

The pst-pdf package*

Rolf Niepraschk[†] Hubert Gäßlein

2008/09/09

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` v1.1t, dated 2008/09/09. Thanks to Peter Dybala for the translation.

[†]`Rolf.Niepraschk@gmx.de`

least the size of the whole page. To be able to make use of the graphics' in a later pdfL^AT_EX 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`¹ 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^AT_EX environments behave?)

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

2.2 Program calls

The following table shows the course necessary to create a PDF document containing PostScript graphics². 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^AT_EX 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^AT_EX 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}`

¹CTAN: support/pdfcrop/

²The T_EX distribution “teT_EX” 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^AT_EX 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[⟨keys⟩](⟨pfxadd⟩)<⟨ovpfgd⟩>[⟨ovpbgd⟩]{⟨filename⟩}</code> Wie im Paket <code>psfragx</code> definiert zu verwenden.
<code>\savepicture</code>	<code>\savepicture{⟨name⟩}</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[⟨keys⟩]{⟨name⟩}</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>&</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^AT_EX 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 $\text{te}\TeX$ / 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@Ginclude@graphics\Ginclude@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_EX 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}%
72   \pr@outerfalse\ppf@Ginclude@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 \ifppf@PST@used\RequirePackage{pstricks}\fi
83 \RequirePackage{active,dvips,tightpage}[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 \def\pdftexversion{121}\input{pdftex.def}%
90 \edef\x{\endgroup\def\noexpand\ppf@other@extensions{\Gin@extensions}
91 }%
92 \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.

```

93 \AtBeginDocument{%
94   \@ifpackageloaded{keyval}{%
95     \def\KV@errx#1{\PackageInfo{keyval}{#1}}%
96     }{}%
97   \@ifpackageloaded{xkeyval}{%
98     \def\XKV@err#1{\PackageInfo{xkeyval}{#1}}%
99     }{}%

```

In this mode undefined keys should not be an error.

```

100 \@for\@tempa:=\ppf@other@extensions\do{%
101   \expandafter\let\csname Gin@rule@\@tempa\endcsname\relax}%
102 \DeclareGraphicsRule{*}{eps}{*}{}%

```

No function in this mode.

```

103 \define@key{Gin}{innerframe}[true]{}%
104 \define@key{Gin}{frame}[true]{}%
105 \define@key{Gin}{ignore}[true]{}%
106 \define@key{Gin}{showname}[true]{}%
107 \define@key{Gin}{namefont}{}%
108 \ifundefined{GPT@page}{\define@key{Gin}{page}{}{}}{}
109 \ifppf@tightpage\else
110   \def\PreviewBbAdjust{%
111     -600pt -600pt 600pt 600pt}%

```

```

112 \AtEndDocument{%
113 \PackageWarningNoLine{pst-pdf}{Picture container needs cropping.}}%
114 \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).
115 \newenvironment{postscript}[1][]{%
116 {%
117 \global\let\ppf@PreviewBbAdjust\PreviewBbAdjust
118 \if@ppf@tightpage
119 \begingroup
120 \setkeys{Gin}{#1}%
121 \xdef\PreviewBbAdjust{%
122 -\Gin@vllx bp -\Gin@villy bp \Gin@vurx bp \Gin@vury bp}%
123 \endgroup
124 \fi
125 \ignorespaces
126 }%
127 {\aftergroup\ppf@RestoreBbAdjust}}%

128 \PreviewEnvironment{postscript}%
129 \AtBeginDocument{%
130 \@ifundefined{PSTricksLoaded}{}%
131 {%

pspicture Announce preview original definition.
132 \PreviewEnvironment{pspicture}%

psmatrix Announce preview original definition.
133 \@ifundefined{psmatrix}{}%
134 {%
135 \PreviewEnvironment{psmatrix}%
136 \newcommand*\ppf@set@mode{}%
137 \newcommand*\ppf@test@mmode{%
138 \ifmmode
139 \ifinner
140 \let\ppf@set@mode=$%
141 \else
142 \def\ppf@set@mode{$$}%
143 \fi
144 \else
145 \let\ppf@set@mode=\@empty
146 \fi
147 }%
148 \let\ppf@psmatrix=\psmatrix
149 \expandafter\let\expandafter\ppf@pr@psmatrix%
150 \expandafter=\cename pr@\string\psmatrix\endcsname
151 \let\ppf@endpsmatrix=\endpsmatrix
152 \def\psmatrix{\ppf@test@mmode\ppf@psmatrix}
153 \expandafter\def\cename pr@\string\psmatrix\endcsname{%
154 \ppf@set@mode\ppf@pr@psmatrix}%
155 \def\endpsmatrix{\ppf@endpsmatrix\ppf@set@mode}%
156 }%

```



```

182   \fi
183 }%

184 \PreviewMacro[{}]{\ppf@Ginclude@graphics}%
185 \let\pdfliteral\@gobble%
186 \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.

```

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

188 \if@ppf@PST@used
189   \let\ppf@temp\AtBeginDvi\let\AtBeginDvi\@gobble
190   \RequirePackage{pstricks}\let\AtBeginDvi\ppf@temp
191 \fi

192 \@temptokena{%
193   \let\Gin@PS@file@header\@gobble\let\Gin@PS@literal@header\@gobble
194   \let\Gin@PS@raw\@gobble\let\Gin@PS@restored\@gobble
195   \@ifundefined{PSTricksLoaded}{-}{%

```

Necessary if PSTricks < 2.0.

```

196   \PSTricksOff
197   \@ifundefined{c@lor@to@ps}{\def\c@lor@to@ps#1 #2\@{}}{}}%

    PostScript output is now inhibited and later once again.

198 \the\@temptokena
199 \expandafter\AtBeginDocument\expandafter
200   {\the\@temptokena\@temptokena{}}%
201 \@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.

```

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

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

210   \long\def\pr@startbox#1#2{%
211     \ifpr@outer
212       \toks@{#2}%
213       \edef\pr@cleanup{\the\toks@}%
214       \setbox\@tempboxa\vbox\bgroup

```

```

215     \everydisplay{}%
216     \pr@outerfalse%
217     \expandafter\@firstofone
218     \else
219     \expandafter\@gobble
220     \fi{#1}}%
221 \def\pr@endbox{%
222     \egroup
223     \setbox\@tempboxa\box\voidb@x
224     \ppf@getpicture
225     \pr@cleanup}%

```

(See also the identical definition in DVI mode.)

```

226 \AtBeginDocument{%
227     \ifundefined{pst@object}{}%
228     {%
229         \PreviewMacro[{}*[]%
230         ?\bgroup{#{#1}{#{1}}}{}%
231         ?\bgroup{#{#1}{#{1}}}{}%
232         ?({#{#1})}({#{1})}{}%
233         ?({#{#1})}({#{1})}{}%
234         ?({#{#1})}({#{1})}{}%
235         }]\pst@object}}%
236     }%
237 }%

```

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

```

238 \begingroup
239     \input{dvips.def}%
240     \edef\x{\endgroup\def\noexpand\ppf@other@extensions{\Gin@extensions}}%
241     \x

```

Dummy definition for in DVI mode supported file formats.

```

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

```

(For internal use only!)

```

250     \edef\@tempa{\toks@{\noexpand\frame{\the\toks@}}}%
251     \ifcase#1\relax
252     \ifGin@innerframe\else\let\@tempa\relax\fi
253     \or
254     \ifGin@frame\else\let\@tempa\relax\fi
255     \fi
256     \@tempa
257 }%
258 \define@key{Gin}{showname}[true]{%
259     \lowercase{\Gin@boolkey{#1}}{showname}}%
260 \define@key{Gin}{namefont}{%
261     \begingroup

```

```

262     \@temptokena\expandafter{\ppf@namefont#1}%
263     \edef\x{\endgroup\def\noexpand\ppf@namefont{\the\@temptokena}}%
264     \x
265 }%
266 \newcommand*\ppf@filename{}%
267 \newcommand*\ppf@namefont{\tiny\ttfamily}%
268 \newcommand*\ppf@Gin@keys{}%
269 \let\ppf@Gin@setfile\Gin@setfile

\Gin@setfile Save real file name and, if applicable, page number for later use.
270 \def\Gin@setfile#1#2#3{\ppf@Gin@setfile{#1}{#2}{#3}%
271 \xdef\ppf@filename{%
272     #3\ifx\GPT@page\empty\else(\GPT@page)\fi}}%

\Gin@ii Examine the options “frame”, “ignore”, etc. as soon as other special cases.
273 \def\Gin@ii[#1]#2{%
274     \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.
275     \setkeys{Gin}{#1}%
276     \@temptokena{#1}\def\@tempb{#2}%
    Finds empty file name when calling \usepicture.
277     \ifx\@tempb\empty\else
278     \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.
279     {%
280         \ifx\@tempb\PDFcontainer
281             \@temptokena{page=\GPT@page}%
282         \fi
283     }%
284     {%
285         \refstepcounter{pspicture}%
286         \@temptokena{page=\the\c@pspicture}\def\@tempb{\PDFcontainer}%
287     }%
288     \fi
289     \ifGin@ignore\else
    “frame@@=0” = inner frame, “frame@@=1” = outer frame.
290     \edef\@tempa{\noexpand\ppf@Gin@ii[frame@@=0,\the\@temptokena,
291         frame@@=1]{\@tempb}}%
292     \@tempa
293     \ifGin@showname
294         \ppf@namefont
295         \raisebox{-\ht\strutbox}[0pt][0pt]{\llap{\ppf@filename}}%
296         \gdef\ppf@filename{}%
297     \fi
298     \fi
299 \endgroup
300 }%

```

```

301 \IfFileExists{\PDFcontainer}%
302 {%
\ppf@container@max The number of pages as contained in \PDFcontainer file.
303 \pdfximage{\PDFcontainer}%
304 \edef\ppf@container@max{\the\pdflastximagepages}%
305 \AtEndDocument{%
306 \ifnum\c@pspicture>\z@
A warning only makes sense when a graphics is needed at all.
307 \ifnum\c@pspicture=\ppf@container@max\else
308 \PackageWarningNoLine{pst-pdf}{%
309 '\PDFcontainer' contains \ppf@container@max\space pages
310 \MessageBreak but \the\c@pspicture\space pages are requested:
311 \MessageBreak File '\PDFcontainer' is no more valid!
312 \MessageBreak Recreate it
313 }%
314 \fi
315 \fi
316 }%
317 }%
318 {%
319 \def\ppf@container@max{0}%
320 \AtEndDocument{%
321 \ifnum\c@pspicture>\z@
322 \filename@parse{\PDFcontainer}%
323 \PackageWarningNoLine{pst-pdf}{%
324 File '\PDFcontainer' not found.\MessageBreak
325 Use the following commands to create it:\MessageBreak
326 -----
327 \MessageBreak
328 latex \jobname.tex\MessageBreak
329 dvips -o \filename@base.ps \jobname.dvi\MessageBreak
330 ps2pdf \filename@base.ps\MessageBreak
331 -----
332 }%
333 \fi
334 }%
335 }%

\ppf@isnum If parameter #1 is numeric, the instructions in #2, otherwise those in #3 are executed (see bibtopic.sty).
336 \newcommand\ppf@isnum[1]{%
337 \if!\ifnum9<1#1!\else_\fi\expandafter\@firstoftwo
338 \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.

pspicture

postscript
339 \newcommand*\ppf@set@mode{%
340 \newcommand*\ppf@test@mmode{%

```

```

341 \ifmode
342 \ifinner
343 \let\ppf@set@mode=$%
344 \else
345 \def\ppf@set@mode{$$}%
346 \fi
347 \else
348 \let\ppf@set@mode=\@empty
349 \fi
350 }

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

The useless redefinition of \includegraphics in pdfTeX mode (package psfragx)
is leading to double insertion of the result. We go back to the original meaning.

370 \let\includegraphics=\pfx@includegraphics
371 \def\pfx@includegraphicx#1#2{\ppf@getpicture}%
372 }%
373 }%

```

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

```

374 \def\savepicture#1{%
375 \expandafter\xdef\csname ppf@@@#1\endcsname{\the\pdflastximage}}%

```

\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.

```

376 \renewcommand*\usepicture[2] [] {%
377 \@ifundefined{ppf@@@#2}%
378 {%
379 \ppf@isnum{#2}%
380 {\ppf@getpicture{#1}{#2}}%
381 {\@latexerror{picture '#2' undefined}\@ehc}%
382 }%
383 {%

```

```

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

```

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

```

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

```

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

```

414 \newcommand*\ppf@@getpicture{%
415 \ifpr@outer
416 \refstepcounter{pspicture}%
417 \expandafter\ppf@getpicture\expandafter{\ppf@Gin@keys}%
418 {\the\c@pspicture}%
419 \fi}%

```

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

```

420 \renewenvironment*{pst-pdf-defs}%
421 {%
422 \endgroup
423 % ??? \@currencline
424 \chardef\ppf@temp=\catcode'\&%
425 \@makeother\&%
426 }{%
427 \catcode'\&=\ppf@temp
428 \begingroup
429 \def\@currenvir{pst-pdf-defs}%

```

```

430 }
431 \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 \TeX compiler force this mode.

```

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

436 \InputIfFileExists{pst-pdf.cfg}{%
437 \PackageInfo{pst-pdf}{Local config file pst-pdf.cfg used}}{}
438 \</package>

```

Change History

v1.0a	General: Initial version. 1	does accept a numerical parameter. (RN) 13
v1.0b	General: Some code and documentation cleaning. (RN) 1	v1.0h <code>\psmatrix</code> : Based no more on the comment environment from the verbatim package. (RN) 12
v1.0c	General: New options “pstricks”, “nopstricks”, “draft” and “final”. (RN) 3	v1.0i <code>\ppf@is@pdfTeX@graphic</code> : No more errors for given files without extensions. (RN) 5
v1.0d	General: Redefinition of <code>\includegraphics</code> in modes 0 und 1. Now using of eps graphics directly in pdf \LaTeX is possible. (RN) 1	v1.0j General: Check <code>AtBeginDocument</code> for package ‘pstricks’ even if “nopstricks” is given. (RN) . . . 1
v1.0e	<code>\postscript</code> : “trim” option added. (RN) 7	v1.0k <code>\Gin@setfile</code> : Show also the pagenumber if exists. (RN) 11
v1.0f	General: Config file loading added. (RN) 15	<code>\Gin@setfile</code> : Prevent division by zero. (RN) 8
	<code>\savepicture</code> : New macro <code>\savepspicture</code> . (RN) 13	v1.0l General: Options “framesep”, “framerule”, “linewidth” removed, “fname” and “innerframe” added. (RN) 1
	<code>\usepicture</code> : New macro <code>\usepspicture</code> . Useful for putting a PSTricks graphic in a box or something else. (RN) 13	v1.0m General: New package option “notightpage” added. (RN) . . . 1
v1.0g	General: Definition of <code>\PDFcontainer</code> now with <code>\edef</code> . (RN) 5	v1.0n General: Changed marco names (<code>\savepicture</code> and <code>\usepicture</code>). (RN) 1
	<code>\usepicture</code> : Now <code>\usepspicture</code>	

	Some code cleaning. (RN)	1	v1.1g	postscript: “psmatrix” environ- ment (preserve math mode). (RN/HjG)	12
v1.0o	General: New code for “notight- page”. (RN)	6		pspicture: pspicture environment must still parse its arguments. (RN/HjG)	12
	Option “fname” renamed to “showname”. (RN)	1			
v1.0p	General: Some code and documen- tation cleaning. (RN)	1	v1.1h	\Ginclud@graphics: Check if in- side of a PS-related environ- ment (correct graphic inclu- sion). (RN)	8
v1.0q	\usepicture: Now \usepspicture works for all kind of graphics. (RN)	13	v1.1i	General: \ifpr@outer must be pre- defined. (HjG/RN)	5
v1.0r	\ppf@is@pdfTeX@graphic: Changed \ppf@is@known@graphic to \ppf@is@pdfTeX@graphic . Now pdfTeX graphics are pref- ered. (RN)	5		Package option “final” also for “graphicx”. (RN)	4
v1.0s	General: Scaling e.g. of PostScript pictures now only in extraction mode. Some code cleaning. (RN)	1	v1.1k	\Ginclud@graphics: Correction of the inside check. (RN/HjG) . . .	8
	\Gin@ii: Rewritten. (RN)	11		General: New environment pst-pdf- defs: Support for PSTricks envi- ronment “psmatrix” inside user definitions. (RN,HjG)	1
v1.1a	General: Support for the internal PSTricks macro \pst@object . (HjG/RN)	8	v1.1l	General: Support for the package “psfragx”. (RN)	8
v1.1b	General: Ignore the call of \nofiles inside of preview . (RN)	9	v1.1m	General: Merge english and german version of the documentation. (RN)	1
	Some code and documentation cleaning. (RN)	1	v1.1n	General: \nofiles added (sugges- tion of Torsten Bronger).	6
v1.1c	General: New package option “tight- page” added. (RN)	1	v1.1o	\Gscale@@box: Disable scaling. (RN)	8
	Special support for “tabularx”. (RN)	8	v1.1p	General: \nofiles makes \makeindex and \makeglossary to \relax . \@empty is better be- cause of later \renewcommand ’s. . .	6
	Supress handling of pdfL ^A T _E X graphic formats in DVI mode. (RN)	6	v1.1p1	General: \let\output\@gobble be- fore loading of “preview” added. (RN)	9
v1.1d	postscript: Support for PSTricks environment “psmatrix”. (RN) . . .	12	v1.1q	General: Problem with “tabularx” and “threeparttable” solved. (RN)	8
v1.1e	General: New option “display- math” (see preview package). (HjG/RN)	3	v1.1r	General: Fixed values for \PreviewBbAdjust because \paperwidth is not allways	
v1.1f	General: Package option “ignore” reimplemented. Now the com- pilation of the dtx file in L ^A T _E X mode is possible. (RN)	3			

	defined (suggested by Will Robertson).	6	“\let\output@gobble” be- cause of bad side effects. (RN) .	9
v1.1s	General: Dummy definition of the page key in DVI mode.	6	postscript: Using <code>environ</code> the en- vironment postscript is now sim- ple and more robust. (RN) . .	13
v1.1t	General: Remove the line			

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
\& 424, 425, 427 <i>3</i> , <u>46</u> , <u>420</u>	\if@ppf@pdf@tex@graphic
\@currenvir 51, 354, 429	\everydisplay 215 57, 74
\@currenvline .. 48, 423	F	\if@ppf@PST@used ..
\@ehc 381	\filename@base 329, 330 4, 82, 188
\@firstofone 217	\filename@parse ... 322	\if@ppf@tightpage .
\@ifpackageloaded 94, 97	\frame 250 5, 109, 118
\@latex@error 381	G	\ifGin@frame ... 58, 254
\@makeoother 425	\Gin@bboxtrue 392	\ifGin@ignore .. 61, 289
\@ppf@PST@usedfalse 10	\Gin@boolkey 244, 246, 248, 259	\ifGin@innerframe .
\@ppf@PST@usedtrue 4, 9	\Gin@defaultbp 390, 391 59, 252
\@ppf@pdf@tex@graphic@false	\Gin@extensions 55, 71, 90, 177, 240, 278	\ifGin@showname 60, 293
..... 64	\Gin@framefalse ... 58	\ifinner 139, 342
\@ppf@pdf@tex@graphic@true	\Gin@ignorefalse .. 61	\ifmmode 138, 341
..... 70	\Gin@ii 56, <u>273</u>	\ifpr@outer 62, 176, 211, 415
\@ppf@tightpage@false 16	\Gin@innerframefalse 59	\includegraphics ..
\@ppf@tightpage@true	\Gin@llx 389, 396	.. 2, 370, 401, 410
..... 5, 17	\Gin@lly 389, 394	\includegraphicsx ... <i>3</i>
A	\Gin@nat@height 388, 391, 393, 394	J
\AtBeginDvi ... 189, 190	\Gin@nat@width 388, 390, 395, 396	\jobname .. 41, 328, 329
C	\Gin@PS@file@header 193	K
\color@to@ps 197	\Gin@PS@literal@header 193	\KV@errx 95
\c@p@picture 286, 306, 307, 310, 321, 418	\Gin@PS@raw 194	L
\catcode 424, 427	\Gin@PS@restored .. 194	\leavevmode 399
\Collect@Body 360	\Gin@req@height ... 398	\long 210
\CurrentOption 12, 15, 19	\Gin@req@sizes 397	M
D	\Gin@req@width 398	\makeglossary 80
\DeclareGraphicsRule	\Gin@setfile 67, 269, <u>270</u>	\makeindex 80
..... 102, 242	\Gin@shownamefalse . 60	\mbox 173
\define@key 103–108, 243, 245, 247, 249, 258, 260, 400	\Gin@urx 390, 395	N
E	\Gin@ury 391, 393	\newcolumn@type 165
\endpostscript 364, 367	\Gin@viewport@code . 392	\nofiles .. 79, 204, 208
\endpsmatrix 151, 155, 367	\Gin@vllx 122	O
\endpspicture 364	\Gin@vllly 122	\OptionNotUsed 6
\endtabularx 167	\Gin@vurx 122	\output 203, 209
environments:	\Gin@vury 122	\overfullrule 81
postscript 2, <u>115</u> , <u>339</u>	\Gin@vury 122	P
psmatrix .. <u>133</u> , <u>339</u>	\Gin@vury 122	\PassOptionsToPackage
pspicture 2, <u>132</u> , <u>339</u>	\Gin@vury 122 12, 15, 19
	\Gin@vury 122	\PDF@container 40, 41, 280, 286,

301, 303, 309, 311, 322, 324, 411	\ppf@is@pdfTeX@graphic . 63, 177, 278, 434	\PreviewMacro 157, 170, 184, 229
\pdflastximage 375	\ppf@isnum ... 336, 379	\psmatrix 148, 150, 152, 153, 366
\pdflastximagepages 304	\ppf@namefont 262, 263, 267, 294	psmatrix (environ- ment) ... 133 , 339
\pdfliteral 185	\ppf@nofiles .. 204, 208	pspicture (environ- ment) . 2, 132 , 339
\pdfoutput 26	\ppf@other@extensions 43, 90, 100, 177, 240, 278	pst-pdf-defs (environ- ment) .. 3, 46 , 420
\pdfrefximage 387	\ppf@output 202, 203, 209	\pst@@@picture 363
\pdfTeXtext 66, 69	\ppf@pr@psmatrix 149, 154	\pst@object ... 163, 235
\pdfTeXversion 89	\ppf@PreviewBbAdjust 84, 86, 117	\PSTricksOff 196
\pdfximage 303	\ppf@psmatrix . 148, 152	
\pfx@includedgraphics 370	\ppf@RestoreBbAdjust 85, 127	R
\pfx@includedgraphicsx 170, 371	\ppf@set@mode 136, 140, 142, 145, 154, 155, 339, 343, 345, 348	\raisebox 295
\postscript ... 363, 366	\ppf@shipout .. 205, 207	\refstepcounter 285, 416
postscript (environ- ment) . 2, 115 , 339	\ppf@temp 189, 190, 424, 427	\rule 178
\ppf@getpicture .. . 224, 359, 371, 414	\ppf@test@mode 137, 152, 340	S
\ppf@container@max 303 , 307, 309, 319, 406	\ppf@TeX@mode . 2, 7, 8, 21, 22, 33, 35, 37, 76, 78, 187, 432	\savepicture . 3, 45, 374
\ppf@draft 3, 13, 14, 410	\pr@cleanup ... 213, 225	\setkeys 120, 275
\ppf@endpsmatrix 151, 155	\pr@endbox 221	\shipout 205, 207
\ppf@filename . 266, 271, 295, 296, 386	\pr@outerfalse . 72, 216	\string 150, 153
\ppf@getpicture 380, 404 , 417	\pr@startbox 210	\strutbox 295
\ppf@Gin@extensions 55	\PreviewBbAdjust .. . 86, 110, 117, 121	T
\ppf@Gin@ii 56, 290	\PreviewEnvironment 128, 132, 135	\tabularx 166
\ppf@Gin@keys . 268, 355, 357, 413, 417		U
\ppf@Gin@setfile 269, 270		\usepicture .. 3, 44, 376
\ppf@Gin@include@graphics 54, 72, 179, 181, 184		V
		\voidb@x 223
		X
		\XKV@err 98