the title page of each document than the first page. This can be done with the ⟨*page spec*⟩ specifier. The following example inserts the second page of *doc1.pdf* and the third page of *doc2.pdf* and *doc3.pdf*:

```
\includepdfmerge[nup=1x3, landscape, linktodoc]
                {doc1.pdf, 2, doc2.pdf, 3, doc3.pdf, 3}
```

Here is an example of more complex ⟨*page spec*⟩ specifiers:

```
\includepdfmerge[nup=1x3, landscape, linktodoc]
                {doc1.pdf, 1-3,
                 doc2.pdf, 3, 5, 9,
                 doc3.pdf, 3-5, 7}
```

`\includepdfset` If you need the same options for `\includepdf` all the time, it is possible to define global options with `\includepdfset`. The argument of `\includepdfset` is a comma separated list of options, using the ⟨*key*⟩=⟨*value*⟩ syntax. These options are processed each time `\includepdf` is called. Local options (passed as an optional argument directly to `\includepdf`) are overwriting global options:

```
\includepdfset{⟨global options⟩}
\includepdf[⟨local options⟩]{pdf-file}
```

Only options specific to this package can be made global by `\includepdfset`. Options of the graphicx package are not concerned.

`\threadinfodict` When using the option `thread` to create an article thread, it may be useful to create a thread information dictionary, too, which contains informations about the thread, such as its title, author, and creation date. The macro `\threadinfodict` is used to set these informations. It can be redefined and must contain the thread information dictionary in low-level PDF commands. (See [2] for more information.)

```
\renewcommand*{\threadinfodict}
  {/I << /Title (My first thread) /Author (That's me!) >>}
```

2.3 The Layout

The default layout can be changed by the options `delta` and `offset`. Figure 1 shows the meaning of these options.

The inserted logical pages are being centered on the sheet of paper by default. To displace them use the `offset` option, which argument should be two dimensions. E.g. `offset=10mm 14mm` means that the logical pages are displaced by 10 mm in horizontal direction and by 14 mm in vertical direction.

By default logical pages are being arranged side by side. To put some space between them, use the `delta` option, whose argument should be two dimensions. Figure 1 shows the meaning of `delta`.

The layout options `delta` and `offset` *always* refer to a sheet of paper in portrait orientation. No matter whether you have set the `landscape` option to `true`, or not.

If you are confused about horizontal (`x`) and vertical (`y`) directions, just set the option `turn=false`. Now your PDF viewer shows the pages in the *same* orientation as in Figure 1. And the options `delta` and `offset` have the *same* meaning as in Figure 1. Regardless of any other options.

2.4 Pitfalls

pagecolor When setting the background color with `\pagecolor` (a command from *color.sty*), the first `\pagecolor` *must* precede `\usepackage{pdfpages}`.

```
\usepackage{color}
\pagecolor{white}
\usepackage{pdfpages}
```

The color is nonrelevant, it can be changed afterwards by using `\pagecolor` again. Just the order (first `\pagecolor` before `\usepackage{pdfpages}`) is important.

3 Required Packages

The `pdfpages` package requires the following packages:

eso-pic `CTAN:macros/latex/contrib/supported/ms/contrib/` Download the whole `ms/` directory, because eso-pic.sty requires everyshi.sty from that directory.

graphicx, ifthen, calc These packages belong to the standard LaTeX distribution.

Furthermore it requires a recent version of:

pdftex.def `http://www.tug.org/applications/pdftex/pdftex.def`

Since pdfTeX, Version 3.14159-1.00a-pretest-20010806, PDF import has improved a lot. This results in much smaller file sizes, faster processing and the intuitively correct treatment of landscape pages. The latest version of pdfTeX can be found at: `ftp://ftp.muni.cz/pub/tex/local/cstug/thanh/pdftex`.

4 Acknowledgment

I would like to thank Rolf Niepraschk and Heiko Oberdiek for their useful hints and suggestions. As well as Ross Moore, who encouraged me to implement the hypertext features.

References

[1] Hàn Thế Thành, Sebastian Rahtz, Hans Hagen, *The pdfTeX user manual*, http://www.tug.org/applications/pdftex/pdftex-s.pdf

[2] *PDF Reference*, Third Edition, Version 1.4, Adobe Systems Incorporated, http://partners.adobe.com/asn/developer/acrosdk/docs/filefmtspecs/PDFReference.pdf