

The ppr-prv class: Prosper Preview

Mathieu GOUTELLE

v 0.13c, 2004/03/21

Contents

1	Introduction	1
1.1	Presentation	1
1.2	Use of <code>ppr-prv</code>	1
2	Details of the code	2
2.1	Preamble and options	2
2.2	Main code	4
2.3	AtBeginDocument code	11
2.4	Support for HA-prosper	13

1 Introduction

1.1 Presentation

This class is used with LaTeX presentations using the `prosper` class. `ppr-prv` stands for "Prosper Preview". The aim of this class is to produce a printable version of the slides written with `prosper`, with two slides per page.

The result tries to be as accurate as possible. The major problem is the support of the starred version of the `{only,until,from}Slide` macros: the result is approximately what you can expect... It is far from perfect, just because the content of this macros are heaped over each other. But, it is difficult to handle these overlays on a sheet of paper!

You can send comments, bugs report or proposed improvements to the author by email: `mgoutell[at]users[dot]sourceforge[dot]net`.

1.2 Use of `ppr-prv`

The installation of `ppr-prv` files follows the same scheme as any other package installation on your distrib. You can consult this page from the UK TeX FAQ: <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=instpackages>.

To use `ppr-prv`, you just have to change the `\documentclass` from `prosper` to `ppr-prv`. You don't need to change another thing to your `prosper` document: all the macros and options of `prosper` are normally fully supported by

ppr-prv. You can change the footer of each "slide" by changing `\slidename` and `\no`. By default, it is defined as: `\newcommand{\slidename}{slide}` and `\newcommand{\no}{\#}`. Since version 0.07, it uses Babel to support various languages (english, french, german and all their variants): if Babel is loaded, the footer is adapted to the chosen language. You can of course change it manually with `\renewcommand{\slidename}{...}` and `\renewcommand{\no}{...}` after `\begin{document}`.

There is a new option for ppr-prv since v0.11: `nopagebreak`. This can be used to let \LaTeX decide where to break the page instead of every two slides. It may be used for producing a more compact document. The macro `\NoPageBreak` has exactly the same meaning, except that it can be used at any time in the document. ppr-prv also used since v0.11 the `geometry` package to define the layout of the page. You can change it by using `\geometry{...}` (see the documentation of this package for more details).

If you use the patch for Prosper written by Hendri Adriaens (<http://stuwwww.uvt.nl/~hendri/Downloads/haproper.html>), you can use ppr-prv as well since v. 0.09: features like toc, toc section and notes are fully supported. The use is exactly the same if you either load the patch or not. You just have to change your `documentclass` from `prosper` to `ppr-prv` and let the class do the work of redefining the macros for you. Since v. 0.09 is the first release with a support for the patch, it may remain some bugs. Don't hesitate to report them to the author of ppr-prv.

I have tried not to depend on `pstricks` in ppr-prv. It means that, if you don't use `pstricks` in your slide, you can use `pdf \LaTeX` to compile your ppr-prv document, even if the compilation can lead to `Non-PDF special ignored!` warnings. Obviously, it doesn't work if you have some `pstricks` stuff in your slide contents or if you don't provide the figures in a format `pdf \LaTeX` can understand (PDF, PNG, JPG, MetaPost e.g.): in these cases, you still need to use \LaTeX to produce your document.

A common mistake is to use floating environment in slides. It may work with regular prosper document. But with ppr-prv, it may leads to error message such as `Not in outer par mode` or `Float lost`. ppr-prv tries to handle this issue by defining the floats as non-floating. But, if you specify some placement options (like `[hbtpr]`), you will get these errors. By the way, you **really** don't need floats in slides!

2 Details of the code

2.1 Preamble and options

We define first the version and all the stuff we need (or not) to begin

```

1 \*class
2 \def\PPRPRV@Version{v. 0.13}
3 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
4 \ProvidesClass{ppr-prv}[2005/09/02, \PPRPRV@Version]

```

```

5 \typeout{(c) 2003-2005 Mathieu Goutelle}
6 \end{class}

```

Then we define the prosper options to handle the one useful in our case (e.g. draft, footer) and ignore the others. There is also the declaration of the nopagebreak option specific to ppr-prv.

```

7 \end{class}
8 \newif\ifDVIToPS
9 \DeclareOption{ps}{\global\DVIToPStrue}
10 \DeclareOption{pdf}{\global\DVIToPSfalse}
11
12 \newif\ifisDraft
13 \DeclareOption{draft}{\global\isDrafttrue}
14 \DeclareOption{final}{\global\isDraftfalse}
15
16 \newif\ifinColor
17 \DeclareOption{slideColor}{}
18 \DeclareOption{slideBW}{}
19
20 \newif\ifallPages
21 \DeclareOption{total}{}
22 \DeclareOption{nototal}{}
23
24 \newif\ifcolorBG
25 \DeclareOption{nocolorBG}{}
26 \DeclareOption{colorBG}{}
27
28 \newif\ifAccumulate
29 \DeclareOption{accumulate}{}
30 \DeclareOption{noaccumulate}{}
31
32 \newif\ifnoFooter
33 \DeclareOption{noFooter}{\global\noFootertrue}
34 \noFooterfalse
35
36 \newif\if@distiller
37 \@distillerfalse
38 \DeclareOption{distiller}{\global\@distillertrue}
39
40 \newif\if@nopagebreak
41 \@nopagebreakfalse
42 \DeclareOption{nopagebreak}{\global\@nopagebreaktrue}
43
44 \DeclareOption{YandY}{\gdef\@pdf@driver{dvipsone}}
45 \DeclareOption{ps2pdf}{\gdef\@pdf@driver{ps2pdf}}
46 \DeclareOption{vtex}{\gdef\@pdf@driver{vtex}}
47
48 %% Default style file = default
49 \def\Style@chosen{default}
50 \DeclareOption*{\global\let\Style@chosen=\CurrentOption}

```

```

51 </class>
    Here we pass all options a user want to add to specify the paper and font sizes
    to the article class and geometry package.
52 <*class>
53 \DeclareOption{a4paper}{%
54   \PassOptionsToClass{\CurrentOption}{article}%
55   \PassOptionsToPackage{\CurrentOption}{geometry}%
56 }
57 \DeclareOption{a5paper}{%
58   \PassOptionsToClass{\CurrentOption}{article}%
59   \PassOptionsToPackage{\CurrentOption}{geometry}%
60 }
61 \DeclareOption{b5paper}{%
62   \PassOptionsToClass{\CurrentOption}{article}%
63   \PassOptionsToPackage{\CurrentOption}{geometry}%
64 }
65 \DeclareOption{letterpaper}{%
66   \PassOptionsToClass{\CurrentOption}{article}%
67   \PassOptionsToPackage{\CurrentOption}{geometry}%
68 }
69 \DeclareOption{legalpaper}{%
70   \PassOptionsToClass{\CurrentOption}{article}%
71   \PassOptionsToPackage{\CurrentOption}{geometry}%
72 }
73 \DeclareOption{executivepaper}{%
74   \PassOptionsToClass{\CurrentOption}{article}%
75   \PassOptionsToPackage{\CurrentOption}{geometry}%
76 }
77 \DeclareOption{landscape}{%
78   \PassOptionsToClass{\CurrentOption}{article}%
79   \PassOptionsToPackage{\CurrentOption}{geometry}%
80 }
81
82 \PassOptionsToClass{10pt,11pt,12pt,oneside,twoside,final,draft,
83   leqno,fleqn,openbib}{article}
84 </class>

```

Then, we define a default behaviour, which is exactly the same as the prosper one, except we add options for article (a4paper and 10pt).

```

85 <*class>
86 \ExecuteOptions{final,slideBW,total,nocolorBG,noaccumulate,ps,ps2pdf,a4paper,10pt}
87 \ProcessOptions\relax
88 </class>

```

2.2 Main code

As a first action, we load the article class and save some macros for further use.

```

89 <*class>
90 %% Preliminary stuff

```

```

91 \LoadClass{article}
92
93 %% I save here the old value of the macros I need and which may be
94 %% modified by HA-prosper and define a test for detecting HA-prosper.
95 \newif\if@HAprosper\global\@HAprosperfalse
96 \let\old@section\section
97 \let\old@subsection\subsection
98 \let\old@itemize\itemize
99 \let\old@enditemize\enditemize
100 \let\old@enumerate\enumerate
101 \let\old@endenumerate\endenumerate
102 \</class>

```

We then load `pstricks`, `graphicx` and `hyperref` to exactly mimic the `prosper` environment and some others for internal use (`textcomp` and `geometry`). The `keyval` package is for HA-prosper support.

```

103 \<class>
104 \RequirePackage{pstricks}
105 \RequirePackage{textcomp}
106 \RequirePackage{graphicx}
107 \RequirePackage[hmargin=2cm,vmargin=2cm,noheadfoot]{geometry}
108 \RequirePackage{keyval,ifpdf}
109 \ifpdf\gdef\@pdf@driver{pdftex}\fi
110 \RequirePackage[\@pdf@driver,plainpages=false,colorlinks=true]{hyperref}
111 \</class>

```

Here we want to suppress the common error of using floats in slides: we redefine them as non-floats.

```

112 \<class>
113 \RequirePackage{float}
114 \newfloat{figure}{H}{lof}
115 \newfloat{table}{H}{lot}
116 \</class>

```

`NoPageBreak` The `\NoPageBreak` macro has the same meaning as the `nopagebreak` option, except it can be used at any time in the document. To revert the effect of `\NoPageBreak`, the `\PageBreak` macro is defined. We define also the `indent` to 0pt and we don't want folio nor head/foot.

```

117 \<class>
118 \newcommand{\NoPageBreak}{\global\@nopagebreaktrue}
119 \newcommand{\PageBreak}{\global\@nopagebreakfalse}
120
121 \setlength{\parindent}{0pt}
122 \pagestyle{empty}
123 \</class>

```

For the support for the HA-prosper package, we need to define some `prosper` internals to avoid the HA patch to complain about a missing length or macro.

```

124 \<class>
125 %% Support for the HA-prosper package: we try here to mimic the prosper
126 %% class to avoid the HA patch to complain about a missing length or

```

```

127 %% macro.
128 \let\normal@write\write
129 \newlength{\slideWidth}
130 \setlength{\slideWidth}{\linewidth}
131 \newlength{\slidewidth}
132 \setlength{\slidewidth}{\linewidth}
133 \newlength{\slideheight}
134 \newcounter{trueSlideCounter}
135 \newcounter{overlaysCount}
136 \newcounter{item@step}
137 \newcommand{\slidetitle}[1]{}
138 \newcommand{\slideframe}[1]{}
139 \newcommand{\NewSlideStyle}[4] [] {}
140 \newcommand{\newslideframe}[2]{}
141 \newcommand{\slidepagestyle}[1]{}
142 \def\@fontTextColor{}
143 \def\@fontTextBW{}
144 \def\@fontTitleColor{}
145 \def\@fontTitleBW{}
146 \def\@fontSubtitleColor{}
147 \def\@fontSubtitleBW{}
148 \def\ptsize#1{}
149 \newenvironment{slide@seminar}{}{}
150 \def\@DefMyItem{}
151 \def\@titleSpacing{\vspace*{10pt}}
152 \newif\ifInOverlays\InOverlaysfalse
153 \newcounter{limitOverlays}
154 \def\@cleanupOverlay{}
155 \end{class}

```

Here are some seminar internal we need also for HA-prosper.

```

156 \end{class}
157 %% Commands from Seminar
158 \newcounter{slide}
159 \newcommand{\renewpagestyle}[3]{}
160 \def\slidefonts{}
161 \newif\ifarticle\articletrue
162 \end{class}

```

This test will be used to decide if we need to insert a page break before printing the current slide.

```

163 \end{class}
164 % Test to see if we need a pageskip
165 \newif\if@needsnewpage\global\@needsnewpagetrue
166 \end{class}

```

part We redefine the `\part` command first to suppress the optional transition argument and second to handle pagebreak and slides counter.

```

167 \end{class}
168 %% Definition of the part command to suppress the optional transition argument
169 \let\part\relax

```

```

170 \newcommand{\part}[2] [] {%
171   \if@needsnewpage
172     \if@nopagebreak\else\ifodd\theslideii\else\mbox{}\clearpage\fi\fi
173     \global\@needsnewpagefalse
174   \fi
175   \refstepcounter{slidei}%
176   \old@section*{#2}%
177 }
178 \end{class}

maketitle   The \maketitle command is completely redefined since the article one isn't
            relevant: prosper define some extra field we need to handle here.

179 \begin{class}
180 %% Handle all the maketitle stuff
181 \let\@cartouche\@empty
182 \newcommand{\slideCaption}[1]{\gdef\@cartouche{#1}}
183 \let\@Subtitle\@empty
184 \newcommand{\subtitle}[1]{\gdef\@Subtitle{#1}}
185 \let\old@Title\title
186 \renewcommand{\title}[1]{\gdef\@Title{#1}}
187
188 \let\@email\@empty
189 \newcommand{\email}[1]{\gdef\@email{#1}}
190 \let\@institution\@empty
191 \newcommand{\institution}[1]{\gdef\@institution{#1}}
192
193 \gdef\@Author{}
194 \renewcommand{\author}[1]{\gdef\@Author{#1}}
195
196 \renewcommand{\maketitle}{
197   \thispagestyle{empty}\setcounter{page}{0}%
198   \refstepcounter{slidei}%
199   \mbox{}\vfill
200   \begin{center}%
201     {\LARGE \@Title \par}%
202     \ifx\@Subtitle\@empty\else\vskip 2em{\large\@Subtitle\par}\fi
203     \vskip 6em%
204     {\large\@Author
205     \ifx\@email\@empty\else\vskip 1em\@email\fi
206     \ifx\@institution\@empty\else\vskip 1em\@institution\par\fi}%
207   \end{center}%
208   \vfill\mbox{}\clearpage
209   \global\@needsnewpagefalse
210 }
211 \end{class}

Itemize     We redefine the Itemize and itemstep. They have the same behaviour as
itemstep    the original itemize of article. We have to ignore the optional argument of
            itemstep.

212 \begin{class}

```

```

213 %% Definition of the Itemize environment ;)
214 \let\Itemize\itemize
215 \let\endItemize\enditemize
216
217 %% Definition of the itemstep environment to gobble the optional argument.
218 \newenvironment{itemstep}[1] []%
219   {\itemize}
220   {\enditemize}
221 \end{class}

```

slide We define here the `slide` environment. We first define `prv@slide` because we will need it later on, for the support of HA-prosper.

`\@size@overlay` will contain the height of the current slide and will be used to correctly draw the frame around slides. This height is computed as the maximum real height of the different overlays (especially those defined with a `Slide*` macros) of the slide and is equal at least to 4cm, to avoid a really little frame when the user wants exactly two slides per page. The `\@slide@content` box contains the content of the slide. The first counter is for the number printed in footer, i.e. exactly the same as the one in `prosper`. The second is used internally to print exactly two slides per page (without counting part for example).

```

222 \begin{class}
223 %% \@size@overlay contains the height of the current slide if we are using
224 %% overlays. It is equal at least to 4cm to try to avoid really little frame
225 %% when the user wants exactly two slides per page.
226 \newlength{\@size@tmp}
227 \global\@size@tmp\z@
228 \newlength{\@size@overlay}
229 %% Definition of the slide environment
230 \newsavebox{\@slide@content} % will contain the slide
231 %% Slide counter to be printed on the bottom of each slide
232 \newcounter{slidei}\setcounter{slidei}{0}
233 %% Slide counter to print just two slides per page (even when there is a
234 %% \part...)
235 \newcounter{slideii}\setcounter{slideii}{0}
236
237 \newenvironment{prv@slide}[1]{%
238   \if@needsnewpage
239     \if@nopagebreak\else\ifodd\theslideii\mbox{}\clearpage\fi\fi
240   \fi
241   \global\@needsnewpagetrue
242   \if@nopagebreak
243     \global\setlength{\@size@overlay}{\z@}%
244   \else
245     \global\setlength{\@size@overlay}{4cm}%
246   \fi
247   \begin{lrbox}{\@slide@content}\begin{minipage}{\linewidth}%
248     \old@subsection*{#1}%
249   }\fi
250   \end{minipage}\end{lrbox}%

```

```

251 \settoheight{\@size@tmp}{\@slide@content}%
252 \begin{minipage}{\linewidth}
253 \fbox{%
254   \usebox{\@slide@content}%
255   \ifdim\@size@overlay>\@size@tmp
256     \addtolength{\@size@overlay}{\baselineskip}%
257     \rule[-\@size@overlay]{1pt}{0pt}%
258   \fi
259 }%
260 \ifnoFooter\else
261   \ifhmode\\\fi
262   \makebox[\linewidth][r]{%
263     \footnotesize
264     \ifx\@cartouche\@empty\else\@cartouche~---~\fi
265     \slidename~\no\theslidei}%
266   \fi
267 \end{minipage}
268 \par\if@nopcodebreak\bigskip\else\vfill\fi
269 }
270 \newenvironment{slide}[2]{}
271 {}
272 \refstepcounter{slidei}\stepcounter{slideii}%
273 \begin{prv@slide}{#2}%
274 }
275 {\end{prv@slide}}
276 \end{class}

```

At the end of the document, we print an invisible box to attach the `\vfill` and be sure the last slide doesn't flush to the bottom of the page.

```

277 \end{class}
278 \AtEndDocument{\if@nopcodebreak\else\ifodd\theslideii\vfill\mbox{} \else\fi\fi}
279 \end{class}

```

Here, we ignore a lot of prosper macros.

```

280 \end{class}
281 %% For the \Logo, don't do anything !
282 \newcommand{\Logo}{%
283   \@ifnextchar({\Logo@one}{\Logo@two}%
284 }
285 \def\Logo@one(#1)#2{}
286 \def\Logo@two#1{}
287
288 %% For the \Font*, don't do anything !
289 \newcommand{\FontTitle}[2]{}
290 \newcommand{\FontSubtitle}[2]{}
291 \newcommand{\FontText}[2]{}
292 \end{class}

```

In the following code, we handle the overlays. The tricky part is the definition of the starred version of `{only,until,from}Slide` macros. For these, we need to measure the content of the material before smashing it in a `Opt` height box. We

put this height in the \@size@overlay length we use previously in prv@slide to draw a frame with the correct height.

```

293 <*class>
294 %% Definition of the whole overlay stuff: a bit annoying...
295 \newcommand{\overlays}[2]{#2} % We ignore overlays
296
297 \def\FromSlide#1{\relax}
298 \def\UntilSlide#1{\relax}
299 \def\OnlySlide#1{\relax}
300
301 \long\def\fromSlide@NE#1#2{#2}
302 \long\def\untilSlide@NE#1#2{#2}
303 \long\def\onlySlide@NE#1#2{#2}
304
305 \newbox\PRV@tempbox
306 \long\def\fromSlide@E#1#2{%
307   \sbox\PRV@tempbox{\begin{minipage}{\linewidth}#2\end{minipage}}%
308   \setlength{\@size@tmp}{\the\ht\PRV@tempbox}%
309   \addtolength{\@size@tmp}{\the\dp\PRV@tempbox}%
310   \ifdim\@size@tmp>\@size@overlay\global\@size@overlay\@size@tmp\fi
311   \vbox to Opt\bgroup\usebox{\PRV@tempbox}\egroup
312   \vskip -\baselineskip\ignorespaces
313 }
314 \long\def\untilSlide@E#1#2{%
315   \sbox\PRV@tempbox{\begin{minipage}{\linewidth}#2\end{minipage}}%
316   \setlength{\@size@tmp}{\the\ht\PRV@tempbox}%
317   \addtolength{\@size@tmp}{\the\dp\PRV@tempbox}%
318   \ifdim\@size@tmp>\@size@overlay\global\@size@overlay\@size@tmp\fi
319   \vbox to Opt\bgroup\usebox{\PRV@tempbox}\egroup
320   \vskip -\baselineskip\ignorespaces
321 }
322 \long\def\onlySlide@E#1#2{%
323   \sbox\PRV@tempbox{\begin{minipage}{\linewidth}#2\end{minipage}}%
324   \setlength{\@size@tmp}{\the\ht\PRV@tempbox}%
325   \addtolength{\@size@tmp}{\the\dp\PRV@tempbox}%
326   \ifdim\@size@tmp>\@size@overlay\global\@size@overlay\@size@tmp\fi
327   \vbox to Opt\bgroup\usebox{\PRV@tempbox}\egroup
328   \vskip -\baselineskip\ignorespaces
329 }
330
331 \def\fromSlide{\@ifstar{\fromSlide@E}{\fromSlide@NE}}
332 \def\untilSlide{\@ifstar{\untilSlide@E}{\untilSlide@NE}}
333 \def\onlySlide{\@ifstar{\onlySlide@E}{\onlySlide@NE}}
334 </class>

```

We define some prosper macros we don't really need in ppr-prv. But we obviously need to define it...

```

335 <*class>
336 %% All the things I may have forgotten if I haven't read the docs
337 %% carefully enough

```

```

338 \newcommand{\PDFtransition}[1]{}
339 \newcommand{\DefaultTransition}[1]{}
340 \def\NoFrenchBabelItemize{\relax}
341 \newcommand{\ColorFoot}[1]{}
342 \newif\ifshowVersion
343 \showVersionfalse
344 \newcommand{\displayVersion}{\global\showVersiontrue}
345
346 \def\fontTitle#1{#1}
347 \def\fontText#1{#1}
348
349 \newcommand{\myitem}[2]{}
350
351 %% Stolen from the prosper.cls code
352 \newcommand{\PDFforPS}{%
353   \ifDVItoPS
354     \let\prosper@next\@secondoftwo
355   \else
356     \let\prosper@next\@firstoftwo
357   \fi
358   \prosper@next}
359 %% #1 appears on the slide only on PS mode
360 \newcommand{\onlyInPS}{%
361   \ifDVItoPS
362     \let\prosper@next\@iden
363   \else
364     \let\prosper@next\@gobble
365   \fi
366   \prosper@next}
367 %% #1 appears on the slide only on PDF mode
368 \newcommand{\onlyInPDF}{%
369   \ifDVItoPS
370     \let\prosper@next\@gobble
371   \else
372     \let\prosper@next\@iden
373   \fi
374   \prosper@next}
375 \end{class}

```

2.3 AtBeginDocument code

Finally, we add some `\AtBeginDocument` hackery. The purpose is first to support multi-language with Babel and we need to know if this package is loaded.

```

376 \begin{class}
377 \AtBeginDocument{% Multi-language support with Babel
378 \end{class}

```

If we detect Babel, we test the loaded language and define the footer macros according to this.

```

379 <*class>
380 \ifpackageloaded{babel}{%
381   \def\@test{french}%
382   \ifx\language\@test
383     \newcommand{\slidename}{transparent}%
384     \providecommand{\no}{n\textdegree}%
385   \fi
386   \def\@test{english}%
387   \ifx\language\@test
388     \newcommand{\slidename}{slide}%
389     \providecommand{\no}{\#}%
390   \fi
391   \def\@test{british}%
392   \ifx\language\@test
393     \newcommand{\slidename}{slide}%
394     \providecommand{\no}{\#}%
395   \fi
396   \def\@test{canadian}%
397   \ifx\language\@test
398     \newcommand{\slidename}{slide}%
399     \providecommand{\no}{\#}%
400   \fi
401   \def\@test{USenglish}%
402   \ifx\language\@test
403     \newcommand{\slidename}{slide}%
404     \providecommand{\no}{\#}%
405   \fi
406   \def\@test{UKenglish}%
407   \ifx\language\@test
408     \newcommand{\slidename}{slide}%
409     \providecommand{\no}{\#}%
410   \fi
411   \def\@test{german}%
412   \ifx\language\@test
413     \newcommand{\slidename}{transparent}%
414     \providecommand{\no}{Nr.}%
415   \fi
416   \def\@test{germanb}%
417   \ifx\language\@test
418     \newcommand{\slidename}{transparent}%
419     \providecommand{\no}{Nr.}%
420   \fi
421   \def\@test{austrian}%
422   \ifx\language\@test
423     \newcommand{\slidename}{transparent}%
424     \providecommand{\no}{Nr.}%
425   \fi
426   \def\@test{ngerman}%
427   \ifx\language\@test
428     \newcommand{\slidename}{transparent}%

```

```

429     \providecommand{\no}{Nr.}%
430     \fi
431     \def\@test{naustrian}%
432     \ifx\language\@test
433         \newcommand{\slidename}{transparent}%
434         \providecommand{\no}{Nr.}%
435     \fi
436 }{}
437 \providecommand{\slidename}{slide}%
438 \providecommand{\no}{\#}%
439 }
440 \end{class}

```

2.4 Support for HA-prosper

If we detect HA-prosper, we need to redefine all the macros added by this patch. All this stuff goes into the HAP-ppr-prv.def file, which is loaded at the end of HA-prosper if it detects ppr-prv.

```

441 \begin{HAPdef}
442 %% Support for the HA-prosper package: redefinition of the
443 %% functionalities added by this patch.
444 \ifpackage\loaded{HA-prosper}{%
445     \global\@HAProspertrue
446     % \def\HAP@ReqVersion{v3.9}
447     % \ifx\HAP@Version\undefined\else%
448     % \ifx\HAP@ReqVersion\HAP@Version\else%
449     % \ClassError{ppr-prv}{HA-prosper \HAP@Version\space and
450     % ppr-prv \PPRPRV@Version\space are incompatible}%
451     % {Read the documentation of HA-prosper.}%
452     % \fi%
453     % \fi
454
455     \renewenvironment{itemstep}[1][\old@itemize]{\old@enditemize}%
456     \renewenvironment{enumstep}[1][\old@enumerate]{\old@endenumerate}%
457
458     \define@key{PRV@keys@slide}{toc}{\def\@tockey{#1}}
459     \define@key{PRV@keys@slide}{bm}{}
460     \define@key{PRV@keys@slide}{trans}{}
461     \define@key{PRV@keys@slide}{sound}{}
462     \define@key{PRV@keys@slide}{template}{}
463     \def\xitem{\@ifnextchar<{\PRV@xitem}{\PRV@xitem<>}}
464     \def\PRV@xitem<#1>{\item}
465     \def\email#1{#1}
466     \def\institution#1{#1}
467
468     \renewcommand{\NewSlideStyle}[4][{}]{
469 \end{HAPdef}

```

`slide` We first redefine the `slide` environment to first handle the optional keys for

defining toc entries. But it is roughly `prv@slide...` The `wideslide` environment is set to `slide`.

There is also some stuff here in case HA-prosper is loaded: since this package change `itemize`-like environments `AtBeginDocument`, we need to redefine it after... and after `AtBeginDocument` means just before printing the slides. We do this only in the first slide by testing the value of `\itemize`.

```

470 ⟨*HAPdef⟩
471 \renewenvironment{slide}[2][ ]{%
472   \refstepcounter{slidei}\stepcounter{slideii}%
473   \setcounter{trueSlideCounter}{\theslidei}%
474   \let\tockey\@undefined
475   \setkeys{PRV@keys@slide}{#1}%
476   \def\@tempa{#2}%
477   \ifx\@tempa\@empty
478     \ifx\@tockey\@undefined\else
479       \ifx\@tockey\@empty\else
480         \HAP@contentsline{0}{\@tockey}{\theslidei}{}%
481       \fi
482     \fi
483   \else
484     \ifx\@tockey\@undefined
485       \HAP@contentsline{0}{#2}{\theslidei}{}%
486     \else
487       \ifx\@tockey\@empty\else
488         \HAP@contentsline{0}{\@tockey}{\theslidei}{}%
489       \fi
490     \fi
491   \fi
492   \ifx\itemize\old@itemize\else
493     \global\let\itemize\old@itemize
494     \global\let\enditemize\old@enditemize
495     \global\let\enumerate\old@enumerate
496     \global\let\endenumerate\old@endenumerate
497   \fi
498   \begin{prv@slide}{#2}%
499 }
500 {%
501   \end{prv@slide}%
502 }
503
504 \renewenvironment{wideslide}[2][ ]{\begin{slide}[#1]{#2}}{\end{slide}}
505 ⟨/HAPdef⟩

```

`dualslide` The `dualslide` environment is a bit more complicated to redefine, just because we want to get rid of `pstricks`. So we do everything with \LaTeX boxes and some spaces adjustments.

```

506 ⟨*HAPdef⟩
507 \newsavebox{\@slide@left}%
508 \newsavebox{\@slide@right}%

```

```

509 \newlength{\ht@slide}%
510
511 \def\dualslide{\@ifnextchar[\@dualslide{\@@dualslide [] [] []}}%
512 \def\@dualslide[#1]{%
513   \@ifnextchar[{\@@dualslide[#1]}{\@@@dualslide[#1] [#1] [#1]}}%
514 \def\@@dualslide[#1] [#2]{%
515   \@ifnextchar[{\@@@dualslide[#1] [#2]}{\@@@@dualslide[#1] [#2] [#1]}}%
516 \long\def\@@@@dualslide[#1] [#2] [#3] #4#5#6{%
517   \setlength{\HAP@colsep}{.04\linewidth}%
518   \setlength{\HAP@lcolwidth}{.47\linewidth}%
519   \setlength{\HAP@rcolwidth}{.47\linewidth}%
520   \setlength{\HAP@topsep}{0cm}%
521   \setlength{\HAP@bottomsep}{.3cm}%
522   \setlength{\HAP@indent}{0cm}%
523   \setkeys{HAP@keys@dual}{#4}%
524
525   \sbox{\@slide@left}{\begin{minipage}[t]{\HAP@lcolwidth}#5\end{minipage}}%
526   \sbox{\@slide@right}{\begin{minipage}[t]{\HAP@rcolwidth}#6\end{minipage}}%
527
528   \setlength{\ht@slide}{\the\ht\@slide@left}%
529   \addtolength{\ht@slide}{-0.5\baselineskip}%
530   \addtolength{\ht@slide}{\HAP@topsep}%
531   \leavevmode\kern\the\HAP@indent
532   \raisebox{-\ht@slide}{\usebox{\@slide@left}}%
533   \kern\the\HAP@colsep
534   \setlength{\ht@slide}{\the\ht\@slide@right}%
535   \addtolength{\ht@slide}{-0.5\baselineskip}%
536   \addtolength{\ht@slide}{\HAP@topsep}%
537   \raisebox{-\ht@slide}{\usebox{\@slide@right}}%
538   \par\vspace*{\HAP@bottomsep}%
539 }
540 \end{HAPdef}

```

We ignore again overlays...

```

541 \begin{HAPdef}
542 \renewcommand{\overlays}[2]{#2} % We ignore overlays (again...)
543 \end{HAPdef}

```

maketitle The redefinition of `\maketitle` is exactly the same as the previous one, except that we add the support for the optional toc entry definition.

```

544 \begin{HAPdef}
545 \renewcommand{\maketitle}[1] [] {%
546   \refstepcounter{slidei}%
547   \setcounter{trueSlideCounter}{\theslidei}%
548   \let\@tockey\@undefined
549   \setkeys{PRV@keys@slide}{#1}%
550   \ifx\@tockey\@undefined
551   \else
552     \ifx\@tockey\@empty\else
553       \HAP@contentsline{0}{\@tockey}{\theslidei}{}%
554     \fi

```

```

555 \fi
556 \thispagestyle{empty}\setcounter{page}{0}%
557 \mbox{}\vfill
558 \begin{center}%
559 {\LARGE \@title \par}%
560 \ifx\@Subtitle\@empty\else\vskip 2em{\large\@Subtitle\par}\fi
561 \vskip 6em%
562 {\large\begin{tabular}[t]{c}\@author\end{tabular}\par}%
563 \end{center}%
564 \vfill\HAP@toc
565 \vfill\mbox{}\clearpage
566 \global\@needsnewpagefalse
567 }
568 \HAPdef

```

pcontentsline Here is defined the \pcontentsline macro for defining the toc layout printed on the first page.

```

569 (*HAPdef)
570 \renewcommand{\fontToc}{}
571 \newlength{\PRV@indenttoc}
572 \setlength{\PRV@indenttoc}{0pt}
573 \renewcommand{\pcontentsline}[5]{%
574 \if1#1%
575 \setlength{\PRV@indenttoc}{0pt}%
576 \fi
577 \hspace*{\PRV@indenttoc}#2%
578 \if1#5\else\leaders\hbox{\hss.\ \hss}\hfill\slidename~\no#3\fi\\
579 \if1#1%
580 \setlength{\PRV@indenttoc}{10pt}%
581 \fi
582 }%
583 \HAPdef

```

All the sectioning commands of HA-prosper are redefined here. We only handle the toc entry, so it is much less complicated than in HA-prosper!

```

584 (*HAPdef)
585 % Redefinition of tsection command to remove what is not needed
586 \newcounter{MG@tempTrueSlideCounter}
587 \renewcommand{\tsection}{%
588 \ifstar
589 {\@ifnextchar[{\@tsection{0}{2}}{\@tsection{0}{2}[]}}%
590 {\@ifnextchar[{\@tsection{0}{1}}{\@tsection{0}{1}[]}}%
591 }%
592 \def\@tsection#1#2[#3]#4{%
593 \setcounter{MG@tempTrueSlideCounter}{\thetrueSlideCounter}%
594 \stepcounter{MG@tempTrueSlideCounter}%
595 \HAP@contentsline{1}{#4}{\theMG@tempTrueSlideCounter}{1}%
596 }%
597
598 % Redefinition of the part command to deal with the toc key

```

```

599 \renewcommand{\part}[2] [] {%
600   \if@needsnewpage
601     \if@nopagebreak\else\ifodd\theslidei\else\mbox{}\clearpage\fi\fi
602     \global\@needsnewpagefalse
603   \fi
604   \refstepcounter{slidei}%
605   \setcounter{trueSlideCounter}{\theslidei}%
606   \let\@tockey\@undefined
607   \setkeys{PRV@keys@slide}{#1}%
608   \ifx\@tockey\@undefined
609     \HAP@contentsline{0}{#2}{\theslidei}{}%
610   \else
611     \ifx\@tockey\@empty\else
612       \HAP@contentsline{0}{\@tockey}{\theslidei}{}%
613     \fi
614   \fi
615   \old@section*{#2}%
616 }%
617 \renewcommand{\tsectionandpart}{\@ifstar{\tsectionandpart@E}{\tsectionandpart@NE}}
618 \newcommand{\tsectionandpart@E}[2] [] {%
619   \let\@tockey\@undefined
620   \setkeys{PRV@keys@slide}{#1}%
621   \ifcase
622     \ifx\@tockey\@undefined\@ne\fi
623     \ifx\@tockey\@empty\@ne\fi
624     \z@
625     \tsection*\{\@tockey}%
626   \else
627     \tsection*{#2}%
628   \fi
629   \part[toc={}] {#2}%
630 }
631 \newcommand{\tsectionandpart@NE}[2] [] {%
632   \let\@tockey\@undefined
633   \setkeys{PRV@keys@slide}{#1}%
634   \ifcase
635     \ifx\@tockey\@undefined\@ne\fi
636     \ifx\@tockey\@empty\@ne\fi
637     \z@
638     \tsection{\@tockey}%
639   \else
640     \tsection{#2}%
641   \fi
642   \part[toc={}] {#2}%
643 }
644 \
```

Since HA-prosper enhances the footer (left and right), we try to take advantages of this.

```

645 \
```

```

646 \let\l@foot\@empty
647 \let\r@foot\@empty
648 \ifundefined{HAP@leftfoot}{\gdef\l@foot{\HAP@leftfoot}}
649 \ifundefined{HAP@rightfoot}{\gdef\r@foot{\HAP@rightfoot}}
650 \gdef\@cartouche{\l@foot\hfill\r@foot}
651 \renewcommand{\LeftFoot}[1]{%
652   \gdef\l@foot{#1}%
653   \gdef\@cartouche{\l@foot\hfill\r@foot}%
654 }%
655 \renewcommand{\RightFoot}[1]{%
656   \gdef\r@foot{#1}%
657   \gdef\@cartouche{\l@foot\hfill\r@foot}%
658 }%
659 \end{HAPdef}

```

`notes` Again, HA-prosper adds a `notes` environment. We print it just between the slides, in a quote environment.

```

660 \begin{HAPdef}
661   \ifHAP@slidesonly\else
662     \renewenvironment{notes}[1]{%
663       \old@subsection*{#1}\bgroup\quote\footnotesize\sffamily
664     }{%
665       \endquote\egroup
666     \ifHAP@notesonly\else
667       \par
668       \if@nopcodebreak
669         \bigskip
670       \else
671         \vfill
672         \ifodd\theslideii\else\mbox{} \clearpage\fi
673       \fi
674     \fi
675     \global\@needsnewpagefalse
676   }%
677 \fi
678 \end{HAPdef}

```

`OnSlide` HA-prosper adds also two extra macros for overlays: `\onSlide` and `\OnSlide`.
`onSlide` To handle it is relatively simple since we only have to “flatten” overlays (see `\onlySlide` definition above).

```

679 \begin{HAPdef}
680   \def\OnSlide#1{\relax}
681   \long\def\onSlide@NE#1#2{#2}
682   \long\def\onSlide@E#1#2{%
683     \sbox\PRV@tempbox{\begin{minipage}{\linewidth}#2\end{minipage}}%
684     \setlength{\@size@tmp}{\the\ht\PRV@tempbox}%
685     \addtolength{\@size@tmp}{\the\dp\PRV@tempbox}%
686     \ifdim\@size@tmp>\@size@overlay\global\@size@overlay\@size@tmp\fi
687     \vbox to 0pt\bgroup\usebox{\PRV@tempbox}\egroup
688     \vskip -\baselineskip\ignorespaces

```

```

689 }
690 \def\onSlide{\@ifstar\onSlide@E\onSlide@NE}
691 }%
692 {%
693 \global\HAProsperfalse%
694 }
695 \HAPdef

```

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	
\#	389, 394, 399, 404, 409, 438
\@@dualslide	511, 513, 515, 516
\@dualslide	513, 514
\@Author	193, 194, 204
\@DefMyItem	150
\@HAProsperfalse	95, 693
\@HAProspertrue	445
\@Subtitle	183, 184, 202, 560
\@Title	186, 201
\@author	562
\@cartouche	181, 182, 264, 650, 653, 657
\@cleanupOverlay	154
\@distillerfalse	37
\@distillertrue	38
\@dualslide	511, 512
\@email	188, 189, 205
\@empty	181, 183, 188, 190, 202, 205, 206, 264, 477, 479, 487, 552, 560, 611, 623, 636, 646, 647
\@firstoftwo	356
\@fontSubtitleBW	147
\@fontSubtitleColor	146
\@fontTextBW	143
\@fontTextColor	142
\@fontTitleBW	145
\@fontTitleColor	144
\@gobble	364, 370
\@iden	362, 372
\@ifnextchar	283, 463, 511, 513, 515, 589, 590
\@ifpackageloaded	380, 444
\@ifstar	331–333, 588, 617, 690
\@ifundefined	648, 649
\@institution	190, 191, 206
\@ne	622, 623, 635, 636
\@needsnewpagefalse	173, 209, 566, 602, 675
\@needsnewpagetrue	165, 241
\@nopagebreakfalse	41, 119
\@nopagebreaktrue	42, 118
\@pdf@driver	44–46, 109, 110
\@secondoftwo	354
\@size@overlay	223, 228, 243, 245, 255–257, 310, 318, 326, 686
\@size@tmp	226, 227, 251, 255, 308– 310, 316–318, 324–326, 684–686
\@slide@content	230, 247, 251, 254
\@slide@left	507, 525, 528, 532
\@slide@right	508, 526, 534, 537
\@tempa	476, 477
\@test	381, 382, 386, 387, 391, 392, 396, 397, 401, 402, 406, 407, 411, 412, 416, 417, 421, 422, 426, 427, 431, 432
\@title	559
\@titleSpacing	151
\@tockey	458, 474, 478–480, 484, 487, 488, 548, 550, 552, 553, 606, 608, 611, 612, 619, 622, 623, 625, 632, 635, 636, 638
\@tsection	589, 590, 592
\@undefined	474, 478, 484, 548, 550, 606, 608, 619, 622, 632, 635
\@	261, 578
_	578

A	
<code>\addtolength</code>	256, 309, 317, 325, 529, 530, 535, 536, 685
<code>\articlettrue</code>	161
<code>\AtBeginDocument</code>	377
<code>\AtEndDocument</code>	278
<code>\author</code>	194
B	
<code>\baselineskip</code> 256, 312, 320, 328, 529, 535, 688
<code>\begin</code>	200, 247, 252, 273, 307, 315, 323, 498, 504, 525, 526, 558, 562, 683
<code>\bgroup</code>	311, 319, 327, 663, 687
<code>\bigskip</code>	268, 669
C	
<code>\ClassError</code>	449
<code>\clearpage</code>	172, 208, 239, 565, 601, 672
<code>\ColorFoot</code>	341
<code>\CurrentOption</code>	50, 54, 55, 58, 59, 62, 63, 66, 67, 70, 71, 74, 75, 78, 79
D	
<code>\DeclareOption</code>	9, 10, 13, 14, 17, 18, 21, 22, 25, 26, 29, 30, 33, 38, 42, 44–46, 50, 53, 57, 61, 65, 69, 73, 77
<code>\def</code>	2, 49, 142– 148, 150, 151, 154, 160, 285, 286, 297–299, 301–303, 306, 314, 322, 331–333, 340, 346, 347, 381, 386, 391, 396, 401, 406, 411, 416, 421, 426, 431, 446, 458, 463–466, 476, 511, 512, 514, 516, 592, 680–682, 690
<code>\DefaultTransition</code>	339
<code>\define@key</code>	458–462
<code>\displayVersion</code>	344
<code>\dp</code>	309, 317, 325, 685
<code>\dualslide</code>	511
<code>dualslide</code> (environment)	14
<code>\DVIToPSfalse</code>	10
<code>\DVIToPStrue</code>	9
E	
<code>\egroup</code>	311, 319, 327, 665, 687
<code>\else</code>	172, 202, 205, 206, 239, 244, 260, 264, 268, 278, 355, 363, 371, 447, 448, 478, 479, 483, 486, 487, 492, 551, 552, 560, 578, 601, 610, 611, 626, 639, 661, 666, 670, 672
<code>\email</code>	189, 465
<code>\end</code>	207, 250, 267, 275, 307, 315, 323, 501, 504, 525, 526, 562, 563, 683
<code>\endenumerate</code>	101, 496
<code>\endItemize</code>	215
<code>\enditemize</code>	99, 215, 220, 494
<code>\endquote</code>	665
<code>\enumerate</code>	100, 495
environments:dualslide	
<code>dualslide</code>	14
environments:Itemize	
<code>Itemize</code>	7
environments:itemstep	
<code>itemstep</code>	7
environments:notes	
<code>notes</code>	18
environments:slide	
<code>slide</code>	8, 13
<code>\ExecuteOptions</code>	86
F	
<code>\fbox</code>	253
<code>\fi</code>	109, 172, 174, 202, 205, 206, 239, 240, 246, 258, 261, 264, 266, 268, 278, 310, 318, 326, 357, 365, 373, 385, 390, 395, 400, 405, 410, 415, 420, 425, 430, 435, 452, 453, 481, 482, 489–491, 497, 554, 555, 560, 576, 578, 581, 601, 603, 613, 614, 622, 623, 628, 635, 636, 641, 672–674, 677, 686
<code>\Font</code>	288
<code>\FontSubtitle</code>	290
<code>\FontText</code>	291
<code>\fontText</code>	347
<code>\FontTitle</code>	289
<code>\fontTitle</code>	346
<code>\fontToc</code>	570
<code>\footnotesize</code>	263, 663
<code>\FromSlide</code>	297
<code>\fromSlide</code>	331
<code>\fromSlide@E</code>	306, 331
<code>\fromSlide@NE</code>	301, 331

G	
<code>\gdef</code>	44–46, 109, 182, 184, 186, 189, 191, 193, 194, 648–650, 652, 653, 656, 657
<code>\global</code>	9, 10, 13, 14, 33, 38, 42, 50, 95, 118, 119, 165, 173, 209, 227, 241, 243, 245, 310, 318, 326, 344, 445, 493–496, 566, 602, 675, 686, 693
H	
<code>\HAP@bottomsep</code>	521, 538
<code>\HAP@colsep</code>	517, 533
<code>\HAP@contentsline</code>	480, 485, 488, 553, 595, 609, 612
<code>\HAP@indent</code>	522, 531
<code>\HAP@lcolwidth</code>	518, 525
<code>\HAP@leftfoot</code>	648
<code>\HAP@rcolwidth</code>	519, 526
<code>\HAP@ReqVersion</code>	446, 448
<code>\HAP@rightfoot</code>	649
<code>\HAP@toc</code>	564
<code>\HAP@topsep</code>	520, 530, 536
<code>\HAP@Version</code>	447–449
<code>\hbox</code>	578
<code>\hfill</code>	578, 650, 653, 657
<code>\hspace</code>	577
<code>\hss</code>	578
<code>\ht</code>	308, 316, 324, 528, 534, 684
<code>\ht@slide</code> ..	509, 528–530, 532, 534–537
I	
<code>\if</code>	574, 578, 579
<code>\if@distiller</code>	36
<code>\if@HAProsper</code>	95
<code>\if@needsnewpage</code> ..	165, 171, 238, 600
<code>\if@nopagebreak</code>	40, 172, 239, 242, 268, 278, 601, 668
<code>\ifAccumulate</code>	28
<code>\ifallPages</code>	20
<code>\ifarticle</code>	161
<code>\ifcase</code>	621, 634
<code>\ifcolorBG</code>	24
<code>\ifdim</code>	255, 310, 318, 326, 686
<code>\ifDVItOtoPS</code>	8, 353, 361, 369
<code>\ifHAP@notesonly</code>	666
<code>\ifHAP@slidesonly</code>	661
<code>\ifhmode</code>	261
<code>\ifinColor</code>	16
<code>\ifInOverlays</code>	152
<code>\ifisDraft</code>	12
<code>\ifnoFooter</code>	32, 260
<code>\ifodd</code>	172, 239, 278, 601, 672
<code>\ifpdf</code>	109
<code>\ifshowVersion</code>	342
<code>\ifx</code> ...	202, 205, 206, 264, 382, 387, 392, 397, 402, 407, 412, 417, 422, 427, 432, 447, 448, 477– 479, 484, 487, 492, 550, 552, 560, 608, 611, 622, 623, 635, 636
<code>\ignorespaces</code>	312, 320, 328, 688
<code>\InOverlaysfalse</code>	152
<code>\institution</code>	191, 466
<code>\isDraftfalse</code>	14
<code>\isDrafttrue</code>	13
<code>\item</code>	464
<code>\Itemize</code>	214
<code>Itemize</code> (environment)	7
<code>\itemize</code>	98, 214, 219, 492, 493
<code>itemstep</code> (environment)	7
K	
<code>\kern</code>	531, 533
L	
<code>\l@foot</code> ...	646, 648, 650, 652, 653, 657
<code>\languagename</code> ..	382, 387, 392, 397, 402, 407, 412, 417, 422, 427, 432
<code>\LARGE</code>	201, 559
<code>\large</code>	202, 204, 560, 562
<code>\leaders</code>	578
<code>\leavevmode</code>	531
<code>\LeftFoot</code>	651
<code>\let</code>	50, 96–101, 128, 169, 181, 183, 185, 188, 190, 214, 215, 354, 356, 362, 364, 370, 372, 474, 493– 496, 548, 606, 619, 632, 646, 647
<code>\linewidth</code>	130, 132, 247, 252, 262, 307, 315, 323, 517–519, 683
<code>\LoadClass</code>	91
<code>\Logo</code>	281, 282
<code>\Logo@one</code>	283, 285
<code>\Logo@two</code>	283, 286
<code>\long</code> 301–303, 306, 314, 322, 516, 681, 682	
M	
<code>\makebox</code>	262
<code>\maketitle</code>	7, 15, 196, 545

<code>\mbox</code>	172, 199, 208, 239, 278, 557, 565, 601, 672	<code>\onSlide@E</code>	682, 690
<code>\myitem</code>	349	<code>\onSlide@NE</code>	681, 690
		<code>\overlays</code>	295, 542
N		P	
<code>\NeedsTeXFormat</code>	3	<code>\PageBreak</code>	119
<code>\newbox</code>	305	<code>\pagestyle</code>	122
<code>\newcommand</code>	118, 119, 137–141, 159, 170, 182, 184, 189, 191, 282, 289–291, 295, 338, 339, 341, 344, 349, 352, 360, 368, 383, 388, 393, 398, 403, 408, 413, 418, 423, 428, 433, 618, 631	<code>\par</code>	201, 202, 206, 268, 538, 559, 560, 562, 667
<code>\newcounter</code>		<code>\parindent</code>	121
. 134–136, 153, 158, 232, 235, 586		<code>\part</code> ...	6, 169, 170, 234, 599, 629, 642
<code>\newenvironment</code> ...	149, 218, 237, 270	<code>\PassOptionsToClass</code>	
<code>\newfloat</code>	114, 115	54, 58, 62, 66, 70, 74, 78, 82
<code>\newif</code>	8, 12, 16, 20, 24, 28, 32, 36, 40, 95, 152, 161, 165, 342	<code>\PassOptionsToPackage</code>	
<code>\newlength</code>	55, 59, 63, 67, 71, 75, 79
. 129, 131, 133, 226, 228, 509, 571		<code>\pcontentsline</code>	16, 573
<code>\newsavebox</code>	230, 507, 508	<code>\PDFforPS</code>	352
<code>\newslideframe</code>	140	<code>\PDFtransition</code>	338
<code>\NewSlideStyle</code>	139, 468	<code>\PPRPRV@Version</code>	2, 4, 450
<code>\no</code> 265, 384, 389, 394, 399, 404, 409, 414, 419, 424, 429, 434, 438, 578		<code>\ProcessOptions</code>	87
<code>\noFooterfalse</code>	34	<code>\prosper@next</code>	354, 356, 358, 362, 364, 366, 370, 372, 374
<code>\noFootertrue</code>	33	<code>\providecommand</code>	
<code>\NoFrenchBabelItemize</code>	340	. 384, 389, 394, 399, 404, 409, 414, 419, 424, 429, 434, 437, 438	
<code>\NoPageBreak</code>	5, 118	<code>\ProvidesClass</code>	4
<code>\normal@write</code>	128	<code>\PRV@indenttoc</code>	571, 572, 575, 577, 580
notes (environment)	18	<code>\PRV@tempbox</code>	
<code>\null</code>	531	. 305, 307–309, 311, 315–317, 319, 323–325, 327, 683–685, 687	
		<code>\PRV@xitem</code>	463, 464
		<code>\ptsize</code>	148
		Q	
		<code>\quote</code>	663
O		R	
<code>\old@endenumerate</code>	101, 456, 496	<code>\r@foot</code> ...	647, 649, 650, 653, 656, 657
<code>\old@enditemize</code>	99, 455, 494	<code>\raisebox</code>	532, 537
<code>\old@enumerate</code>	100, 456, 495	<code>\refstepcounter</code>	
<code>\old@itemize</code>	98, 455, 492, 493	175, 198, 272, 472, 546, 604
<code>\old@section</code>	96, 176, 615	<code>\relax</code>	87, 169, 297–299, 340, 680
<code>\old@subsection</code>	97, 248, 663	<code>\renewcommand</code>	
<code>\old@Title</code>	185	. 186, 194, 196, 468, 542, 545, 570, 573, 587, 599, 617, 651, 655	
<code>\onlyInPDF</code>	368	<code>\renewenvironment</code>	
<code>\onlyInPS</code>	360	455, 456, 471, 504, 662
<code>\OnlySlide</code>	299	<code>\renewpagestyle</code>	159
<code>\onlySlide</code>	333	<code>\RequirePackage</code> ...	104–108, 110, 113
<code>\onlySlide@E</code>	322, 333	<code>\RightFoot</code>	655
<code>\onlySlide@NE</code>	303, 333	<code>\rule</code>	257
<code>\OnSlide</code>	18, 680		
<code>\onSlide</code>	18, 690		

S	
<code>\sbox</code>	307, 315, 323, 525, 526, 683
<code>\section</code>	96
<code>\setcounter</code>	197, 232, 235, 473, 547, 556, 593, 605
<code>\setkeys</code>	475, 523, 549, 607, 620, 633
<code>\setlength</code>	121, 130, 132, 243, 245, 308, 316, 324, 517– 522, 528, 534, 572, 575, 580, 684
<code>\settoheight</code>	251
<code>\sffamily</code>	663
<code>\showVersionfalse</code>	343
<code>\showVersiontrue</code>	344
<code>slide</code> (environment)	8, 13
<code>\slideCaption</code>	182
<code>\slidefonts</code>	160
<code>\slideframe</code>	138
<code>\slideheight</code>	133
<code>\slidename</code>	265, 383, 388, 393, 398, 403, 408, 413, 418, 423, 428, 433, 437, 578
<code>\slidepagestyle</code>	141
<code>\slidetitle</code>	137
<code>\slideWidth</code>	129, 130
<code>\slidewidth</code>	131, 132
<code>\space</code>	449, 450
<code>\stepcounter</code>	272, 472, 594
<code>\Style@chosen</code>	49, 50
<code>\subsection</code>	97
<code>\subtitle</code>	184
T	
<code>\textdegree</code>	384
<code>\the</code>	308, 309, 316, 317, 324, 325, 528, 531, 533, 534, 684, 685
<code>\theMG@tempTrueSlideCounter</code>	595
<code>\theslidei</code>	265, 473, 480, 485, 488, 547, 553, 605, 609, 612
<code>\theslideii</code>	172, 239, 278, 601, 672
<code>\thetrueSlideCounter</code>	593
<code>\thispagestyle</code>	197, 556
<code>\title</code>	185, 186
<code>\tsection</code>	587, 625, 627, 638, 640
<code>\tsectionandpart</code>	617
<code>\tsectionandpart@E</code>	617, 618
<code>\tsectionandpart@NE</code>	617, 631
<code>\typeout</code>	5
U	
<code>\undefined</code>	447
<code>\UntilSlide</code>	298
<code>\untilSlide</code>	332
<code>\untilSlide@E</code>	314, 332
<code>\untilSlide@NE</code>	302, 332
<code>\usebox</code>	254, 311, 319, 327, 532, 537, 687
V	
<code>\vbox</code>	311, 319, 327, 687
<code>\vfill</code>	199, 208, 268, 278, 557, 564, 565, 671
<code>\vskip</code>	202, 203, 205, 206, 312, 320, 328, 560, 561, 688
<code>\vspace</code>	151, 538
W	
<code>\write</code>	128
X	
<code>\xitem</code>	463
Z	
<code>\z@</code>	227, 243, 624, 637