

Das Paket `pst-pdf`*

Rolf Niepraschk[†] Hubert Gäßlein

2017/06/22

1 Einleitung

Das Paket `pst-pdf` vereinfacht die Verwendung von PSTricks-Grafiken und anderem PostScript-Code in PDF-Dokumenten. Ähnlich wie beim Erstellen des Literaturverzeichnisses mit `bibTeX` werden zusätzlich externe Programme aufgerufen. Sie dienen in diesem Fall dazu, eine PDF-Datei, die sämtliche Grafiken enthält, zu erzeugen. Ihr Inhalt wird im endgültigen Dokument statt des ursprünglichen PostScript-Codes eingefügt.

2 Anwendung

2.1 Paketoptionen

active Aktiviert den Extraktionsmodus (DVI-Ausgabe). Die explizite Angabe ist normalerweise unnötig (Standard im `LATEX`-Modus).

inactive Keine besonderen Aktionen; es werden nur die Pakete `pstricks` und `graphicx` geladen (Standard bei Verwendung von `VTEX`). Kann dazu benutzt werden, um das Dokument mit `LATEX` in eine DVI-Datei zu wandeln und dabei die automatische Verwendung des Extraktionsmodus' zu vermeiden.

pstricks Das Paket `pstricks` wird geladen (Standard).

nopstricks Das Paket `pstricks` wird nicht geladen. Wird später festgestellt, dass `pstricks` doch noch anderweitig geladen wurde, wird die Umgebung `pspicture` nachträglich in der Weise behandelt, als wäre die Option "pstricks" doch angegeben worden.

draft Im `pdfLATEX`-Modus werden aus der Containerdatei eingefügte Grafiken nur als Rahmen dargestellt.

final Im `pdfLATEX`-Modus werden aus der Containerdatei eingefügte Grafiken vollständig dargestellt (Standard).

tightpage Die Abmessung Grafiken in der Containerdatei entsprechen denen der zugehörigen `TEX`-Boxen (Standard).

*Dieses Dokument bezieht sich auf `pst-pdf` v1.2d vom 2017/06/22.

[†]`Rolf.Niepraschk@gmx.de`

notightpage Die Abmessung der zur Grafik gehörenden \TeX -Box ist manchmal nicht korrekt, da PostScript-Anweisungen auch außerhalb der Box zeichnen können. Die Option “notightpage” führt dazu, dass die Grafiken in der Containerdatei mindestens die Größe des gesamten Blattes einnehmen. Um die Grafiken im späteren pdf \LaTeX -Lauf verwenden zu können, muss die Containerdatei nachbearbeitet werden, so dass die Größe der Grafiken auf die der sichtbaren Bestandteile reduziert ist. Dazu kann z. B. das Programm `pdfcrop`¹ dienen. Die Anwendung dieses Verfahrens kann die Angabe der Option “trim” erübrigen (siehe Abschnitt 2.4).

displaymath Es werden zusätzlich die mathematischen Umgebungen `displaymath`, `eqnarray` und `$$` extrahiert und im pdf-Modus als Grafik eingefügt. So können zusätzliche PSTricks-Ergänzungen leicht dem Inhalt dieser Umgebungen zugefügt werden. (Frage: Wie verhalten sich die AMS \LaTeX -Umgebungen?)

(*other*) Alle anderen Optionen werden an das Paket `pstricks` weitergereicht.

2.2 Programmaufrufe

Die folgende Tabelle zeigt den Ablauf, der nötig ist, um ein PDF-Dokument mit PostScript-Grafiken zu erzeugen². Im Vergleich dazu ist der analoge Ablauf für Literaturverzeichnisse angegeben.

PostScript-Grafiken	Literaturverzeichnis
<code>pdflatex document.tex</code>	<code>pdflatex document.tex</code>
<i>Hilfsaufrufe</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>

Bei der Erzeugung wird nur Code berücksichtigt, der sich innerhalb der Umgebungen `pspicture` oder `postscript` befindet. Ebenfalls werden PostScript-Grafiken, die als Parameter von `\includegraphics` angegeben wurden, der Containerdatei hinzugefügt. Der Name dieser Datei ist standardmäßig `\jobname-pics.pdf`. Er kann durch Umdefinieren des Makros `\PDFcontainer` geändert werden.

2.3 Nutzeranweisungen

<code>pspicture</code>	<code>\begin{pspicture}[(keys)] (<x0,x1>)<(y0,y1)> ... \end{pspicture}</code>
	Die <code>pspicture</code> -Umgebung steht zur Verfügung, wenn nicht die Option “nopstricks” angegeben wurde. Sie wird so wie in PSTricks üblich verwendet. Im pdf \LaTeX -Modus wird ihr Inhalt nur dann dargestellt, wenn vorher die Containerdatei erzeugt wurde.
<code>postscript</code>	<code>\begin{postscript}[(keys)] ... \end{postscript}</code>

¹CTAN: `support/pdfcrop/`

²Die Shell-Skripte `ps4pdf` bzw. `ps4pdf.bat` führen alle angegebenen Programmaufrufe automatisch aus.

Die `postscript`-Umgebung kann beliebigen Code mit Ausnahme von Gleitumgebungen aufnehmen. Im `pdfLATEX`-Modus wird ihr Inhalt ebenfalls der Containerdatei entnommen. Ist diese Datei nicht vorhanden, wird – anders als bei der `pspicture`-Umgebung – der später benötigte Platz möglicherweise nicht korrekt frei gehalten.

<code>\includegraphics</code>	<code>\includegraphics[<i>keys</i>]{<i>filename</i>}</code> Wie in <code>graphics/graphics</code> definiert zu verwenden. Zusätzlich ist es nun möglich, auch im <code>pdfL^AT_EX</code> -Modus EPS-Dateien als Argument anzugeben und ihren Inhalt darzustellen. Er wird dazu ebenfalls der Containerdatei entnommen.
<code>\includegraphicx</code>	<code>\includegraphicx[<i>keys</i>](<i>pxadd</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> Die zuletzt ausgegebene Grafik (Ergebnisse der Umgebungen <code>pspicture</code> , <code>postscript</code> und der <code>\includegraphics</code> -Anweisungen mit PostScript-Dateien) wird unter dem als Parameter übergebenen Namen gespeichert.
<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 entpricht 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

Das Verhalten der Anweisungen `\includegraphics`, `\usepicture` und der Umgebung `postscript` kann mit den folgenden optionalen Parametern beeinflusst werden (key-value-Syntax):

frame=`(true|false)` Es wird – ähnlich wie bei der Anweisung `\fbox` – ein Rahmen um die Grafik gezeichnet. Die durch Rotation geänderte Gesamtgröße wird dabei berücksichtigt. Das Zeichnen geschieht im `pdfLATEX`-Modus; vorher beim Erzeugen der Containerdatei wird dieser Parameter ignoriert. Standard: `false`.

innerframe=`(true|false)` Wie “**frame**” jedoch wird der Rahmen nur um die Grafik selbst, nicht aber um die resultierende Box gezeichnet.

ignore=`(true|false)` Bei “**true**” wird die Grafik nicht ausgegeben. Bei Angabe von `\savepicture{name}` kann sie später jedoch an anderer Stelle mit `\usepicture` verwendet werden. Standard: `false`.

showname=`(true|false)` Gibt in kleiner Schrift den tatsächlich verwendeten Dateinamen unter der Grafik aus. Standard: `false`.

namefont=`(font commands)` Beeinflusst die Schriftart, die bei “**showname=true**” benutzt wird. Standard: `\ttfamily\tiny`

Alle Parameter können auch global per `\setkeys{Gin}{key=value}` gesetzt werden.

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

Es wird getestet, welcher $\text{T}_{\text{E}}\text{X}$ compiler in welchem Modus läuft (siehe ‘graphics.cfg’ von $\text{t}_{\text{E}}\text{X}/\text{T}_{\text{E}}\text{XLive}$). Entsprechend dem Ergebnis bekommen die Umgebungen `pspicture` und `postscript` unterschiedliche Funktionalität. Der Test wird nur ausgeführt, wenn nicht die Paketoptionen `active` oder `inactive` angegeben wurden.

```
22 \RequirePackage{ifpdf,ifxetex,ifvtex}
23 \ifnum\ppf@TeX@mode=-1\relax
24 \ifpdf
    ⇒ pdf $\text{T}_{\text{E}}\text{X}$  or Lua $\text{T}_{\text{E}}\text{X}$  are running in PDF mode
25 \def\ppf@TeX@mode{1}%
26 \RequirePackage{luatex85}%
27 \else
28 \ifvtex
    ⇒ V $\text{T}_{\text{E}}\text{X}$ 
29 \def\ppf@TeX@mode{9}%
30 \else
31 \ifxetex
    ⇒ Xe $\text{T}_{\text{E}}\text{X}$ 
32 \def\ppf@TeX@mode{9}%
33 \else
    ⇒ DVI mode
34 \def\ppf@TeX@mode{0}%
35 \fi
```

```

36   \fi
37   \fi
38 \fi

39 \newcommand*\PDFcontainer{}
40 \edef\PDFcontainer{\jobname-pics.pdf}
41 \newcounter{pspicture}
42 \newcommand*\ppf@other@extensions[1]{}
43 \newcommand*\usepicture[2][1]{}
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}[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 wird eigentlich im Paket preview definiert. Wir müssen es aber bereits hier zusätzlich tun, da sonst T_EX u. U. beim Parsen der \ifcase-Struktur “außer Tritt” kommt.

```
61 \newif\ifpr@outer
```

\ppf@is@pdfTeX@graphic

Parameter #1 ist der Name einer Grafikdatei mit oder ohne Endung, Parameter #2 enthält die gültigen Dateieindungen im pdf-Modus, Parameter #3 enthält die gültigen Dateieindungen im dvi-Modus. Ist es möglich, die Grafik im pdf-Modus zu verarbeiten, werden die Anweisungen in #4 ausgeführt, sonst die in #5.

```

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

```

Statt Einladen einer identifizierten Grafik nur Test der Grafikendung.

```

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

```

Es müssen Dateitypen beider Moden gefunden werden, um die Fehlermeldung “File ‘#1’ not found” zu vermeiden.

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

```

Testaufruf. Dabei Ausgabe vollständig verhindern.

```
71 \pr@outerfalse\ppf@Ginclude@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)

Die Umgebung `pspicture` behält die Definition aus `pstricks.tex`. Ausschließlich der Code der Umgebungen `pspicture` und `postscript` sowie `\includegraphics` mit PS-Dateien bewirken Einträge in die DVI-Datei. Der restliche Code des Dokuments wird bei der Ausgabe der DVI-Datei ignoriert. Nach Wandlung der DVI-Datei über PostScript (“dvips”) nach PDF (Datei `\PDFcontainer`) nimmt jede Grafik genau eine Seite der pdf-Datei ein. Der $\text{T}_{\text{E}}\text{X}$ -Compiler mit DVI-Ausgabe sowie die Paketoption “active” erzwingen diesen Modus.

```
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 \if@ppf@PST@used\RequirePackage{pstricks}\fi
82 \RequirePackage[active,dvips,tightpage]{preview}[2005/01/29]%
83 \newcommand*\ppf@PreviewBbAdjust{}
84 \newcommand*\ppf@RestoreBbAdjust{}
85 \let\PreviewBbAdjust\ppf@PreviewBbAdjust}%
```

Es werden auch die im pdf \LaTeX -Modus erlaubten Endungen von Grafikdateien benötigt.

```
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
```

Für die im PDF-Modus möglichen Grafikformate dürfen keine speziellen Regeln definiert sein (z. B. wegen ‘dvips’-Erweiterungen). Für sie wird die universelle EPS-Regel verwendet, damit sie zumindest gefunden werden.

```
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 diesem Modus sollten undefinierte keys keinen Fehler bewirken.

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

In diesem Modus keine Funktion.

```
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{Gin@page}{\define@key{Gin}{page}{}{}}{}

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

postscript Die Umgebung `postscript` wertet die `trim`-Option in derselben Weise wie `\includegraphics` aus (Angaben ohne Maßeinheit werden als `bp` interpretiert).

```
117 \newenvironment{postscript}[1][]{%
118   {%
119     \global\let\ppf@PreviewBbAdjust\PreviewBbAdjust
120     \ifppf@tightpage
121       \begingroup
122         \setkeys{Gin}{#1}%
123         \xdef\PreviewBbAdjust{%
124           -\Gin@vllx bp -\Gin@villy 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 Originaldefinition `preview` bekannt machen.

```
134   \PreviewEnvironment{pspicture}%
```

psmatrix Originaldefinition `preview` bekannt machen.

```
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@mode\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     }%

```

Internes Makro `\pst@object` bekanntmachen, um manchen PSTricks-Code außerhalb von `pspicture`-Umgebungen ebenfalls verwenden zu können. Derzeit sind Aufrufe der folgenden Art möglich:

```

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

```

Mehr als drei optionale Argumente am Ende des Aufrufs, wie beispielsweise bei `\psline` denkbar, sind noch nicht möglich.

```

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

```

Mehrfaches testweises Setzen von Tabelleninhalten durch “`tabularx`” verhindern.

```

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

```

Unterstützung von `\includegraphicx` aus dem Paket `psfragx`.

```

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

```

`\Gscale@@@box` Skalieren verhindern.

```

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

```

`\Ginclude@graphics` Alle Grafiken mit bekanntem Format (z. B. EPS-Dateien) werden normal verarbeitet, was in diesem Modus bedeutet, dass sie der Preview-Funktionalität unterliegen. Andere Grafiken (z. B. PDF-Dateien) werden ignoriert.

```

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

```

Im allgemeinen Fall sollen pdf_{TEX}-Grafiken bevorzugt werden (Einfügen erst im pdf_{TEX}-Modus). Ist nur eine DVIPS-Graphik vorhanden, dann wirkt wieder die Originaldefinition und Registrierung beim `preview`-Paket muss erfolgen.

```

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

```

Dummy-Box, um Division durch Null bei Skalierung/Rotation zu vermeiden. Wird ansonsten ignoriert.

```

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

Innerhalb von PS-Umgebungen (pspicture usw.) muss sich \includegraphics
wie die Originaldefinition verhalten (nur die DVIPS-Graphik-Typen sind gültig).
183     \ppf@Ginclude@graphics{#1}%
184     \fi
185   }%

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

```

3.4 pdfL^AT_EX mode (pdf output)

Ist die Datei \PDFcontainer (default: \jobname)-pics.pdf vorhanden, so wird der Inhalt der Umgebungen pspicture und postscript ignoriert. Stattdessen wird die zugehörige Grafik aus der Datei \PDFcontainer eingebunden.

```

189   \PackageInfo{pst-pdf}{MODE: \ppf@TeX@mode\space (pdfTeX mode)}%
Verhindert pdfTEXs Warnung Non-PDF special ignored!.
190   \ifppf@PST@used
191     \let\ppf@temp\AtBeginDvi\let\AtBeginDvi@gobble
192     \def\c@lor@to@ps#1 #2@@@{}
193     \RequirePackage{pstricks}\let\AtBeginDvi\ppf@temp
194   \fi

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

```

Für PSTricks < 2.0 nötig.

```

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

```

PostScript-Ausgabe jetzt verhindern und später noch einmal.

```

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

```

Zum Parsen der Argumente von PSTricks' \pst@object laden wir preview im active-Modus, restaurieren aber die standardmäßigen Definitionen von \output und \shipout. \pr@startbox und \pr@endbox dienen hier nur dazu, um \pst@object wirkungslos zu machen und stattdessen die zugehörige Grafik aus der Containerdatei einzuladen. Derzeit werden nur maximal 3 optionale Parameter in runden Klammern am Ende von \pst@object unterstützt, was für viele, aber nicht für alle Fälle ausreichend ist.

```

205   \newtoks\ppf@output
206   \ppf@output\expandafter{\the\output}%
207   \let\ppf@nofiles=\nofiles \let\nofiles=\relax

```

```

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

```

\pr@startbox, \pr@endbox: Gegenüber Originaldefinition vereinfacht.

```

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

```

(Siehe auch identische Definition im DVI-Modus.)

```

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

```

Es werden auch die im DVI-Modus erlaubten Endungen von Grafikdateien benötigt.

```

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

```

Dummy-Definition für die im DVI-Modus gültigen Dateitypen.

```

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

```

(Nur intern zu benutzen!)

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

`\Gin@setfile` Realen Dateinamen und ggf. Seitenzahl zur späteren Verwendung merken.

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

`\Gin@ii` Auswertung der Optionen “frame”, “ignore” usw. sowie weiterer Spezialfälle.

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

Der Wert `\ifGin@innerframe` muss bereits vor Zeichnen des inneren Rahmens bekannt sein. Die Werte für `\ifGin@showname` und `\ppf@namefont` müssen auch nach Darstellung der Grafik verfügbar sein. Daher durch eine Gruppe geschützt vorher Auswertung der Optionen.

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

Leerer Dateiname beim Aufruf von `\usepicture` aus.

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

Grafiken aus Containerdatei sind bereits skaliert usw. Nicht noch einmal, daher optionalen Parameter ignorieren.

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

“frame@@=0” = innerer Rahmen, “frame@@=1” = äußerer Rahmen.

```
293     \edef\@tempa{\noexpand\ppf@Gin@ii[frame@@=0,\the\@temptokena,
294     frame@@=1]{\ppf@tempb}}%
295     \@tempa
296     \ifGin@showname
297     \ppf@namefont
298     \raisebox{-\ht\strutbox}[Opt][Opt]{\llap{\ppf@filename}}%
299     \gdef\ppf@filename{}%
300     \fi
301     \fi
302 \endgroup
303 }%

304 \IfFileExists{\PDFcontainer}%
305 {%
```

\ppf@container@max Die Anzahl der in der Containerdatei enthaltenen Seiten.

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

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

Warnung ist nur sinnvoll, wenn überhaupt Grafiken benötigt wurden.

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

`\ppf@isnum` Ist Parameter #1 numerisch, werden Anweisungen in #2 sonst die in #3 ausgeführt (siehe `bibtopic.sty`).

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

`psmatrix` Beide Umgebungen ignorieren ihren Inhalt und laden stattdessen die zugehörige Grafik aus der Containerdatei. Auf den Wert des dabei benutzten Zählers (`pspicture`) kann per `\label/\ref` zugegriffen werden.

`postscript`

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

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

Die im pdfTeX-Modus unnütze Umdefinition von `\includegraphics` (Paket `psfrag`) führt zu zweifachem Einfügen des Ergebnisses, weshalb die Originaldefinition wiederhergestellt wird.

```
373 \let\includegraphics=\pfx@includegraphics
374 \def\pfx@includegraphicx#1#2{\ppf@@getpicture}%
375 }%
376 }%
```

`\savepicture` Speichert die Nummer der aktuellen Grafik in einem Makro mit Namen `\ppf@@@#1`.

```

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

```

`\usepicture` Fügt Grafik mit symbolischem Namen #2 ein. Der Name muss vorher mit `\savepicture{<Name>}` vereinbart worden sein. Statt des Namens kann auch eine Zahl angegeben werden, die dann direkt eine Grafik aus der Containerdatei adressiert. Der optionale Parameter #1 entspricht dem bei `\includegraphics`.

```

379 \renewcommand*\usepicture[2] []{%
380 \ifundefined{ppf@@#2}%
381 {%
382 \ppf@isnum{#2}%
383 {\ppf@getpicture{#1}{#2}}%
384 {\@latex@error{picture ‘#2’ undefined}\@ehc}%
385 }%
386 {%
387 \begingroup
388 \def\Gin@include@graphics##1{%
389 \xdef\ppf@filename{#2}%
390 \setbox\z@\hbox{\pdfrefximage\@nameuse{ppf@@#2}}%
391 \Gin@nat@height\ht\z@ \Gin@nat@width\wd\z@
392 \def\Gin@llx{0} \let\Gin@lly\Gin@llx
393 \Gin@defaultbp\Gin@urx{\Gin@nat@width}%
394 \Gin@defaultbp\Gin@ury{\Gin@nat@height}%
395 \Gin@bboxtrue\Gin@viewport@code
396 \Gin@nat@height\Gin@ury bp%
397 \advance\Gin@nat@height-\Gin@lly bp%
398 \Gin@nat@width\Gin@urx bp%
399 \advance\Gin@nat@width-\Gin@llx bp%
400 \Gin@req@sizes
401 \ht\z@\Gin@req@height \wd\z@\Gin@req@width
402 \leavevmode\box\z@}%
403 \define@key{Gin}{type}{}%
404 \includegraphics[scale=1,#1]{}%
405 \endgroup
406 }}%

```

`\ppf@getpicture` Fügt die Seite (Grafik) mit Nummer #2 aus der Containerdatei ein. Parameter #1: Optionen wie bei `\includegraphics`.

```

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

```

`\ppf@@getpicture` Fügt die nächste Seite (Grafik) aus der Containerdatei ein.

```

417 \newcommand*\ppf@@getpicture{%
418 \ifpr@outer
419 \refstepcounter{pspicture}%

```

```

420     \expandafter\ppf@getpicture\expandafter{\ppf@Gin@keys}%
421     {\the\c@pspicture}%
422     \fi}%

```

`pst-pdf-defs` Umgebung, die keine eigene Gruppe aufmacht. Innerhalb der Umgebung bekommt das Zeichen `&` den Kategoriecode „other“. Gedacht für eigene Makrodefinitionen, die z. B. eine `psmatrix` enthalten.

```

423 \renewenvironment*{pst-pdf-defs}%
424 {%
425   \endgroup
426 %   ??? \@currentvline
427   \chardef\ppf@temp=\catcode'\&%
428   \@makeother\&%
429 }{%
430   \catcode'\&=\ppf@temp
431   \begingroup
432   \def\@currentvir{pst-pdf-defs}%
433 }
434 \else

```

3.5 Inactiver Modus

Es werden nur die Pakete `pstricks` und `graphicx` geladen – keine weitere Einflussnahme. Die Paketoption „inactive“ sowie der \TeX -Compiler erzwingen diesen Modus.

```

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

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

```

Change History

v1.0a	possible. (RN)	1
General: Initial version.		1
v1.0b	<code>postscript</code> : “trim” option added. (RN)	7
General: Some code and documentation cleaning. (RN)		1
v1.0c	<code>\savepicture</code> : New macro <code>\savepspicture</code> . (RN)	13
General: New options “pstricks”, “nopstricks”, “draft” and “final”. (RN)	<code>\usepicture</code> : New macro <code>\usepspicture</code> . Useful for putting a PSTricks graphic in a box or something else. (RN)	4
v1.0d	General: Redefinition of <code>\includegraphics</code> in modes 0 und 1. Now using of eps graphics directly in pdf \TeX is (RN)	14
	General: Config file loading added. (RN)	15

v1.0g	\usepicture: Now \usepspicture does accept a numerical parameter. (RN) 14	Now pdf \TeX graphics are preferred. (RN) 5
	General: Definition of \PDFcontainer now with \edef. (RN) 5	v1.0s
v1.0h	psmatrix: Based no more on the comment environment from the verbatim package. (RN) 13	\Gin@ii: Rewritten. (RN) 11
v1.0i	\ppf@is@pdfTeX@graphic: No more errors for given files without extensions. (RN) 5	General: Scaling e.g. of PostScript pictures now only in extraction mode. Some code cleaning. (RN) 1
v1.0j	General: Check AtBeginDocument for package ‘pstricks’ even if “nopstricks” is given. (RN) . . . 1	v1.1a
v1.0k	\Gin@setfile: Show also the pagenumber if exists. (RN) . . 11	General: Support for the internal PSTricks macro \pst@object. (HjG/RN) 8
	\Gin@include@graphics: Prevent division by zero. (RN) 8	v1.1b
v1.0l	General: Options “framesep”, “framerule”, “linewidth” removed, “fname” and “innerframe” added. (RN) 1	General: Ignore the call of \nofiles inside of preview. (RN) 9
v1.0m	General: New package option “notightpage” added. (RN) . . . 1	Some code and documentation cleaning. (RN) 1
v1.0n	General: Changed macro names (\savepicture and \usepicture). (RN) 1	v1.1c
	Some code cleaning. (RN) 1	General: New package option “tightpage” added. (RN) 1
v1.0o	General: New code for “notightpage”. (RN) 7	Special support for “tabularx”. (RN) 8
	Option “fname” renamed to “showname”. (RN) 1	Supress handling of pdf \TeX graphic formats in DVI mode. (RN) 6
v1.0p	General: Some code and documentation cleaning. (RN) . 1	v1.1d
v1.0q	\usepicture: Now \usepspicture works for all kind of graphics. (RN) 14	postscript: Support for PSTricks environment “psmatrix”. (RN) 13
v1.0r	\ppf@is@pdfTeX@graphic: Changed \ppf@is@known@graphic to \ppf@is@pdfTeX@graphic.	v1.1e
		General: New option “displaymath” (see preview package). (HjG/RN) 4
		v1.1f
		General: Package option “ignore” reimplemented. Now the compilation of the dtx file in \TeX mode is possible. (RN) . 4
		v1.1g
		postscript: “psmatrix” environment (preserve math mode). (RN/HjG) 13
		pspicture: pspicture environment must still parse its arguments. (RN/HjG) 13
		v1.1h
		\Gin@include@graphics: Check if inside of a PS-related environment (correct graphic inclusion). (RN) 8
		v1.1i
		\Gin@include@graphics: Correction of the inside check. (RN/HjG) . 8

General: <code>\ifpr@outer</code> must be predefined. (HjG/RN)	5	defined (suggested by Will Robertson).	7
Package option “final” also for “graphicx”. (RN)	4	v1.1s General: Dummy definition of the page key in DVI mode.	7
v1.1k General: New environment <code>pst-pdf-defs</code> : Support for PSTricks environment “psmatrix” inside user definitions. (RN,HjG)	1	v1.1t General: Remove the line “ <code>\let\output@gobble</code> ” because of bad side effects. (RN)	9
v1.1l General: Support for the package “psfrag”. (RN)	8	<code>postscript</code> : Using <code>environ</code> the environment <code>postscript</code> is now simple and more robust. (RN)	13
v1.1m General: Merge english and german version of the documentation. (RN)	1	v1.1u General: <code>\pdfoutput</code> must be set when loading “ <code>pdftex.def</code> ” in DVI mode. (RN)	6
v1.1n General: <code>\nofiles</code> added (suggestion of Torsten Bronger).	6	v1.1v <code>\Gin@ii</code> : Key settings only for pdf graphics. (RN)	11
v1.1o <code>\Gscale@@box</code> : Disable scaling. (RN)	8	General: Local redefinition of <code>\pdfoutput</code> to be a counter. (RN)	6
v1.1p General: <code>\nofiles</code> makes <code>\makeindex</code> and <code>\makeglossary</code> to <code>\relax</code> . <code>\@empty</code> is better because of later <code>\renewcommand</code> ’s.	6	v1.2a General: Engine tests changed (RN)	4
v1.1p1 General: <code>\let\output@gobble</code> before loading of “preview” added. (RN)	9	v1.2b General: Loading packages ‘if*’ at wrong place (RN)	4
v1.1q General: Problem with “ <code>tabularx</code> ” and “ <code>threeparttable</code> ” solved. (RN)	8	v1.2c General: “ <code>postscript</code> ” environment no more allowed before <code>\begin{document}</code> (changed example file).	1
v1.1r General: Fixed values for <code>\PreviewBbAdjust</code> because <code>\paperwidth</code> is not always		v1.2d General: <code>\color@to@ps</code> must not be undefined before loading ‘PSTricks’	9
		Version parameter for ‘ <code>graphicx</code> ’ and rename <code>\GPT@page</code> to <code>\Gin@page</code> at several places	5

Index

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

Symbols	pst-pdf-defs ...	I
<code>\&</code> 427, 428, 430	<code>\everydisplay</code> 218	<code>\if@ppf@pdftex@graphic</code> 56, 73
<code>\@currenvir</code> 50, 357, 432		<code>\if@ppf@PST@used</code> 4, 81, 190
<code>\@currenvline</code> .. 47, 426		<code>\if@ppf@tightpage</code> 5, 111, 120
<code>\@ehc</code> 384	F	<code>\ifGin@frame</code> ... 57, 257
<code>\@firstofone</code> 220	<code>\filename@base</code> 332, 333	<code>\ifGin@ignore</code> .. 60, 292
<code>\@ifpackageloaded</code> 96, 99	<code>\filename@parse</code> ... 325	<code>\ifGin@innerframe</code> 58, 255
<code>\@latex@error</code> 384	<code>\frame</code> 253	<code>\ifGin@showname</code> 59, 296
<code>\@makeoother</code> 428		<code>\ifinner</code> 141, 345
<code>\@ppf@PST@usedfalse</code> 10	G	<code>\ifmmode</code> 140, 344
<code>\@ppf@PST@usedtrue</code> 4, 9	<code>\Gin@bboxtrue</code> 395	<code>\ifpdf</code> 24
<code>\@ppf@pdftex@graphicfalse</code> 63	<code>\Gin@boolkey</code> 247, 249, 251, 262	<code>\ifpr@outer</code> 61, 178, 214, 418
<code>\@ppf@pdftex@graphictrue</code> 69	<code>\Gin@defaultbp</code> 393, 394	<code>\ifvtex</code> 28
<code>\@ppf@tightpagefalse</code> 16	<code>\Gin@extensions</code> 54, 70, 92, 179, 243, 280	<code>\ifxetex</code> 31
<code>\@ppf@tightpagetrue</code> 5, 17	<code>\Gin@framefalse</code> ... 57	<code>\includegraphics</code> 3, 373, 404, 413
	<code>\Gin@ignorefalse</code> .. 60	<code>\includegraphicsx</code> ... 3
A	<code>\Gin@ii</code> 55, <u>276</u>	
<code>\AtBeginDvi</code> ... 191, 193	<code>\Gin@innerframefalse</code> 58	J
	<code>\Gin@llx</code> 392, 399	<code>\jobname</code> .. 40, 331, 332
	<code>\Gin@lly</code> 392, 397	K
C	<code>\Gin@nat@height</code> 391, 394, 396, 397	<code>\KV@errx</code> 97
<code>\c@lor@to@ps</code> .. 192, 200	<code>\Gin@nat@width</code> 391, 393, 398, 399	L
<code>\c@pspicture</code> 289, 309, 310, 313, 324, 421	<code>\Gin@page</code> 275, 284	<code>\leavevmode</code> 402
<code>\catcode</code> 427, 430	<code>\Gin@PS@file@header</code> 196	<code>\long</code> 213
<code>\Collect@Body</code> 363	<code>\Gin@PS@literal@header</code> 196	M
<code>\CurrentOption</code> 12, 15, 19	<code>\Gin@PS@raw</code> 197	<code>\makeglossary</code> 79
	<code>\Gin@PS@restored</code> .. 197	<code>\makeindex</code> 79
D	<code>\Gin@req@height</code> ... 401	<code>\mbox</code> 175
<code>\DeclareGraphicsRule</code> 104, 245	<code>\Gin@req@sizes</code> 400	
<code>\define@key</code> 105–110, 246, 248, 250, 252, 261, 263, 403	<code>\Gin@req@width</code> 401	N
	<code>\Gin@setfile</code> 66, 272, <u>273</u>	<code>\newcolumntype</code> 167
E	<code>\Gin@shownamefalse</code> . 59	<code>\newcount</code> 89
<code>\endpostscript</code> 367, 370	<code>\Gin@curx</code> 393, 398	<code>\nofiles</code> .. 78, 207, 211
<code>\endpsmatrix</code> 153, 157, 370	<code>\Gin@cury</code> 394, 396	
<code>\endpspicture</code> 367	<code>\Gin@viewport@code</code> . 395	O
<code>\endtabularx</code> 169	<code>\Gin@vllx</code> 124	<code>\OptionNotUsed</code> 6
environments:	<code>\Gin@vllly</code> 124	<code>\output</code> 206, 212
<code>postscript</code> 2, <u>117</u> , <u>342</u>	<code>\Gin@vurx</code> 124	<code>\overfullrule</code> 80
<code>psmatrix</code> .. <u>135</u> , <u>342</u>	<code>\Gin@vury</code> 124	
<code>pspicture</code> 2, <u>134</u> , <u>342</u>	<code>\Gin@viewports@code</code> . 395	
	<code>\Gin@vllx</code> 124	
	<code>\Gin@vllly</code> 124	
	<code>\Gin@vurx</code> 124	
	<code>\Gin@vury</code> 124	
	<code>\Gin@vury</code> 124	
	<code>\Gin@vury</code> 124	
	<code>\Ginclude@graphics</code> 53, <u>177</u> , 388	
	<code>\Gscale@@box</code> <u>174</u>	

P	
<code>\PassOptionsToPackage</code>	12, 15, 19
<code>\PDFcontainer</code>	39, 40, 283, 289, 304, 306, 312, 314, 325, 327, 414
<code>\pdflastximage</code>	378
<code>\pdflastximagepages</code>	307
<code>\pdfliteral</code>	187
<code>\pdfoutput</code>	89, 90
<code>\pdfrefximage</code>	390
<code>\pdfTeXtext</code>	65, 68
<code>\pdftexversion</code>	88
<code>\pdfximage</code>	306
<code>\pfx@includgraphics</code>	373
<code>\pfx@includgraphicx</code>	172, 374
<code>\postscript</code>	366, 369
<code>postscript</code> (environment)	2, 117, 342
<code>\ppf@getpicture</code>	227, 362, 374, 417
<code>\ppf@container@max</code>	306, 310, 312, 322, 409
<code>\ppf@draft</code>	3, 13, 14, 413
<code>\ppf@endpsmatrix</code>	153, 157
<code>\ppf@filename</code>	269, 274, 298, 299, 389
<code>\ppf@getpicture</code>	383, 407, 420
<code>\ppf@Gin@extensions</code>	54
<code>\ppf@Gin@ii</code>	55, 293
<code>\ppf@Gin@keys</code>	271, 358, 360, 416, 420
<code>\ppf@Gin@setfile</code>	272, 273
<code>\ppf@Gin@graphics</code>	53, 71, 181, 183, 186
<code>\ppf@is@pdfTeX@graphic</code>	62, 179, 280, 437
<code>\ppf@isnum</code>	339, 382
<code>\ppf@namefont</code>	265, 266, 270, 297
<code>\ppf@nofiles</code>	207, 211
<code>\ppf@other@extensions</code>	42, 92, 102, 179, 243, 280
<code>\ppf@output</code>	205, 206, 212
<code>\ppf@pr@psmatrix</code>	151, 156
<code>\ppf@PreviewBbAdjust</code>	83, 85, 119
<code>\ppf@psmatrix</code>	150, 154
<code>\ppf@RestoreBbAdjust</code>	84, 129
<code>\ppf@set@mode</code>	138, 142, 144, 147, 156, 157, 342, 346, 348, 351
<code>\ppf@shipout</code>	208, 210
<code>\ppf@temp</code>	191, 193, 427, 430
<code>\ppf@tempb</code>	278, 279, 283, 289, 294
<code>\ppf@test@mmode</code>	139, 154, 343
<code>\ppf@TeX@mode</code>	2, 7, 8, 21, 23, 25, 29, 32, 34, 75, 77, 189, 435
<code>\pr@cleanup</code>	216, 228
<code>\pr@endbox</code>	224
<code>\pr@outerfalse</code>	71, 219
<code>\pr@startbox</code>	213
<code>\PreviewBbAdjust</code>	85, 112, 119, 123
<code>\PreviewEnvironment</code>	130, 134, 137
<code>\PreviewMacro</code>	159, 172, 186, 232
<code>\psmatrix</code>	150, 152, 154, 155, 369
<code>psmatrix</code> (environment)	135, 342
<code>pspicture</code> (environment)	2, 134, 342
<code>pst-pdf-defs</code> (environment)	3, 45, 423
<code>\pst@opicture</code>	366
<code>\pst@object</code>	165, 238
<code>\PSTricksOff</code>	199
R	
<code>\raisebox</code>	298
<code>\refstepcounter</code>	288, 419
<code>\rule</code>	180
S	
<code>\savepicture</code>	3, 44, 377
<code>\setkeys</code>	122, 282
<code>\shipout</code>	208, 210
<code>\string</code>	152, 155
<code>\strutbox</code>	298
T	
<code>\tabularx</code>	168
U	
<code>\usepicture</code>	3, 43, 379
V	
<code>\voidbox</code>	226
X	
<code>\XKV@err</code>	100