

# The pst-pdf package\*

Rolf Niepraschk<sup>†</sup>      Hubert Gäßlein

2008/10/09

## 1 Introduction

The package `pst-pdf` simplifies the use of graphics from PSTricks and other PostScript code in PDF documents. As in building a bibliography with `BIBTEX` additional external programmes are being invoked. In this case they are used to create a PDF file (`\PDFcontainer`) that will contain all this graphics material. In the final document this contents will be inserted instead of the original PostScript code.

## 2 Usage

### 2.1 Package options

**active** Activates the extraction mode (DVI output). An explicit declaration usually is not necessary (default in `LATEX` mode).

**inactive** No special actions; only the packages `pstricks` and `graphicx` are loaded (default in `VTEX`). Can be used to just convert the document with `LATEX` into a DVI file while avoiding the automatic extraction mode.

**pstricks** The package `pstricks` is loaded (default).

**nopstricks** The package `pstricks` does not get loaded. Once it is detected that `pstricks` was loaded however in some other way, the `pspicture` environment is treated as if the option “`pstricks`” was given.

**draft** From the `\PDFcontainer` file included graphics is displayed as frame in `pdfLATEX` mode.

**final** From the `\PDFcontainer` file included graphics is correctly displayed in `pdfLATEX` mode (default).

**tightpage** The graphics’ dimensions in the `\PDFcontainer` file match exactly those of the corresponding `TEX` boxes (default).

**notightpage** The dimensions of the `TEX` box corresponding to its graphics is not always correct, since a PostScript statement can draw outside its box. The option “`notightpage`” makes the graphics in the `\PDFcontainer` file to be at

---

\*This document corresponds to `pst-pdf` v1.1v, dated 2008/10/09. Thanks to Peter Dyballa for the translation.

<sup>†</sup>`Rolf.Niepraschk@gmx.de`

least the size of the whole page. To be able to make use of the graphics' in a later pdfL<sup>A</sup>T<sub>E</sub>X run, the `\PDFcontainer` file needs to be finished in a way that each graphics gets reduced in size to its visible part. For this an external programme like `pdfcrop`<sup>1</sup> can be useful. Its use can save declaring the option “trim” (see also section 2.4).

**displaymath** In PDF mode the mathematical environments `displaymath`, `eqnarray`, and `$$` get also extracted and included as graphics. This way additional PSTricks extensions can easily be added to the contents of these environments. (Question: how do AMSL<sup>A</sup>T<sub>E</sub>X environments behave?)

**⟨other⟩** All other options are passed to `psctricks` package.

## 2.2 Program calls

The following table shows the course necessary to create a PDF document containing PostScript graphics<sup>2</sup>. As comparison the analogous course for a bibliography is shown.

PostScript graphics	bibliography
<code>pdflatex document.tex</code>	<code>pdflatex document.tex</code>
<i>auxiliary calls</i>	
<code>latex document.tex</code>	
<code>dvips -o document-pics.ps document.dvi</code>	
<code>ps2pdf document-pics.ps</code>	<code>bibtex document.aux</code>
<code>pdflatex document.tex</code>	<code>pdflatex document.tex</code>

While creating the output only code from inside a `pspicture` or `postscript` environment is considered. PostScript graphics files, which are passed as parameter of an `\includegraphics` statement, too are included into the `\PDFcontainer` file. This file's name is by default `\jobname-pics.pdf`. It can be changed by re-defining the macro `\PDFcontainer`.

## 2.3 User commands

`pspicture`      `\begin{pspicture}[\langle keys \rangle] (\langle x0,x1 \rangle) (\langle y0,y1 \rangle) ... \end{pspicture}`  
 The `pspicture` environment is not available when the option “nopstricks” was given. It is to be used the same way as if in PSTricks. In pdfL<sup>A</sup>T<sub>E</sub>X mode this environment's contents is only displayed when the `\PDFcontainer` file was created before.

`postscript`    `\begin{postscript}[\langle keys \rangle] ... \end{postscript}`  
 The `postscript` environment can contain any code except floats. In pdfL<sup>A</sup>T<sub>E</sub>X mode its contents is take too off the `\PDFcontainer` file. Other as in the `pspicture` environment the necessary space is not always preserved when the `\PDFcontainer` file does not exist yet.

`\includegraphics`      `\includegraphics[\langle keys \rangle]{\langle filename \rangle}`

<sup>1</sup>CTAN: support/pdfcrop/

<sup>2</sup>The T<sub>E</sub>X distribution “teT<sub>E</sub>X” contains a UNIX shell script `ps4pdf` which executes all the necessary steps. See: CTAN: macros/latex/contrib/ps4pdf/

To be used as in `graphics/graphicx` defined. In pdfL<sup>A</sup>T<sub>E</sub>X mode it is now additionally feasible to pass the name of an EPS file. Its visible contents too is taken from the `\PDFcontainer` file.

<code>\includegraphics</code>	<code>\includegraphics[<i>keys</i>](<i>pfxadd</i>)&lt;<i>ovpfgd</i>&gt;[<i>ovpbgd</i>]{<i>filename</i>}</code> Wie im Paket <code>psfragx</code> definiert zu verwenden.
<code>\savepicture</code>	<code>\savepicture{<i>name</i>}</code> The last output graphics (result of the <code>pspicture</code> or <code>postscript</code> environments or the <code>\includegraphics</code> statement with an PostScript file as argument) is being saved in a file under the name as given by the parameter.
<code>\usepicture</code>	<code>\usepicture[<i>keys</i>]{<i>name</i>}</code> Die zuvor mit <code>\savepicture</code> gespeicherte Grafik wird ausgegeben. Der optionale Parameter entspricht dem bei der Anweisung <code>\includegraphics</code> möglichen.
<code>pst-pdf-defs</code>	<code>\begin{pst-pdf-defs} ... \end{pst-pdf-defs}</code> Sollen eigene Makros oder Umgebungen definiert werden, die das Zeichen <code>&amp;</code> (andere?) im Ersetzungstext enthalten, so müssen diese Definitionen von der Umgebung <code>pst-pdf-defs</code> umschlossen werden.

## 2.4 Command options

The behaviour of the `\includegraphics` and `\usepicture` statements and the `postscript` environment can be modified with any of the following parameters (key value syntax):

- frame**=`<true|false>` As with the `\fbox` statement a frame is drawn around the graphics. Any change of size due to rotation is taken into account. Drawing happens in pdfL<sup>A</sup>T<sub>E</sub>X mode; before, in creating the `\PDFcontainer` file, it is ignored. Default: `false`.
- innerframe**=`<true|false>` As in “`frame`”, but the frame is drawn around the graphics, not its box.
- ignore**=`<true|false>` If “`true`” no graphics is output. With `\savepicture{name}` the graphics can be used later in a different place via `\usepicture`. Default: `false`.
- showname**=`<true|false>` A caption of minimal font size records the used file’s name. Default: `false`.
- namefont**=`<font commands>` Controls the font used when “`showname=true`” is set. Default: `\ttfamily\tiny`

All parameters can be set globally as in `\setkeys{Gin}{key=value}`.

## 3 Implementation

1 `{*package}`

### 3.1 Package options

2 `\newcommand*\ppf@TeX@mode{-1}`

```

3 \newcommand*\ppf@draft{false}
4 \newif\if@ppf@PST@used\@ppf@PST@usedtrue
5 \newif\if@ppf@tightpage \@ppf@tightpagetrue
6 \DeclareOption{active}{\OptionNotUsed}
7 \DeclareOption{inactive}{\def\ppf@TeX@mode{9}}
8 \DeclareOption{ignore}{\def\ppf@TeX@mode{999}}
9 \DeclareOption{pstricks}{\@ppf@PST@usedtrue}
10 \DeclareOption{nopstricks}{\@ppf@PST@usedfalse}
11 \DeclareOption{displaymath}{%
12   \PassOptionsToPackage\CurrentOption{preview}}
13 \DeclareOption{draft}{\def\ppf@draft{true}}
14 \DeclareOption{final}{\def\ppf@draft{false}}%
15   \PassOptionsToPackage\CurrentOption{graphicx}}

16 \DeclareOption{notightpage}{\@ppf@tightpagefalse}%
17 \DeclareOption{tightpage}{\@ppf@tightpagetrue}%
18 \DeclareOption*{%
19   \PassOptionsToPackage\CurrentOption{pstricks}}
20 \ProcessOptions\relax
21 \ifnum\ppf@TeX@mode=999\relax\expandafter\endinput\fi

```

### 3.2 Compiler tests

It is tested which  $\TeX$  compiler in which mode of operation is actually used (see ‘graphics.cfg’ in  $\text{te}\TeX/\text{TeX}$  Live). Accordingly the environments `pspicture` and `postscript` gain each a different range of functions. This test is only executed when the options `active` or `inactive` were not given.

```

22 \ifnum\ppf@TeX@mode=-1\relax
23   \begingroup
Default ( $\TeX$  with a dvi-to-ps converter)
24   \chardef\x=0 %
Check pdf $\TeX$ 
25   \@ifundefined{pdfoutput}{}{%
26     \ifcase\pdfoutput\else
27       \chardef\x=1 %
28       \fi
29   }%
Check V $\TeX$ 
30   \@ifundefined{OpMode}{}{\chardef\x=2 }%
31   \expandafter\endgroup
32   \ifcase\x
⇒ DVI mode
33   \def\ppf@TeX@mode{0}%
34   \or
⇒ pdf $\TeX$  is running in PDF mode
35   \def\ppf@TeX@mode{1}%
36   \else
⇒ V $\TeX$  is running
37   \def\ppf@TeX@mode{9}%
38   \fi
39 \fi

```

```

40 \newcommand*{\PDFcontainer{}}
41 \edef\PDFcontainer{\jobname-pics.pdf}
42 \newcounter{pspicture}
43 \newcommand*\ppf@other@extensions[1]{}
44 \newcommand*\usepicture[2][1]{}
45 \newcommand*\savepicture[1]{}

```

pst-pdf-defs

```

46 \newenvironment*{pst-pdf-defs}{%
47   \endgroup
48   %   ??? \@currentline
49 }{%
50   \begingroup
51   \def\@currentvir{pst-pdf-defs}%
52 }

53 \RequirePackage{graphicx}%
54 \let\ppf@Gininclude@graphics\Gininclude@graphics
55 \let\ppf@Gin@extensions\Gin@extensions
56 \let\ppf@Gin@ii\Gin@ii

57 \newif\if@ppf@pdftex@graphic
58 \newif\if@Gin@frame\Gin@framefalse
59 \newif\if@Gin@innerframe\Gin@innerframefalse
60 \newif\if@Gin@showname\Gin@shownamefalse
61 \newif\if@Gin@ignore\Gin@ignorefalse

```

\ifpr@outer in fact is defined in package preview. We have to do it here too since otherwise T<sub>E</sub>X could “stumble and fall” while parsing the \ifcase structure.

```
62 \newif\ifpr@outer
```

\ppf@is@pdfTeX@graphic

Parameter #1 is the name of a graphics file with or without extension, parameter #2 contains the valid extensions in PDF mode, parameter #3 contains the valid extensions in DVI mode. If it works to process the graphics in PDF mode, then the statements in #4 are executed, otherwise those in #5.

```

63 \newcommand*\ppf@is@pdfTeX@graphic[5]{}
64   \@ppf@pdftex@graphicfalse%
65   \begingroup
66     \edef\pdfTeXext{#2}%

```

Instead of loading the found graphics, only a test on file name extension.

```

67   \def\Gin@setfile##1##2##3{%
68     \edef\@tempb{##2}%
69     \@for\@tempa:=\pdfTeXext\do{%
70       \ifx\@tempa\@tempb\global\@ppf@pdftex@graphictrue\fi}}%

```

File types for both modes need to be determined to prevent a wrong error message “File ‘#1’ not found”.

```

71   \edef\Gin@extensions{#2,#3}%

Trial invocation. Output is completely inhibited.

72   \pr@outerfalse\ppf@Gininclude@graphics{#1}%
73   \endgroup
74   \if@ppf@pdftex@graphic#4\else#5\fi
75 }

76 \ifcase\ppf@TeX@mode\relax

```

### 3.3 Extraction mode (DVI output)

The `pspicture` environment retains any definition from `pstricks.tex`. Only the code from the environments `pspicture` and `postscript` as well as `\includegraphics` with PostScript files leads to records into the DVI file. The remainder of the document's code is ignored for output. After conversion of the DVI file via PostScript (“`dvips`”) into PDF (`\PDFcontainer` file) each graphics takes exactly one page in the `\PDFcontainer` file. The  $\TeX$  compiler with DVI output and the package option “`active`” both force this mode.

```
77 \PackageInfo{pst-pdf}{%
78   MODE: \ppf@TeX@mode\space (dvi -- extraction mode)}
79 \nofiles
80 \let\makeindex\@empty \let\makeglossary\@empty
81 \AtBeginDocument{\overfullrule=\z@}%
82 \if@ppf@PST@used\RequirePackage{pstricks}\fi
83 \RequirePackage[active,dvips,tightpage]{preview}[2005/01/29]%
84 \newcommand*\ppf@PreviewBbAdjust{}
85 \newcommand*\ppf@RestoreBbAdjust{}
86 \let\PreviewBbAdjust\ppf@PreviewBbAdjust%
```

The pdf $\LaTeX$  mode compliant graphics file formats are needed too.

```
87 \begingroup
88 \let\AtBeginDocument\@gobble \let\PackageWarningNoLine\@gobbletwo
89 \chardef\pdfTeXversion=121 %
90 \newcount\pdfoutput
91 \pdfoutput=1 %
92 \input{pdftex.def}%
93 \edef\x{\endgroup\def\noexpand\ppf@other@extensions{\Gin@extensions}
94 }%
95 \x
```

In PDF mode no rules must be defined for its compliant (PNG, JPEG, PDF) graphics file formats (because of for example ‘`dvips`’ extensions). The universal EPS rule is used to at least find these files.

```
96 \AtBeginDocument{%
97   \@ifpackageloaded{keyval}{%
98     \def\KV@errx#1{\PackageInfo{keyval}{#1}}%
99   }{}%
100  \@ifpackageloaded{xkeyval}{%
101    \def\XKV@err#1{\PackageInfo{xkeyval}{#1}}%
102  }{}%
```

In this mode undefined keys should not be an error.

```
103 \@for\@tempa:=\ppf@other@extensions\do{%
104   \expandafter\let\csname Gin@rule@\@tempa\endcsname\relax}%
105 \DeclareGraphicsRule*{eps}{*}{*}{*}%
```

No function in this mode.

```
106 \define@key{Gin}{innerframe}[true]{}%
107 \define@key{Gin}{frame}[true]{}%
108 \define@key{Gin}{ignore}[true]{}%
109 \define@key{Gin}{showname}[true]{}%
110 \define@key{Gin}{namefont}{}%
111 \@ifundefined{GPT@page}{\define@key{Gin}{page}{}{}}
```

```

112 \if@ppf@tightpage\else
113 \def\PreviewBbAdjust{%
114     -600pt -600pt 600pt 600pt}%
115 \AtEndDocument{%
116     \PackageWarningNoLine{pst-pdf}{Picture container needs cropping.}}%
117 \fi

```

**postscript** The postscript environment utilises the trim option in the same manner as does `\includegraphics` (any specification without dimension is interpreted as if given in bp).

```

118 \newenvironment{postscript}[1] []%
119 {%
120     \global\let\ppf@PreviewBbAdjust\PreviewBbAdjust
121     \if@ppf@tightpage
122         \begingroup
123             \setkeys{Gin}{#1}%
124             \xdef\PreviewBbAdjust{%
125                 -\Gin@vllx bp -\Gin@villy bp \Gin@vurx bp \Gin@vury bp}%
126         \endgroup
127     \fi
128     \ignorespaces
129 }%
130 {\aftergroup\ppf@RestoreBbAdjust}%

131 \PreviewEnvironment{postscript}%
132 \AtBeginDocument{%
133     \@ifundefined{PSTricksLoaded}{}%
134     {%

```

**pspicture** Announce preview original definition.

```

135     \PreviewEnvironment{pspicture}%

```

**psmatrix** Announce preview original definition.

```

136     \@ifundefined{psmatrix}{}%
137     {%
138         \PreviewEnvironment{psmatrix}%
139         \newcommand*\ppf@set@mode{}%
140         \newcommand*\ppf@test@mmode{%
141             \ifmmode
142                 \ifinner
143                     \let\ppf@set@mode=$%
144                 \else
145                     \def\ppf@set@mode{$$}%
146                 \fi
147             \else
148                 \let\ppf@set@mode=\@empty
149             \fi
150         }%
151         \let\ppf@psmatrix=\psmatrix
152         \expandafter\let\expandafter\ppf@pr@psmatrix%
153         \expandafter=\csname pr@\string\psmatrix\endcsname
154         \let\ppf@endpsmatrix=\endpsmatrix
155         \def\psmatrix{\ppf@test@mmode\ppf@psmatrix}
156         \expandafter\def\csname pr@\string\psmatrix\endcsname{%

```

```

157     \ppf@set@mode\ppf@pr@psmatrix}%
158     \def\endpsmatrix{\ppf@endpsmatrix\ppf@set@mode}%
159     }%

```

Announce internal macro `\pst@object` to enable the use of some PSTricks code outside of `pspicture` environments. At the moment invocations of the following kind are feasible:

```

\pst@object {<m>}<*>[<o>]{<o>}{<o>}<(o)><(o)><(o)>
(m = necessary, * = optional, o = optional)

```

More than three optional arguments at the call's end, as in `\psline` possible, do not work yet.

```

160     \PreviewMacro[{}*[]%
161     ?\bgroup{#{#1}{#1}}{}%
162     ?\bgroup{#{#1}{#1}}{}%
163     ?(#{#1}){({#1})}{}%
164     ?(#{#1}){({#1})}{}%
165     ?(#{#1}){({#1})}{}%
166     ]{\pst@object}

```

Prevent multiple test-wise setting of table contents by “`tabularx`”.

```

167     \@ifundefined{tabularx}{}{%
168     \newcolumntype{X}{c}%
169     \expandafter\let\expandafter\tabularx\csname tabular*\endcsname
170     \expandafter\let\expandafter\endtabularx\csname endtabular*\endcsname
171     }%

```

Support of `\includegraphicx` from the package `psfragx`.

```

172     \@ifundefined{pfx@includegraphicx}{}{%
173     \PreviewMacro[{}{}]{\pfx@includegraphicx}}%
174     }%

```

`\Gscale@@box` Disable scaling.

```

175     \def\Gscale@@box#1#2#3{%
176     \toks@{\mbox}%
177     }

```

`\Ginclude@graphics` All graphics content of well known format (for instance EPS files) is treated in a regular way, which in this mode denotes that it is subject to preview functions. Other graphics content (for instance PDF files) is ignored.

```

178     \def\Ginclude@graphics#1{%
179     \ifpr@outer

```

Generally pdf $\TeX$  supported graphics formats are intended to be preferred (inclusion in final pdf $\TeX$  run). If it's a PostScript type graphics, then the original definition is in function again and registration for the `preview` package is necessary in order to convert this PostScript type graphics into PDF.

```

180     \ppf@is@pdfTeX@graphic{#1}{\ppf@other@extensions}{\Gin@extensions}%

```

Dummy box to prevent a division by zero while scaling or rotating. Otherwise ignored.

```

181     {\rule{10pt}{10pt}}%
182     {\ppf@Ginclude@graphics{#1}}%
183     \else

```



Inside a PostScript environment (pspicture etc.) `\includegraphics` has to behave as in its original definition (only DVIPS supported graphics formats are allowed).

```

184     \ppf@Ginclude@graphics{#1}%
185     \fi
186   }%

187   \PreviewMacro[{}]{\ppf@Ginclude@graphics}%
188   \let\pdfliteral@gobble%
189 \or

```

### 3.4 pdf $\LaTeX$ mode (PDF output)

When the `\PDFcontainer` file (default: `\jobname`-pics.pdf) exists, the contents of the environments `pspicture` and `postscript` is ignored. Instead the corresponding graphics from the `\PDFcontainer` file is used.

```

190   \PackageInfo{pst-pdf}{MODE: \ppf@TeX@mode\space (pdfTeX mode)}%
   Prevent pdf $\TeX$ 's message Non-PDF special ignored!.

191   \ifppf@PST@used
192     \let\ppf@temp@AtBeginDvi\let\AtBeginDvi@gobble
193     \RequirePackage{pstricks}\let\AtBeginDvi\ppf@temp
194   \fi

195   \@temptokena{%
196     \let\Gin@PS@file@header@gobble\let\Gin@PS@literal@header@gobble
197     \let\Gin@PS@raw@gobble\let\Gin@PS@restored@gobble
198     \@ifundefined{PSTricksLoaded}{-}{%

```

Necessary if `PSTricks < 2.0`.

```

199     \PSTricksOff
200     \@ifundefined{color@to@ps}{\def\color@to@ps#1 #2@@{}-}}}%

```

PostScript output is now inhibited and later once again.

```

201   \the\@temptokena
202   \expandafter\AtBeginDocument\expandafter
203     {\the\@temptokena\@temptokena{}}%
204   \@ifundefined{PSTricksLoaded}{-}{%

```

To parse the arguments of `PSTricks'` `\pst@object` we load `preview` in active mode, but restore the default definitions of `\output` and `\shipout`. `\pr@startbox` and `\pr@endbox` serve here only to disable `\pst@object` and to load the corresponding graphics from the `\PDFcontainer` file. At present a maximum of three optional parameters in round braces (parenthesis) at the end of `\pst@object` is supported, which is sufficient, but not always enough.

```

205   \newtoks\ppf@output
206   \ppf@output\expandafter{\the\output}%
207   \let\ppf@nofiles=\nofiles \let\nofiles=\relax
208   \let\ppf@shipout=\shipout
209   \RequirePackage[active]{preview}[2005/01/29]%
210   \let\shipout=\ppf@shipout \let\ppf@shipout=\relax
211   \let\nofiles=\ppf@nofiles \let\ppf@nofiles=\relax
212   \output\expandafter{\the\ppf@output} \ppf@output{}%

   \pr@startbox, \pr@endbox: simpler over original definitions.

213   \long\def\pr@startbox#1#2{%

```

```

214     \ifpr@outer
215     \toks@{#2}%
216     \edef\pr@cleanup{\the\toks@}%
217     \setbox\@tempboxa\ vbox\bgroup
218     \everydisplay{}%
219     \pr@outerfalse%
220     \expandafter\@firstofone
221     \else
222     \expandafter\@gobble
223     \fi{#1}}%
224 \def\pr@endbox{%
225     \egroup
226     \setbox\@tempboxa\ box\voidb@x
227     \ppf@getpicture
228     \pr@cleanup}%

```

(See also the identical definition in DVI mode.)

```

229     \AtBeginDocument{%
230     \ifundefined{pst@object}{}%
231     {%
232     \PreviewMacro[{}* []%
233     ?\bgroup{#1}{#1}}{}%
234     ?\bgroup{#1}{#1}}{}%
235     ?({#1}){({#1})}}{}%
236     ?({#1}){({#1})}}{}%
237     ?({#1}){({#1})}}{}%
238     }]{\pst@object}}%
239     }%
240 }%

```

Too the supported file name extensions from DVI mode are needed.

```

241 \begingroup
242 \input{dvips.def}%
243 \edef\x{\endgroup\def\noexpand\ppf@other@extensions{\Gin@extensions}}%
244 \x

```

Dummy definition for in DVI mode supported file formats.

```

245 \DeclareGraphicsRule{*}{eps}{*}{}%
246 \define@key{Gin}{innerframe}[true]{%
247 \lowercase{\Gin@boolkey{#1}}{innerframe}}%
248 \define@key{Gin}{frame}[true]{%
249 \lowercase{\Gin@boolkey{#1}}{frame}}%
250 \define@key{Gin}{ignore}[true]{%
251 \lowercase{\Gin@boolkey{#1}}{ignore}}%
252 \define@key{Gin}{frame@}@{}%

```

(For internal use only!)

```

253 \edef\@tempa{\toks@{\noexpand\frame{\the\toks@}}}%
254 \ifcase#1\relax
255 \ifGin@innerframe\else\let\@tempa\relax\fi
256 \or
257 \ifGin@frame\else\let\@tempa\relax\fi
258 \fi
259 \@tempa
260 }%

```

```

261 \define@key{Gin}{showname}[true]{%
262   \lowercase{\Gin@boolkey{#1}}{showname}}%
263 \define@key{Gin}{namefont}{%
264   \begingroup
265     \@temptokena\expandafter{\ppf@namefont#1}%
266     \edef\x{\endgroup\def\noexpand\ppf@namefont{\the\@temptokena}}%
267   \x
268 }%
269 \newcommand*\ppf@filename{}%
270 \newcommand*\ppf@namefont{\tiny\ttfamily}%
271 \newcommand*\ppf@Gin@keys{}%
272 \let\ppf@Gin@setfile\Gin@setfile

```

`\Gin@setfile` Save real file name and, if applicable, page number for later use.

```

273 \def\Gin@setfile#1#2#3{\ppf@Gin@setfile{#1}{#2}{#3}%
274   \xdef\ppf@filename{%
275     #3\ifx\GPT@page\@empty\else(\GPT@page)\fi}}%

```

`\Gin@ii` Examine the options “frame”, “ignore”, etc. as soon as other special cases.

```

276 \def\Gin@ii[#1]#2{%
277   \begingroup

```

The value of `\ifGin@innerframe` has to be known before the inner frame is drawn. The values for `\ifGin@showname` and `\ppf@namefont` need to be available after rendering the graphics too. Thus beforehand and protected inside a group examine the options.

```

278   \@temptokena{#1}\def\ppf@tempb{#2}%

```

Finds empty file name when calling `\usepicture`.

```

279   \ifx\ppf@tempb\@empty\else
280     \ppf@is@pdfTeX@graphic{#2}{\Gin@extensions}{\ppf@other@extensions}%

```

Graphics out of `\PDFcontainer` are complete – scaled, rotated, etc. Don’t apply these things again and therefore ignore the optional parameters.

```

281   {%
282     \setkeys{Gin}{#1}%
283     \ifx\ppf@tempb\PDFcontainer
284       \@temptokena{page=\GPT@page}%
285     \fi
286   }%
287   {%
288     \refstepcounter{pspicture}%
289     \@temptokena{page=\the\c@pspicture}\def\ppf@tempb{\PDFcontainer}%
290   }%
291   \fi
292   \ifGin@ignore\else

```

“frame@@=0” = inner frame, “frame@@=1” = outer frame.

```

293     \edef\@tempa{\noexpand\ppf@Gin@ii[frame@@=0,\the\@temptokena,
294       frame@@=1]{\ppf@tempb}}%
295   \@tempa
296   \ifGin@showname
297     \ppf@namefont
298     \raisebox{-\ht\strutbox}[0pt][0pt]{\llap{\ppf@filename}}%
299   \gdef\ppf@filename{}%

```

```

300     \fi
301     \fi
302   \endgroup
303 }%

304 \IfFileExists{\PDFcontainer}%
305 {%

```

`\ppf@container@max` The number of pages as contained in `\PDFcontainer` file.

```

306   \pdfximage{\PDFcontainer}%
307   \edef\ppf@container@max{\the\pdflastximagepages}%

308   \AtEndDocument{%
309     \ifnum\c@pspicture>\z@

    A warning only makes sense when a graphics is needed at all.

310     \ifnum\c@pspicture=\ppf@container@max\else
311       \PackageWarningNoLine{pst-pdf}{%
312         '\PDFcontainer' contains \ppf@container@max\space pages
313         \MessageBreak but \the\c@pspicture\space pages are requested:
314         \MessageBreak File '\PDFcontainer' is no more valid!
315         \MessageBreak Recreate it
316       }%
317     \fi
318   \fi
319 }%
320 }%
321 {%
322   \def\ppf@container@max{0}%
323   \AtEndDocument{%
324     \ifnum\c@pspicture>\z@
325       \filename@parse{\PDFcontainer}%
326       \PackageWarningNoLine{pst-pdf}{%
327         File '\PDFcontainer' not found.\MessageBreak
328         Use the following commands to create it:\MessageBreak
329         -----
330         \MessageBreak
331         latex \jobname.tex\MessageBreak
332         dvips -o \filename@base.ps \jobname.dvi\MessageBreak
333         ps2pdf \filename@base.ps\MessageBreak
334         -----
335       }%
336     \fi
337   }%
338 }%

```

`\ppf@isnum` If parameter #1 is numeric, the instructions in #2, otherwise those in #3 are executed (see `bibtopic.sty`).

```

339   \newcommand\ppf@isnum[1]{%
340     \if!\ifnum9<1#1!\else_\fi\expandafter\@firstoftwo
341     \else\expandafter\@secondoftwo\fi}%

```

`psmatrix` Both environments ignore their contents and load instead the corresponding graphics out of the `\PDFcontainer` file. The value of the herein used `pspicture` counter's value can be used in `\label/\ref`.

postscript

```
342 \newcommand*\ppf@set@mode{}%
343 \newcommand*\ppf@test@mode{}%
344 \ifmode
345 \ifinner
346 \let\ppf@set@mode=$%
347 \else
348 \def\ppf@set@mode{$$}%
349 \fi
350 \else
351 \let\ppf@set@mode=@empty
352 \fi
353 }

354 \RequirePackage{environ}%
355 \newenvironment{postscript}[1] []{%
356 \def\@tempa{postscript}%
357 \ifx\@tempa\@currenvr
358 \def\ppf@Gin@keys{#1}%
359 \else
360 \def\ppf@Gin@keys{}%
361 \fi
362 \ppf@@getpicture
363 \Collect@Body@gobble}{}%
364 \AtBeginDocument{%
365 \@ifundefined{PSTricksLoaded}{}{%
366 \def\pst@@picture[#1](#2,#3)(#4,#5){\postscript}%
367 \def\endpspicture{\endpostscript\endgroup}%
368 \@ifundefined{psmatrix}{}{%
369 \let\psmatrix=\postscript
370 \let\endpsmatrix=\endpostscript}%
371 }%
372 \@ifundefined{pfx@includegraphicx}{}{%

The useless redefinition of \includegraphics in pdfTeX mode (package psfrag)
is leading to double insertion of the result. We go back to the original meaning.
373 \let\includegraphics=\pfx@includegraphics
374 \def\pfx@includegraphicx#1#2{\ppf@@getpicture}%
375 }%
376 }%
```

`\savepicture` Saves the recent graphics' number in a macro named `\ppf@@@#1`.

```
377 \def\savepicture#1{%
378 \expandafter\xdef\csname ppf@@@#1\endcsname{\the\pdfastximage}}%
```

`\usepicture` Inserts graphics with symbolic name #2. This name has to be declared beforehand in `\savepicture{<name>}`. Instead of a name a number can be used too, which directly addresses a graphics in the `\PDFcontainer` file. The optional parameter #1 corresponds to the one in `\includegraphics`.

```
379 \renewcommand*\usepicture[2] []{%
380 \@ifundefined{ppf@@@#2}%
381 {%
382 \ppf@isnum{#2}%
383 {\ppf@getpicture{#1}{#2}}%
```

```

384     {\@latex@error{picture ‘#2’ undefined}\@ehc}%
385   }%
386   {%
387     \begingroup
388     \def\Gininclude@graphics##1{%
389       \xdef\ppf@filename{#2}%
390       \setbox\z@\hbox{\pdfrefximage\@nameuse{ppf@@@#2}}%
391       \Gin@nat@height\ht\z@ \Gin@nat@width\wd\z@
392       \def\Gin@llx{0} \let\Gin@lly\Gin@llx
393       \Gin@defaultbp\Gin@urx{\Gin@nat@width}%
394       \Gin@defaultbp\Gin@ury{\Gin@nat@height}%
395       \Gin@bboxtrue\Gin@viewport@code
396       \Gin@nat@height\Gin@ury bp%
397       \advance\Gin@nat@height-\Gin@lly bp%
398       \Gin@nat@width\Gin@urx bp%
399       \advance\Gin@nat@width-\Gin@llx bp%
400       \Gin@req@sizes
401       \ht\z@\Gin@req@height \wd\z@\Gin@req@width
402       \leavevmode\box\z@}%
403     \define@key{Gin}{-type}{-}%
404     \includegraphics[scale=1,#1]{-}%
405     \endgroup
406   }}%

```

`\ppf@getpicture` Inserts the page (graphics) with number #2 from the `\PDFcontainer` file. Parameter #1: any option as in `\includegraphics`.

```

407 \newcommand*\ppf@getpicture[2]{%
408   \@tempcnta=#2\relax%
409   \ifnum\@tempcnta>\ppf@container@max
410     \PackageWarningNoLine{pst-pdf}{%
411       pspicture No. \the\@tempcnta\space undefined}%
412   \else
413     \includegraphics[draft=\ppf@draft,#1,page=\the\@tempcnta]%
414       {\PDFcontainer}%
415   \fi
416   \gdef\ppf@Gin@keys{-}%

```

`\ppf@@getpicture` Inserts next page (graphics) from the `\PDFcontainer` file.

```

417 \newcommand*\ppf@@getpicture{%
418   \ifpr@outer
419     \refstepcounter{pspicture}%
420     \expandafter\ppf@getpicture\expandafter{\ppf@Gin@keys}%
421     {\the\c@pspicture}%
422   \fi}%

```

`pst-pdf-defs` Environment without grouping. The character & has the catcode “other”. Useful for user-defined macro definitions with e.g. `psmatrix` inside.

```

423 \renewenvironment*{pst-pdf-defs}%
424   {%
425     \endgroup
426   % ??? \@currentvline
427   \chardef\ppf@temp=\catcode‘\&%
428   \@makeother\&%
429   }%

```

```

430 \catcode'\&=\ppf@temp
431 \begingroup
432 \def\@currenvir{pst-pdf-defs}%
433 }
434 \else

```

### 3.5 Inactive Mode

Only the packages `pstricks` and `graphicx` are loaded – no further exertion of influence. The package option “inactive” as soon as the  $\text{V}\text{T}\text{E}\text{X}$  compiler force this mode.

```

435 \PackageInfo{pst-pdf}{MODE: \ppf@TeX@mode\space (inactive mode)}%
436 \newenvironment{postscript}[1][\ignorespaces]{}
437 \let\ppf@is@pdfTeX@graphic\relax
438 \fi

439 \InputIfFileExists{pst-pdf.cfg}{%
440 \PackageInfo{pst-pdf}{Local config file pst-pdf.cfg used}}{}
441 \</package>

```

## Change History

v1.0a	General: Initial version. . . . . 1	v1.0g	General: Definition of <code>\PDFcontainer</code> now with <code>\edef</code> . (RN) . . . . . 5
v1.0b	General: Some code and documentation cleaning. (RN) . . . . . 1		<code>\usepicture</code> : Now <code>\usepspicture</code> does accept a numerical parameter. (RN) . . . . . 13
v1.0c	General: New options “pstricks”, “nopstricks”, “draft” and “final”. (RN) . . . . . 3	v1.0h	<code>psmatrix</code> : Based no more on the comment environment from the verbatim package. (RN) . . . . 12
v1.0d	General: Redefinition of <code>\includegraphics</code> in modes 0 und 1. Now using of eps graphics directly in pdf $\text{L}\text{A}\text{T}\text{E}\text{X}$ is possible. (RN) . . . . . 1	v1.0i	<code>\ppf@is@pdfTeX@graphic</code> : No more errors for given files without extensions. (RN) . . . . . 5
v1.0e	<code>postscript</code> : “trim” option added. (RN) . . . . . 7	v1.0j	General: Check <code>AtBeginDocument</code> for package ‘pstricks’ even if “nopstricks” is given. (RN) . . . 1
v1.0f	General: Config file loading added. (RN) . . . . . 15	v1.0k	<code>\Ginsetfile</code> : Show also the page number if exists. (RN) . . . . 11
	<code>\savepicture</code> : New macro <code>\savepspicture</code> . (RN) . . . . 13		<code>\Gincludgraphics</code> : Prevent division by zero. (RN) . . . . . 8
	<code>\usepicture</code> : New macro <code>\usepspicture</code> . Useful for putting a PSTricks graphic in a box or something else. (RN) 13	v1.0l	General: Options “framesep”, “framerule”, “linewidth” removed, “fname” and “innerframe” added. (RN) . . . . . 1

v1.0m	General: New package option “notightpage” added. (RN) . . .	1	v1.1e	General: New option “display-math” (see preview package). (HjG/RN) . . . . .	3
v1.0n	General: Changed macro names ( <code>\savepicture</code> and <code>\usepicture</code> ). (RN) . . . . .	1	v1.1f	General: Package option “ignore” reimplemented. Now the compilation of the dtx file in L <sup>A</sup> T <sub>E</sub> X mode is possible. (RN) . . . . .	3
	Some code cleaning. (RN) . . . . .	1	v1.1g	<code>postscript</code> : “psmatrix” environment (preserve math mode). (RN/HjG) . . . . .	13
v1.0o	General: New code for “notightpage”. (RN) . . . . .	7		<code>pspicture</code> : pspicture environment must still parse its arguments. (RN/HjG) . . . . .	12
	Option “fname” renamed to “showname”. (RN) . . . . .	1	v1.1h	<code>\Gininclude@graphics</code> : Check if inside of a PS-related environment (correct graphic inclusion). (RN) . . . . .	8
v1.0p	General: Some code and documentation cleaning. (RN) . . . . .	1	v1.1i	General: <code>\ifpr@outer</code> must be predefined. (HjG/RN) . . . . .	5
v1.0q	<code>\usepicture</code> : Now <code>\usepspicture</code> works for all kind of graphics. (RN) . . . . .	13		Package option “final” also for “graphicx”. (RN) . . . . .	4
v1.0r	<code>\ppf@is@pdfTeX@graphic</code> : Changed <code>\ppf@is@known@graphic</code> to <code>\ppf@is@pdfTeX@graphic</code> . Now pdfT <sub>E</sub> X graphics are preferred. (RN) . . . . .	5		<code>\Gininclude@graphics</code> : Correction of the inside check. (RN/HjG) . . .	8
v1.0s	General: Scaling e.g. of PostScript pictures now only in extraction mode. Some code cleaning. (RN) . . . . .	1	v1.1k	General: New environment <code>pst-pdfdefs</code> : Support for PSTricks environment “psmatrix” inside user definitions. (RN,HjG) . . . . .	1
	<code>\Gin@ii</code> : Rewritten. (RN) . . . . .	11	v1.1l	General: Support for the package “psfragx”. (RN) . . . . .	8
v1.1a	General: Support for the internal PSTricks macro <code>\pst@object</code> . (HjG/RN) . . . . .	8	v1.1m	General: Merge english and german version of the documentation. (RN) . . . . .	1
v1.1b	General: Ignore the call of <code>\nofiles</code> inside of <code>preview</code> . (RN) . . . . .	9	v1.1n	General: <code>\nofiles</code> added (suggestion of Torsten Bronger). . . . .	6
	Some code and documentation cleaning. (RN) . . . . .	1	v1.1o	<code>\Gscale@@box</code> : Disable scaling. (RN) . . . . .	8
v1.1c	General: New package option “tightpage” added. (RN) . . . . .	1	v1.1p	General: <code>\nofiles</code> makes <code>\makeindex</code> and <code>\makeglossary</code> to <code>\relax</code> . <code>\@empty</code> is better because of later <code>\renewcommand</code> ’s. . . . .	6
	Special support for “tabularx”. (RN) . . . . .	8	v1.1pl	General: <code>\let\output\@gobble</code> before loading of “preview” added.	
	Supress handling of pdfL <sup>A</sup> T <sub>E</sub> X graphic formats in DVI mode. (RN) . . . . .	6			
v1.1d	<code>postscript</code> : Support for PSTricks environment “psmatrix”. (RN) . . . . .	13			



	(RN) .....	9	“\let\output@gobble” be-		
v1.1q	General: Problem with “tabularx”		cause of bad side effects. (RN) .	9	
	and “threeparttable” solved.		<b>postscript:</b> Using <code>environ</code> the en-		
	(RN) .....	8	vironment <b>postscript</b> is now sim-		
v1.1r	General: Fixed values for		ple and more robust. (RN) ..	13	
	<code>\PreviewBbAdjust</code> because				
	<code>\paperwidth</code> is not always		v1.1u		
	defined (suggested by Will		General: <code>\pdfoutput</code> must be set		
	Robertson). .....	7	when loading “pdf <code>tex.def</code> ” in		
v1.1s	General: Dummy definition of the		DVI mode. (RN) .....	6	
	page key in DVI mode. ....	6	v1.1v		
v1.1t	General: Remove the line		General: Local redefinition of		
			<code>\pdfoutput</code> to be a counter.		
			(RN) .....	6	
			<code>\Gin@ii:</code> Key settings only for pdf		
			graphics. (RN) .....	11	

# Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols	pst-pdf-defs ...	I
<code>\&amp;</code> ..... 427, 428, 430	<code>\everydisplay</code> ..... 218	<code>\if@ppf@pdftex@graphic</code> ..... 57, 74
<code>\@currenenvir</code> 51, 357, 432		<code>\if@ppf@PST@used</code> .. ..... 4, 82, 191
<code>\@currenvrline</code> .. 48, 426	<b>F</b>	<code>\if@ppf@tightpage</code> . ..... 5, 112, 121
<code>\@ehc</code> ..... 384	<code>\filename@base</code> 332, 333	<code>\ifGin@frame</code> ... 58, 257
<code>\@firstofone</code> ..... 220	<code>\filename@parse</code> ... 325	<code>\ifGin@ignore</code> .. 61, 292
<code>\@ifpackageloaded</code> . ..... 97, 100	<code>\frame</code> ..... 253	<code>\ifGin@innerframe</code> . ..... 59, 255
<code>\@latex@error</code> ..... 384	<b>G</b>	<code>\ifGin@showname</code> 60, 296
<code>\@makeoother</code> ..... 428	<code>\Gin@bboxtrue</code> ..... 395	<code>\ifinner</code> ..... 142, 345
<code>\@ppf@PST@usedfalse</code> 10	<code>\Gin@boolkey</code> ..... ..... 247, 249, 251, 262	<code>\ifmmode</code> ..... 141, 344
<code>\@ppf@PST@usedtrue</code> 4, 9	<code>\Gin@defaultbp</code> 393, 394	<code>\ifpr@outer</code> ..... ..... 62, 179, 214, 418
<code>\@ppf@pdftex@graphicfalse</code> ..... 64	<code>\Gin@extensions</code> 55, 71, 93, 180, 243, 280	<code>\includegraphics</code> .. .. 2, 373, 404, 413
<code>\@ppf@pdftex@graphictrue</code> ..... 70	<code>\Gin@framefalse</code> ... 58	<code>\includegraphicsx</code> ... 3
<code>\@ppf@tightpagefalse</code> 16	<code>\Gin@ignorefalse</code> .. 61	
<code>\@ppf@tightpagetrue</code> ..... 5, 17	<code>\Gin@ii</code> ..... 56, <u>276</u>	<b>J</b>
	<code>\Gin@innerframefalse</code> 59	<code>\jobname</code> .. 41, 331, 332
<b>A</b>	<code>\Gin@llx</code> ..... 392, 399	<b>K</b>
<code>\AtBeginDvi</code> ... 192, 193	<code>\Gin@lly</code> ..... 392, 397	<code>\KV@errx</code> ..... 98
	<code>\Gin@nat@height</code> ... ..... 391, 394, 396, 397	<b>L</b>
<b>C</b>	<code>\Gin@nat@width</code> .... ..... 391, 393, 398, 399	<code>\leavevmode</code> ..... 402
<code>\color@to@ps</code> ..... 200	<code>\Gin@PS@file@header</code> 196	<code>\long</code> ..... 213
<code>\c@pspicture</code> 289, 309, 310, 313, 324, 421	<code>\Gin@PS@literal@header</code> ..... 196	<b>M</b>
<code>\catcode</code> ..... 427, 430	<code>\Gin@PS@raw</code> ..... 197	<code>\makeglossary</code> ..... 80
<code>\Collect@Body</code> ..... 363	<code>\Gin@PS@restored</code> .. 197	<code>\makeindex</code> ..... 80
<code>\CurrentOption</code> 12, 15, 19	<code>\Gin@req@height</code> ... 401	<code>\mbox</code> ..... 176
	<code>\Gin@req@sizes</code> .... 400	
<b>D</b>	<code>\Gin@req@width</code> .... 401	<b>N</b>
<code>\DeclareGraphicsRule</code> ..... 105, 245	<code>\Gin@setfile</code> 67, 272, <u>273</u>	<code>\newcolumntype</code> .... 168
<code>\define@key</code> 106–111, 246, 248, 250, 252, 261, 263, 403	<code>\Gin@shownamefalse</code> . 60	<code>\newcount</code> ..... 90
	<code>\Gin@curx</code> ..... 393, 398	<code>\nofiles</code> .. 79, 207, 211
<b>E</b>	<code>\Gin@cury</code> ..... 394, 396	<b>O</b>
<code>\endpostscript</code> 367, 370	<code>\Gin@viewport@code</code> . 395	<code>\OptionNotUsed</code> ..... 6
<code>\endpsmatrix</code> ..... ..... 154, 158, 370	<code>\Gin@vllx</code> ..... 125	<code>\output</code> ..... 206, 212
<code>\endpspicture</code> ..... 367	<code>\Gin@vllly</code> ..... 125	<code>\overfullrule</code> ..... 81
<code>\endtabularx</code> ..... 170	<code>\Gin@vurx</code> ..... 125	
environments:	<code>\Gin@vury</code> ..... 125	<b>P</b>
<code>postscript</code> 2, <u>118</u> , <u>342</u>	<code>\Gin@vury</code> ..... 125	<code>\PassOptionsToPackage</code> ..... 12, 15, 19
<code>psmatrix</code> .. <u>136</u> , <u>342</u>	<code>\Ginclude@graphics</code> . ..... 54, <u>178</u> , 388	
<code>pspicture</code> 2, <u>135</u> , <u>342</u>	<code>\GPT@page</code> .... 275, 284	
	<code>\Gscale@@box</code> ..... <u>175</u>	

