

The pst-pdf package*

Rolf Niepraschk[†] Hubert Gäßlein

2016/07/11

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` v1.2a, dated 2016/07/11. 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[<i><keys></i>](<i><pfxadd></i>)<<i><ovpfgd></i>>[<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>&</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**=** 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 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   \RequirePackage{ifpdf,ifxetex,ifvtex}%
24   \ifpdf
25     ⇒ pdf $\text{\TeX}$  or Lua $\text{\TeX}$  are running in PDF mode
26     \def\ppf@TeX@mode{1}%
27     \RequirePackage{luatex85}%
28   \else
29     \ifvtex
30     ⇒ V $\text{\TeX}$ 
31     \def\ppf@TeX@mode{9}%
32   \else
33     \ifxetex
34     ⇒ Xe $\text{\TeX}$ 
35     \def\ppf@TeX@mode{9}%
36   \else
37     ⇒ DVI mode
38     \def\ppf@TeX@mode{0}%
39     \fi
40   \fi
41 \fi
42 \fi

39 \newcommand*\PDFcontainer{}
40 \edef\PDFcontainer{\jobname-pics.pdf}
41 \newcounter{pspicture}
42 \newcommand*\ppf@other@extensions[1]{

```

```

43 \newcommand*\usepicture[2] [] {}
44 \newcommand*\savepicture[1] {}

```

pst-pdf-defs

```

45 \newenvironment*{pst-pdf-defs}{%
46   \endgroup
47   %   ??? \@currenvline
48 }{%
49   \begingroup
50   \def\@currenvir{pst-pdf-defs}%
51 }

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

56 \newif\if@ppf@pdftex@graphic
57 \newif\if@Gin@frame\Gin@framefalse
58 \newif\if@Gin@innerframe\Gin@innerframefalse
59 \newif\if@Gin@showname\Gin@shownamefalse
60 \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.

```

61 \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.

```

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

```

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

```

66   \def\Gin@setfile##1##2##3{%
67     \edef\@tempb{##2}%
68     \@for\@tempa:=\pdfTeXext\do{%
69       \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”.

```

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

```

Trial invocation. Output is completely inhibited.

```

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

```

```

75 \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.

```

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

```

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

```

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

```

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

```

In this mode undefined keys should not be an error.

```

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

```

No function in this mode.

```

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

```

```

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

```

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

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

```

pspicture Announce preview original definition.

```

134 \PreviewEnvironment{pspicture}%

```

psmatrix Announce preview original definition.

```

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

```

```

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

```

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.

```

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

```

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

```

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

```

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

```

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

```

`\Gscale@@box` Disable scaling.

```

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

```

`\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.

```

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

```

Generally pdfTeX supported graphics formats are intended to be preferred (inclusion in final pdfTeX 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.

```

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

```

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

```

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

```


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

```

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

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

```

3.4 pdf_{La}TeX 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.

```

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

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

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

```

Necessary if PSTricks < 2.0.

```

198     \PSTricksOff
199     \@ifundefined{c@lor@to@ps}{\def\c@lor@to@ps#1 #2\@{}}{}%

```

PostScript output is now inhibited and later once again.

```

200 \the\@temptokena
201 \expandafter\AtBeginDocument\expandafter
202     {\the\@temptokena\@temptokena{}}%
203 \@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.

```

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

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

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

```

```

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

```

(See also the identical definition in DVI mode.)

```

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

```

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

```

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

```

Dummy definition for in DVI mode supported file formats.

```

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

```

(For internal use only!)

```

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

```

```

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

\Gin@setfile Save real file name and, if applicable, page number for later use.
272 \def\Gin@setfile#1#2#3{\ppf@Gin@setfile{#1}{#2}{#3}%
273 \xdef\ppf@filename{%
274 #3\ifx\GPT@page\empty\else(\GPT@page)\fi}}%

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

```

```

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

303 \IfFileExists{\PDFcontainer}%
304 {%

\ppf@container@max The number of pages as contained in \PDFcontainer file.
305   \pdfximage{\PDFcontainer}%
306   \edef\ppf@container@max{\the\pdflastximagepages}%

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

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

```

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

```

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

```

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

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

```

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.

```

372 \let\includegraphics=\pfx@includegraphics
373 \def\pfx@includegraphics#1#2{\ppf@@getpicture}%
374 }%
375 }%

```

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

```

376 \def\savepicture#1{%
377 \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`.

```

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

```

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

```

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

```

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

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

```

Change History

v1.0a	General: Initial version. 1	v1.0g	<code>\usepicture</code> : Now <code>\usepspicture</code> does accept a numerical parameter. (RN) 13
v1.0b	General: Some code and documentation cleaning. (RN) . 1		General: Definition of <code>\PDFcontainer</code> now with <code>\edef</code> . (RN) 4
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 \LaTeX 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	<code>\savepicture</code> : New macro <code>\savepspicture</code> . (RN) 13	v1.0k	<code>\Gin@setfile</code> : Show also the pagenumber if exists. (RN) . . 11
	<code>\usepicture</code> : New macro <code>\usepspicture</code> . Useful for putting a PSTricks graphic in a box or something else. (RN) . 13		<code>\Gin@cl@graph</code> : Prevent division by zero. (RN) 8
	General: Config file loading added. (RN) 15	v1.0l	General: Options “framesep”, “framerule”, “linewidth” removed, “fname” and

	“innerframe” added. (RN)	1	v1.1d	
v1.0m	General: New package option			postscript : Support for PSTricks
	“notightpage” added. (RN) . . .	1	v1.1e	environment “psmatrix”. (RN) 13
v1.0n	General: Changed macro names			General: New option “displaymath”
	(\savepicture and			(see preview package).
	\usepicture). (RN)	1	v1.1f	(HjG/RN) 3
	Some code cleaning. (RN)	1		General: Package option “ignore”
v1.0o	General: New code for			reimplemented. Now the
	“notightpage”. (RN)	7	v1.1g	compilation of the dtx file in
	Option “fname” renamed to			L ^A T _E X mode is possible. (RN) . 3
	“showname”. (RN)	1		postscript : “psmatrix”
v1.0p	General: Some code and			environment (preserve math
	documentation cleaning. (RN) .	1		mode). (RN/HjG) 13
v1.0q	\usepicture: Now \usepspicture			pspicture : pspicture environment
	works for all kind of graphics.			must still parse its arguments.
	(RN)	13	v1.1h	(RN/HjG) 12
v1.0r	\ppf@is@pdfTeX@graphic:			\Ginclude@graphics: Check if
	Changed			inside of a PS-related
	\ppf@is@known@graphic to			environment (correct graphic
	\ppf@is@pdfTeX@graphic.			inclusion). (RN) 8
	Now pdfT _E X graphics are		v1.1i	\Ginclude@graphics: Correction
	preferred. (RN)	5		of the inside check. (RN/HjG) . 8
v1.0s	\Gin@ii: Rewritten. (RN)	11		General: \ifpr@outer must be
	General: Scaling e.g. of PostScript			predefined. (HjG/RN) 5
	pictures now only in extraction			Package option “final” also for
	mode. Some code cleaning.			“graphicx”. (RN) 4
	(RN)	1	v1.1k	General: New environment
v1.1a	General: Support for the internal			pst-pdf-defs: Support for
	PSTricks macro \pst@object.			PSTricks environment
	(HjG/RN)	8	v1.1l	“psmatrix” inside user
v1.1b	General: Ignore the call of			definitions. (RN,HjG) 1
	\nofiles inside of preview.			General: Support for the package
	(RN)	9		“psfragx”. (RN) 8
	Some code and documentation		v1.1m	General: Merge english and
	cleaning. (RN)	1		german version of the
v1.1c	General: New package option			documentation. (RN) 1
	“tightpage” added. (RN)	1	v1.1n	General: \nofiles added
	Special support for “tabularx”.			(suggestion of Torsten
	(RN)	8	v1.1o	Bronger). 6
	Supress handling of pdfL ^A T _E X			\Gscale@@box: Disable scaling.
	graphic formats in DVI mode.			(RN) 8
	(RN)	6	v1.1p	General: \nofiles makes
				\makeindex and

	<code>\makeglossary</code> to <code>\relax</code> .		
	<code>\@empty</code> is better because of		
	later <code>\renewcommand</code> 's.	6	
v1.1p1	General: <code>\let\output\@gobble</code>		
	before loading of "preview"		
	added. (RN)	9	
v1.1q	General: Problem with "tabularx"		
	and "threeparttable" solved.		
	(RN)	8	
v1.1r	General: Fixed values for		
	<code>\PreviewBbAdjust</code> because		
	<code>\paperwidth</code> is not always		
	defined (suggested by Will		
	Robertson).	7	
v1.1s	General: Dummy definition of the		
	page key in DVI mode.	6	
			v1.1t
			General: Remove the line
			" <code>\let\output\@gobble</code> "
			because of bad side effects.
			(RN)
			9
			<code>postscript</code> : Using <code>environ</code> the
			environment <code>postscript</code> is now
			simple and more robust. (RN)
			13
			v1.1u
			General: <code>\pdfoutput</code> must be set
			when loading "pdftex.def" in
			DVI mode. (RN)
			6
			v1.1v
			<code>\Gin@ii</code> : Key settings only for pdf
			graphics. (RN)
			11
			General: Local redefinition of
			<code>\pdfoutput</code> to be a counter.
			(RN)
			6
			v1.2a
			General: Engine tests changed
			(RN)
			4

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
\& 426, 427, 429 <i>3</i> , <i>45</i> , <i>422</i>	\if@ppf@pdftex@graphic
\@currenvir 50, 356, 431	\everydisplay 217 56, 73
\@currenvline . . 47, 425		\if@ppf@PST@used . .
\@ehc 383	F 4, 81, 190
\@firstofone 219	\filename@base 331, 332	\if@ppf@tightpage .
\@ifpackageloaded 96, 99	\filename@parse . . . 324 5, 111, 120
\@latex@error 383	\frame 252	\ifGin@frame . . . 57, 256
\@makeother 427		\ifGin@ignore . . 60, 291
\@ppf@PST@usedfalse 10	G	\ifGin@innerframe .
\@ppf@PST@usedtrue 4, 9	\Gin@bboxtrue 394 58, 254
\@ppf@pdftex@graphicfalse	\Gin@boolkey	\ifGin@showname 59, 295
. 63 246, 248, 250, 261	\ifinner 141, 344
\@ppf@pdftex@graphictrue	\Gin@defaultbp 392, 393	\ifmmode 140, 343
. 69	\Gin@extensions 54,	\ifpdf 24
\@ppf@tightpagefalse 16	70, 92, 179, 242, 279	\ifpr@outer
\@ppf@tightpagetrue	\Gin@framefalse . . . 57 61, 178, 213, 417
. 5, 17	\Gin@ignorefalse . . 60	\ifvtex 28
	\Gin@ii 55, <i>275</i>	\ifxetex 31
A	\Gin@innerframefalse 58	\includegraphics . .
\AtBeginDvi . . . 191, 192	\Gin@llx 391, 398 2, 372, 403, 412
	\Gin@lly 391, 396	\includegraphicsx . . . 3
C	\Gin@nat@height . . .	
\color@to@ps 199 390, 393, 395, 396	J
\c@psspicture 288, 308,	\Gin@nat@width	\jobname . . . 40, 330, 331
309, 312, 323, 420 390, 392, 397, 398	
\catcode 426, 429	\Gin@PS@file@header 195	K
\Collect@Body 362	\Gin@PS@literal@header	\KV@errx 97
\CurrentOption 12, 15, 19 195	
	\Gin@PS@raw 196	L
D	\Gin@PS@restored . . 196	\leavevmode 401
\DeclareGraphicsRule	\Gin@req@height . . . 400	\long 212
. 104, 244	\Gin@req@sizes 399	
\define@key 105–110,	\Gin@req@width 400	M
245, 247, 249,	\Gin@setfile 66, <i>271</i> , <i>272</i>	\makeglossary 79
251, 260, 262, 402	\Gin@shownamefalse . 59	\makeindex 79
	\Gin@urx 392, 397	\mbox 175
E	\Gin@ury 393, 395	
\endpostscript 366, 369	\Gin@viewport@code . 394	N
\endpsmatrix	\Gin@vllx 124	\newcolumnmtype . . . 167
. 153, 157, 369	\Gin@vllly 124	\newcount 89
\endpspicture 366	\Gin@vurx 124	\nofiles . . . 78, 206, 210
\endtabularx 169	\Gin@vury 124	
environments:	\Gin@vury 124	O
postscript 2, <i>117</i> , <i>341</i>	\Gin@vury 124	\OptionNotUsed 6
psmatrix . . . <i>135</i> , <i>341</i>	\Gin@vury 124	\output 205, 211
pspicture 2, <i>134</i> , <i>341</i>	\Gin@vury 124	\overfullrule 80
	\Gin@vury 124	

P		
\PassOptionsToPackage		
.....	12, 15, 19	
\PDFcontainer	
	39, 40, 282, 288,	
	303, 305, 311,	
	313, 324, 326, 413	
\pdflastximage	377
\pdflastximagepages		306
\pdfliteral	187
\pdfoutput	89, 90
\pdfrefximage	389
\pdfTeXtext	65, 68
\pdftexversion	88
\pdfximage	305
\pfx@includgraphics		
.....	372	
\pfx@includgraphicx		
.....	172, 373	
\postscript	...	365, 368
postscript (environ-		
ment)	. 2, <u>117</u> , <u>341</u>	
\ppf@getpicture	..	
	. 226, 361, 373, <u>416</u>	
\ppf@container@max		
.....	305,	
	309, 311, 321, 408	
\ppf@draft	3, 13, 14, 412	
\ppf@endpsmatrix	..	
.....	153, 157	
\ppf@filename	. 268,	
	273, 297, 298, 388	
\ppf@getpicture	...	
....	382, <u>406</u> , 419	
\ppf@Gin@extensions		54
\ppf@Gin@ii	55, 292
\ppf@Gin@keys	. 270,	
	357, 359, 415, 419	
\ppf@Gin@setfile	..	
.....	271, 272	
\ppf@Gin@graphics		
.....	53,	
	71, 181, 183, 186	
\ppf@is@pdfTeX@graphic		
	. 62, 179, 279, 436	
\ppf@isnum	...	<u>338</u> , 381
\ppf@namefont	
	. 264, 265, 269, 296	
\ppf@nofiles	..	206, 210
\ppf@other@extensions		
.....	42, 92,	
	102, 179, 242, 279	
\ppf@output	204, 205, 211	
\ppf@pr@psmatrix	..	
.....	151, 156	
\ppf@PreviewBbAdjust		
.....	83, 85, 119	
\ppf@psmatrix	. 150, 154	
\ppf@RestoreBbAdjust		
.....	84, 129	
\ppf@set@mode	
	. 138, 142, 144,	
	147, 156, 157,	
	341, 345, 347, 350	
\ppf@shipout	..	207, 209
\ppf@temp	
	. 191, 192, 426, 429	
\ppf@tempb	... 277,	
	278, 282, 288, 293	
\ppf@test@mmode	...	
....	139, 154, 342	
\ppf@TeX@mode	
	. 2, 7, 8, 21,	
	22, 25, 29, 32,	
	34, 75, 77, 189, 434	
\pr@cleanup	...	215, 227
\pr@endbox	223
\pr@outerfalse	. 71, 218	
\pr@startbox	212
\PreviewBbAdjust	..	
	. 85, 112, 119, 123	
\PreviewEnvironment	130, 134, 137
\PreviewMacro	
	. 159, 172, 186, 231	
\psmatrix	150,
	152, 154, 155, 368	
psmatrix (environ-		
ment)	... <u>135</u> , <u>341</u>	
pspicture (environ-		
ment)	. 2, <u>134</u> , <u>341</u>	
pst-pdf-defs (environ-		
ment)	.. 3, <u>45</u> , <u>422</u>	
\pst@@picture	365
\pst@object	...	165, 237
\PSTricksOff	198
R		
\raisebox	297
\refstepcounter		287, 418
\rule	180
S		
\savepicture	. 3, 44, <u>376</u>	
\setkeys	122, 281
\shipout	207, 209
\string	152, 155
\strutbox	297
T		
\tabularx	168
U		
\usepicture	.. 3, 43, <u>378</u>	
V		
\voidbox	225
X		
\XKV@err	100