

# The pst-pdf package\*

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

2019/11/15

## 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` v1.2e, dated 2019/11/15. Thanks to Peter Dybala 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 `pstricks` 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>&lt;keys&gt;</i>](<i>&lt;pfxadd&gt;</i>)&lt;<i>&lt;ovpfgd&gt;</i>&gt;[<i>&lt;ovpbgd&gt;</i>]{<i>&lt;filename&gt;</i>}</code> To be used like defined in <code>packagepsfragx</code> .
<code>\savepicture</code>	<code>\savepicture{<i>&lt;name&gt;</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>&lt;keys&gt;</i>]{<i>&lt;name&gt;</i>}</code> The graphic previously stored with <code>\savepicture</code> is outputted. The optional parameter corresponds to <code>\includegraphics</code> .
<code>pst-pdf-defs</code>	<code>\begin{pst-pdf-defs} ... \end{pst-pdf-defs}</code> For defining macros or environments, which contain character <code>&amp;</code> (others?) in the output, these definitions have to be wrapped with environment <code>pst-pdf-defs</code> .

## 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 set to “**true**” no graphics are outputted. With macro `\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  $\text{\TeX}$  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 \RequirePackage{ifpdf,ifxetex,ifvtex}
23 \ifnum\ppf@TeX@mode=-1\relax
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
43 \fi
44 \newcommand*\PDFcontainer{}
45 \edef\PDFcontainer{\jobname-pics.pdf}
46 \newcounter{pspicture}
47 \newcommand*\ppf@other@extensions[1]{%
48 \newcommand*\usepicture[2][{}]{%

```

```

44 \newcommand*\savepicture[1]{}

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

52 \RequirePackage{graphicx}[2017/06/01]%
53 \let\ppf@Gininclude@graphics\Gininclude@graphics
54 \let\ppf@Gin@extensions\Gin@extensions
55 \let\ppf@Gin@ii\Gin@ii

56 \newif\ifppf@pdftex@graphic
57 \newif\ifGin@frame\Gin@framefalse
58 \newif\ifGin@innerframe\Gin@innerframefalse
59 \newif\ifGin@showname\Gin@shownamefalse
60 \newif\ifGin@ignore\Gin@ignorefalse

\ifpr@outer in fact is defined in package preview. We have to do it here too since
otherwise TEX 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\pdfTeXtext{,#2,}%
66   \edef\pdfTeXtext{\detokenize\expandafter{\pdfTeXtext}}%

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

67   \def\Gin@setfile##1##2##3{%
68     \edef\@tempa{,##2,}%
69     \@expandtwoargs\in@{\detokenize\expandafter{\@tempa}}{\pdfTeXtext}%
70     \ifin@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   \ifppf@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 \let\printindex\@empty
81 \renewcommand*\makeindex[1][]{}%
82 \renewcommand*\makeglossary[1][]{}%
83 \renewcommand*\printindex[1][]{}%
84 \AtBeginDocument{\overfullrule=\z@}%
85 \ifppf@PST@used\RequirePackage{pstricks}\fi
86 \RequirePackage{active,dvips,tightpage}{preview}[2005/01/29]%
87 \newcommand*\ppf@PreviewBbAdjust{}
88 \newcommand*\ppf@RestoreBbAdjust{%
89   \let\PreviewBbAdjust\ppf@PreviewBbAdjust}%

```

The pdfLaTeX mode compliant graphics file formats are needed too.

```

90 \begingroup
91   \let\AtBeginDocument\@gobble \let\PackageWarningNoLine\@gobbletwo
92   \chardef\pdftexversion=121 %
93   \newcount\pdfoutput
94   \pdfoutput=1 %
95   \input{pdftex.def}%
96   \edef\x{\endgroup\def\noexpand\ppf@other@extensions{\Gin@extensions}}
97   }%
98 \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.

```

99 \AtBeginDocument{%
100   \@ifpackageloaded{keyval}{%
101     \def\KV@errx#1{\PackageInfo{keyval}{#1}}%
102     }{}%
103   \@ifpackageloaded{xkeyval}{%
104     \def\XKV@err#1{\PackageInfo{xkeyval}{#1}}%
105     }{}%

```

In this mode undefined keys should not be an error.

```

106   \@for\@tempa:=\ppf@other@extensions\do{%
107     \expandafter\let\csname Gin@rule@\@tempa\endcsname\relax}%
108   \DeclareGraphicsRule{*}{eps}{*}{}%

```

No function in this mode.

```

109 \define@key{Gin}{innerframe}[true]{}%
110 \define@key{Gin}{frame}[true]{}%
111 \define@key{Gin}{ignore}[true]{}%
112 \define@key{Gin}{showname}[true]{}%

```

```

113 \define@key{Gin}{namefont}{}%
114 \ifundefined{Gin@page}{\define@key{Gin}{page}{}{}}{}

115 \if@ppf@tightpage\else
116 \def\PreviewBbAdjust{%
117 -600pt -600pt 600pt 600pt}%
118 \AtEndDocument{%
119 \PackageWarningNoLine{pst-pdf}{Picture container needs cropping.}}%
120 \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).

```

121 \newenvironment{postscript}[1][]{%
122 {%
123 \global\let\ppf@PreviewBbAdjust\PreviewBbAdjust
124 \if@ppf@tightpage
125 \begingroup
126 \setkeys{Gin}{#1}%
127 \xdef\PreviewBbAdjust{%
128 -\Gin@vllx bp -\Gin@vllly bp \Gin@vurx bp \Gin@vury bp}%
129 \endgroup
130 \fi
131 \ignorespaces
132 }%
133 {\aftergroup\ppf@RestoreBbAdjust}%

134 \PreviewEnvironment{postscript}%
135 \AtBeginDocument{%
136 \ifundefined{PSTricksLoaded}{}%
137 {%

```

**pspicture** Announce preview original definition.

```

138 \PreviewEnvironment{pspicture}%

```

**psmatrix** Announce preview original definition.

```

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

```

```

157     \let\ppf@endpsmatrix=\endpsmatrix
158     \def\psmatrix{\ppf@test@mode\ppf@psmatrix}
159     \expandafter\def\csname pr@string\psmatrix\endcsname{%
160         \ppf@set@mode\ppf@pr@psmatrix}%
161     \def\endpsmatrix{\ppf@endpsmatrix\ppf@set@mode}%
162 }%

```

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.

```

163     \PreviewMacro[{}*[]%
164     ?\bggroup{##1}{\#1}}{}%
165     ?\bggroup{##1}{\#1}}{}%
166     ?({\#1}){({\#1})}}{}%
167     ?({\#1}){({\#1})}}{}%
168     ?({\#1}){({\#1})}}{}%
169     ]{\pst@object}}

```

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

```

170     \@ifundefined{tabularx}{}{}%
171     \newcolumntype{X}{c}%
172     \expandafter\let\expandafter\tabularx\csname tabular*\endcsname
173     \expandafter\let\expandafter\endtabularx\csname endtabular*\endcsname
174 }%

```

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

```

175     \@ifundefined{pfx@includegraphicx}{}{}%
176     \PreviewMacro[{}{}]{\pfx@includegraphicx}%
177 }%

```

`\Gscale@@box` Disable scaling.

```

178     \def\Gscale@@box#1#2#3{%
179         \toks@{\mbox}%
180     }

```

`\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.

```

181     \def\Ginclude@graphics#1{%
182         \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.

```

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

```



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

```
184     {\rule{10pt}{10pt}}%
185     {\ppf@Ginclude@graphics{#1}}%
186     \else
```

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

```
187     \ppf@Ginclude@graphics{#1}%
188     \fi
189   }%

190   \PreviewMacro[{}]{\ppf@Ginclude@graphics}%
191   \let\pdfliteral\@gobble%
192 \or
```

### 3.4 pdfL<sup>A</sup>T<sub>E</sub>X 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.

```
193   \PackageInfo{pst-pdf}{MODE: \ppf@TeX@mode\space (pdfTeX mode)}%

Prevent pdfTEX's message Non-PDF special ignored!.

194   \if@ppf@PST@used
195     \let\ppf@temp\AtBeginDvi\let\AtBeginDvi\@gobble
196     \def\c@lor@to@ps#1 #2\@{}
197     \RequirePackage{pstricks}\let\AtBeginDvi\ppf@temp
198   \fi

199   \@temptokena{%
200     \let\Gin@PS@file@header\@gobble\let\Gin@PS@literal@header\@gobble
201     \let\Gin@PS@raw\@gobble\let\Gin@PS@restored\@gobble
202     \@ifundefined{PSTricksLoaded}{-}{%

Necessary if PSTricks < 2.0.
```

```
203       \PSTricksOff
204       \def\c@lor@to@ps#1 #2\@{}
205     }%
206   }%

PostScript output is now inhibited and later once again.
```

```
207   \the\@temptokena
208   \expandafter\AtBeginDocument\expandafter
209   {\the\@temptokena\@temptokena{}}%
210   \@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.
```

```
211   \newtoks\ppf@output
212   \ppf@output\expandafter{\the\output}%
```

```

213 \let\ppf@nofiles=\nofiles \let\nofiles=\relax
214 \let\ppf@shipout=\shipout
215 \RequirePackage[active]{preview}[2005/01/29]%
216 \let\shipout=\ppf@shipout \let\ppf@shipout=\relax
217 \let\nofiles=\ppf@nofiles \let\ppf@nofiles=\relax
218 \output\expandafter{\the\ppf@output} \ppf@output{}%

```

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

```

219 \long\def\pr@startbox#1#2{%
220   \ifpr@outer
221     \toks@{#2}%
222     \edef\pr@cleanup{\the\toks@}%
223     \setbox\@tempboxa\vbox\bgroup
224     \everydisplay{}%
225     \pr@outerfalse%
226     \expandafter\@firstofone
227   \else
228     \expandafter\@gobble
229   \fi{#1}}%
230 \def\pr@endbox{%
231   \egroup
232   \setbox\@tempboxa\box\voidb@x
233   \ppf@getpicture
234   \pr@cleanup}%

```

(See also the identical definition in DVI mode.)

```

235 \AtBeginDocument{%
236   \ifundefined{pst@object}{}%
237   {%
238     \PreviewMacro[{}*[]%
239     ?\bgroup{#{#1}{#{#1}}}{}%
240     ?\bgroup{#{#1}{#{#1}}}{}%
241     ?({#{#1}){({#{1})}}{)%
242     ?({#{#1}){({#{1})}}{)%
243     ?({#{#1}){({#{1})}}{)%
244     }]{\pst@object}}%
245   }%
246 }%

```

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

```

247 \begingroup
248   \input{dvips.def}%
249   \edef\x{\endgroup\def\noexpand\ppf@other@extensions{\Gin@extensions}}%
250   \x

```

Dummy definition for in DVI mode supported file formats.

```

251 \DeclareGraphicsRule{*}{eps}{*}{}%
252 \define@key{Gin}{innerframe}[true]{%
253   \lowercase{\Gin@boolkey{#1}}{innerframe}}%
254 \define@key{Gin}{frame}[true]{%
255   \lowercase{\Gin@boolkey{#1}}{frame}}%
256 \define@key{Gin}{ignore}[true]{%
257   \lowercase{\Gin@boolkey{#1}}{ignore}}%
258 \define@key{Gin}{frame@@}{%

```

(For internal use only!)

```

259 \edef\@tempa{\toks@{\noexpand\frame{\the\toks@}}}%
260 \ifcase#1\relax
261 \ifGin@innerframe\else\let\@tempa\relax\fi
262 \or
263 \ifGin@frame\else\let\@tempa\relax\fi
264 \fi
265 \@tempa
266 }%
267 \define@key{Gin}{showname}[true]{%
268 \lowercase{\Gin@boolkey{#1}}{showname}}%
269 \define@key{Gin}{namefont}{%
270 \begingroup
271 \@temptokena\expandafter{\ppf@namefont#1}%
272 \edef\x{\endgroup\def\noexpand\ppf@namefont{\the\@temptokena}}%
273 \x
274 }%
275 \newcommand*\ppf@filename{}%
276 \newcommand*\ppf@namefont{\tiny\ttfamily}%
277 \newcommand*\ppf@Gin@keys{}%
278 \let\ppf@Gin@setfile\Gin@setfile

```

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

```

279 \def\Gin@setfile#1#2#3{\ppf@Gin@setfile{#1}{#2}{#3}%
280 \xdef\ppf@filename{%
281 #3\ifx\Gin@page\empty\else(\Gin@page)\fi}}%

```

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

```

282 \def\Gin@ii[#1]#2{%
283 \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.

```

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

```

Finds empty file name when calling `\usepicture`.

```

285 \ifx\ppf@tempb\empty\else
286 \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.

```

287 {%
288 \setkeys{Gin}{#1}%
289 \ifx\ppf@tempb\PDFcontainer
290 \@temptokena{page=\Gin@page}%
291 \fi
292 }%
293 {%
294 \refstepcounter{pspicture}%
295 \@temptokena{page=\the\c@pspicture}\def\ppf@tempb{\PDFcontainer}%
296 }%
297 \fi
298 \ifGin@ignore\else

```

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

```

299     \edef\@tempa{\noexpand\ppf@Gin@ii[frame@@=0,\the\@temptokena,
300         frame@@=1]{\ppf@tempb}}}%
301     \@tempa
302     \ifGin@showname
303         \ppf@namefont
304         \raisebox{-\ht\strutbox}[Opt][Opt]{\llap{\ppf@filename}}%
305         \gdef\ppf@filename{}%
306     \fi
307 \fi
308 \endgroup
309 }%

310 \IfFileExists{\PDFcontainer}%
311 {%

```

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

```

312     \pdfximage{\PDFcontainer}%
313     \edef\ppf@container@max{\the\pdflastximagepages}%

314     \AtEndDocument{%
315         \ifnum\c@pspicture>\z@

```

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

```

316         \ifnum\c@pspicture=\ppf@container@max\else
317             \PackageWarningNoLine{pst-pdf}{%
318                 ‘\PDFcontainer’ contains \ppf@container@max\space pages
319                 \MessageBreak but \the\c@pspicture\space pages are requested:
320                 \MessageBreak File ‘\PDFcontainer’ is no more valid!
321                 \MessageBreak Recreate it
322             }%
323         \fi
324     \fi
325 }%
326 }%
327 {%
328     \def\ppf@container@max{0}%
329     \AtEndDocument{%
330         \ifnum\c@pspicture>\z@
331             \filename@parse{\PDFcontainer}%
332             \PackageWarningNoLine{pst-pdf}{%
333                 File ‘\PDFcontainer’ not found.\MessageBreak
334                 Use the following commands to create it:\MessageBreak
335                 -----
336                 \MessageBreak
337                 latex \jobname.tex\MessageBreak
338                 dvips -o \filename@base.ps \jobname.dvi\MessageBreak
339                 ps2pdf \filename@base.ps\MessageBreak
340                 -----
341             }%
342         \fi
343     }%
344 }%

```

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

```
345 \newcommand\ppf@isnum[1]{%
346   \if!\ifnum9<1#1!\else_\fi\expandafter\@firstoftwo
347   \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`

```
348 \newcommand*\ppf@set@mode{%
349 \newcommand*\ppf@test@mmode{%
350 \ifmmode
351   \ifinner
352     \let\ppf@set@mode=$%
353   \else
354     \def\ppf@set@mode{${$}%
355   \fi
356 \else
357   \let\ppf@set@mode=\@empty
358 \fi
359 }

360 \RequirePackage{environ}%
361 \newenvironment{postscript}[1][]{%
362   \def\@tempa{postscript}%
363   \ifx\@tempa\@currenvir
364     \def\ppf@Gin@keys{#1}%
365   \else
366     \def\ppf@Gin@keys{}%
367   \fi
368   \ppf@@getpicture
369   \Collect@Body\@gobble}{}%
370 \AtBeginDocument{%
371   \@ifundefined{PSTricksLoaded}{-}{%
372     \def\pst@@@picture[#1](#2,#3)(#4,#5){\postscript}%
373     \def\endpspicture{\endpostscript\endgroup}%
374     \@ifundefined{psmatrix}{-}{%
375       \let\psmatrix=\postscript
376       \let\endpsmatrix=\endpostscript}%
377   }%
378   \@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.

```
379   \let\includegraphics=pfx@includegraphics
380   \def\pfx@includegraphicx#1#2{\ppf@@getpicture}%
381   }%
382 }%
```

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

```
383 \def\savepicture#1{%
384   \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`.

```

385 \renewcommand*\usepicture[2][]{%
386   \@ifundefined{ppf@@@#2}%
387   {%
388     \ppf@isnum{#2}%
389     {\ppf@getpicture{#1}{#2}}%
390     {\@latex@error{picture '#2' undefined}\@ehc}%
391   }%
392   {%
393     \begingroup
394     \def\Gin@include@graphics##1{%
395       \xdef\ppf@filename{#2}%
396       \setbox\z@\hbox{\pdfrefximage\@nameuse{ppf@@@#2}}%
397       \Gin@nat@height\ht\z@ \Gin@nat@width\wd\z@
398       \def\Gin@llx{0} \let\Gin@lly\Gin@llx
399       \Gin@defaultbp\Gin@urx{\Gin@nat@width}%
400       \Gin@defaultbp\Gin@ury{\Gin@nat@height}%
401       \Gin@bboxtrue\Gin@viewport@code
402       \Gin@nat@height\Gin@ury bp%
403       \advance\Gin@nat@height-\Gin@lly bp%
404       \Gin@nat@width\Gin@urx bp%
405       \advance\Gin@nat@width-\Gin@llx bp%
406       \Gin@req@sizes
407       \ht\z@\Gin@req@height \wd\z@\Gin@req@width
408       \leavevmode\box\z@}%
409     \define@key{Gin}{type}{}%
410     \includegraphics[scale=1,#1]{}%
411   \endgroup
412   }}%

```

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

```

413 \newcommand*\ppf@getpicture[2]{%
414   \@tempcnta=#2\relax%
415   \ifnum\@tempcnta>\ppf@container@max
416     \PackageWarningNoLine{pst-pdf}{%
417       pspicture No. \the\@tempcnta\space undefined}%
418   \else
419     \includegraphics[draft=\ppf@draft,#1,page=\the\@tempcnta]%
420     {\PDFcontainer}%
421   \fi
422   \gdef\ppf@Gin@keys{}}%

```

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

```

423 \newcommand*\ppf@@getpicture{%
424   \ifpr@outer
425     \refstepcounter{pspicture}%
426     \expandafter\ppf@getpicture\expandafter{\ppf@Gin@keys}%
427     {\the\c@pspicture}%
428   \fi}%

```

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

```

429 \renewenvironment*{pst-pdf-defs}%
430 {%
431   \endgroup
432 %   ??? \@currentline
433   \chardef\ppf@temp=\catcode'\&%
434   \@makeother\&%
435 }{%
436   \catcode'\&=\ppf@temp
437   \begingroup
438   \def\@currentvir{pst-pdf-defs}%
439 }

440 \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  $\text{\TeX}$  compiler force this mode.

```

441 \PackageInfo{pst-pdf}{MODE: \ppf@TeX@mode\space (inactive mode)}%
442 \newenvironment{postscript}[1][\ignorespaces]{%
443   \let\ppf@is@pdf@TeX@graphic\relax
444 }{fi

445 \InputIfFileExists{pst-pdf.cfg}{%
446   \PackageInfo{pst-pdf}{Local config file pst-pdf.cfg used}}{}
447 \</package>

```

## Change History

v1.0a	General: Initial version. . . . . 1	v1.0f	<code>\savepicture</code> : New macro <code>\savepicture</code> . (RN) . . . . . 13
v1.0b	General: Some code and documentation cleaning. (RN) . 1		<code>\usepicture</code> : New macro <code>\usepicture</code> . Useful for putting a PSTricks graphic in a box or something else. (RN) . 14
v1.0c	General: New options “pstricks”, “nopstricks”, “draft” and “final”. (RN) . . . . . 3		General: Config file loading added. (RN) . . . . . 15
v1.0d	General: Redefinition of <code>\includegraphics</code> in modes 0 und 1. Now using of eps graphics directly in pdf $\text{\TeX}$ is possible. (RN) . . . . . 1	v1.0g	<code>\usepicture</code> : Now <code>\usepicture</code> does accept a numerical parameter. (RN) . . . . . 14
v1.0e	<code>postscript</code> : “trim” option added. (RN) . . . . . 7		General: Definition of <code>\PDFcontainer</code> now with <code>\edef</code> . (RN) . . . . . 4
		v1.0h	<code>psmatrix</code> : Based no more on the comment environment from the

	verbatim package. (RN) . . . .	13		General: Scaling e.g. of PostScript pictures now only in extraction mode. Some code cleaning. (RN) . . . . .	1
v1.0i	<code>\ppf@is@pdfTeX@graphic</code> : No more errors for given files without extensions. (RN) . . . .	5	v1.1a	General: Support for the internal PSTricks macro <code>\pst@object</code> . (HjG/RN) . . . . .	8
v1.0j	General: Check <code>AtBeginDocument</code> for package ‘pstricks’ even if “nopstricks” is given. (RN) . . .	1	v1.1b	General: Ignore the call of <code>\nofiles</code> inside of <code>preview</code> . (RN) . . . . .	9
	For <code>\includegraphics</code> <code>\usepicture</code> and <code>postscript</code> the new options “frame”, “framesep”, “framerule”, “linewidth”, and “ignore” added. (RN) . . . . .	1	v1.1c	General: New package option “tightpage” added. (RN) . . . .	1
v1.0k	<code>\Gin@setfile</code> : Show also the pagenumber if exists. (RN) . .	11		Special support for “tabularx”. (RN) . . . . .	8
	<code>\Gin@include@graphics</code> : Prevent division by zero. (RN) . . . . .	8		Supress handling of pdfLaTeX graphic formats in DVI mode. (RN) . . . . .	6
v1.0l	General: Options “framesep”, “framerule”, “linewidth” removed, “fname” and “innerframe” added. (RN) . . . .	1	v1.1d	<code>postscript</code> : Support for PSTricks environment “psmatrix”. (RN)	13
v1.0m	General: New package option “notightpage” added. (RN) . . .	1	v1.1e	General: New option “displaymath” (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 LaTeX 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> : <code>pspicture</code> environment must still parse its arguments. (RN/HjG) . . . . .	13
	Option “fname” renamed to “showname”. (RN) . . . . .	1	v1.1h	<code>\Gin@include@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	<code>\Gin@include@graphics</code> : Correction of the inside check. (RN/HjG) .	8
v1.0q	<code>\usepicture</code> : Now <code>\usepspicture</code> works for all kind of graphics. (RN) . . . . .	14		General: <code>\ifpr@outer</code> must be predefined. (HjG/RN) . . . . .	5
v1.0r	<code>\ppf@is@pdfTeX@graphic</code> : Changed <code>\ppf@is@known@graphic</code> to <code>\ppf@is@pdfTeX@graphic</code> . Now pdfTeX graphics are preferred. (RN) . . . . .	5		Package option “final” also for “graphicx”. (RN) . . . . .	4
v1.0s	<code>\Gin@ii</code> : Rewritten. (RN) . . . .	11			



v1.1k	General: New environment pst-pdf-defs: Support for PSTricks environment “psmatrix” inside user definitions. (RN,HjG) . . . . .	1	“\let\output\@gobble” because of bad side effects. (RN) . . . . .	9
v1.1l	General: Support for the package “psfragx”. (RN) . . . . .	8	postscript: Using environ the environment postscript is now simple and more robust. (RN)	13
v1.1m	General: Merge english and german version of the documentation. (RN) . . . . .	1	v1.1u General: \pdfoutput must be set when loading “pdftex.def” in DVI mode. (RN) . . . . .	6
v1.1n	General: \nofiles added (suggestion of Torsten Bronger). . . . .	6	v1.1v \Gin@ii: Key settings only for pdf graphics. (RN) . . . . .	11
v1.1o	\Gscale@@box: Disable scaling. (RN) . . . . .	8	General: Local redefinition of \pdfoutput to be a counter. (RN) . . . . .	6
v1.1p	General: \let\output\@gobble before loading of “preview” added. (RN) . . . . .	9	v1.2a General: Engine tests changed (RN) . . . . .	4
v1.1q	\nofiles makes \makeindex and \makeglossary to \relax. \@empty is better because of later \renewcommand’s. . . . .	6	v1.2b General: Loading packages ‘if*’ at wrong place (RN) . . . . .	4
v1.1r	General: Problem with “tabularx” and “threeparttable” solved. (RN) . . . . .	8	v1.2c General: “postscript” environment no more allowed before \begin{document} (changed example file). . . . .	1
v1.1s	General: Fixed values for \PreviewBbAdjust because \paperwidth is not allways defined (suggested by Will Robertson). . . . .	7	v1.2d General: \c@lor@to@ps must not be undefined before loading ‘PSTricks’ . . . . .	9
v1.1t	General: Dummy definition of the page key in DVI mode. . . . .	6	Version parameter for ‘graphicx’ and rename \GPT@page to \Gin@page at several places . . .	5
	General: Remove the line		v1.2e \ppf@is@pdfTeX@graphic: Parameter #2 is detokenized when expanded to \pdfTeXext .	5
			General: gobble optional argument for \makeindex, \makeglossary and \printindex . . . . .	6

# 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.

<b>Symbols</b>	<code>psmatrix</code> .. 139, 348	<code>\Gscale@@box</code> ..... 178
<code>\&amp;</code> ..... 433, 434, 436	<code>pspicture</code> 2, <u>138</u> , <u>348</u>	
<code>\@currenvir</code> 50, 363, 438	<code>pst-pdf-defs</code> ...	<b>I</b>
<code>\@currenvline</code> .. 47, 432	..... 3, <u>45</u> , <u>429</u>	<code>\if@ppf@pdfTeX@graphic</code>
<code>\@ehc</code> ..... 390	<code>\everydisplay</code> ..... 224	..... 56, 74
<code>\@expandtwoargs</code> ... 69	<b>F</b>	<code>\if@ppf@PST@used</code> ..
<code>\@firstofone</code> ..... 226		..... 4, 85, 194
<code>\@ifpackageloaded</code> .	<code>\filename@base</code> 338, 339	<code>\if@ppf@tightpage</code> .
..... 100, 103	<code>\filename@parse</code> ... 331	..... 5, 115, 124
<code>\@latex@error</code> ..... 390	<code>\frame</code> ..... 259	<code>\ifGin@frame</code> ... 57, 263
<code>\@makeother</code> ..... 434	<b>G</b>	<code>\ifGin@ignore</code> .. 60, 298
<code>\@ppf@PST@usedfalse</code> 10		<code>\ifGin@innerframe</code> .
<code>\@ppf@PST@usedtrue</code> 4, 9	<code>\Gin@bboxtrue</code> ..... 401	..... 58, 261
<code>\@ppf@pdfTeX@graphicfalse</code>	<code>\Gin@boolkey</code> .....	<code>\ifGin@showname</code> 59, 302
..... 63	. 253, 255, 257, 268	<code>\ifin@</code> ..... 70
<code>\@ppf@pdfTeX@graphictrue</code>	<code>\Gin@defaultbp</code> 399, 400	<code>\ifinner</code> ..... 145, 351
..... 70	<code>\Gin@extensions</code> 54,	<code>\ifmmode</code> ..... 144, 350
<code>\@ppf@tightpagefalse</code> 16	71, 96, 183, 249, 286	<code>\ifpdf</code> ..... 24
<code>\@ppf@tightpagetrue</code>	<code>\Gin@framefalse</code> ... 57	<code>\ifpr@outer</code> .....
..... 5, 17	<code>\Gin@ignorefalse</code> .. 60	. 61, 182, 220, 424
<b>A</b>	<code>\Gin@ii</code> ..... 55, <u>282</u>	<code>\ifvtex</code> ..... 28
<code>\AtBeginDvi</code> ... 195, 197	<code>\Gin@innerframefalse</code> 58	<code>\ifxetex</code> ..... 31
	<code>\Gin@llx</code> ..... 398, 405	<code>\in@</code> ..... 69
	<code>\Gin@lly</code> ..... 398, 403	<code>\includegraphics</code> ..
	<code>\Gin@nat@height</code> ...	.. 2, 379, 410, 419
	. 397, 400, 402, 403	<code>\includegraphicsx</code> ... 3
<b>C</b>	<code>\Gin@nat@width</code> ....	
<code>\color@to@ps</code> .. 196, 204	. 397, 399, 404, 405	<b>J</b>
<code>\c@pspicture</code> 295, 315,	<code>\Gin@page</code> ..... 281, 290	<code>\jobname</code> .. 40, 337, 338
316, 319, 330, 427	<code>\Gin@PS@file@header</code> 200	
<code>\catcode</code> ..... 433, 436	<code>\Gin@PS@literal@header</code>	<b>K</b>
<code>\Collect@Body</code> ..... 369	..... 200	<code>\KV@errx</code> ..... 101
<code>\CurrentOption</code> 12, 15, 19	<code>\Gin@PS@raw</code> ..... 201	
	<code>\Gin@PS@restored</code> .. 201	<b>L</b>
<b>D</b>	<code>\Gin@req@height</code> ... 407	<code>\leavevmode</code> ..... 408
<code>\DeclareGraphicsRule</code>	<code>\Gin@req@sizes</code> .... 406	<code>\long</code> ..... 219
..... 108, 251	<code>\Gin@req@width</code> .... 407	
<code>\define@key</code> 109–114,	<code>\Gin@setfile</code> 67, 278, <u>279</u>	<b>M</b>
252, 254, 256,	<code>\Gin@shownamefalse</code> . 59	<code>\makeglossary</code> ... 80, 82
258, 267, 269, 409	<code>\Gin@urx</code> ..... 399, 404	<code>\makeindex</code> ..... 80, 81
<code>\detokenize</code> ..... 66, 69	<code>\Gin@ury</code> ..... 400, 402	<code>\mbox</code> ..... 179
<b>E</b>	<code>\Gin@viewport@code</code> . 401	
<code>\endpostscript</code> 373, 376	<code>\Gin@vllx</code> ..... 128	<b>N</b>
<code>\endpsmatrix</code> .....	<code>\Gin@vllly</code> ..... 128	<code>\newcolumnType</code> .... 171
.... 157, 161, 376	<code>\Gin@vurx</code> ..... 128	<code>\newcount</code> ..... 93
<code>\endpspicture</code> ..... 373	<code>\Gin@vury</code> ..... 128	<code>\nofiles</code> .. 79, 213, 217
<code>\endtabularx</code> ..... 173	<code>\Gin@vury</code> ..... 128	
environments:	<code>\Gin@vury</code> ..... 128	<b>O</b>
<code>postscript</code> 2, <u>121</u> , <u>348</u>	<code>\Gin@vury</code> ..... 128	<code>\OptionNotUsed</code> ..... 6
	..... 53, <u>181</u> , 394	

\output . . . . .	212, 218	\ppf@Gin@setfile . .		\PreviewBbAdjust . .	
\overfullrule . . . . .	84		278, 279		89, 116, 123, 127
<b>P</b>					
\PassOptionsToPackage		\ppf@Gin@graphics		\PreviewEnvironment	
. . . . .	12, 15, 19	. . . . .	53,	. . . .	134, 138, 141
\PDFcontainer . . . . .		72, 185, 187, 190		\PreviewMacro . . . . .	
39, 40, 289, 295,		\ppf@is@pdfTeX@graphic		. 163, 176, 190, 238	
310, 312, 318,		. 62, 183, 286, 443		\printindex . . . . .	80, 83
320, 331, 333, 420		\ppf@isnum . . .	345, 388	\psmatrix . . . . .	154,
\pdflastximage . . . . .	384	\ppf@namefont . . . . .		156, 158, 159, 375	
\pdflastximagepages	313	. 271, 272, 276, 303		psmatrix (environ-	
\pdfliteral . . . . .	191	\ppf@nofiles . .	213, 217	ment) . . .	139, 348
\pdfoutput . . . . .	93, 94	\ppf@other@extensions		pspicture (environ-	
\pdfrefximage . . . . .	396	. . . . .	42, 96,	ment) . 2, 138, 348	
\pdfTeXtext . . . . .	65, 66, 69	106, 183, 249, 286		pst-pdf-defs (environ-	
\pdfTeXversion . . . . .	92	\ppf@output 211, 212, 218		ment) . . 3, 45, 429	
\pdfximage . . . . .	312	\ppf@pr@psmatrix . .		\pst@@@picture . . . . .	372
\pfx@includegraphics		. . . . .	155, 160	\pst@object . . . . .	169, 244
. . . . .	379	\ppf@PreviewBbAdjust		\PSTricksOff . . . . .	203
\pfx@includegraphicx		. . . . .	87, 89, 123	<b>R</b>	
. . . . .	176, 380	\ppf@psmatrix . .	154, 158	\raisebox . . . . .	304
\postscript . . . . .	372, 375	\ppf@RestoreBbAdjust		\refstepcounter 294, 425	
postscript (environ-		. . . . .	88, 133	\rule . . . . .	184
ment) . 2, 121, 348		\ppf@set@mode . . . . .		<b>S</b>	
\ppf@getpicture . .		. 142, 146, 148,		\savepicture . . 3, 44, 383	
. 233, 368, 380, 423		151, 160, 161,		\setkeys . . . . .	126, 288
\ppf@container@max .		348, 352, 354, 357		\shipout . . . . .	214, 216
. . . . .	312,	\ppf@shipout . .	214, 216	\string . . . . .	156, 159
316, 318, 328, 415		\ppf@temp . . . . .		\strutbox . . . . .	304
\ppf@draft 3, 13, 14, 419		. 195, 197, 433, 436		<b>T</b>	
\ppf@endpsmatrix . .		\ppf@tempb . . . . .	284,	\tabularx . . . . .	172
. . . . .	157, 161	285, 289, 295, 300		<b>U</b>	
\ppf@filename . 275,		\ppf@test@mmode . . .		\usepicture . . 3, 43, 385	
280, 304, 305, 395		. . . . .	143, 158, 349	<b>V</b>	
\ppf@getpicture . . .		\ppf@TeX@mode . . . . .		\voidb@x . . . . .	232
. . . . .	389, 413, 426	. . . 2, 7, 8, 21,		<b>X</b>	
\ppf@Gin@extensions 54		23, 25, 29, 32,		\XKV@err . . . . .	104
\ppf@Gin@ii . . . . .	55, 299	34, 76, 78, 193, 441			
\ppf@Gin@keys . 277,		\pr@cleanup . . . . .	222, 234		
364, 366, 422, 426		\pr@endbox . . . . .	230		
		\pr@outerfalse . 72, 225			
		\pr@startbox . . . . .	219		