

Das Paket `pst-pdf`*

Rolf Niepraschk[†] 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 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` v1.2e vom 2019/11/15.

[†]`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 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/graphics</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}{\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 Compilertests

Es wird getestet, welcher \TeX compiler in welchem Modus läuft (siehe ‘graphics.cfg’ von \textTeX /T \TeX Live). 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 $\text{\TeX}$  or Lua $\text{\TeX}$  are running in PDF mode
26     \def\ppf@TeX@mode{1}%
27     \RequirePackage{luatex85}%
28   \else
29     \ifvtex
30       ⇒ V $\text{\TeX}$ 
31       \def\ppf@TeX@mode{9}%
32     \else
33       \ifxetex
34         ⇒ Xe $\text{\TeX}$ 
35         \def\ppf@TeX@mode{9}%
36       \else
37         ⇒ DVI mode
38         \def\ppf@TeX@mode{0}%
39       \fi
```

```

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][{}]{
44 \newcommand*\savepicture[1]{}

```

pst-pdf-defs

```

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

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

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

```

\ifpr@outer wird eigentlich im Paket preview definiert. Wir müssen es aber bereits hier zusätzlich tun, da sonst T_EX u. U. beim Parsen der \ifcase-Struktur “außer Tritt” kommt.

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

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

```

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

```

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}}{\pdfTeXext}%
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@Gininclude@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 Paketooption “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 pdf \LaTeX -Modus erlaubten Endungen von Grafikdateien benötigt.

```

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

```

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.

```

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

```

In diesem Modus sollten undefinierte keys keinen Fehler bewirken.

```

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

In diesem Modus keine Funktion.

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

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

```

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

```

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

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

```

pspicture Originaldefinition preview bekannt machen.

```

138 \PreviewEnvironment{pspicture}%

```

psmatrix Originaldefinition preview bekannt machen.

```

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

```

```

148         \def\ppf@set@mode{$$}%
149         \fi
150     \else
151         \let\ppf@set@mode=\@empty
152     \fi
153 }%
154 \let\ppf@psmatrix=\psmatrix
155 \expandafter\let\expandafter\ppf@pr@psmatrix%
156     \expandafter=\csname pr@\string\psmatrix\endcsname
157 \let\ppf@endpsmatrix=\endpsmatrix
158 \def\psmatrix{\ppf@test@mode\ppf@psmatrix}
159 \expandafter\def\csname pr@\string\psmatrix\endcsname{%
160     \ppf@set@mode\ppf@pr@psmatrix}%
161 \def\endpsmatrix{\ppf@endpsmatrix\ppf@set@mode}%
162 }%

```

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.

```

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

```

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

```

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

```

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

```

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

```

`\Gscale@@box` Skalieren verhindern.

```

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

```

`\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.

```

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

```

183     \ppf@is@pdfTeX@graphic{#1}{\ppf@other@extensions}{\Gin@extensions}%
    Dummy-Box, um Division durch Null bei Skalierung/Rotation zu vermeiden. Wird
    ansonsten ignoriert.
184     {\rule{10pt}{10pt}}%
185     {\ppf@Gin@include@graphics{#1}}%
186     \else
    Innerhalb von PS-Umgebungen (pspicture usw.) muss sich \includegraphics
    wie die Originaldefinition verhalten (nur die DVIPS-Graphik-Typen sind gültig).
187     \ppf@Gin@include@graphics{#1}%
188     \fi
189 }%

190 \PreviewMacro[{}]{\ppf@Gin@include@graphics}%
191 \let\pdfliteral\@gobble%
192 \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.

```

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

```

Für PSTricks < 2.0 nötig.

```

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

```

PostScript-Ausgabe jetzt verhindern und später noch einmal.

```

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

```

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

```

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

```

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

```

(Siehe auch identische Definition im DVI-Modus.)

```

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

```

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

```

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

```

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

```

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

```

```

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

```

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

```

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

```

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

```

282 \def\Gin@ii[#1]#2{%
283   \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.
284   \@temptokena{#1}\def\ppf@tempb{#2}%
  Leerer Dateiname beim Aufruf von \usepicture aus.
285   \ifx\ppf@tempb\@empty\else
286     \ppf@is@pdfTeX@graphic{#2}{\Gin@extensions}{\ppf@other@extensions}%
  Grafiken aus Containerdatei sind bereits skaliert usw. Nicht noch einmal, daher
  optionalen Parameter ignorieren.
287     {%
288       \setkeys{Gin}{#1}%
289       \ifx\ppf@tempb\PDFcontainer
290         \@temptokena{page=\Gin@page}%
291       \fi
292     }%

```

```

293     {%
294       \refstepcounter{pspicture}%
295       \@temptokena{page=\the\c@pspicture}\def\ppf@tempb{\PDFcontainer}%
296     }%
297   \fi
298   \ifGin@ignore\else
    “frame@@=0” = innerer Rahmen, “frame@@=1” = äußerer Rahmen.
299     \edef\@tempa{\noexpand\ppf@Gin@ii[frame@@=0,\the\@temptokena,
300       frame@@=1]{\ppf@tempb}}%
301     \@tempa
302     \ifGin@showname
303       \ppf@namefont
304       \raisebox{-\ht\strutbox}[0pt][0pt]{\llap{\ppf@filename}}%
305       \gdef\ppf@filename{}%
306     \fi
307   \fi
308   \endgroup
309 }%

310 \IfFileExists{\PDFcontainer}%
311 {%

```

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

```

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

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

    Warnung ist nur sinnvoll, wenn überhaupt Grafiken benötigt wurden.

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

```

```

340      -----
341      }%
342      \fi
343      }%
344      }%

```

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

```

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

```

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

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

```

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

```

379   \let\includegraphics=pfx@includegraphics

```

```

380     \def\pfx@includegraphicx#1#2{\ppf@getpicture}%
381   }%
382 }%

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

`pst-pdf-defs` 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.

```

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

440 \else

```

3.5 Inactiver Modus

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

```

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

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

```

Change History

v1.0a	General: Initial version. 1	v1.0d	General: Redefinition of <code>\includegraphics</code> in modes 0 und 1. Now using of eps graphics directly in pdf \LaTeX 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 \usepspicture	
\usepicture: Now \usepspicture		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 ^A T _E 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 ^A T _E X mode is possible. (RN) . . .	4		
v1.1g	postscript : “psmatrix” environment (preserve math mode). (RN/HjG)	13		
	pspicture : pspicture environment must still parse its arguments. (RN/HjG)	13		
v1.1h	\Gininclude@graphics : Check if inside of a PS-related environment (correct graphic inclusion). (RN)	8		
v1.1i	\Gininclude@graphics : Correction of the inside check. (RN/HjG) . . .	8		
	General: \ifpr@outer must be predefined. (HjG/RN)	5		
	Package option “final” also for “graphicx”. (RN)	4		
v1.1k	General: New environment pst-pdf-defs: Support for PSTricks environment “psmatrix” inside user definitions. (RN,HjG)	1		
v1.1l	General: Support for the package “psfragx”. (RN)	8		
v1.1m	General: Merge english and german version of the documentation. (RN)	1		
v1.1n	General: \nofiles added (suggestion of Torsten Bronger).	6		
v1.1o	\Gscale@@box : Disable scaling. (RN)	8		
v1.1p	General: \let\output\@gobble before loading of “preview” added. (RN)	10		
	\nofiles makes \makeindex and \makeglossary to \relax . \@empty is better because of later \renewcommand ’s.	6		
v1.1q	General: Problem with “tabularx” and “threeparttable” solved. (RN)	8		
			v1.1r	General: Fixed values for \PreviewBbAdjust because \paperwidth is not allways defined (suggested by Will Robertson).
			v1.1s	General: Dummy definition of the page key in DVI mode.
			v1.1t	General: Remove the line “ \let\output\@gobble ” because of bad side effects. (RN)
				postscript : Using environ the environment postscript is now simple and more robust. (RN) . . .
			v1.1u	General: \pdfoutput must be set when loading “pdf _{tex} .def” in DVI mode. (RN)
			v1.1v	\Gin@ii : Key settings only for pdf graphics. (RN)
				General: Local redefinition of \pdfoutput to be a counter. (RN)
			v1.2a	General: Engine tests changed (RN)
			v1.2b	General: Loading packages ‘if*’ at wrong place (RN)
			v1.2c	General: “postscript” environment no more allowed before \begin{document} (changed example file).
			v1.2d	General: \c@lor@to@ps must not be undefined before loading ‘PSTricks’
				Version parameter for ‘graphicx’ and rename \GPT@page to \Gin@page at several places . . .
			v1.2e	\ppf@is@pdfTeX@graphic : Parameter #2 is detokenized when expanded to \pdfTeXext . .
				General: gobble optional argument for \makeindex , \makeglossary and \printindex

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

\output	212, 218	\ppf@Gin@setfile . .	\PreviewBbAdjust . .
\overfullrule	84	89, 116, 123, 127
P			
\PassOptionsToPackage		\ppf@Gin@include@graphics	\PreviewEnvironment
.	12, 15, 19	134, 138, 141
\PDFcontainer	163, 176, 190, 238
39, 40, 289, 295,		\ppf@is@pdfTeX@graphic	\printindex
310, 312, 318,		80, 83
320, 331, 333, 420		\ppf@isnum	\psmatrix
\pdflastximage	384	156, 158, 159, 375
\pdflastximagepages	313	\ppf@nofiles	psmatrix (environ-
\pdfliteral	191	ment)
\pdfoutput	93, 94	\ppf@other@extensions	pspicture (environ-
\pdfrefximage	396	ment)
\pdfTeXtext	65, 66, 69	pst-pdf-defs (environ-
\pdfTeXversion	92	\ppf@output	ment)
\pdfximage	312	\ppf@pr@psmatrix . .	\pst@@@picture
\pfx@includegraphics		\pst@object
.	379	\ppf@PreviewBbAdjust	\PSTricksOff
\pfx@includegraphicx		
.	176, 380	\ppf@psmatrix	R
\postscript	372, 375	\ppf@RestoreBbAdjust	\raisebox
postscript (environ-		294, 425
ment)	2, 121, 348	\ppf@set@mode	\rule
\ppf@getpicture	
.	233, 368, 380, 423	S
\ppf@container@max .		\ppf@shipout	\savepicture
.	312,	\ppf@temp	\setkeys
316, 318, 328, 415		\shipout
\ppf@draft 3, 13, 14, 419		\ppf@tempb	\string
\ppf@endpsmatrix	\strutbox
.	157, 161	\ppf@tempb	
\ppf@filename	275,	T
280, 304, 305, 395		\ppf@test@mmode	\tabularx
\ppf@getpicture	
.	389, 413, 426	\ppf@TeX@mode	U
\ppf@Gin@extensions	54	\usepicture
\ppf@Gin@ii	55, 299	
\ppf@Gin@keys	277,	V
364, 366, 422, 426		\pr@cleanup	\voidb@x
		\pr@endbox	
		\pr@outerfalse	X
		\pr@startbox	\XKV@err