

# Das Paket `pst-pdf`\*

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

2019/11/15

## 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` v1.2e vom 2019/11/15.

<sup>†</sup>`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`<sup>1</sup> 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 erzeugen<sup>2</sup>. 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)] (&lt;x0,x1&gt;)&lt;(y0,y1)&gt; ... \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>

<sup>1</sup>CTAN: `support/pdfcrop/`

<sup>2</sup>Die 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<sup>A</sup>T<sub>E</sub>X</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>)&lt;<i>ovpfgd</i>&gt;[<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>&amp;</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<sub>E</sub>X 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 Dateierendungen im pdf-Modus, Parameter #3 enthält die gültigen Dateierendungen 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{\detokenize\expandafter{#2}}%

```

Statt Einladen einer identifizierten Grafik nur Test der Grafikendung.

```

66 \def\Gin@setfile##1##2##3{%
67 \expandtwoargs\in@{\detokenize\expandafter{##2}},{\, \pdfTeXtext,}%
68 \ifin@global\@ppf@pdftex@graphictrue\fi}%

```

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

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

```

Testaufruf. Dabei Ausgabe vollständig verhindern.

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

74 \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 Paketooption “`active`” erzwingen diesen Modus.

```
75 \PackageInfo{pst-pdf}{%
76   MODE: \ppf@TeX@mode\space (dvi -- extraction mode)}
77 \nofiles
78 \let\makeindex@empty \let\makeglossary@empty \let\printindex@empty
79 \renewcommand*\makeindex[1] [] {}%
80 \renewcommand*\makeglossary[1] [] {}%
81 \renewcommand*\printindex[1] [] {}%
82 \AtBeginDocument{\overfullrule=\z@}%
83 \if@ppf@PST@used\RequirePackage{pstricks}\fi
84 \RequirePackage[active,dvips,tightpage]{preview}[2005/01/29]%
85 \newcommand*\ppf@PreviewBbAdjust{}
86 \newcommand*\ppf@RestoreBbAdjust{}
87 \let\PreviewBbAdjust\ppf@PreviewBbAdjust}%
```

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

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

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

In diesem Modus sollten undefinierte keys keinen Fehler bewirken.

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

In diesem Modus keine Funktion.

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

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

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

```
119 \newenvironment{postscript}[1][]{%
120 {%
121 \global\let\ppf@PreviewBbAdjust\PreviewBbAdjust
122 \ifppf@tightpage
123 \begingroup
124 \setkeys{Gin}{#1}%
125 \xdef\PreviewBbAdjust{%
126 -\Gin@vllx bp -\Gin@vlyy bp \Gin@vurx bp \Gin@vury bp}%
127 \endgroup
128 \fi
129 \ignorespaces
130 }%
131 {\aftergroup\ppf@RestoreBbAdjust}}%

132 \PreviewEnvironment{postscript}%
133 \AtBeginDocument{%
134 \@ifundefined{PSTricksLoaded}{}%
135 {%
```

**pspicture** Originaldefinition preview bekannt machen.

```
136 \PreviewEnvironment{pspicture}%
```

**psmatrix** Originaldefinition preview bekannt machen.

```
137 \@ifundefined{psmatrix}{}%
138 {%
139 \PreviewEnvironment{psmatrix}%
140 \newcommand*\ppf@set@mode{}%
141 \newcommand*\ppf@test@mmode{%
142 \ifmmode
143 \ifinner
144 \let\ppf@set@mode=$%
145 \else
```

```

146         \def\ppf@set@mode{$$}%
147         \fi
148     \else
149         \let\ppf@set@mode=\@empty
150     \fi
151 }%
152 \let\ppf@psmatrix=\psmatrix
153 \expandafter\let\expandafter\ppf@pr@psmatrix%
154     \expandafter=\csname pr@\string\psmatrix\endcsname
155 \let\ppf@endpsmatrix=\endpsmatrix
156 \def\psmatrix{\ppf@test@mode\ppf@psmatrix}
157 \expandafter\def\csname pr@\string\psmatrix\endcsname{%
158     \ppf@set@mode\ppf@pr@psmatrix}%
159 \def\endpsmatrix{\ppf@endpsmatrix\ppf@set@mode}%
160 }%

```

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.

```

161     \PreviewMacro[{}*[]%
162     ?\bgroup{##1}{##1}}{}%
163     ?\bgroup{##1}{##1}}{}%
164     ?({#{(##1)}({#{##1})})}{}%
165     ?({#{(##1)}({#{##1})})}{}%
166     ?({#{(##1)}({#{##1})})}{}%
167     ]{\pst@object}

```

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

```

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

```

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

```

173     \@ifundefined{pfx@includegraphicx}{}{}%
174     \PreviewMacro[{}{}]{\pfx@includegraphicx}%
175 }%

```

`\Gscale@@box` Skalieren verhindern.

```

176     \def\Gscale@@box#1#2#3{%
177         \toks@{\mbox}%
178     }

```

`\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.

```

179     \def\Ginclude@graphics#1{%
180         \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.

```

181     \ppf@is@pdfTeX@graphic{#1}{\ppf@other@extensions}{\Gin@extensions}%
    Dummy-Box, um Division durch Null bei Skalierung/Rotation zu vermeiden. Wird
    ansonsten ignoriert.
182     {\rule{10pt}{10pt}}%
183     {\ppf@Gin@include@graphics{#1}}%
184     \else
    Innerhalb von PS-Umgebungen (pspicture usw.) muss sich \includegraphics
    wie die Originaldefinition verhalten (nur die DVIPS-Graphik-Typen sind gültig).
185     \ppf@Gin@include@graphics{#1}%
186     \fi
187 }%

188 \PreviewMacro[{}]{\ppf@Gin@include@graphics}%
189 \let\pdfliteral@gobble%
190 \or

```

### 3.4 pdf $\LaTeX$ 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.

```

191 \PackageInfo{pst-pdf}{MODE: \ppf@TeX@mode\space (pdfTeX mode)}%
    Verhindert pdf $\TeX$ s Warnung Non-PDF special ignored!.
192 \ifppf@PST@used
193   \let\ppf@temp@AtBeginDvi\let\AtBeginDvi@gobble
194   \def\c@lor@to@ps#1 #2\@{}
195   \RequirePackage{pstricks}\let\AtBeginDvi\ppf@temp
196   \fi
197 \@temptokena{%
198   \let\Gin@PS@file@header@gobble\let\Gin@PS@literal@header@gobble
199   \let\Gin@PS@raw@gobble\let\Gin@PS@restored@gobble
200   \@ifundefined{PSTricksLoaded}{-}%

```

Für PSTricks < 2.0 nötig.

```

201     \PSTricksOff
202     \def\c@lor@to@ps#1 #2\@{}
203     }%
204 }%

```

PostScript-Ausgabe jetzt verhindern und später noch einmal.

```

205 \the\@temptokena
206 \expandafter\AtBeginDocument\expandafter
207   {\the\@temptokena\@temptokena{}}%
208 \@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.

```

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

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

```

(Siehe auch identische Definition im DVI-Modus.)

```

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

```

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

```

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

```

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

```

249 \DeclareGraphicsRule{*}{eps}{*}{}%
250 \define@key{Gin}{innerframe}[true]{%
251   \lowercase{\Gin@boolkey{#1}}{innerframe}}%

```

```

252 \define@key{Gin}{frame}[true]{%
253   \lowercase{\Gin@boolkey{#1}}{frame}}%
254 \define@key{Gin}{ignore}[true]{%
255   \lowercase{\Gin@boolkey{#1}}{ignore}}%
256 \define@key{Gin}{frame@@}{%

(Nur intern zu benutzen!)

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

```

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

```

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

```

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

```

280 \def\Gin@ii[#1]#2{%
281   \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.

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

Leerer Dateiname beim Aufruf von \usepicture aus.

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

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

285   {%
286     \setkeys{Gin}{#1}%
287     \ifx\ppf@tempb\PDFcontainer
288       \@temptokena{page=\Gin@page}%
289     \fi
290   }%

```

```

291     {%
292       \refstepcounter{pspicture}%
293       \@temptokena{page=\the\c@pspicture}\def\ppf@tempb{\PDFcontainer}%
294     }%
295     \fi
296     \ifGin@ignore\else
    “frame@@=0” = innerer Rahmen, “frame@@=1” = äußerer Rahmen.
297     \edef\@tempa{\noexpand\ppf@Gin@ii[frame@@=0,\the\@temptokena,
298       frame@@=1]{\ppf@tempb}}%
299     \@tempa
300     \ifGin@showname
301       \ppf@namefont
302       \raisebox{-\ht\strutbox}[0pt][0pt]{\llap{\ppf@filename}}%
303       \gdef\ppf@filename{}%
304     \fi
305     \fi
306     \endgroup
307 }%

308 \IfFileExists{\PDFcontainer}%
309 {%

```

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

```

310   \pdfximage{\PDFcontainer}%
311   \edef\ppf@container@max{\the\pdflastximagepages}%

312   \AtEndDocument{%
313     \ifnum\c@pspicture>\z@

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

```

```

338 -----
339     }%
340     \fi
341 }%
342 }%

```

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

```

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

```

346 \newcommand*\ppf@set@mode{}%
347 \newcommand*\ppf@test@mode{}%
348 \ifmode
349   \ifinner
350     \let\ppf@set@mode=$%
351   \else
352     \def\ppf@set@mode{$$}%
353   \fi
354 \else
355   \let\ppf@set@mode=\@empty
356 \fi
357 }

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

```

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

```

377   \let\includegraphics=\pfx@includegraphics

```

```

378     \def\pfx@includegraphicx#1#2{\ppf@getpicture}%
379     }%
380 }%

```

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

```

381 \def\savepicture#1{%
382   \expandafter\xdef\csname ppf@@@#1\endcsname{\the\pdfastximage}}%

```

`\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`.

```

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

```

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

```

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

```

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

```

421 \newcommand*\ppf@getpicture{%
422   \ifpr@outer
423     \refstepcounter{pspicture}%
424     \expandafter\ppf@getpicture\expandafter{\ppf@Gin@keys}%
425     {\the\c@pspicture}%
426   \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.

```

427 \renewenvironment*{pst-pdf-defs}%
428   {%
429   \endgroup
430 %   ??? \@currenvline
431   \chardef\ppf@temp=\catcode'\&%
432   \@makeother\&%
433 }{%
434   \catcode'\&=\ppf@temp
435   \begingroup
436   \def\@currenvir{pst-pdf-defs}%
437 }
438 \else

```

### 3.5 Inactiver Modus

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

```

439 \PackageInfo{pst-pdf}{MODE: \ppf@TeX@mode\space (inactive mode)}%
440 \newenvironment{postscript}[1][\ignorespaces]{%
441 \let\ppf@is@pdfTeX@graphic\relax
442 \fi
443 \InputIfFileExists{pst-pdf.cfg}{%
444 \PackageInfo{pst-pdf}{Local config file pst-pdf.cfg used}}{}
445 \endpackage}

```

## Change History

v1.0a	General: Initial version. . . . . 1	v1.0d	General: Redefinition of <code>\includegraphics</code> in modes 0 und 1. Now using of eps graphics directly in pdf $\text{\TeX}$ is possible. (RN) . . . . . 1
v1.0b	General: Some code and documentation cleaning. (RN) . 1		
v1.0c	General: New options “pstricks”, “nopstricks”, “draft” and “final”. (RN) . . . . . 4	v1.0e	<code>postscript</code> : “trim” option added. (RN) . . . . . 7

v1.0f		v1.0o	
\savepicture: New macro		General: New code for	
\savepicture. (RN) . . . . .	14	“notightpage”. (RN) . . . . .	7
\usepicture: New macro		Option “fname” renamed to	
\usepicture. Useful for		“showname”. (RN) . . . . .	1
putting a PSTricks graphic in a		v1.0p	
box or something else. (RN) .	14	General: Some code and	
General: Config file loading added.		documentation cleaning. (RN) .	1
(RN) . . . . .	15	v1.0q	
v1.0g		\usepicture: Now \usepicture	
\usepicture: Now \usepicture		works for all kind of graphics.	
does accept a numerical		(RN) . . . . .	14
parameter. (RN) . . . . .	14	v1.0r	
General: Definition of		\ppf@is@pdfTeX@graphic:	
\PDFcontainer now with		Changed	
\edef. (RN) . . . . .	5	\ppf@is@known@graphic to	
v1.0h		\ppf@is@pdfTeX@graphic.	
psmatrix: Based no more on the		Now pdfTeX graphics are	
comment environment from the		preferred. (RN) . . . . .	5
verbatim package. (RN) . . . .	13	v1.0s	
v1.0i		\Gin@ii: Rewritten. (RN) . . . . .	11
\ppf@is@pdfTeX@graphic: No		General: Scaling e.g. of PostScript	
more errors for given files		pictures now only in extraction	
without extensions. (RN) . . . .	5	mode. Some code cleaning.	
v1.0j		(RN) . . . . .	1
General: Check AtBeginDocument		v1.1a	
for package ‘pstricks’ even if		General: Support for the internal	
“nopstricks” is given. (RN) . . .	1	PSTricks macro \pst@object.	
For \includegraphics		(HjG/RN) . . . . .	8
\usepicture and postscript		v1.1b	
the new options “frame”,		General: Ignore the call of	
“framesep”, “framerule”,		\nofiles inside of preview.	
“linewidth”, and “ignore”		(RN) . . . . .	10
added. (RN) . . . . .	1	Some code and documentation	
v1.0k		cleaning. (RN) . . . . .	1
\Gin@setfile: Show also the		v1.1c	
pagenumber if exists. (RN) . . .	11	General: New package option	
\Gin@include@graphics: Prevent		“tightpage” added. (RN) . . . . .	1
division by zero. (RN) . . . . .	8	Special support for “tabularx”.	
v1.0l		(RN) . . . . .	8
General: Options “framesep”,		Supress handling of pdfL <sup>A</sup> T <sub>E</sub> X	
“framerule”, “linewidth”		graphic formats in DVI mode.	
removed, “fname” and		(RN) . . . . .	6
“innerframe” added. (RN) . . . .	1	v1.1d	
v1.0m		postscript: Support for PSTricks	
General: New package option		environment “psmatrix”. (RN) .	13
“notightpage” added. (RN) . . .	1	v1.1e	
v1.0n		General: New option “displaymath”	
General: Changed macro names		(see preview package).	
(\savepicture and		(HjG/RN) . . . . .	4
\usepicture). (RN) . . . . .	1	v1.1f	
Some code cleaning. (RN) . . . .	1	General: Package option “ignore”	
		reimplemented. Now the	



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

# 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>\&amp;</code> . . . . .	431, 432, 434
<code>\@currentenv</code> . . . . .	50, 361, 436
<code>\@currentline</code> . . . . .	47, 430
<code>\@ehc</code> . . . . .	388
<code>\@expandtwoargs</code> . . . . .	67
<code>\@firstofone</code> . . . . .	224
<code>\@ifpackageloaded</code> . . . . .	98, 101
<code>\@latex@error</code> . . . . .	388
<code>\@makeother</code> . . . . .	432
<code>\@ppf@PST@usedfalse</code> . . . . .	10
<code>\@ppf@PST@usedtrue</code> . . . . .	4, 9
<code>\@ppf@pdftex@graphicfalse</code> . . . . .	63
<code>\@ppf@pdftex@graphictrue</code> . . . . .	68
<code>\@ppf@tightpagefalse</code> . . . . .	16
<code>\@ppf@tightpagetrue</code> . . . . .	5, 17
<b>A</b>	
<code>\AtBeginDvi</code> . . . . .	193, 195
<b>C</b>	
<code>\color@to@ps</code> . . . . .	194, 202
<code>\c@pspicture</code> . . . . .	293, 313, 314, 317, 328, 425
<code>\catcode</code> . . . . .	431, 434
<code>\Collect@Body</code> . . . . .	367
<code>\CurrentOption</code> . . . . .	12, 15, 19
<b>D</b>	
<code>\DeclareGraphicsRule</code> . . . . .	106, 249
<code>\define@key</code> . . . . .	107–112, 250, 252, 254, 256, 265, 267, 407
<code>\detokenize</code> . . . . .	65, 67
<b>E</b>	
<code>\endpostscript</code> . . . . .	371, 374
<code>\endpsmatrix</code> . . . . .	155, 159, 374
<code>\endpspicture</code> . . . . .	371
<code>\endtabularx</code> . . . . .	171
environments:	
<code>postscript</code> . . . . .	2, 119, 346
<code>psmatrix</code> . . . . .	137, 346
<code>pspicture</code> . . . . .	2, 136, 346
<code>pst-pdf-defs</code> . . . . .	3, 45, 427
<code>\everydisplay</code> . . . . .	222
<b>F</b>	
<code>\filename@base</code> . . . . .	336, 337
<code>\filename@parse</code> . . . . .	329
<code>\frame</code> . . . . .	257
<b>G</b>	
<code>\Gin@bboxtrue</code> . . . . .	399
<code>\Gin@boolkey</code> . . . . .	251, 253, 255, 266
<code>\Gin@defaultbp</code> . . . . .	397, 398
<code>\Gin@extensions</code> . . . . .	54, 69, 94, 181, 247, 284
<code>\Gin@framefalse</code> . . . . .	57
<code>\Gin@ignorefalse</code> . . . . .	60
<code>\Gin@ii</code> . . . . .	55, 280
<code>\Gin@innerframefalse</code> . . . . .	58
<code>\Gin@llx</code> . . . . .	396, 403
<code>\Gin@lly</code> . . . . .	396, 401
<code>\Gin@nat@height</code> . . . . .	395, 398, 400, 401
<code>\Gin@nat@width</code> . . . . .	395, 397, 402, 403
<code>\Gin@page</code> . . . . .	279, 288
<code>\Gin@PS@file@header</code> . . . . .	198
<code>\Gin@PS@literal@header</code> . . . . .	198
<code>\Gin@PS@raw</code> . . . . .	199
<code>\Gin@PS@restored</code> . . . . .	199
<code>\Gin@req@height</code> . . . . .	405
<code>\Gin@req@sizes</code> . . . . .	404
<code>\Gin@req@width</code> . . . . .	405
<code>\Gin@setfile</code> . . . . .	66, 276, 277
<code>\Gin@shownamefalse</code> . . . . .	59
<code>\Gin@urx</code> . . . . .	397, 402
<code>\Gin@ury</code> . . . . .	398, 400
<code>\Gin@viewport@code</code> . . . . .	399
<code>\Gin@vllx</code> . . . . .	126
<code>\Gin@vly</code> . . . . .	126
<code>\Gin@vurx</code> . . . . .	126
<code>\Gin@vury</code> . . . . .	126
<code>\Gin@width@code</code> . . . . .	399
<code>\Ginclude@graphics</code> . . . . .	53, 179, 392
<code>\Gscale@@@box</code> . . . . .	176
<b>I</b>	
<code>\if@ppf@pdftex@graphic</code> . . . . .	56, 72
<code>\if@ppf@PST@used</code> . . . . .	4, 83, 192
<code>\if@ppf@tightpage</code> . . . . .	5, 113, 122
<code>\ifGin@frame</code> . . . . .	57, 261
<code>\ifGin@ignore</code> . . . . .	60, 296
<code>\ifGin@innerframe</code> . . . . .	58, 259
<code>\ifGin@showname</code> . . . . .	59, 300
<code>\ifin@</code> . . . . .	68
<code>\ifinner</code> . . . . .	143, 349
<code>\ifmmode</code> . . . . .	142, 348
<code>\ifpdf</code> . . . . .	24
<code>\ifpr@outer</code> . . . . .	61, 180, 218, 422
<code>\ifvtex</code> . . . . .	28
<code>\ifxetex</code> . . . . .	31
<code>\in@</code> . . . . .	67
<code>\includegraphics</code> . . . . .	3, 377, 408, 417
<code>\includegraphicsx</code> . . . . .	3
<b>J</b>	
<code>\jobname</code> . . . . .	40, 335, 336
<b>K</b>	
<code>\KV@errx</code> . . . . .	99
<b>L</b>	
<code>\leavevmode</code> . . . . .	406
<code>\long</code> . . . . .	217
<b>M</b>	
<code>\makeglossary</code> . . . . .	78, 80
<code>\makeindex</code> . . . . .	78, 79
<code>\mbox</code> . . . . .	177
<b>N</b>	
<code>\newcolumn</code> . . . . .	169
<code>\newcount</code> . . . . .	91
<code>\nofiles</code> . . . . .	77, 211, 215
<b>O</b>	
<code>\OptionNotUsed</code> . . . . .	6

<code>\output</code> . . . . .	210, 216	<code>\ppf@Gin@setfile</code> ..	<code>\PreviewBbAdjust</code> ..
<code>\overfullrule</code> . . . . .	82	. . . . .	87, 114, 121, 125
<b>P</b>			
<code>\PassOptionsToPackage</code>		<code>\ppf@Gin@graphics</code>	<code>\PreviewEnvironment</code>
. . . . .	12, 15, 19	. . . . .	. . . . .
<code>\PDFcontainer</code> . . . . .		53,	132, 136, 139
39, 40, 287, 293,		70, 183, 185, 188	<code>\PreviewMacro</code> . . . . .
308, 310, 316,		<code>\ppf@is@pdfTeX@graphic</code>	. 161, 174, 188, 236
318, 329, 331, 418		. 62, 181, 284, 441	<code>\printindex</code> . . . . .
<code>\pdflastximage</code> . . . . .	382	<code>\ppf@isnum</code> . . .	78, 81
<code>\pdflastximagepages</code>	311	<u>343</u> , 386	<code>\psmatrix</code> . . . . .
<code>\pdfliteral</code> . . . . .	189	<code>\ppf@namefont</code> . . . . .	154, 156, 157, 373
<code>\pdfoutput</code> . . . . .	91, 92	. 269, 270, 274, 301	<code>psmatrix</code> (environ-
<code>\pdfrefximage</code> . . . . .	394	<code>\ppf@nofiles</code> ..	ment) . . . <u>137</u> , <u>346</u>
<code>\pdfTeXtext</code> . . . . .	65, 67	. 211, 215	<code>pspicture</code> (environ-
<code>\pdfTeXversion</code> . . . . .	90	<code>\ppf@other@extensions</code>	ment) . 2, <u>136</u> , <u>346</u>
<code>\pdfximage</code> . . . . .	310	. . . . .	<code>pst-pdf-defs</code> (environ-
<code>\pfx@includegraphics</code>		42, 94,	ment) . . 3, <u>45</u> , <u>427</u>
. . . . .	377	104, 181, 247, 284	<code>\pst@@@picture</code> . . . . .
<code>\pfx@includegraphicx</code>		<code>\ppf@output</code> 209, 210, 216	<code>\pst@object</code> . . . 167, 242
. . . . .	174, 378	<code>\ppf@pr@psmatrix</code> ..	<code>\PSTricksOff</code> . . . . .
<code>\postscript</code> . . . 370, 373		. . . . .	201
<code>postscript</code> (environ-		<code>\ppf@PreviewBbAdjust</code>	
ment) . 2, <u>119</u> , <u>346</u>		. . . . .	<b>R</b>
<code>\ppf@getpicture</code> ..		85, 87, 121	<code>\raisebox</code> . . . . .
. 231, 366, 378, <u>421</u>		<code>\ppf@psmatrix</code> . 152, 156	302
<code>\ppf@container@max</code> .		<code>\ppf@RestoreBbAdjust</code>	<code>\refstepcounter</code> 292, 423
. . . . .	310,	. . . . .	<code>\rule</code> . . . . .
314, 316, 326, 413		86, 131	182
<code>\ppf@draft</code> 3, 13, 14, 417		<code>\ppf@set@mode</code> . . . . .	<b>S</b>
<code>\ppf@endpsmatrix</code> ..		. 140, 144, 146,	<code>\savepicture</code> . 3, 44, <u>381</u>
. . . . .	155, 159	149, 158, 159,	<code>\setkeys</code> . . . . .
<code>\ppf@filename</code> . 273,		346, 350, 352, 355	124, 286
278, 302, 303, 393		<code>\ppf@shipout</code> ..	<code>\shipout</code> . . . . .
<code>\ppf@getpicture</code> . . .		212, 214	212, 214
. . . . .	387, <u>411</u> , 424	<code>\ppf@temp</code> . . . . .	<code>\string</code> . . . . .
<code>\ppf@Gin@extensions</code>	54	. 193, 195, 431, 434	154, 157
<code>\ppf@Gin@ii</code> . . . . .	55, 297	<code>\ppf@tempb</code> . . . . .	<code>\strutbox</code> . . . . .
<code>\ppf@Gin@keys</code> . 275,		. 282,	302
362, 364, 420, 424		283, 287, 293, 298	<b>T</b>
		<code>\ppf@test@mmode</code> . . .	<code>\tabularx</code> . . . . .
		. . . . .	170
		<code>\ppf@TeX@mode</code> . . . . .	<b>U</b>
		. . . 2, 7, 8, 21,	<code>\usepicture</code> . . 3, 43, <u>383</u>
		23, 25, 29, 32,	<b>V</b>
		34, 74, 76, 191, 439	<code>\voidb@x</code> . . . . .
		<code>\pr@cleanup</code> . . . 220, 232	230
		<code>\pr@endbox</code> . . . . .	<b>X</b>
		228	<code>\XKV@err</code> . . . . .
		<code>\pr@outerfalse</code> . 70, 223	102
		<code>\pr@startbox</code> . . . . .	
		217	