

Das Paket `pst-pdf`*

Rolf Niepraschk[†] Hubert Gäßlein

2020/10/10

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` v1.2f vom 2020/10/10.

[†]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 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

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
25     ⇒ pdf $\text{T}_{\text{E}}\text{X}$  or Lua $\text{T}_{\text{E}}\text{X}$  are running in PDF mode
26     \def\ppf@TeX@mode{1}%
27     \RequirePackage{luatex85}%
28 \else
29 \ifvtex
30     ⇒ V $\text{T}_{\text{E}}\text{X}$ 
31     \def\ppf@TeX@mode{9}%
32 \else
33 \ifxetex
34     ⇒ Xe $\text{T}_{\text{E}}\text{X}$ 
35     \def\ppf@TeX@mode{9}%
36 \else
37     ⇒ DVI mode
38     \def\ppf@TeX@mode{0}%
39 \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 Dateierweiterungen im pdf-Modus, Parameter #3 enthält die gültigen Dateierweiterungen 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,}%
66 \edef\pdfTeXtext{\detokenize\expandafter{\pdfTeXtext}}%

```

Statt Einladen einer identifizierten Grafik nur Test der Grafikendung.

```

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}%

```

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

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

Testaufruf. Dabei Ausgabe vollständig verhindern.

```

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

76 \ifcase\ppf@TeX@mode\relax

```

3.3 Extraction mode (dvi output)

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 \TeX -Compiler mit DVI-Ausgabe sowie die Paketoption “active” erzwingen diesen Modus.

```

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 \if@ppf@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}%

```

Es werden auch die im pdfLaTeX-Modus erlaubten Endungen von Grafikdateien benötigt (Argument aus ‘pdftex.def’ – 2020/10/05 – kopiert).

```

90 \def\ppf@other@extensions{%
91   .pdf,.png,.jpg,.mps,.jpeg,.jbig2,.jb2,%
92   .PDF,.PNG,.JPG,.JPEG,.JBIG2,.JB2%
93 }

```

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.

```

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

```

In diesem Modus sollten undefinierte keys keinen Fehler bewirken.

```

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

```

In diesem Modus keine Funktion.

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

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

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

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

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

pspicture Originaldefinition `preview` bekannt machen.

```
133   \PreviewEnvironment{pspicture}%
```

psmatrix Originaldefinition `preview` bekannt machen.

```
134   \@ifundefined{psmatrix}{}%
135   {%
136     \PreviewEnvironment{psmatrix}%
137     \newcommand*\ppf@set@mode{}%
138     \newcommand*\ppf@test@mmode{%
139       \ifmmode
140         \ifinner
141           \let\ppf@set@mode=$%
142         \else
143           \def\ppf@set@mode{$$}%
144         \fi
145       \else
146         \let\ppf@set@mode=\@empty
147       \fi
```

```

148     }%
149     \let\ppf@psmatrix=\psmatrix
150     \expandafter\let\expandafter\ppf@pr@psmatrix%
151     \expandafter=\csname pr@\string\psmatrix\endcsname
152     \let\ppf@endpsmatrix=\endpsmatrix
153     \def\psmatrix{\ppf@test@mode\ppf@psmatrix}%
154     \expandafter\def\csname pr@\string\psmatrix\endcsname{%
155     \ppf@set@mode\ppf@pr@psmatrix}%
156     \def\endpsmatrix{\ppf@endpsmatrix\ppf@set@mode}%
157     }%

```

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>>
(m = notwendig, * = optional, o = optional)

```

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

```

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

```

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

```

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

```

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

```

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

```

`\Gscale@@box` Skalieren verhindern.

```

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

```

`\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.

```

176     \def\Ginclude@graphics#1{%
177     \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.

```

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

```


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

```

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

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

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

```

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

Verhindert pdfTEXs Warnung Non-PDF special ignored!.
189 \ifppf@PST@used
190   \let\ppf@temp\AtBeginDvi\let\AtBeginDvi@gobble
191   \def\c@lor@to@ps#1 #2\@{}%
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}{-}{%

```

Für PSTricks < 2.0 nötig.

```

198   \PSTricksOff%
199   \def\c@lor@to@ps#1 #2\@{}%
200 }%
201 }%

```

PostScript-Ausgabe jetzt verhindern und später noch einmal.

```

202 \the\@temptokena%
203 \expandafter\AtBeginDocument\expandafter
204   {\the\@temptokena\@temptokena{}}%
205 \@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.

```

206   \newtoks\ppf@output

```

```

207 \ppf@output\expandafter{\the\output}%
208 \let\ppf@nofiles=\nofiles \let\nofiles=\relax
209 \let\ppf@shipout=\shipout
210 \RequirePackage[active]{preview}[2005/01/29]%
211 \let\shipout=\ppf@shipout \let\ppf@shipout=\relax
212 \let\nofiles=\ppf@nofiles \let\ppf@nofiles=\relax
213 \output\expandafter{\the\ppf@output} \ppf@output{}}%
\pr@startbox, \pr@endbox: Gegenüber Originaldefinition vereinfacht.
214 \long\def\pr@startbox#1#2{%
215   \ifpr@outer
216     \toks@{#2}%
217     \edef\pr@cleanup{\the\toks@}%
218     \setbox\@tempboxa\vbox\bgroup
219     \everydisplay{}%
220     \pr@outerfalse%
221     \expandafter\@firstofone
222   \else
223     \expandafter\@gobble
224   \fi{#1}}%
225 \def\pr@endbox{%
226   \egroup
227   \setbox\@tempboxa\box\voidb@x
228   \ppf@getpicture
229   \pr@cleanup}%

```

(Siehe auch identische Definition im DVI-Modus.)

```

230 \AtBeginDocument{%
231   \@ifundefined{pst@object}{}%
232   {%
233     \PreviewMacro[{}*[]%
234     ?\bgroup{##1}{##1}}{}%
235     ?\bgroup{##1}{##1}}{}%
236     ?({##1}){({##1})}}{}%
237     ?({##1})-({##1})}}{}%
238     ?({##1})-({##1})}}{}%
239     }]\pst@object}}%
240   }%
241 }%

```

Es werden auch die im DVI-Modus erlaubten Endungen von Grafikdateien benötigt (Argument aus ‘dvips.def’ – 2017/06/20 – kopiert).

```

242 \def\ppf@other@extensions{.eps,.ps,.eps.gz,.ps.gz,.eps.Z,.mps}%
Dummy-Definition für die im DVI-Modus gültigen Dateitypen.
243 \DeclareGraphicsRule*{eps}{*}{}%
244 \define@key{Gin}{innerframe}[true]{%
245   \lowercase{\Gin@boolkey{#1}}{innerframe}}%
246 \define@key{Gin}{frame}[true]{%
247   \lowercase{\Gin@boolkey{#1}}{frame}}%
248 \define@key{Gin}{ignore}[true]{%
249   \lowercase{\Gin@boolkey{#1}}{ignore}}%
250 \define@key{Gin}{frame@}{}%

```

(Nur intern zu benutzen!)

```

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

```

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

```

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

```

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

```

274 \def\Gin@ii[#1]#2{%
275 \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.

```

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

```

Leerer Dateiname beim Aufruf von `\usepicture` aus.

```

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

```

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

```

279 {%
280 \setkeys{Gin}{#1}%
281 \ifx\ppf@tempb\PDFcontainer
282 \@temptokena{page=\Gin@page}%
283 \fi
284 }%
285 {%
286 \refstepcounter{pspicture}%
287 \@temptokena{page=\the\c@pspicture}\def\ppf@tempb{\PDFcontainer}%
288 }%
289 \fi
290 \ifGin@ignore\else

```

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

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

302 \IfFileExists{\PDFcontainer}%
303 {%
```

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

```
304     \pdfximage{\PDFcontainer}%
305     \edef\ppf@container@max{\the\pdfximagepages}%

306     \AtEndDocument{%
307         \ifnum\c@pspicture>\z@
```

Warnung ist nur sinnvoll, wenn überhaupt Grafiken benötigt wurden.

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

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

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

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

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

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

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

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

```

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

```

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

```

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

```

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

```

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

```

415 \newcommand*\ppf@@getpicture{%
416 \ifpr@outer
417 \refstepcounter{pspicture}%

```

```

418     \expandafter\ppf@getpicture\expandafter{\ppf@Gin@keys}%
419     {\the\c@pspicture}%
420     \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.

```

421 \renewenvironment*{pst-pdf-defs}%
422 {%
423   \endgroup
424 %   ??? \@currentvline
425   \chardef\ppf@temp=\catcode'\&%
426   \@makeother\&%
427 }{%
428   \catcode'\&=\ppf@temp
429   \begingroup
430   \def\@currentvir{pst-pdf-defs}%
431 }
432 \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.

```

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

437 \InputIfFileExists{pst-pdf.cfg}{%
438   \PackageInfo{pst-pdf}{Local config file pst-pdf.cfg used}}{%
439 \>/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)		4
v1.0d	<code>\usepicture</code> : New macro <code>\usepspicture</code> . Useful for putting a PSTricks graphic in a box or something else. (RN)	14
General: Redefinition of <code>\includegraphics</code> in modes 0 und 1. Now using of eps graphics directly in pdf \TeX is (RN)		15

	environment (correct graphic inclusion). (RN)	8	v1.1t	General: Remove the line “\let\output@gobble” because of bad side effects. (RN)	9
v1.1i	\Ginclude@graphics: Correction of the inside check. (RN/HjG)	8		postscript: Using environ the environment postscript is now simple and more robust. (RN)	13
	General: \ifpr@outer must be predefined. (HjG/RN)	5	v1.1u	General: \pdfoutput must be set when loading “pdftex.def” in DVI mode. (RN)	6
	Package option “final” also for “graphicx”. (RN)	4	v1.1v	\Gin@ii: Key settings only for pdf graphics. (RN)	11
v1.1k	General: New environment pst-pdf-defs: Support for PSTricks environment “psmatrix” inside user definitions. (RN,HjG)	1		General: Local redefinition of \pdfoutput to be a counter. (RN)	6
v1.1l	General: Support for the package “psfragx”. (RN)	8	v1.2a	General: Engine tests changed (RN)	4
v1.1m	General: Merge english and german version of the documentation. (RN)	1	v1.2b	General: Loading packages ‘if*’ at wrong place (RN)	4
v1.1n	General: \nofiles added (suggestion of Torsten Bronger).	6	v1.2c	General: “postscript” environment no more allowed before \begin{document} (changed example file).	1
v1.1o	\Gscale@@box: Disable scaling. (RN)	8	v1.2d	General: \c@lor@to@ps must not be undefined before loading ‘PSTricks’	9
v1.1p	General: \let\output@gobble before loading of “preview” added. (RN)	9		Version parameter for ‘graphicx’ and rename \GPT@page to \Gin@page at several places . . .	5
	\nofiles makes \makeindex and \makeglossary to \relax. \@empty is better because of later \renewcommand’s.	6	v1.2e	\ppf@is@pdfTeX@graphic: Parameter #2 is detokenized when expanded to \pdfTeXext . . .	5
v1.1q	General: Problem with “tabularx” and “threeparttable” solved. (RN)	8		General: gobble optional argument for \makeindex, \makeglossary and \printindex	6
v1.1r	General: Fixed values for \PreviewBbAdjust because \paperwidth is not always defined (suggested by Will Robertson).	7	v1.2f	General: No more loading of file ‘dvisps.def’ (RN)	10
v1.1s	General: Dummy definition of the page key in DVI mode.	7		No more loading of file ‘pdftex.def’ (RN)	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.

Symbols	<code>pspicture</code> 2, <u>133</u> , <u>340</u>	I
<code>\&</code> 425, 426, 428	<code>pst-pdf-defs</code> ...	<code>\if@ppf@pdf@graphic</code>
<code>\@currenvir</code> 50, 355, 430 <u>3</u> , <u>45</u> , <u>421</u> 56, 74
<code>\@currencline</code> .. 47, 424	<code>\everydisplay</code> 219	<code>\if@ppf@PST@used</code> ..
<code>\@ehc</code> 382	F 4, 85, 189
<code>\@expandtwoargs</code> ... 69	<code>\filename@base</code> 330, 331	<code>\if@ppf@tightpage</code> .
<code>\@firstofone</code> 221	<code>\filename@parse</code> ... 323 5, 110, 119
<code>\@ifpackageloaded</code> 95, 98	<code>\frame</code> 251	<code>\ifGin@frame</code> ... 57, 255
<code>\@latex@error</code> 382	G	<code>\ifGin@ignore</code> .. 60, 290
<code>\@makeoother</code> 426	<code>\Gin@bboxtrue</code> 393	<code>\ifGin@innerframe</code> .
<code>\@ppf@PST@usedfalse</code> 10	<code>\Gin@boolkey</code> 245, 247, 249, 260 58, 253
<code>\@ppf@PST@usedtrue</code> 4, 9	<code>\Gin@defaultbp</code> 391, 392	<code>\ifGin@showname</code> 59, 294
<code>\@ppf@pdf@graphicfalse</code>	<code>\Gin@extensions</code> ...	<code>\ifin@</code> 70
..... 63	.. 54, 71, 178, 278	<code>\ifinner</code> 140, 343
<code>\@ppf@pdf@graphictrue</code>	<code>\Gin@framefalse</code> ... 57	<code>\ifmmode</code> 139, 342
..... 70	<code>\Gin@ignorefalse</code> .. 60	<code>\ifpdf</code> 24
<code>\@ppf@tightpagefalse</code> 16	<code>\Gin@ii</code> 55, <u>274</u>	<code>\ifpr@outer</code> 61, 177, 215, 416
<code>\@ppf@tightpagetrue</code>	<code>\Gin@innerframefalse</code> 58	<code>\ifvtex</code> 28
..... 5, 17	<code>\Gin@llx</code> 390, 397	<code>\ifxetex</code> 31
A	<code>\Gin@lly</code> 390, 395	<code>\in@</code> 69
<code>\AtBeginDvi</code> ... 190, 192	<code>\Gin@nat@height</code> ... 389, 392, 394, 395	<code>\includegraphics</code> ..
C	<code>\Gin@nat@width</code> ... 389, 391, 396, 397	.. <u>3</u> , 371, 402, 411
<code>\color@to@ps</code> .. 191, 199	<code>\Gin@page</code> 273, 282	<code>\includegraphicsx</code> ... 3
<code>\c@pspicture</code> 287, 307, 308, 311, 322, 419	<code>\Gin@PS@file@header</code> 195	J
<code>\catcode</code> 425, 428	<code>\Gin@PS@literal@header</code> 195	<code>\jobname</code> .. 40, 329, 330
<code>\Collect@Body</code> 361	<code>\Gin@PS@raw</code> 196	K
<code>\CurrentOption</code> 12, 15, 19	<code>\Gin@PS@restored</code> .. 196	<code>\KV@errx</code> 96
D	<code>\Gin@req@height</code> ... 399	L
<code>\DeclareGraphicsRule</code>	<code>\Gin@req@sizes</code> ... 398	<code>\leavevmode</code> 400
..... 103, 243	<code>\Gin@req@width</code> ... 399	<code>\long</code> 214
<code>\define@key</code> 104–109, 244, 246, 248, 250, 259, 261, 401	<code>\Gin@setfile</code> 67, 270, <u>271</u>	M
<code>\detokenize</code> 66, 69	<code>\Gin@shownamefalse</code> . 59	<code>\makeglossary</code> ... 80, 82
E	<code>\Gin@urx</code> 391, 396	<code>\makeindex</code> 80, 81
<code>\endpostscript</code> 365, 368	<code>\Gin@ury</code> 392, 394	<code>\mbox</code> 174
<code>\endpsmatrix</code> 152, 156, 368	<code>\Gin@viewport@code</code> . 393	N
<code>\endpspicture</code> 365	<code>\Gin@vllx</code> 123	<code>\newcolumnmtype</code> 166
<code>\endtabularx</code> 168	<code>\Gin@vllly</code> 123	<code>\nofiles</code> .. 79, 208, 212
environments:	<code>\Gin@vurx</code> 123	O
<code>postscript</code> 2, <u>116</u> , <u>340</u>	<code>\Gin@vury</code> 123	<code>\OptionNotUsed</code> 6
<code>psmatrix</code> .. <u>134</u> , <u>340</u>	<code>\Gin@viewport@code</code> . 393	<code>\output</code> 207, 213
	<code>\Gin@vllx</code> 123	<code>\overfullrule</code> 84
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code> 123	
	<code>\Gin@vury</code> 123	
	<code>\Gin@viewport@code</code> . 393	
	<code>\Gin@vllx</code> 123	
	<code>\Gin@vllly</code> 123	
	<code>\Gin@vurx</code>	

P

\PassOptionsToPackage 12, 15, 19

\PDFcontainer 39, 40, 281, 287, 302, 304, 310, 312, 323, 325, 412

\pdflastximage 376

\pdflastximagepages 305

\pdfliteral 186

\pdfrefximage 388

\pdfTeXtext 65, 66, 69

\pdfximage 304

\pfx@includegraphics 371

\pfx@includegraphicx 171, 372

\postscript 364, 367

postscript (environment) 2, 116, 340

\ppf@getpicture 228, 360, 372, 415

\ppf@container@max 304, 308, 310, 320, 407

\ppf@draft 3, 13, 14, 411

\ppf@endpsmatrix 152, 156

\ppf@filename 267, 272, 296, 297, 387

\ppf@getpicture 381, 405, 418

\ppf@Gin@extensions 54

\ppf@Gin@ii 55, 291

\ppf@Gin@keys 269, 356, 358, 414, 418

\ppf@Gin@setfile 270, 271

\ppf@Gin@include@graphics 53, 72, 180, 182, 185

\ppf@is@pdfTeX@graphic 62, 178, 278, 435

\ppf@isnum 337, 380

\ppf@namefont 263, 264, 268, 295

\ppf@nofiles 208, 212

\ppf@other@extensions 42, 90, 101, 178, 242, 278

\ppf@output 206, 207, 213

\ppf@pr@psmatrix 150, 155

\ppf@PreviewBbAdjust 87, 89, 118

\ppf@psmatrix 149, 153

\ppf@RestoreBbAdjust 88, 128

\ppf@set@mode 137, 141, 143, 146, 155, 156, 340, 344, 346, 349

\ppf@shipout 209, 211

\ppf@temp 190, 192, 425, 428

\ppf@tempb 276, 277, 281, 287, 292

\ppf@test@mmode 138, 153, 341

\ppf@TeX@mode 2, 7, 8, 21, 23, 25, 29, 32, 34, 76, 78, 188, 433

\pr@cleanup 217, 229

\pr@endbox 225

\pr@outerfalse 72, 220

\pr@startbox 214

\PreviewBbAdjust 89, 111, 118, 122

\PreviewEnvironment 129, 133, 136

\PreviewMacro 158, 171, 185, 233

\printindex 80, 83

\psmatrix 149, 151, 153, 154, 367

psmatrix (environment) 134, 340

pspicture (environment) 2, 133, 340

pst-pdf-defs (environment) 3, 45, 421

\pst@@@picture 364

\pst@object 164, 239

\PSTricksOff 198

R

\raisebox 296

\refstepcounter 286, 417

\rule 179

S

\savepicture 3, 44, 375

\setkeys 121, 280

\shipout 209, 211

\string 151, 154

\strutbox 296

T

\tabularx 167

U

\usepicture 3, 43, 377

V

\voidb@x 227

X

\XKV@err 99