

# The pdfpages Package\*

Andreas MATTHIAS  
amat@kabsi.at

2001/03/10

## Abstract

This package makes it easy to include pages of external PDF documents. It is based on pdfLaTeX and does *not* work with L<sup>A</sup>T<sub>E</sub>X.

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Usage</b>	<b>2</b>
2.1	Package Options . . . . .	2
2.2	Commands . . . . .	2
2.3	Remarks on Landscape Format . . . . .	3
<b>3</b>	<b>Acknowledgment</b>	<b>4</b>
<b>4</b>	<b>Implementation</b>	<b>4</b>
4.1	Checking . . . . .	4
4.2	Required Packages . . . . .	5
4.3	Package Options . . . . .	5
4.4	Counters and Booleans . . . . .	5
4.5	Commands . . . . .	6
4.6	Options of <code>\includepdf</code> . . . . .	10
<b>5</b>	<b>History</b>	<b>11</b>

## 1 Introduction

When creating PDF documents it is sometimes useful to include pages from 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

---

\*This file has version number v0.1e, last revised 2001/03/10.

the included page does not match to the print space. To circumvent this, one have to scale (reduce) the included page. However, this is sometimes not desirable.

The `pdfpages` package makes it easy to include one or more pages of external PDF documents without worrying about the print space. It even gives you the possibility to arrange two or four logical pages on one physical sheet of paper. (Like ‘psnup’.)

## 2 Usage

### 2.1 Package Options

`\usepackage[options]{pdfpages}`

*option* – **final**: Include pages. This is the default.

**draft**: Do not include pages, but print a box and the filename instead.

### 2.2 Commands

`\includepdf` Pages of external PDF documents are included as follows:

`\includepdf[key=val]{pdf-file}`

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

*pdf-file* – The filename of the PDF document.

Possible options specific to `\includepdf` are:

**firstpage** Page number of the first page to include. (Default: `firstpage=1`)

**lastpage** Page number of the last page to include.  
(Default: `lastpage=firstpage`)

**nup** Put multiple logical pages on one physical sheet of paper. Possible values are: 1, 2 and 4. (Default: `nup=1`)

**landscape** Include pages in landscape format. See section 2.3 for more information on landscape format.

**turn** Some pages are displayed in landscape orientation. With `turn=false` this can be prohibit. Either ‘true’ or ‘false’ (or no value, which is equivalent to ‘true’). (Default: `turn=true`)

**openright** Put first page on the right side. (Just in combination with `nup=2` and `nup=4`.) Either ‘true’ or ‘false’ (or no value, which is equivalent to ‘true’). (Default: `openright=false`)

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

**deltax** Put some space between the logical pages in horizontal direction. The argument should be one dimension. (Default: `deltax=0`)

**deltay** Put some space between the logical pages in vertical direction. The argument should be one dimension. (Default: `deltay=0`)

**pagecommand** L<sup>A</sup>T<sub>E</sub>X commands, which are executed on each physical sheet of paper. (Default: `pagecommand={\thispagestyle{empty}}`)

**offset** Displace the origin of the included pages. The argument should be two dimensions, separated by space. (Default: `offset=0 0`)

The `\includepdf` command calls the `\includegraphics` command from the `graphicx` (actually `graphics`) package, which makes it possible to use its options too. All options which are not interpreted by `\includepdf` are passed to `\includegraphics`. This way it is possible e. g. to use the ‘trim’ and ‘clip’ options of `\includegraphics` by passing them to `\includepdf`, to include only parts of a page. (Maybe to cut off the header and footer of the included pages.)

`\includepdfset` If one needs 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.

### 2.3 Remarks on Landscape Format

To include pages in landscape format use the option `landscape`. However, this only works satisfyingly, if the included pages are in ‘real’ landscape format. This means that the pages are wider than high. In the PDF file such pages are described e. g. as:

```
/MediaBox [0 0 841.887 595.273]
```

Nevertheless there exists another way, how to describe a landscape page in a PDF file. E. g.:

```
/MediaBox [0 0 595.273 841.887]
/Rotate 90
```

This means that the page is in portrait format (the page is higher than wide), but it is rotated by 90 degrees. I call this a ‘fake’ landscape. This is what you get with the following L<sup>A</sup>T<sub>E</sub>X (pdfLaTeX) code fragment:

```
\usepackage{lscape}
...
\begin{landscape}
\pdfpageattr{/Rotate 90}
...
\end{landscape}
```

Keep in mind that ‘real’ and ‘fake’ landscape pages look the same in the PDF Viewer. And unfortunately until now pdfLaTeX does not support the user with any information about the /Rotate entry.

Here is an example, how to include ‘real’ landscape pages:

```
\includepdf [page=4, landscape]{\file}
```

To include ‘fake’ landscape pages you have to rotate the pages by using the option entry `angle=-90` like in the following example:

```
\includepdf [page=4, landscape, angle=-90]{\file}
```

Sorry, but there is no way to do this automatically.

### 3 Acknowledgment

I would like to thank ROLF NIEPRASCHK and HEIKO OBERDIEK for their useful hints and suggestions.

## 4 Implementation

### 4.1 Checking

Check for using pdfLaTeX. – If `graphics.sty` is loaded before this package, `pdfoutput` may be undefined, even if pdfLaTeX is used, because some `graphic.cfg` undefine `pdfoutput` (`\let\pdfoutput\@undefined`), if the document contains the line `\pdfoutput=0`.

```
1 \begingroup
2 \@ifundefined{pdfoutput}
3   {\PackageError{pdfpages}
4     {You're either not using pdfLaTeX\MessageBreak
5     or have turn off PDF output.\MessageBreak
6     'pdfpages' works only with pdfLaTeX\MessageBreak
7     and PDF output}
8     {Use 'pdflatex' instead of 'latex'\MessageBreak
9     and set output to PDF (use: \pdfoutput = 1)!}
10  \expandafter\endinput
11  }{}
```

Check for producing PDF output.

```
12 \ifnum\pdfoutput=0
13   \PackageError{pdfpages}
14     {You don't produce PDF output, which\MessageBreak
15      is required for the 'pdfpages' package}
16     {Set output to PDF. (use: \pdfoutput = 1)}
17   \expandafter\endinput
18 \fi
19 \endgroup
```

Check for pdftex.def version: This is done with the following macro, which is derived from \@ifpackagelater from the standard L<sup>A</sup>T<sub>E</sub>X kernel.

```
20 \def\AM@ifdriverlater{\@ifl@ter{def}}
```

Check for pdftex.def version 2000/05/12 v0.02u or higher:

\* support for multipage PDF files, option 'page' added (HO).

```
21 \AtBeginDocument{%
22   \AM@ifdriverlater{pdftex}{2000/05/12}{-}{%
23     \PackageError{pdfpages}{%
24       You have an old 'pdftex.def'.\MessageBreak%
25       There is no multipage support\MessageBreak%
26       in 'pdftex.def' < 0.02u.\MessageBreak%
27       Update your 'pdftex.def'
28       {Update your 'pdftex.def'!}}
```

Check for pdftex.def version 2000/09/14 v0.03d or higher:

\* Fixes for 'viewport' and 'trim' (HO).

\* Clip support added for viewport and trim (HO).

```
29   \AM@ifdriverlater{pdftex}{2000/09/14}{-}{%
30     \PackageWarning{pdfpages}{%
31       You have an old 'pdftex.def'.\MessageBreak%
32       There might be problems with\MessageBreak%
33       the 'trim' and 'clip' options.\MessageBreak%
34       Update your 'pdftex.def'!\MessageBreak}}
35 }% AtBeginDocument
```

## 4.2 Required Packages

```
36 \RequirePackage{ifthen, graphicx, eso-pic, calc}
```

## 4.3 Package Options

```
37 \newcommand{\AM@draft}{false}
38 \DeclareOption{draft}{\def\AM@draft{true}}
39 \DeclareOption{final}{\def\AM@draft{false}}
40 \ProcessOptions
```

## 4.4 Counters and Booleans

```
41 \newcounter{AM@firstpage}\setcounter{AM@firstpage}{1}
42 \newcounter{AM@lastpage}\setcounter{AM@lastpage}{2}
```

```

43 \newcounter{AM@nup}\setcounter{AM@nup}{1}
44 \newboolean{AM@openright}\setboolean{AM@openright}{false}
45 \newboolean{AM@frame}\setboolean{AM@frame}{false}
46 \newboolean{AM@landscape}\setboolean{AM@landscape}{false}
47 \newboolean{AM@turn}\setboolean{AM@turn}{true}
48 \newcounter{AM@page}\setcounter{AM@page}{1}

```

## 4.5 Commands

```

49 \newcommand{\AM@xoffset}{Opt}\newcommand{\AM@yoffset}{Opt}
50 \newcommand{\AM@deltax}{Opt}\newcommand{\AM@deltay}{Opt}
51 \newcommand{\AM@xpos}{Opt}\newcommand{\AM@ypos}{Opt}
52 \newcommand{\AM@rotate}{0}
53 \newcommand{\AM@lscap@roti}{0}
54 \newcommand{\AM@lscap@rotii}{0}
55 \newcommand{\AM@pagecommand}{\thispagestyle{empty}}
56 \newcommand{\AM@fbox}{}
57 \newsavebox{\AM@pagebox}

```

`\includepdfset` This command sets global options for `\includepdf`.

```
58 \newcommand*{\includepdfset}[1]{\setkeys{Gin}{#1}}
```

`\includepdf` This command includes one or more pages of an external PDF document.

```
59 \newcommand*{\includepdf}[2] []{%
60 \begingroup% setkey
```

Save counters.

```
61 \newcommand{\saveAM@firstpage}{}
62 \edef\saveAM@firstpage{\arabic{AM@firstpage}}
63 \newcommand{\saveAM@lastpagexxx}{}
64 \edef\saveAM@lastpage{\arabic{AM@lastpage}}
65 \newcommand{\saveAM@nup}{}
66 \edef\saveAM@nup{\arabic{AM@nup}}
```

Read in optional parameters.

```
67 \setkeys{Gin}{draft=\AM@draft, #1}
68 \ifthenelse{\boolean{AM@frame}}
69     {\let\AM@fbox\fbox}{\let\AM@fbox\relax}
```

This loop arranges one or more PDF pages onto each sheet of paper.

```
70 \setcounter{AM@page}{\value{AM@firstpage}}
71 \whiledo{\value{AM@page} < \value{AM@lastpage}}{%
72 \newpage
73 \AddToShipoutPicture{%
74 \setlength{\unitlength}{1pt}
75 \setlength{\fboxsep}{Opt}
```

Arrange one PDF pages per sheet. (nup=1)

```
76 \ifthenelse{\value{AM@nup} = 1}
77     {\savebox{\AM@pagebox}{\AM@fbox{%
78 \includegraphics [page=\theAM@page, angle=\AM@lscap@rotii,
79 \setlength{\fboxsep}{Opt}
angle=\AM@rotate, #1]{#2}}%

```

```

80         \stepcounter{AM@page}}
81     {}
Arrange two PDF pages per sheet. (nup=2)
82     \ifthenelse{\value{AM@nup} = 2}{%
83         \def\AM@smallpage{%
84             \ifthenelse{\value{AM@page} < \value{AM@lastpage}}
85                 {\AM@fbox{%
86                     \includegraphics[page=\theAM@page, angle=90,
87                                     angle=\AM@landscape@roti, angle=\AM@rotate,
88                                     scale=.7071, #1]{#2}}
89                 {\phantom{\AM@fbox{%
90                     \includegraphics[page=\theAM@firstpage, angle=90,
91                                     angle=\AM@landscape@roti, angle=\AM@rotate,
92                                     scale=.7071, #1]{#2}}}}
93     \ifthenelse{\boolean{AM@openright}\and%
94                 \value{AM@page}=\value{AM@firstpage}}
95         {\savebox{\AM@pagebox}{%
96             \vbox{%
97                 \hbox{%
98                     \AM@smallpage%
99                     \stepcounter{AM@page}%
100                }% hbox
101                \nointerlineskip\vskip-\fboxrule\vskip\AM@deltay
102                \hbox{%
103                    \phantom{\AM@smallpage}%
104                }% hbox
105            }}}% vbox, savebox, ifthen
106         {\savebox{\AM@pagebox}{%
107             \vbox{%
108                 \hbox{%
109                     \ifthenelse{\boolean{AM@landscape}}
110                         {\AM@smallpage%
111                         \stepcounter{AM@page}}
112                         {\stepcounter{AM@page}%
113                         \AM@smallpage%
114                         \addtocounter{AM@page}{-1}}%
115                 }% hbox
116                \nointerlineskip\vskip-\fboxrule\vskip\AM@deltay
117                \hbox{%
118                    \ifthenelse{\boolean{AM@landscape}}
119                        {\AM@smallpage%
120                        \stepcounter{AM@page}}
121                        {\AM@smallpage%
122                        \addtocounter{AM@page}{2}}%
123                }% hbox
124            }}}% vbox, savebox, ifthen
125     {}
Arrange four PDF pages per sheet. (nup=4)
126     \ifthenelse{\value{AM@nup} = 4}{%

```

```

127 \def\AM@smallpage{%
128 \ifthenelse{\value{AM@page} < \value{AM@lastpage}}
129 {\AM@fbox{%
130 \includegraphics [page=\theAM@page,
131 angle=\AM@lscap@rotii, angle=\AM@rotate,
132 scale=.5, #1]{#2}}
133 {\phantom{\AM@fbox{%
134 \includegraphics [page=\theAM@firstpage,
135 angle=\AM@lscap@rotii, angle=\AM@rotate,
136 scale=.5, #1]{#2}}}}
137 \ifthenelse{\boolean{AM@openright}\and%
138 \value{AM@page}=\value{AM@firstpage}}
139 {\ifthenelse{\boolean{AM@landscape}}
140 {\savebox{\AM@pagebox}{%
141 \vbox{%
142 \hbox{%
143 \AM@smallpage%
144 \addtocounter{AM@page}{2}%
145 \hskip-\fboxrule\hskip\AM@deltax%
146 \AM@smallpage%
147 \addtocounter{AM@page}{-1}%
148 }% hbox
149 \nointerlineskip\vskip-\fboxrule\vskip\AM@deltay
150 \hbox{%
151 \phantom{\AM@smallpage}%
152 \hskip-\fboxrule\hskip\AM@deltax%
153 \AM@smallpage%
154 \addtocounter{AM@page}{2}%
155 }% hbox
156 }}}% vbox, savebox, ifthen
157 {\savebox{\AM@pagebox}{%
158 \vbox{%
159 \hbox{%
160 \phantom{\AM@smallpage}%
161 \hskip-\fboxrule\hskip\AM@deltax%
162 \AM@smallpage%
163 \stepcounter{AM@page}%
164 }% hbox
165 \nointerlineskip\vskip-\fboxrule\vskip\AM@deltay
166 \hbox{%
167 \AM@smallpage%
168 \stepcounter{AM@page}%
169 \hskip-\fboxrule\hskip\AM@deltax%
170 \AM@smallpage%
171 \stepcounter{AM@page}%
172 }% hbox
173 }}}% vbox, savebox, ifthen, ifthen
174 {\ifthenelse{\boolean{AM@landscape}}
175 {\savebox{\AM@pagebox}{%
176 \vbox{%

```

```

177         \hbox{%
178             \stepcounter{AM@page}%
179             \AM@smallpage%
180             \addtocounter{AM@page}{2}%
181             \hskip-\fboxrule\hskip\AM@deltax%
182             \AM@smallpage%
183             \addtocounter{AM@page}{-3}%
184         }% hbox
185         \nointerlineskip\vskip-\fboxrule\vskip\AM@deltay
186         \hbox{%
187             \AM@smallpage%
188             \addtocounter{AM@page}{2}%
189             \hskip-\fboxrule\hskip\AM@deltax%
190             \AM@smallpage%
191             \addtocounter{AM@page}{2}%
192         }% hbox
193     }}}} vbox, savebox, ifthen
194     {\savebox{\AM@pagebox}{%
195         \vbox{%
196             \hbox{%
197                 \AM@smallpage%
198                 \stepcounter{AM@page}%
199                 \hskip-\fboxrule\hskip\AM@deltax%
200                 \AM@smallpage%
201                 \stepcounter{AM@page}%
202             }% hbox
203             \nointerlineskip\vskip-\fboxrule\vskip\AM@deltay
204             \hbox{%
205                 \AM@smallpage%
206                 \stepcounter{AM@page}%
207                 \hskip-\fboxrule\hskip\AM@deltax%
208                 \AM@smallpage%
209                 \stepcounter{AM@page}%
210             }% hbox
211         }}}} vbox, savebox, ifthen, ifthen
212     }{}

```

Page output.

```

213     \def\AM@add{+}
214     \if@twoside\ifodd\value{page}\else\def\AM@add{-}\fi\fi
215     \def\AM@xpos{.5\paperwidth-.5\wd\AM@pagebox \AM@add\AM@xoffset}
216     \def\AM@ypos{.5\paperheight-.5\ht\AM@pagebox+%
217         .5\dp\AM@pagebox+\AM@yoffset}
218     \setlength{\@tempdima}{\AM@xpos}
219     \edef\AM@xpos{\strip@pt\@tempdima}
220     \setlength{\@tempdima}{\AM@ypos}
221     \edef\AM@ypos{\strip@pt\@tempdima}
222     \put(\AM@xpos,\AM@ypos){\usebox{\AM@pagebox}}
223     }% AddToShipoutPicture

```

Some pages are displayed in landscape format, if the turn-Option is ‘true’.

```

224 \ifthenelse{\boolean{AM@turn}}
225   {\ifthenelse{\boolean{AM@landscape}}
226     {\ifthenelse{\value{AM@nup}=1}{\pdfpageattr{/Rotate 90}}{}
227     \ifthenelse{\value{AM@nup}=4}{\pdfpageattr{/Rotate 90}}{}}
228     {\ifthenelse{\value{AM@nup}=2}{\pdfpageattr{/Rotate 90}}{}}
229   }{}

230 \AM@pagecommand
231 \hskipOpt
232 \newpage
233 \ClearShipoutPicture
234 }% whiledo

Write back saved counters.

235 \setcounter{AM@firstpage}{\saveAM@firstpage}
236 \setcounter{AM@lastpage}{\saveAM@lastpage}
237 \setcounter{AM@nup}{\saveAM@nup}

238 \endgroup% setkey
239 \ClearShipoutPicture
240 }% includepdf

```

## 4.6 Options of \includepdf

```

241 \define@key{Gin}{firstpage}{%
242     \setcounter{AM@firstpage}{#1}}
243 \define@key{Gin}{lastpage}{%
244     \setcounter{AM@lastpage}{#1}%
245     \stepcounter{AM@lastpage}}
246 \define@key{Gin}{offset}{\AM@parse@offset#1 \}
247 \define@key{Gin}{nup}{\setcounter{AM@nup}{#1}}
248 \define@key{Gin}{deltax}{\AM@parse@deltax#1 \}
249 \define@key{Gin}{deltay}{\AM@parse@deltay#1 \}
250 \define@key{Gin}{frame}[true]{\setboolean{AM@frame}{#1}}
251 \define@key{Gin}{openright}[true]{\setboolean{AM@openright}{#1}}
252 \define@key{Gin}{pagecommand}{\renewcommand{\AM@pagecommand}{#1}}
253 \define@key{Gin}{rotate}{\def\AM@rotate{#1}}
254 \define@key{Gin}{landscape}[true]{\setboolean{AM@landscape}{#1}%
255     \def\AM@lscape@roti{-90}%
256     \def\AM@lscape@rotii{90}}
257 \define@key{Gin}{turn}[true]{\setboolean{AM@turn}{#1}}

```

Parsing option entries. These lines are derived from `graphicx.sty`.

```

258 \def\AM@parse@deltax#1 #2\{\AMGin@defaultbp\AM@deltax{#1}}
259 \def\AM@parse@deltay#1 #2\{\AMGin@defaultbp\AM@deltay{#1}}

260 \def\AM@parse@offset#1 #2 #3\{\%
261     \AMGin@defaultbp\AM@xoffset{#1}%
262     \AMGin@defaultbp\AM@yoffset{#2}}

263 \def\AMGin@defaultbp#1#2{\%
264     \afterassignment\AMGin@def@bp\dimen@#2bp\relax{#1}{#2}}

```

```
265 \def\AMGin@def@bp#1\relax#2#3{%
266   \if!#1!%
267     \def#2{#3bp}%
268   \else
269     \dimen@.99626\dimen@
270     \edef#2{\strip@pt\dimen@ bp}%
271   \fi}
```

## 5 History

### 2001/01/17 v0.1b

- The first released version.

### 2001/02/03 v0.1c

- Pages are being centered automatically.

### 2001/03/10 v0.1e

- Support of landscape format.
- The `offset` option recognizes the class option `twoside`.