

Das Paket `pst-pdf`*

Rolf Niepraschk[†] Hubert Gäßlein

2006/07/14

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 Extraktionsmodus¹ 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` v1.1n vom 2006/07/14.

[†]Rolf.Niepraschk@ptb.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 Undefinieren des Makros `\PDFcontainer` geändert werden.

2.3 Nutzeranweisungen

pspicture `\begin{pspicture}[(keys)] (<x0,x1>)(<y0,y1>) ... \end{pspicture}`
 Die `pspicture`-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.

postscript `\begin{postscript}[(keys)] ... \end{postscript}`

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 pdfL^AT_EX-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[⟨keys⟩]{⟨filename⟩}</code> Wie in <code>graphics/graphicx</code> definiert zu verwenden. Zusätzlich ist es nun möglich, auch im pdfL ^A T _E X-Modus EPS-Dateien als Argument anzugeben und ihren Inhalt darzustellen. Er wird dazu ebenfalls der Containerdatei entnommen.
<code>\includegraphicsx</code>	<code>\includegraphicsx[⟨keys⟩](⟨pfxadd⟩)<⟨ovpfgd⟩>[⟨ovpbgd⟩]{⟨filename⟩}</code> Wie im Paket <code>psfragx</code> definiert zu verwenden.
<code>\savepicture</code>	<code>\savepicture{⟨name⟩}</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[⟨keys⟩]{⟨name⟩}</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 pdfL^AT_EX-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}{\def\ppf@TeX@mode{0}}
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 Compilertests

Es wird getestet, welcher T_EX compiler in welchem Modus läuft (siehe ‘graphics.cfg’ von t_EX/L_{IV}e). 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 \ifnum\ppf@TeX@mode=-1\relax
23   \begingroup
Default (TEX with a dvi-to-ps converter)
24   \chardef\x=0 %
Check pdfTEX
25   \@ifundefined{pdfoutput}{}{%
26     \ifcase\pdfoutput\else
27       \chardef\x=1 %
28     \fi
29   }%
Check VTEX
30   \@ifundefined{OpMode}{}{\chardef\x=2 }%
31   \expandafter\endgroup
32   \ifcase\x
⇒ DVI mode
33   \def\ppf@TeX@mode{0}%
34   \or
```

⇒ pdfT_EX is running in PDF mode

```

35 \def\ppf@TeX@mode{1}%
36 \else
    ⇒ VTEX is running
37 \def\ppf@TeX@mode{9}%
38 \fi
39 \fi

40 \newcommand*\PDFcontainer{}
41 \edef\PDFcontainer{\jobname-pics.pdf}
42 \newcounter{pspicture}
43 \newcommand*\ppf@other@extensions[1]{}
44 \newcommand*\usepicture[2][{}]{%
45 \newcommand*\savepicture[1]{}

```

pst-pdf-defs

```

46 \newenvironment*{pst-pdf-defs}%
47 {%
48 \endgroup
49 % ??? \@currentvline
50 }{%
51 \begingroup
52 \def\@currentvir{pst-pdf-defs}%
53 }

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

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

```

63 \newif\ifpr@outer

```

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

```

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

```

Statt Einladen einer identifizierten Grafik nur Test der Grafikendung.

```

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

```

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

```

72 \edef\Gin@extensions{#2,#3}%
Testaufruf. Dabei Ausgabe vollständig verhindern.
73 \pr@outerfalse\ppf@Gininclude@graphics{#1}%
74 \endgroup
75 \if@ppf@pdftex@graphic#4\else#5\fi
76 }

77 \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.

```

78 \PackageInfo{pst-pdf}{%
79   MODE: \ppf@TeX@mode\space (dvi -- extraction mode)}
80 \nofiles
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 \def\pdftexversion{121}\input{pdftex.def}%
89 \edef\x{\endgroup\def\noexpand\ppf@other@extensions{\Gin@extensions}
90 }%
91 \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.

```

92 \AtBeginDocument{%
93   \@for\@tempa:=\ppf@other@extensions\do{%
94     \expandafter\let\csname Gin@rule@\@tempa\endcsname\relax}%
95   \DeclareGraphicsRule{*}{eps}{*}{}%

```

In diesem Modus keine Funktion.

```

96 \define@key{Gin}{innerframe}[true]{}%
97 \define@key{Gin}{frame}[true]{}%
98 \define@key{Gin}{ignore}[true]{}%
99 \define@key{Gin}{showname}[true]{}%
100 \define@key{Gin}{namefont}{}%

```

```

101 \if@ppf@tightpage\else
102 \def\PreviewBbAdjust{%
103   -.5\paperwidth -.5\paperheight .5\paperwidth .5\paperheight}%
104 \AtEndDocument{%
105   \PackageWarningNoLine{pst-pdf}{Picture container needs cropping.}}%
106 \fi

```

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

```

107 \newenvironment{postscript}[1][]%
108 {%
109   \global\let\ppf@PreviewBbAdjust\PreviewBbAdjust
110   \if@ppf@tightpage
111     \begingroup
112       \setkeys{Gin}{#1}%
113       \xdef\PreviewBbAdjust{%
114         -\Gin@vllx bp -\Gin@vllx bp \Gin@vurx bp \Gin@vury bp}%
115     \endgroup
116   \fi
117   \ignorespaces
118 }%
119 {\aftergroup\ppf@RestoreBbAdjust}%

120 \PreviewEnvironment{postscript}%
121 \AtBeginDocument{%
122   \@ifundefined{PSTricksLoaded}{}%
123   {%

```

pspicture Originaldefinition preview bekannt machen.

```

124 \PreviewEnvironment{pspicture}%

```

psmatrix Originaldefinition preview bekannt machen.

```

125 \@ifundefined{psmatrix}{}%
126 {%
127   \PreviewEnvironment{psmatrix}%
128   \newcommand*\ppf@set@mode{}%
129   \newcommand*\ppf@test@mmode{%
130     \ifmmode
131       \ifinner
132         \let\ppf@set@mode=$%
133       \else
134         \def\ppf@set@mode{$$}%
135       \fi
136     \else
137       \let\ppf@set@mode=\@empty
138     \fi
139   }%
140   \let\ppf@psmatrix=\psmatrix
141   \expandafter\let\expandafter\ppf@pr@psmatrix%
142     \expandafter=\csname pr@\string\psmatrix\endcsname
143   \let\ppf@endpsmatrix=\endpsmatrix
144   \def\psmatrix{\ppf@test@mmode\ppf@psmatrix}
145   \expandafter\def\csname pr@\string\psmatrix\endcsname{%
146     \ppf@set@mode\ppf@pr@psmatrix}%

```

```

147     \def\endpsmatrix{\ppf@endpsmatrix\ppf@set@mode}%
148 }%

```

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.

```

149     \PreviewMacro[{}*{}]%
150     ?\bgroup{#{#1}{#{#1}}}%
151     ?\bgroup{#{#1}{#{#1}}}%
152     ?({#{(#1)}{({#{#1})}})%
153     ?({#{(#1)}{({#{#1})}})%
154     ?({#{(#1)}{({#{#1})}})%
155     }\pst@object}%

```

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

```

156     \@ifundefined{tabularx}{}%
157     \def\tabularx#1#2{\tabular{#2}}%
158     \newcolumntype{X}{c}%
159     \let\endtabularx=\endtabular}%

```

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

```

160     \@ifundefined{pfx@includegraphicx}{}%
161     \PreviewMacro[{}{}]{\pfx@includegraphicx}%
162 }%

```

`\Gininclude@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.

```

163     \def\Gininclude@graphics#1{%
164     \ifpr@outer

```

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

```

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

```

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

```

166     {\rule{10pt}{10pt}}%
167     {\ppf@Gininclude@graphics{#1}}%
168     \else

```

Innerhalb von PS-Umgebungen (`pspicture` usw.) muss sich `\includegraphics` wie die Originaldefinition verhalten (nur die DVIPS-Graphik-Typen sind gültig).

```

169     \ppf@Gininclude@graphics{#1}%
170     \fi
171 }%

172 \PreviewMacro[{}]{\ppf@Gininclude@graphics}%
173 \let\pdfliteral\@gobble%
174 \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.

```

175 \PackageInfo{pst-pdf}{MODE: \ppf@TeX@mode\space (pdfTeX mode)}%
176 \@temptokena{%
177   \let\Gin@PS@file@header\@gobble\let\Gin@PS@literal@header\@gobble
178   \let\Gin@PS@raw\@gobble\let\Gin@PS@restored\@gobble
179   \@ifundefined{PSTricksLoaded}{}{}}%
```

Für PSTricks < 2.0 nötig.

```

180   \PSTricksOff
181   \@ifundefined{c@lor@to@ps}{\def\c@lor@to@ps#1 #2\@{}}{}}%
```

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

```

182 \if@ppf@PST@used
183   \let\ppf@temp\AtBeginDvi\let\AtBeginDvi\@gobble
184   \RequirePackage{pstricks}\let\AtBeginDvi\ppf@temp
185 \fi
```

PostScript-Ausgabe jetzt verhindern und später noch einmal.

```

186 \the\@temptokena %%% ???
187 \expandafter\AtBeginDocument\expandafter
188   {\the\@temptokena\@temptokena{}}%
189 \@ifundefined{PSTricksLoaded}{}
190 {}%
```

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.

```

191 \newtoks\ppf@temptoken
192 \ppf@temptoken\expandafter{\the\output}%
193 \let\ppf@nofiles\nofiles \let\nofiles\relax
194 \RequirePackage[active]{preview}[2005/01/29]%
195 \let\shipout=\pr@shipout \let\nofiles\ppf@nofiles
196 \output\expandafter{\the\ppf@temptoken}%
197 \ppf@temptoken{}}%
```

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

```

198 \long\def\pr@startbox#1#2{%
199   \ifpr@outer
200     \toks@{#2}%
201     \edef\pr@cleanup{\the\toks@}%
202     \setbox\@tempboxa\vbox\bgroup
203     \everydisplay{}%
204     \pr@outerfalse%
205     \expandafter\@firstofone
206   \else
207     \expandafter\@gobble
208   \fi{#1}}%
```

```

209 \def\pr@endbox{%
210 \egroup
211 \setbox\@tempboxa\box\voidb@x
212 \ppf@getpicture
213 \pr@cleanup}%

```

(Siehe auch identische Definition im DVI-Modus.)

```

214 \AtBeginDocument{%
215 \ifundefined{pst@object}{}%
216 {%
217 \PreviewMacro[{}*[]%
218 ?\bgroup{#{#1}{#{1}}}{}%
219 ?\bgroup{#{#1}{#{1}}}{}%
220 ?({#{#1}){({#{1})}}{}%
221 ?({#{#1}){({#{1})}}{}%
222 ?({#{#1}){({#{1})}}{}%
223 }]\pst@object}}%
224 }%
225 }%

```

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

```

226 \begingroup
227 \input{dvips.def}%
228 \edef\x{\endgroup\def\noexpand\ppf@other@extensions{\Gin@extensions}}%
229 \x

```

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

```

230 \DeclareGraphicsRule{*}{eps}{*}{}%
231 \define@key{Gin}{innerframe}[true]{%
232 \lowercase{\Gin@boolkey{#1}}{innerframe}}%
233 \define@key{Gin}{frame}[true]{%
234 \lowercase{\Gin@boolkey{#1}}{frame}}%
235 \define@key{Gin}{ignore}[true]{%
236 \lowercase{\Gin@boolkey{#1}}{ignore}}%
237 \define@key{Gin}{frame@@}{%

```

(Nur intern zu benutzen!)

```

238 \edef\@tempa{\toks@{\noexpand\frame{\the\toks@}}}%
239 \ifcase#1\relax
240 \ifGin@innerframe\else\let\@tempa\relax\fi
241 \or
242 \ifGin@frame\else\let\@tempa\relax\fi
243 \fi
244 \@tempa
245 }%
246 \define@key{Gin}{showname}[true]{%
247 \lowercase{\Gin@boolkey{#1}}{showname}}%
248 \define@key{Gin}{namefont}{%
249 \begingroup
250 \temptokena\expandafter{\ppf@namefont#1}%
251 \edef\x{\endgroup\def\noexpand\ppf@namefont{\the\@temptokena}}%
252 \x
253 }%
254 \newcommand*\ppf@filename{%

```

```

255 \newcommand*\ppf@namefont{\tiny\ttfamily}%
256 \newcommand*\ppf@Gin@keys{%
257 \let\ppf@Gin@setfile\Gin@setfile

```

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

```

258 \def\Gin@setfile#1#2#3{\ppf@Gin@setfile{#1}{#2}{#3}%
259 \xdef\ppf@filename{%
260 #3\ifx\GPT@page\empty\else(\GPT@page)\fi}}%

```

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

```

261 \def\Gin@ii[#1]#2{%
262 \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.

```

263 \setkeys{Gin}{#1}%
264 \@temptokena{#1}\def\@tempb{#2}%

```

Leerer Dateiname beim Aufruf von \usepicture aus.

```

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

```

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

```

267 {%
268 \ifx\@tempb\PDFcontainer
269 \@temptokena{page=\GPT@page}%
270 \fi
271 }%
272 {%
273 \refstepcounter{pspicture}%
274 \@temptokena{page=\the\c@pspicture}\def\@tempb{\PDFcontainer}%
275 }%
276 \fi
277 \ifGin@ignore\else

```

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

```

278 \edef\@tempa{\noexpand\ppf@Gin@ii[frame@@=0,\the\@temptokena,
279 frame@@=1]{\@tempb}}%
280 \@tempa
281 \ifGin@showname
282 \ppf@namefont
283 \raisebox{-\ht\strutbox}[0pt][0pt]{\llap{\ppf@filename}}%
284 \gdef\ppf@filename{}%
285 \fi
286 \fi
287 \endgroup
288 }%

```

```

289 \IfFileExists{\PDFcontainer}%
290 {%

```

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

```
291 \pdfximage{\PDFcontainer}%
292 \edef\ppf@container@max{\the\pdflastximagepages}%

293 \AtEndDocument{%
294 \ifnum\c@pspicture>\z@

Warnung ist nur sinnvoll, wenn überhaupt Grafiken benötigt wurden.

295 \ifnum\c@pspicture=\ppf@container@max\else
296 \PackageWarningNoLine{pst-pdf}{%
297 '\PDFcontainer' contains \ppf@container@max\space pages
298 \MessageBreak but \the\c@pspicture\space pages are requested:
299 \MessageBreak File '\PDFcontainer' is no more valid!
300 \MessageBreak Recreate it
301 }%
302 \fi
303 \fi
304 }%
305 }%
306 {%
307 \def\ppf@container@max{0}%
308 \AtEndDocument{%
309 \ifnum\c@pspicture>\z@
310 \filename@parse{\PDFcontainer}%
311 \PackageWarningNoLine{pst-pdf}{%
312 File '\PDFcontainer' not found.\MessageBreak
313 Use the following commands to create it:\MessageBreak
314 -----
315 \MessageBreak
316 latex \jobname.tex\MessageBreak
317 dvips -o \filename@base.ps \jobname.dvi\MessageBreak
318 ps2pdf \filename@base.ps\MessageBreak
319 -----
320 }%
321 \fi
322 }%
323 }%
```

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

```
324 \newcommand\ppf@isnum[1]{%
325 \if!\ifnum9<1#1!\else_\fi\expandafter\@firstoftwo
326 \else\expandafter\@secondoftwo\fi}%
```

postscript 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.

psmatrix

```
327 \newcommand*\ppf@set@mode{%
328 \newcommand*\ppf@test@mmode{%
329 \ifmmode
330 \ifinner
331 \let\ppf@set@mode=$%
```

```

332 \else
333   \def\ppf@set@mode{${$}%
334 \fi
335 \else
336   \let\ppf@set@mode=\@empty
337 \fi
338 }
339 \newenvironment{postscript}[1] []
340 {%
341   \ppf@test@mmode
342   \gdef\ppf@Gin@keys{%
343     \def\@tempa{postscript}\ifx\@tempa\@currentenv\gdef\ppf@Gin@keys{#1}\fi

```

Innerhalb der Umgebung ist das Parsen der Argumente von `\pst@object` unnötig, daher wieder Originaldefinition verwenden.

```

344 \expandafter\let\expandafter\pst@object
345   \csname pr@string\pst@object\endcsname
346 \pr@outerfalse

```

Nötig für `\psmatrix`.

```

347 \@makeother\&%
348 \def\Gin@ii[#1]#2{\setbox\@tempboxa=\vbox\bgroup
349   \ppf@set@mode
350 }%
351 {\ppf@set@mode\egroup\aftergroup\ppf@@getpicture}%
352 \AtBeginDocument{%
353   \ifundefined{PSTricksLoaded}{}%
354   {%
355     \iffalse
356       \PreviewEnvironment{pspicture}% Why doesn't it work?
357       \g@addto@macro\pspicture{%
358         %\pr@outerfalse% necessary, or already there anyway?
359         \@makeother\&% necessary?
360         \def\Gin@ii[#1]#2{%
361           }%
362         \g@addto@macro\endpspicture{\ppf@@getpicture}%
363       \else
364         \def\pst@@@picture[#1](#2,#3)(#4,#5){\postscript}%
365         \def\endpspicture{\endpostscript\endgroup}%
366       \fi
367       \ifundefined{psmatrix}{}%
368       {\let\psmatrix=\postscript\let\endpsmatrix=\endpostscript}%
369     }%
370   \ifundefined{pfx@includegraphics}{}%

```

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@includegraphics#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 {\@latexerror{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 Makros, die z. B. eine `psmatrix` enthalten. (Einen “Hook” verwenden, falls andere Zeichen auch noch benötigt werden!?)

```

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

	a box or something else. (RN)	14		Now pdfTeX graphics are preferred. (RN)	5
v1.0g	General: Definition of <code>\PDFcontainer</code> now with <code>\edef</code> . (RN)	5	v1.0s	General: Scaling e.g. of PostScript pictures now only in extraction mode. Some code cleaning. (RN)	1
	<code>\usepicture</code> : Now <code>\usepspicture</code> does accept a numerical parameter. (RN)	14		<code>\Gin@ii</code> : Rewritten. (RN)	11
v1.0h	<code>postscript</code> : Based no more on the comment environment from the verbatim package. (RN)	12	v1.1a	General: Support for the internal PSTricks macro <code>\pst@object</code> . (HjG/RN)	8
v1.0i	<code>\ppf@is@pdfTeX@graphic</code> : No more errors for given files without extensions. (RN)	5	v1.1b	General: Ignore the call of <code>\nofiles</code> inside of <code>preview</code> . (RN)	9
v1.0j	General: Check <code>AtBeginDocument</code> for package ‘ <code>pstricks</code> ’ even if ‘ <code>nopstricks</code> ’ is given. (RN)	1		Some code and documentation cleaning. (RN)	1
v1.0k	<code>\Gin@setfile</code> : Show also the pagenumber if exists. (RN)	11	v1.1c	General: New package option ‘ <code>tightpage</code> ’ added. (RN)	1
	<code>\Gininclude@graphics</code> : Prevent division by zero. (RN)	8		Special support for ‘ <code>tabularx</code> ’. (RN)	8
v1.0l	General: Options ‘ <code>framesep</code> ’, ‘ <code>framerule</code> ’, ‘ <code>linewidth</code> ’ removed, ‘ <code>fname</code> ’ and ‘ <code>innerframe</code> ’ added. (RN)	1		Supress handling of pdfLaTeX graphic formats in DVI mode. (RN)	6
v1.0m	General: New package option ‘ <code>notightpage</code> ’ added. (RN)	1	v1.1d	<code>psmatrix</code> : Support for PSTricks environment ‘ <code>psmatrix</code> ’. (RN)	12
v1.0n	General: Changed macro names (<code>\savepicture</code> and <code>\usepicture</code>). (RN)	1	v1.1e	General: New option ‘ <code>displaymath</code> ’ (see <code>preview</code> package). (HjG/RN)	4
	Some code cleaning. (RN)	1	v1.1f	General: Package option ‘ <code>ignore</code> ’ reimplemented. Now the compilation of the dtx file in LaTeX mode is possible. (RN)	4
v1.0o	General: New code for ‘ <code>notightpage</code> ’. (RN)	7	v1.1g	<code>psmatrix</code> : ‘ <code>psmatrix</code> ’ environment (preserve math mode). (RN/HjG)	12
	Option ‘ <code>fname</code> ’ renamed to ‘ <code>showname</code> ’. (RN)	1		<code>pspicture</code> : <code>pspicture</code> environment must still parse its arguments. (RN/HjG)	12
v1.0p	General: Some code and documentation cleaning. (RN)	1	v1.1h	<code>\Gininclude@graphics</code> : Check if inside of a PS-related environment (correct graphic inclusion). (RN)	8
v1.0q	<code>\usepicture</code> : Now <code>\usepspicture</code> works for all kind of graphics. (RN)	14	v1.1i	General: <code>\ifpr@outer</code> must be pre-defined. (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> .			Package option ‘ <code>final</code> ’ also for ‘ <code>graphicx</code> ’. (RN)	4

	\Ginclude@graphics: Correction of the inside check. (RN/HjG) . . .	8		"psfragx". (RN)	8
v1.1k			v1.1m		
	General: New environment pst-pdf- defs: Support for PSTricks envi- ronment "psmatrix" inside user definitions. (RN,HjG)	1	v1.1n	General: Merge english and german version of the documentation. (RN)	1
v1.1l				General: \nofiles added (sugges- tion of Torsten Bronger).	6
	General: Support for the package				

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	environments:postscript	<code>\Gininclude@graphics</code> .
<code>\&</code> 347, 359, 425, 426, 428	postscript 2 55, <u>163</u> , 386
<code>\@currenvir</code> 52, 343, 430	environments:pspicture	<code>\GPT@page</code> 260, 269
<code>\@currenvline</code> .. 49, 424	pspicture 2	
<code>\@ehc</code> 382	environments:pst-pdf-	I
<code>\@firstofone</code> 205	defs	<code>\ifppf@pdftex@graphic</code>
<code>\@latex@error</code> 382	pst-pdf-defs 3 58, 75
<code>\@makeother</code> 347, 359, 426	<code>\everydisplay</code> 203	<code>\ifppf@PST@used</code> ..
<code>\@ppf@PST@usedfalse</code> 10	 4, 81, 182
<code>\@ppf@PST@usedtrue</code> 4, 9	F	<code>\ifppf@tightpage</code> .
<code>\@ppf@pdftex@graphicfalse</code>	<code>\filename@base</code> 317, 318 5, 101, 110
..... 65	<code>\filename@parse</code> ... 310	<code>\ifGin@frame</code> ... 59, 242
<code>\@ppf@pdftex@graphictrue</code>	<code>\frame</code> 238	<code>\ifGin@ignore</code> .. 62, 277
..... 71		<code>\ifGin@innerframe</code> .
<code>\@ppf@tightpagefalse</code> 16	G 60, 240
<code>\@ppf@tightpagetrue</code>	<code>\Gin@bboxtrue</code> 393	<code>\ifGin@showname</code> 61, 281
..... 5, 17	<code>\Gin@boolkey</code>	<code>\ifinner</code> 131, 330
	. 232, 234, 236, 247	<code>\ifmmode</code> 130, 329
A	<code>\Gin@defaultbp</code> 391, 392	<code>\ifpr@outer</code>
<code>\AtBeginDvi</code> ... 183, 184	<code>\Gin@extensions</code> 56,	. 63, 164, 199, 416
	72, 89, 165, 228, 266	<code>\includegraphics</code> ..
C	<code>\Gin@framefalse</code> ... 59	.. 3, 371, 402, 411
<code>\c@lor@to@ps</code> 181	<code>\Gin@ignorefalse</code> .. 62	<code>\includegraphicx</code> ... 3
<code>\c@pspicture</code> 274, 294,	<code>\Gin@ii</code> 57, <u>261</u> , 348, 360	
295, 298, 309, 419	<code>\Gin@innerframefalse</code> 60	J
<code>\catcode</code> 425, 428	<code>\Gin@llx</code> 390, 397	<code>\jobname</code> .. 41, 316, 317
<code>\CurrentOption</code> 12, 15, 19	<code>\Gin@lly</code> 390, 395	
	<code>\Gin@nat@height</code> ...	L
D	. 389, 392, 394, 395	<code>\leavevmode</code> 400
<code>\DeclareGraphicsRule</code>	<code>\Gin@nat@width</code>	<code>\long</code> 198
..... 95, 230	. 389, 391, 396, 397	
<code>\define@key</code> . 96–100,	<code>\Gin@PS@file@header</code> 177	N
231, 233, 235,	<code>\Gin@PS@literal@header</code>	<code>\newcolumntype</code> 158
237, 246, 248, 401 177	<code>\nofiles</code> .. 80, 193, 195
	<code>\Gin@PS@raw</code> 178	
E	<code>\Gin@PS@restored</code> .. 178	O
<code>\endpostscript</code> 365, 368	<code>\Gin@req@height</code> ... 399	<code>\output</code> 192, 196
<code>\endpsmatrix</code>	<code>\Gin@req@sizes</code> 398	
.... 143, 147, 368	<code>\Gin@req@width</code> 399	P
<code>\endpspicture</code> . 362, 365	<code>\Gin@setfile</code> 68, 257, <u>258</u>	<code>\paperheight</code> 103
<code>\endtabular</code> 159	<code>\Gin@shownamefalse</code> . 61	<code>\paperwidth</code> 103
<code>\endtabularx</code> 159	<code>\Gin@urx</code> 391, 396	<code>\PassOptionsToPackage</code>
environments:	<code>\Gin@ury</code> 392, 394 12, 15, 19
postscript . <u>107</u> , <u>327</u>	<code>\Gin@viewport@code</code> . 393	<code>\PDFcontainer</code>
psmatrix .. <u>125</u> , <u>327</u>	<code>\Gin@vllx</code> 114	40, 41, 268, 274,
pspicture .. <u>124</u> , <u>327</u>	<code>\Gin@vllly</code> 114	289, 291, 297,
pst-pdf-defs <u>46</u> , <u>421</u>	<code>\Gin@vurx</code> 114	299, 310, 312, 412
	<code>\Gin@vury</code> 114	<code>\pdflastximage</code> 376

<code>\pdflastximagepages</code>	292	<code>\ppf@namefont</code>	<code>\PreviewMacro</code>
<code>\pdfliteral</code> 173		. 250, 251, 255, 282		. 149, 161, 172, 217
<code>\pdfoutput</code> 26	<code>\ppf@nofiles</code>	.. 193, 195	<code>\psmatrix</code> 140,
<code>\pdfrefximage</code> 388	<code>\ppf@other@extensions</code> 43,		142, 144, 145, 368
<code>\pdfTeXtext</code> 67, 70		89, 93, 165, 228, 266	<code>psmatrix</code>	(environ-
<code>\pdftexversion</code> 88	<code>\ppf@pr@psmatrix</code>	..		ment) ... 125 , 327
<code>\pdfximage</code> 291	 141, 146	<code>\pspicture</code> 357
<code>\pfx@includegraphics</code> 371	<code>\ppf@PreviewBbAdjust</code> 83, 85, 109	<code>pspicture</code>	(environ-
<code>\pfx@includegraphicsx</code> 161, 372	<code>\ppf@psmatrix</code>	. 140, 144		ment) . 2 , 124 , 327
<code>\postscript</code>	... 364, 368	<code>\ppf@RestoreBbAdjust</code> 84, 119	<code>pst-pdf-defs</code>	(environ-
<code>postscript</code>	(environ-	<code>\ppf@set@mode</code>		ment) .. 3 , 46 , 421
	ment) . 2 , 107 , 327	 128, 132,	<code>\pst@@@picture</code> 364
<code>\ppf@getpicture</code>	212,		134, 137, 146,	<code>\pst@object</code>
	351, 362, 372, 415		147, 327, 331,		. 155, 223, 344, 345
<code>\ppf@container@max</code>	.		333, 336, 349, 351	<code>\PSTricksOff</code> 180
 291 ,	<code>\ppf@temp</code>		
	295, 297, 307, 407		. 183, 184, 425, 428		
<code>\ppf@draft</code>	3, 13, 14, 411	<code>\ppf@temptoken</code>		
<code>\ppf@endpsmatrix</code>	..		. 191, 192, 196, 197		
 143, 147	<code>\ppf@test@mmode</code>	...		
<code>\ppf@filename</code>	. 254,		. 129, 144, 328, 341		
	259, 283, 284, 387	<code>\ppf@TeX@mode</code>	. 2 , 6 –		
<code>\ppf@getpicture</code>	...		8, 21, 22, 33, 35,	<code>\savepicture</code>	. 3 , 45 , 375
 381, 405 , 418		37, 77, 79, 175, 433	<code>\setkeys</code> 112, 263
<code>\ppf@Gin@extensions</code>	56	<code>\pr@cleanup</code>	... 201, 213	<code>\shipout</code> 195
<code>\ppf@Gin@ii</code> 57, 278	<code>\pr@endbox</code> 209	<code>\string</code>	... 142, 145, 345
<code>\ppf@Gin@keys</code>	. 256,	<code>\pr@outerfalse</code>	<code>\strutbox</code> 283
	342, 343, 414, 418		. 73, 204, 346, 358		
<code>\ppf@Gin@setfile</code>	..	<code>\pr@shipout</code> 195		
 257, 258	<code>\pr@startbox</code> 198		
<code>\ppf@Gin@includegraphics</code> 55,	<code>\PreviewBbAdjust</code>	..		
	73, 167, 169, 172		. 85, 102, 109, 113		
<code>\ppf@is@pdfTeX@graphic</code>	. 64 , 165, 266, 435	<code>\PreviewEnvironment</code>		
<code>\ppf@isnum</code>	... 324 , 380		. 120, 124, 127, 356		

R

`\raisebox` 283
`\refstepcounter` 273, 417
`\rule` 166

S

`\savepicture` . [3](#), [45](#), [375](#)
`\setkeys` 112, 263
`\shipout` 195
`\string` ... 142, 145, 345
`\strutbox` 283

T

`\tabular` 157
`\tabularx` 157

U

`\usepicture` .. [3](#), [44](#), [377](#)

V

`\voidb@x` 211