Making Presentations with \LaTeX\ Guidelines

Xavier Perseguers
xavier.perseguers@epfl.ch

Computer Sciences
Semester Project
March–June 2004

Person in charge
Prof. Serge Vaudenay
serge.vaudenay@epfl.ch
EPFL / LASEC
## Contents

Disclaimer

1 Introduction
   1.1 Packages for \LaTeX

2 Beamer
   2.1 Introduction
   2.2 Frames
   2.3 Overlays
   2.4 Framed Text
   2.5 Interaction
   2.6 Compatibility with Other Packages
   2.7 Presentation Styles
   2.8 Useful Macros
   2.9 Identified Limitations

3 Prosper
   3.1 Introduction
   3.2 Preamble
   3.3 Slides
   3.4 Overlays
   3.5 Presentation Styles
   3.6 How do I
   3.7 Identified Bugs

4 \TeXPower
   4.1 Introduction
   4.2 Preamble
   4.3 Toggling the Logo
   4.4 The Other Three Corners
   4.5 Standard Colors
   4.6 Panels
   4.7 Overlays
   4.8 How do I
   4.9 Useful Macros
   4.10 Identified Bugs

5 Presentation Features
   5.1 Adding Slide Transitions
   5.2 Additional PDF Presentation Features
6 Creation of the PDF Files | 53
6.1 To pdflatex or Not? | 53
6.2 Producing Nice Looking PDF | 54
6.3 PDF Encryption | 56
6.4 Other PDF Features | 58
A Common Problems Resolution | 63
A.1 Invalid Page Format/Orientation | 63
A.2 Multiple Inclusion of a Picture | 64
B Additional Material | 69
B.1 Text following a Sinus Curve | 69
C A few LATEX Explanations | 73
C.1 Fonts and Sizes | 73
C.2 Fragile and Robust Commands | 73
References | 75
Index | 77
Disclaimer

This document is not intended to replace the reference manuals of the corresponding presentation packages available for \LaTeX. It may however help you choosing the best package fitting your needs and then getting as quick as possible a basic understanding on how designing your slides.

The comprehension of this document does not require high \LaTeX skills but assumes nevertheless being familiar with the \LaTeX environment and a basic understanding of macros.

When possible, a try has been made to explain common problems or strange behaviours and how to prevent or correct them. You will find all useful materials presented in this document (such as source code) on the Web:

http://www.perseguers.ch/latex/contrib/presentations/

This document is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.
Index

A
\Acrobatmenu 61
\activateatstep [T] 42
\againframe [P] 25
\alert [B] 8
\alert [T] 8
\alt [B] 7
\animate [B] 9
\animatevalue [B] 10
\AtBeginSection [P] 17
\AtBeginSubsection [P] 17
\author [P] 23
\author [T] 35
lAdvance timing 52
allowframebreaks [B] 6
Auto-advancing 9, 50

B
\backgroundstyle [B] 39
\beamerbutton [B] 11
\beamergotobutton [B] 11
\beamerreturnbutton [B] 12
\beamerskipbutton [B] 11
\beamerstep [T] 43, 44
\BEtaveragebackground [B] 16
\BTgridbackground [B] 16
\BTsolidbackgroundcolor [B] 16
\button [T] 39
Background [B] 38
beamerrahomrounded [B] 10
beamernames 9
Black box in Acrobat Reader 32
Bookmarks 59
break, automatic for frames [B] 6

C
\colorlet [B] 16
Colors [B] 36

D
\DefaultTransition [P] 24
\displaybox [T] 42
\displayidentical [T] 42
\displaystepcontents [T] 42
Display duration 52
Document's informations 59
dvipdf 53
dvipdfm 53
dvip 55

E
\email [P] 23
\email [T] 33

F
\foilhead [T] 33
\framenumber [P] 6
\framesize [P] 12
\FromSlide [P] 25
\fromslide [P] 24
\fromslide.cls is not found 33
Font Size 74
Font, Changing 73
Frame 6
Frame break (automatic) [B] 6

G
Gradient [T] 38

H
\hideignore [T] 42
\hidephantom [T] 42
\hidestepcontents [T] 42
\hidetext [T] 42, 43
\hidetitle [T] 42
highlighted [T] 43
highlightenanced [T] 43
highlighttext [T] 43
\hyperlink [T] 11
\hyperlinkframestartnext [T] 11
\hypersetup 52
Hyperlinks 59
hyperef 61

I
\ifthenelse [T] 41
\imaketitle [P] 24
\invisible [B] 7
Incremental highlight 8, 44

J
\leftheader [T] 37
\liststepwise [T] 41
\logo [P] 24
\logoname [T] 37
\logoframe [T] 37
\logo [T] 37
\mylogo [T] 35
\mktextcolor 29

K

L
\nonboxedsteps [T] 43
\navigation, enhancing 61

M
\only [P] 7
\onlyinPDF [P] 25
\onlyinPS [P] 25

N

O

P

Q

R

S

T

U

V

W

X

Y

Z
1 | Introduction

Contents

1.1 Packages for \LaTeX \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} 1
1.1.1 Beamer \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} 1
1.1.2 Prosper \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} 1
1.1.3 \TeX\hspace{1em}Power \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} \hspace{1em} 2

1.1 Packages for \LaTeX

This document tries to show some possible solutions for creating screen based presentations. As there exists lots of tools for creating screen or online presentations, a choice has been made over solutions like beamer, foiltex, HA-prosper, ifmslide, PPower4, Prosper, seminar.sty, \TeX\hspace{1em}Power and so on, to retain only three of them: Beamer, Prosper and \TeX\hspace{1em}Power. A good document for starting using other PDF based solutions or even HTML based solutions like DocBook slides or latex2slides is available at http://www.miwie.org/presentations/

1.1.1 Beamer

\LaTeX\hspace{1em}Beamer is a really easy-to-use package to create nice PDF presentations. Changing a parameter at the beginning of the document allows you to output either a standard presentation, or handouts or even the whole presentation as a standard \LaTeX article.

Homepage: \hspace{1em} http://latex-beamer.sourceforge.net

Index: \hspace{1em} Beamer-relative commands are followed by the symbol “\hspace{1em}B\hspace{1em}” in the index, starting at page 77.

1.1.2 Prosper

This is a set of macros which allows you to generate PostScript or PDF presentations. There are certain advantages of this package over the others. First, though it has a simple structure, it provides enough options to generate good-looking slides. All the features of a PDF document (such as transitions, overlays, etc.) are available. In addition, it is easy to generate different slide styles, à la PowerPoint. Of course, you still have access to the full power of \LaTeX, so you are free to extend your documents if you have the knowhow. For \LaTeX beginners, however, \TeX\hspace{1em}Power encapsulates a lot of the details in an easy-to-use manner.

Homepage: \hspace{1em} http://prosper.sourceforge.net
Introduction

Good to Know: This package is provided with lots of presentation styles.

Index: Prosper-relative commands are followed by the symbol \[P\] in the index, starting at page 77.

1.1.3 texPower

This is an “all-inclusive” bundle to aid creating presentations. It provides color and font management, basic effects for incremental display, panels, navigation aids. The main distinguish features of texPower are:

- independent of the way PDF is created;
- independent of document class;
- implement display effect by \LaTeX{} programming.

Homepage: http://texpower.sourceforge.net

Index: \LaTeX{}-relative commands are followed by the symbol \[T\] in the index, starting at page 77.

References


Table C.2: Type sizes for size-changing commands for the different document style options

<table>
<thead>
<tr>
<th>Size</th>
<th>\tiny</th>
<th>\scriptsize</th>
<th>\footnotesize</th>
<th>\small</th>
<th>\normalsize</th>
<th>\large</th>
<th>\Large</th>
<th>\huge</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textbf{17pt}</td>
<td>12pt</td>
<td>12pt</td>
<td>14pt</td>
<td>17pt</td>
<td>12pt</td>
<td>20pt</td>
<td>25pt</td>
<td>12pt</td>
</tr>
<tr>
<td>\textbf{25pt}</td>
<td>14pt</td>
<td>12pt</td>
<td>17pt</td>
<td>20pt</td>
<td>12pt</td>
<td>14pt</td>
<td>12pt</td>
<td>12pt</td>
</tr>
<tr>
<td>\textbf{30pt}</td>
<td>17pt</td>
<td>12pt</td>
<td>20pt</td>
<td>25pt</td>
<td>12pt</td>
<td>14pt</td>
<td>12pt</td>
<td>12pt</td>
</tr>
<tr>
<td>\textbf{shortform}</td>
<td>12pt</td>
<td>12pt</td>
<td>12pt</td>
<td>14pt</td>
<td>12pt</td>
<td>12pt</td>
<td>12pt</td>
<td>12pt</td>
</tr>
</tbody>
</table>

C.2.1 Some Fragile Commands

The following list is not exhaustive, but illustrates the kind of commands which are fragile:

- arguments of \caption commands;
- terminal input and output, \typeout and \typein;
- commands that produce page headings;
- the Letter Environment;
- the \thanks command;
- an \@ expression in the Array or Tabular environment.

A fragile command that appears in a moving argument must be preceded by a \protect command. The \protect applies only to the immediately following command; if arguments of this command also contain fragile commands, the latter must be protected with their own \protect.

C.2.2 Some Robust Commands

In general, commands which change Type face or Type style are robust. Length commands are robust and should not be preceded by a \protect command. Nor should a \protect command be used in the argument to \addtocounter or \setcounter command.

2 | Beamer

Contents

2.1 Introduction ........................................... 3
  2.1.1 Options of the Class ............................... 5
2.2 Frames .................................................. 6
2.3 Overlays ............................................... 7
  2.3.1 Dynamically Replacing Text ....................... 7
  2.3.2 Specifying Ranges of Slides ..................... 8
  2.3.3 Incremental highlight ............................ 8
  2.3.4 Incremental Specifications ....................... 8
  2.3.5 Quick Animations ................................ 9
2.4 Framed Text ............................................ 10
2.5 Interaction ............................................. 11
  2.5.1 Jumps .............................................. 11
  2.5.2 Zoom .............................................. 12
2.6 Compatibility with Other Packages .................. 13
2.7 Presentation Styles ................................... 14
  2.7.1 Available Styles ................................ 14
  2.7.2 Choosing Another Color Theme ................... 16
2.8 Useful Macros ........................................... 16
  2.8.1 Removing Navigation symbols ..................... 16
  2.8.2 Incremental Highlight ............................ 16
  2.8.3 Using Packages only when Printing .............. 17
  2.8.4 Structuring Presentation ......................... 17
2.9 Identified Limitations ................................ 18
  2.9.1 Verbatim Environment ............................. 18
  2.9.2 In-line Verbatim .................................. 18

2.1 Introduction

To use the beamer class together with latex or pdflatex, proceed as follows:

1. Specify beamer as document class instead of article.
2. Structure your \LaTeX\ text using section and subsection commands.
3. Place the text of the individual slides inside a frame environment.
4. Run pdflatex on the text.

The following code shows a typical usage of the class (see also Figures 2.1 and 2.2).
\documentclass{beamer}

% Load a theme (graphics, colors, ...) for the presentation
\usepackage{beamerthemesplit}
\title{Example Presentation}
\author{Xavier Persegers}
\date{today}
\begin{document}
\frame{\titlepage}
\section{Outline}
\frame{\tableofcontents}
\section{Introduction}
\subsection{Overview of this class}
\frame{List displayed step–by–step}
\begin{itemize}
\item Normal LaTeX class;
\item Easy overlays;
\item Straightforward use!
\end{itemize}
\section{Current Activities}
\subsection{...}
\subsection{...}
\section{Our Goals}
\subsection{...}
\subsection{...}
\end{document}

C A few \LaTeX\ Explanations

## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.1 Fonts and Sizes</td>
<td>73</td>
</tr>
<tr>
<td>C.2 Fragile and Robust Commands</td>
<td>73</td>
</tr>
<tr>
<td>C.2.1 Some Fragile Commands</td>
<td>74</td>
</tr>
<tr>
<td>C.2.2 Some Robust Commands</td>
<td>74</td>
</tr>
</tbody>
</table>

### C.1 Fonts and Sizes

The default font at \texttt{\normalsize} is a "sans serif font" at size 20pt, unless one of the \texttt{[17pt], [25pt], [30pt]} or shortform\footnote{This class option can be used to generate a document which takes up much less space (paper, mostly).} options have been declared in the \texttt{\documentclass} command. Table C.1 shows the control sequences for other accessible text fonts and the name of the font in a sample of its type. These control sequences give the font at the current size. Font size changing commands for each of the normal point size options are described by Table C.2. Note that \texttt{\bf} and \texttt{\sl} yield sans serif fonts, not the usual variations on roman.

<table>
<thead>
<tr>
<th>Command</th>
<th>Font Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>\sf</td>
<td>Sans Serif</td>
</tr>
<tr>
<td>\it</td>
<td>Text Italic</td>
</tr>
<tr>
<td>\sl</td>
<td>Slanted Sans Serif</td>
</tr>
<tr>
<td>\bf</td>
<td>Bold Sans Serif</td>
</tr>
<tr>
<td>\tt</td>
<td>Typewriter</td>
</tr>
<tr>
<td>\rm</td>
<td>Roman</td>
</tr>
<tr>
<td>\sc</td>
<td>SMALL CAPS</td>
</tr>
</tbody>
</table>

Table C.1: Available fonts and their name

### C.2 Fragile and Robust Commands

\LaTeX\ commands are either fragile or robust. Fragile commands need special care if they are part of a moving argument.

Arguments to some \LaTeX\ commands are called moving arguments because they may be "moved" to other places in the document. For example, the argument of Sectioning commands may be moved to the Table of Contents. Other examples of moving arguments include:
2.1.1 Options of the Class

To make a presentation using the \texttt{beamer} class, you need to specify it in your \texttt{documentclass}. Thus, the first line in the \LaTeX{} file should be of the form:

\begin{verbatim}
\documentclass[⟨options⟩]{beamer}
\end{verbatim}

There are several options that can be specified to the package:

- \texttt{slidestop} puts frame titles on top left corner (default is \texttt{slidescentered}).
- \texttt{compress} makes all navigation bars as small as possible (default is \texttt{uncompressed}).
- \texttt{mathserif/mathsans} uses fonts with serif for maths (default is to use sans-serif fonts as for the text).
- \texttt{sans-serif} uses fonts with or without serif for the text (default is to use sans-serif fonts).
- \texttt{handout} for PDF handouts.
- \texttt{trans} for PDF transparency.

\textbf{Font Size} default is 11pt but may take following values: 8pt, 9pt, 10pt, 11pt, 12pt, 14pt, 17pt, 20pt (see also C.1).

Another important option to specify is which presentation style to use. \texttt{BEAMER} comes with several styles which are described in \texttt{2.7}.

There are also options to specify slide background colors, slide numbers, etc. In general, unless you require black and white slides (e.g., for printing purposes), you will not need to set any color options in the \texttt{documentclass}; the style files will manage them for you.
2.2 Frames

As with most presentation packages, the “unit” in presentations (or slide) is called a frame in Beamer. You have to inform LATEX the contents to be typeset on each frame. This is easily performed with the command \frame as shown below:

\begin{frame}
  \frametitle{Title of the frame}
  \begin{itemize}
    \item Contents such as maths, lists, ...
  \end{itemize}
\end{frame}

If you like the way LATEX is able to deal with automatic page breaks, and you have seen that \texttt{TEXPower} does it too for slides, you may wonder if you could do the same with \texttt{Beamer}.

This is usually considered as a wrong method as in a (good) presentation, you prepare each slide carefully and think twice before putting something on a certain slide rather than on some different slide. With automatic frame-break, you may create endless presentations that look more like a “paper projected on the wall” than a presentation. Nevertheless, if you would like to activate this feature for a certain frame, you may pass the option \allowframebreaks to the frame definition:

\begin{frame}[allowframebreaks]
  \frametitle{References}
  \begin{itemize}
    \item bibitem ...
    \item bibitem ...
    \item bibitem ...
  \end{itemize}
\end{frame}

When the option \allowframebreaks is given, the frame will be automatically broken up into several frames, if it does not fit on a single slide. In details, the following things happen:

1. The option \texttt{containsverbatim} (see \ref{sec:verbatim}) is automatically selected, as a side-effect. Thus, frames with this option set may contain verbatim text;
2. Consequently, overlays (see next section) are not supported;
3. Any footnotes for the frame will be inserted at the last page of the frame;
4. If there is a frame title, each page will have this frame title, with a special note (usually a Roman number) added indicating which page of the frame that page is.

\textbf{Beware:} If a frame needs to be broken into several pages, the material on all but the last page fills only 95\% of each page by default. Thus, there will be some space left at the top and/or bottom. This yields a better visual result than a 100\% filling, which typically looks crowded. However, you can change this percentage using the optional argument \texttt{(fraction)}, where 1 means 100\% and 0.5 means 50\%. This percentage includes the frame title.

\begin{frame}[allowframebreak=0.6]
  \frametitle{Text following a Sinus Curve}
  \begin{itemize}
    \item I'm introducing a new concept...
    \item I'm introducing a new concept...
    \item I'm introducing a new concept...
  \end{itemize}
\end{frame}

\texttt{TEXPower}

File name: \texttt{texpower-wavetext.tex}

\begin{verbatim}
\documentclass[landscape,a4paper]{foils}
\usepackage{color,soul,fixseminar,hyperref}
\usepackage[display]{texpower}
\usepackage[textpower]{texpower}{wavetext}
\rightfooter{}
\begin{document}
\foothead{Text following a Sinus Curve}
\vspace{3cm}
\stepwise{\Large I'm introducing a new concept...\ldots}
\end{document}
\end{verbatim}
When creating overlays, how do you specify on which slides of a series of slides a certain text should be shown? The approach taken by most presentation classes is to introduce new commands, which get a certain slide number as input and which affect text on the slide following this command in a certain way.

BEAMER uses a different approach. The idea is to add overlay specifications to certain commands. These specifications are always given in pointed brackets and follow the command “as soon as possible”. Consider the following example.

```
\begin{slide}
\textbf{<3>}{This line is bold solely on slide 3.}
\end{slide}
```

Another example, using the command \only introduced by BEAMER, lets you “throw away” its contents on slides that are not mentioned. In particular, it occupies no space.

```
\begin{slide}
\only<1>{This line is inserted solely on slide 1.}
\end{slide}
```

There exists other replace commands:

- \textbf{<slides>}: If an overlay specification is present, the text is shown (“uncovered”) only on the specified slides. On other slides, the text still occupies space and it is still typeset, but it is not shown.
- \textbf{<slides>}: Hides the text on the specified slides.
- \textbf{<slides>}: The (main) text is shown on the specified slides, otherwise the (alternative) text.
- \textbf{<slides>}: The (main) text is displayed on the specified slides, otherwise either the parameter \textbf{before} if the current slide is logically before the specified slides, or the parameter \textbf{after} if it is logically after the specified slides.

In case of problems with the heights of replacements, two environments may be used:

\begin{verbatim}
\begin{overlayarea}
\end{overlayarea}
\end{verbatim}

Everything within the environment will be placed in a rectangular area of the specified size. The area will have the same size on all slides of a frame, regardless of its actual contents.
overprint Inside the environment, use \texttt{\onslide} commands to specify different things that should be shown for this environment on different slides. The overlay specifications of the \texttt{\onslide} commands must be disjoint (see next section).

\begin{verbatim}
\begin{overprint}
  \onslide<1> Some text for the first slide.\end{overprint}
\end{verbatim}

2.3.2 Specifying Ranges of Slides

The syntax of (basic) overlay specification is the following: they are comma-separated list of slides and ranges. Ranges are specified like this: 2–5, that means slide two to five. The start or the beginning of a range or the end (but not both of them) may be omitted as it was the case with the itemize environment of the first example, at the beginning of the chapter. An example is 3+ meaning “slides three, four, five, and so on” as -5 is equivalent to 1–5.

2.3.3 Incremental Highlight

The incremental highlight is a way to step through an enumeration of items and displaying in another color (or highlighting) each item as it is introduced (see also § 2.8.2).

\texttt{\alert{\{contents\}}} Emphasizes \texttt{\{contents\}}. If an overlay specification is given, as in the example below, only emphasizes \texttt{\{contents\}} at the corresponding slide(s).

The following example shows the three items of the list starting from slide #2 (<2->) and alerts them one after the other: item foo at step (or “slide”) 2, item foo bar at step 3 and item foo bar bar at steps 4 and greater.

\begin{verbatim}
\begin{itemize}
  \item<2->\alert<2>{foo}
  \item<2->\alert<3>{{foo bar bar}}
\end{itemize}
\end{verbatim}

2.3.4 Incremental Specifications

Often, you want to have overlay specifications that follow a pattern similar to the following:

\begin{verbatim}
\begin{itemize}
  \item<1,2,3> Item foo
  \item<2,3> Item foo bar
  \item<3> Item foo bar bar
\end{itemize}
\end{verbatim}

The problems starts if you decide to insert a new item, for instance, at the beginning. In this case you would have to adjust all overlay specifications. \textsc{Beamer} offers a special \textit{incremental overlay specification}. 

B Additional Material

B.1 Text following a Sinus Curve

This example is written for packages \textsc{Prosper} and \textsc{TeXPower}. It was not extended to \textsc{BEAMER} as this package best works with \textsc{pdflatex} and \textsc{pdflatex} cannot handle package \textsc{psTricks} as said in § 6.1, p. 53.

B.1.1 Package Source Code

Requirements: packages \textsc{pet-text} and \textsc{pet-plot}

File name: \textsc{wavetext.sty}

\begin{verbatim}
% WaveText for Prosper
% Created by Xavier Perseguers
% See the GNU General Public License
\def\fileversion{1.1}
\def\filedate{2004/06/11}
\def\docdate{2004/06/11}
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{wavetext}[\filedate\space\fileversion\space WaveText for Prosper package]
\RequirePackage{pet-text,pet-plot}
\newcommand{\\WT@pkgname}{\wavetext}
\newif\@prosper\@prosperfalse
\newif\@texpower\@texpowerfalse
\newif\@nopackage\@nopackagetrue
\DeclareOption{prosper}{\@prospertrue}
\DeclareOption{texpower}{\@texpowertrue}
\DeclareOption{nopackage}{\@nopackagetrue}
\@prosperfalse\@texpowerfalse\@nopackagetrue
\end{verbatim}
2.3. Overlays

The +−sign may be use in any overlay specification at any point where you would usually use a number. If a +−sign is encountered, it is replaced by the current value of the \Beamer@pauses counter, which is 1 at the beginning of the frame. The counter is increased by 1, at each animation step.

Incremental Highlight with the Incremental Overlay Specification

In the following example, we use the incremental overlay specification to emphasize each item as it is introduced. The special specification +−| alert@+ will be replaced by 1−| alert@1 for the first item, 2−| alert@2 for the second... The notation | alert@1 is a way to specify an special action to be taken at the corresponding step(s). It should be understood as something like that:

\item<+− | alert<+> or 
\item<1− | alert<1> 
\item<2− | alert<2>

Or even shorter...

\begin{itemize}
\item<+− | alert@+>
\item foo
\item foo bar
\item foo bar bar
\end{itemize}

2.3.5 Quick Animations

Chapter 5 explains how you may create advanced animation effects in your presentation. Auto-advancing is a way to create a series of slides shown in rapid succession (see §5.2.1). To facilitate the creation of animations using the auto-advancing feature, the following commands may be used:

\animate⟨overlay specification⟩ Shows the slides specified by the parameter ⟨overlay specification⟩ only as shortly as possible.

Example:

\begin{frame}
\frametitle{A Five Slide Animation}
\animate<2→5>
\begin{itemize}
\item The first slide is shown normally. When the second slide is shown (presumably after pressing a forward key), the second, third, and fourth slides “flash by”. At the end,
A.2.4 Quicker Picture Inclusion

If you include multiple times the same graphics and use each time the includegraphics command, \LaTeX{} will read the picture file for each inclusion, resulting in a longer process.

Manipulating boxed material

Material can be typeset once and then stored inside a named box, whose contents can later be retrieved. Include the declaration in the preamble:

\newsavebox{myLogo}
\usebox{myLogo}

and reuse the box in the document, to include the picture:

\pdfimage width \textwidth
\usebox{myLogo}
A.2.3 Using pdflatex

The syntax of picture inclusion seems somewhat odd at first sight but authorizes the picture to be included only once even if you need to “include” it multiple times. \pdfximage creates a unique identifier for each picture allowing you to reference it for a later use.

\pdfximage \pdfrefximage

\pdfximage creates a new object, containing the specified picture and holds it in memory. Hence, this object is not written to the PDF output unless:

1. the image is referenced by saying \pdfrefximage (object number);
2. \pdfximage is preceded by \immediate.

\pdfximage \pdfrefximage \pdflastximage

\pdflastximage returns the ID of the picture object produced last. Once a picture object has been created, you may use the associated object arbitrarily often again.

It’s highly recommended to wrap \pdfrefximage by a box in order to ensure spacing will be correct:

\begin{itemize}
\item \pdfrefximage \pdflastximage
\end{itemize}

Size and Resolution of the Image

The following examples are taken from [13] and associated documentation.\footnote{The documentation is part of \TeX{} and Mi\TeX{} distributions.}

If \pdfimageresolution is set, the image will be inserted at that resolution (or 72 if \pdfimageresolution is set to zero) given in dots per inch in the output file, but some images may contain data specifying the image resolution, and in such a case the image will be scaled to the original resolution. The dimensions of the image can be accessed by enclosing the \pdfximage command to a box and checking the dimensions of the box:

\begin{itemize}
\item \pdfximage \pdfrefximage \pdflastximage
\item \pdfximage \pdfrefximage \pdflastximage
\end{itemize}

\begin{itemize}
\item \pdfximage width 3cm \pdfximage \pdflastximage
\item \pdfximage \pdfrefximage \pdflastximage
\item \pdfximage \pdfrefximage \pdflastximage
\end{itemize}

2.5 Interaction

During most presentations, you would like to present your slides in a linear fashion. However, there are different reasons why you might have to deviate from this linear order such as:

- go back to an earlier slide;
- present a complicated picture and “zoom out” different part to explain details;

2.5.1 Jumps

To create nonlinear jumps in your presentation, you can add hyperlinks. A hyperlink is usually rendered as a button that, when you click on it, jumps to some other slide. The example below shows how you may create a hyperlink to go to a specific step of the animation of a labeled frame.

\begin{itemize}
\item \hyperlink\frame \begin{itemize}
\item \frame (\textbf{Jump to second step})
\item \frame (\textbf{Jump to second step})
\end{itemize}
\end{itemize}

- First item
- Second item

Figure 2.3: BEAMER: Hyperlink as a button.
\begin{proof}
\begin{itemize}
\item ...
\item ...
\end{itemize}
\end{proof}

\begin{frame}
\frametitle{Multiple Inclusion of a Picture}

\begin{Verbatim}
\documentclass[article]
\usepackage[graphics]
\begin{document}
\includegraphics{foo} \includegraphics{foo} \includegraphics{foo}
\end{document}
\end{Verbatim}

\begin{tabular}{|c|c|c|}
\hline
\textbf{program} & \textbf{te\TeX} & \textbf{Mik\TeX} \\
\hline
dvips & 5.92b & 5.90a \\
pdflatex & 3.14159-1.10b & 2.14542 (1.20a-rc1) \\
ghostscript & APFL 8.14 & APFL 8.14 \\
\hline
\end{tabular}

\end{frame}

A.2.2 Using \texttt{dvipdfm}

\texttt{dvipdfm} includes command to create \texttt{X}Object. The following code shows how you may declare and then reuse a picture. Please note that:

- \texttt{X}Object definition does not display the picture;
- You should encapsulate use of \texttt{X}Object within a box to let \texttt{\LaTeX} create a decent layout. The problem is that \texttt{\LaTeX} is not aware of the dimensions of the picture as this is purely done by the DVI driver;
- \texttt{X}Object dimensions should be manually adjusted or left bigger than what is actually needed.

\textbf{Requirements:} packages graphics and ifthen, \texttt{dvipdfm} to compile the DVI file.

\textbf{File name:} logo-dvipdfm.tex

\begin{Verbatim}
\documentclass[article]
\usepackage[graphics]
\begin{document}
\begin{itemize}
\item \texttt{\LaTeX} automatically.
\item A few tests have been made with the file below. Using either \texttt{pdflatex} or \texttt{latex + dvips + ps2pdf} for producing the PDF file gave always a file as big as if the image had been included only once. The latest distribution of Mik\TeX\ under Windows was not able to create small PDF files. However a Windows PostScript generated file can result in a small PDF file when converted with \texttt{ps2pdf} under Linux, although the version of Ghostscript is equivalent!
\item \\
\end{itemize}
\end{document}
\end{Verbatim}

\begin{Verbatim}
\begin{frame}
\frametitle{Zoom}

Sometimes, a graphic may be complex and you are willing to spend much time explaining it in great detail. In this case, you will often run into the problem that fine details of the graphic are hard to discern. One way to solve this problem is to use the command \texttt{\framezoom}. This command allows you to specify that clicking on a certain area of a frame should zoom out this area.

\begin{Verbatim}
\framezoom{button overlay specification} <\{zoomed overlay specification\} \{options\} ((upper left x),(upper left y)) (\{zoom area width\},(\{zoom area depth\}))
\end{Verbatim}

This command should be given somewhere at the beginning of a frame. When given, two different things will happen, depending on whether the \texttt{(button overlay specification)} applies to the current slide of the frame or whether the \texttt{(zoomed overlay specification)} applies. These overlay specifications should not overlap.

If the \texttt{(button overlay specification)} applies, a clickable area is created inside the frame. The size of this area is given by \texttt{(area width)} and \texttt{(area depth)}. The upper left corner of this area is given by \texttt{(upper left x)} and \texttt{(upper left y)}. They are measures relative to the place where the first normal text of a frame would go. Thus, the location (Opt,Opt) is at the beginning of the normal text (which excludes the headline and the frame title).

By default, the button is clickable, but it will not be indicated in any special way. To draw a border around the button, use the following command:

\begin{Verbatim}
\beamertobutton{[text]} The symbol drawn for this button is usually a left-pointing arrow. Use this button if pressing it will return from a detour.
\end{Verbatim}

\begin{Verbatim}
\begin{frame}
\frametitle{Zoom}

Sometimes, a graphic may be complex and you are willing to spend much time explaining it in great detail. In this case, you will often run into the problem that fine details of the graphic are hard to discern. One way to solve this problem is to use the command \texttt{\framezoom}. This command allows you to specify that clicking on a certain area of a frame should zoom out this area.

\begin{Verbatim}
\framezoom{button overlay specification} <\{zoomed overlay specification\} \{options\} ((upper left x),(upper left y)) (\{zoom area width\},(\{zoom area depth\}))
\end{Verbatim}

This command should be given somewhere at the beginning of a frame. When given, two different things will happen, depending on whether the \texttt{(button overlay specification)} applies to the current slide of the frame or whether the \texttt{(zoomed overlay specification)} applies. These overlay specifications should not overlap.

If the \texttt{(button overlay specification)} applies, a clickable area is created inside the frame. The size of this area is given by \texttt{(area width)} and \texttt{(area depth)}. The upper left corner of this area is given by \texttt{(upper left x)} and \texttt{(upper left y)}. They are measures relative to the place where the first normal text of a frame would go. Thus, the location (Opt,Opt) is at the beginning of the normal text (which excludes the headline and the frame title).

By default, the button is clickable, but it will not be indicated in any special way. To draw a border around the button, use the following command:

\begin{Verbatim}
\beamertobutton{[text]} The symbol drawn for this button is usually a left-pointing arrow. Use this button if pressing it will return from a detour.
\end{Verbatim}

\end{frame}

\begin{frame}
\frametitle{Zoom}

Sometimes, a graphic may be complex and you are willing to spend much time explaining it in great detail. In this case, you will often run into the problem that fine details of the graphic are hard to discern. One way to solve this problem is to use the command \texttt{\framezoom}. This command allows you to specify that clicking on a certain area of a frame should zoom out this area.

\begin{Verbatim}
\framezoom{button overlay specification} <\{zoomed overlay specification\} \{options\} ((upper left x),(upper left y)) (\{zoom area width\},(\{zoom area depth\}))
\end{Verbatim}

This command should be given somewhere at the beginning of a frame. When given, two different things will happen, depending on whether the \texttt{(button overlay specification)} applies to the current slide of the frame or whether the \texttt{(zoomed overlay specification)} applies. These overlay specifications should not overlap.

If the \texttt{(button overlay specification)} applies, a clickable area is created inside the frame. The size of this area is given by \texttt{(area width)} and \texttt{(area depth)}. The upper left corner of this area is given by \texttt{(upper left x)} and \texttt{(upper left y)}. They are measures relative to the place where the first normal text of a frame would go. Thus, the location (Opt,Opt) is at the beginning of the normal text (which excludes the headline and the frame title).

By default, the button is clickable, but it will not be indicated in any special way. To draw a border around the button, use the following command:

\begin{Verbatim}
\beamertobutton{[text]} The symbol drawn for this button is usually a left-pointing arrow. Use this button if pressing it will return from a detour.
\end{Verbatim}

\end{frame}

\begin{frame}
\frametitle{Zoom}

Sometimes, a graphic may be complex and you are willing to spend much time explaining it in great detail. In this case, you will often run into the problem that fine details of the graphic are hard to discern. One way to solve this problem is to use the command \texttt{\framezoom}. This command allows you to specify that clicking on a certain area of a frame should zoom out this area.

\begin{Verbatim}
\framezoom{button overlay specification} <\{zoomed overlay specification\} \{options\} ((upper left x),(upper left y)) (\{zoom area width\},(\{zoom area depth\}))
\end{Verbatim}

This command should be given somewhere at the beginning of a frame. When given, two different things will happen, depending on whether the \texttt{(button overlay specification)} applies to the current slide of the frame or whether the \texttt{(zoomed overlay specification)} applies. These overlay specifications should not overlap.

If the \texttt{(button overlay specification)} applies, a clickable area is created inside the frame. The size of this area is given by \texttt{(area width)} and \texttt{(area depth)}. The upper left corner of this area is given by \texttt{(upper left x)} and \texttt{(upper left y)}. They are measures relative to the place where the first normal text of a frame would go. Thus, the location (Opt,Opt) is at the beginning of the normal text (which excludes the headline and the frame title).

By default, the button is clickable, but it will not be indicated in any special way. To draw a border around the button, use the following command:

\begin{Verbatim}
\beamertobutton{[text]} The symbol drawn for this button is usually a left-pointing arrow. Use this button if pressing it will return from a detour.
\end{Verbatim}

\end{frame}
This package is automatically loaded since package \usepackage{⟨amsmath⟩} is used for typesetting theorems. If you do not wish it to be loaded, which can be necessary especially in \article mode if the package is incompatible with the document class, you can use the class option ⟨noamsthm⟩ to suppress its loading.

\usepackage[⟨french⟩]{⟨babel⟩} When using the ⟨french⟩ style, certain features that clash with the functionality of the \beamer class will be turned off. For instance, enumerations are still produced the way the theme dictates, not the way the ⟨french⟩ style does. Also, the characters : and ! will not be active characters. This means that the little space that is inserted before them in the ⟨french⟩ style is not inserted. You have to do this “by hand”.

2.6 Compatibility with Other Packages

When using certain packages together with the \beamer class, extra options or precautions may be necessary.

\usepackage[⟨french⟩]{⟨babel⟩} This package is automatically loaded since package beamer uses it for typesetting theorems. If you do not wish it to be loaded, which can be necessary especially in \article mode if the package is incompatible with the document class, you can use the class option ⟨noamsthm⟩ to suppress its loading.

\usepackage[⟨french⟩]{⟨babel⟩} When using the ⟨french⟩ style, certain features that clash with the functionality of the \beamer class will be turned off. For instance, enumerations are still produced the way the theme dictates, not the way the ⟨french⟩ style does. Also, the characters : and ! will not be active characters. This means that the little space that is inserted before them in the ⟨french⟩ style is not inserted. You have to do this “by hand”.

A.1.2 If you use package hyperref
You probably noticed that package hyperref only creates working links and other dynamic PDF features if you specify the driver you use to produce the PDF file as one of its option; that is either dvips or ps2pdf or even dvipdfm.

The problem is that as soon as you specify the driver, the page orientation is always set to portrait as shown on Figure A.1.

This is a well-known problem of using package hyperref in landscape page orientation mode. The code below fixes the bug if you use dvips + ps2pdf or dvipdf. You should copy it in the preamble of your document.

\begin{verbatim}
\makeatletter
\def\special{\special{\ps2pdf special \thepaper 297mm 210mm}}
\makeatother
\end{verbatim}

\texttt{dvips} compile your document without any additional option.

\texttt{dvipdf} compile your document as usual.

If you use \texttt{dvipdfm} instead, add option \texttt{dvipdfm} to the package \texttt{hyperref} and compile your document as usual with option \texttt{-l}.

\textbf{Beware:} It is also possible to give the option \texttt{dvips} in addition to the option \texttt{landscape} to the document class declaration to get a landscape paper orientation but you may not be able to define \texttt{specialpaper} as well resulting in a PDF in landscape orientation but not using the A4 paper format.

A.2 Multiple Inclusion of a Picture

Each time you include a picture in your document, the driver processing the DVI file generates code to encapsulate the image’s bitstream into the resulting PostScript file. If for instance you use a logo on each slide, it will be included as many times as there are slides, although it is always the same file. It is even worse if you use overlays as each overlay specification results in a new file!

\textbf{Problem.} The generated PostScript file’s size grows linearly with the number of use of the corresponding picture. If, for instance, you use a 35 KB logo with a 100 slide presentation, your output, as PostScript, will be 3500 KB worth just for the logo!

If you convert the PostScript file to PDF, the presentation will be extremely smaller but its weight will still grow linearly.
2.7 Presentation Styles

All automatically installed Beamer styles have a name of the form `beamerthemestyle`.

In § 2.7.1, a style with a dash is in fact a parametrized style.

E.g., Style `beamerthemesidebar-tab` should be included in the document preamble as \includepackage[tab]{beamerthemesidebar}.

2.7.1 Available Styles

<table>
<thead>
<tr>
<th>Style</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>beamerthemebars</td>
<td>parametrized style</td>
</tr>
<tr>
<td>beamerthemeboxes</td>
<td>parametrized style</td>
</tr>
<tr>
<td>beamerthemeclassic</td>
<td>default style</td>
</tr>
<tr>
<td>beamerthemelined</td>
<td>default style</td>
</tr>
<tr>
<td>beamerthemeplain</td>
<td>default style</td>
</tr>
<tr>
<td>beamerthemeshadow</td>
<td>default style</td>
</tr>
</tbody>
</table>

A. Common Problems Resolution

A.1 Invalid Page Format/Orientation

The PDF file is shown in portrait format while having been prepared for a landscape orientation. Figure A.1 shows the problem. This problem may occur if you use

- \texttt{latex + dvips + ps2pdf},
- \texttt{latex + dvipdf or}
- \texttt{latex + dvipdfm}

If you use \texttt{pdflatex} instead, you should not get this misbehaviour.

Preliminaries

Make sure \texttt{dvips}, \texttt{pdflatex} and other compilation tools involved in a \LaTeX\ compilation process are well configured for your system. In particular, default page format should be set to “A4”. Each \LaTeX\ distribution has its own configuration system so please consult the associated documentation.¹

A.1.1 If you do not use package hyperref

If you did not specify both “landscape” and “a4paper” as document class options, do it yet.

\texttt{dvips} compile your document with option \texttt{-t landscape}.

\texttt{dvipdf} insert \texttt{\special{landscape}} at the beginning of your document and compile it as usual.

\texttt{dvipdfm} compile your document with option \texttt{-l}.

¹If you use \texttt{te\LaTeX}, you may run the command \texttt{texconfig} to configure the whole system.
Figure 6.6: Thumbnails in Adobe Acrobat Reader.
2.7.2 Choosing Another Color Theme

There exists three additional options you may specify with the `\documentclass` definition:

- **red** changes navigation bars and titles to reddish color.
- **brown** changes navigation bars and titles to brownish color.
- **blackandwhite** changes navigation bars and titles to black, white and gray colors.

**Colors**

BEAMER automatically loads `xcolor` package by Uwe Kern, which supports `color` and `pstcol` packages.

**Predefined Colors:** `black`, `blue`, `brown`, `cyan`, `darkgray`, `gray`, `lightgray`, `green`, `magenta`, `orange`, `purple`, `red`, `violet`, `white` and `yellow`.

**Defining new Colors.** You should use `xcolor` definition scheme:

- `\definecolor{lavendar}{rgb}{0.8,0.6,1}`
- `\colorlet{mygreen}{green!60!gray}`
  which means 60% green + 20% gray.
- When a color is needed, you may use directly the method above (e.g., `blue!70` for having a 70% blue).

**Background Color**

- To set solid background color:
  `\beamersetaveragebackground{⟨color⟩}` or `\beamertemplatesolidbackgroundcolor{⟨color⟩}`
- To set gradient background color:
  `\beamertemplateshadingbackground{⟨color1⟩}{⟨color2⟩}`
- To set grid background:
  `\beamertemplategridbackground

2.8 Useful Macros

2.8.1 Removing Navigation symbols

Insert the command `\beamertemplatenavigationsymbolsempty`.

2.8.2 Incremental Highlight

The *incremental highlight* is a way to step through an enumeration of items and displaying in another color (or highlighting) each item as it is introduced (see also § 2.3.3).

6.4 Other PDF Features

6.4.3 Navigation Aids

`\Acrobatmenu{⟨MenuOption⟩}{⟨Text⟩}` allows Adobe Acrobat Reader menu commands to be invoked when clicking on the element (Text) in the PDF document.

This command is provided with the package `hyperref`. Do not forget to specify the driver you use to produce the PDF file as a package option.

E.g., `\usepackage[ps2pdf]{hyperref}`

**Common ⟨MenuOption⟩ Items**

- **File** Open, Close, Print, Quit
- **View** ActualSize, FitVisible, FitWidth, FitPage, FullScreen, FirstPage, PrevPage, NextPage, LastPage, GoToPage, GoBack, GoForward, ShowBookmarks, ShowThumbs
- **Window** ShowHideToolBar, ShowHideMenuBar

**Naming Thumbnails:** The thumbnails, one for each page of the publication, are normally automatically numbered sequentially. However, you may ensure a logical numbering scheme if you use the `hyperref` package with options

```
plainpages = false
pdfpagelabels = true
```

Figure 6.5: Process for including thumbnails in a PDF file.
2.8. Useful Macros

File name: beamer-highlight.tex
Source: taken from the Beamer User Guide

\def\colorize{\protect<#1>{\color{structure!50}(%color{black})\%}}
\def\frame{
  \begin{itemize}
    \item [\colorize{1}] \item First item.
    \item [\colorize{2}] \item Second item.
    \item [\colorize{3}] \item Third item.
    \item [\colorize{4}] \item Fourth item.
  \end{itemize}
}

2.8.3 Using Packages only when Printing

If, for some reason, you wish to include packages only when generating an article out of a presentation, you may include the structure described in the code below in the preamble of the document.

```latex
\usepackage[ps2pdf]{thumbpdf}
\usepackage[pdftex]{thumbpdf}
\usepackage[beamerdefaultarticle]
\usepackage[fullpage]
\usepackage[hyperref]
\usepackage[beamerdefaultpresentation]
```

2.8.4 Structuring Presentation

Often, you may want a certain type of frame to be shown directly after a section or subsection starts. For example, you may wish every subsection to start with a frame showing the table of contents with the current subsection highlighted. To facilitate this, you can use the following commands.

\AtBeginSection[⟨special star text⟩]{{⟨text⟩}}
\AtBeginSubsection[⟨special star text⟩]{{⟨text⟩}}

The given text will be inserted at the beginning of every section. If the ⟨special star text⟩ parameter is specified, this text will be used for starred sections instead. Different calls of this command will not “add up” the given texts (like the \AtBeginDocument command does), but will overwrite any previous text.

Example:

```latex
\AtBeginSection[] \% Do nothing for \section\,.
\AtBeginSubsection[⟨special star text⟩]{{⟨text⟩}}
```

Including Thumbnails

A PDF document may define thumbnail images representing the contents of its pages in miniature form. A viewer application can then display these images on the screen, allowing the user to navigate to a page by clicking its thumbnail image with the mouse (see Figure 6.6).

Recent versions of Adobe Acrobat Reader are able to automatically create thumbnails for an open document. However, these thumbnails will not be saved in the PDF file itself unless you use the full, commercial, version of Adobe Acrobat.

To create thumbnails from \LaTeX, include one of the commands below in the preamble of the document, according to the method you use to produce the PDF.

\usepackage{ps2pdf}{thumbpdf}
\usepackage{pdftex}{thumbpdf}

Then, after having created a first version of the PDF file, run \texttt{thumbpdf}\textsuperscript{6} with the name of the PDF file as argument. This will create the thumbnails.

You then have to redo the whole production process (starting from \LaTeX) to permanently include the generated thumbnails. Figure 6.5 shows the overall process to be done.

\begin{itemize}
  \item \texttt{\usepackage[ps2pdf]{thumbpdf}} \texttt{\usepackage[pdftex]{thumbpdf}}
\end{itemize}

\footnote{With \texttt{--nodea=drips} if you use \texttt{drips} in your production process.}

Figure 6.4: Document Informations in Adobe Acrobat Reader

\begin{tabular}{|c|c|}
  \hline
  \textbf{General Info} & \\
  \hline
  File: \texttt{figs\pev\document.pdf} & Title: Making Presentations with \LaTeX
  \\
  Subject: Guidelines & Author: Xavier Peressburg
  \\
  Keywords: & \hline
  Binding: Left Edge & Creator: \LaTeX
  \\
  Creator: \LaTeX & Producer: \texttt{dvips + AFP Ghostscript 8.14}
  \\
  Modified: Wed May 15 17:06:36 2004 & \hline
  File Size (Bytes): 17024 & Security: None
  \\
  PDF Version: 1.2 (Acrobat 3.x) & Page Size: 219.9 mm x 279.8 mm
  \\
  Number of Pages: 60 & Tagged PDF: No
  \\
  Optimized: No & \hline
\end{tabular}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig6_4.png}
\caption{Document Informations in Adobe Acrobat Reader}
\end{figure}
parameter is specified, this text will be used for starred subsections instead. Different calls of this command will not "add up" the given texts.

Example:

\AtBeginSection[] \% Do nothing for \subsection*
{(\frame<beamer>}
{\frametitle{Outline}}
\{\tableofcontents[ current , currentsubsection]}
)

2.9 Identified Limitations

2.9.1 Verbatim Environment

"\verb" or "verbatim" cannot be directly used in Beamer. However, if there is no overlay, use \frame[containsverbatim] as shown on example below.

2.9.2 In-line Verbatim

Use the command \path{...} instead of \verb.

6.4. Other PDF Features

world. However creating a PDF document should not be associated as only being a solution to provide an easy way to let other people print your La\TeX documents or for you as being the solution for getting animated presentations. One should think of it as a whole entity and give it more informations than just what is "seen".

Figure 6.3: Shell script to set the security of a PDF file.

For this purpose, the PDF file format offers meta-tags for identifying and indexing the documents.

6.4.1 Document’s Informations

Adding more features like links recognition, bookmarks, thumbnails is easily achieved with other options of the package hyperref. We will only show some common features in the rest of this section. A full reference for this package is available at http://www.tug.org/applications/hyperref/manual.html

Bookmarks and Links

The hyperref package extends the functionality of BX\TeX cross references, which in turn can be converted to hyperlinks. Since hyperref overwrites some BX\TeX commands, it is often included as the final package, to make sure that nothing can overwrite it. The backref or pagebackref options in hyperref create so-called back reference to the text passage, and also add links into the bibliography.
#!/bin/sh

title="PDF Security Options"

dialog --title "$title"
    --shadow
    --msgbox "This script allows you to set permissions and a password for a PDF file created from a given PostScript file." on

Choose Permissions:

- Choose Permissions
- Enter (mandatory) Owner Password
- Enter (optional) User Password
- Use the output command line to create your PDF file.

# Permission Check-List
permissions="

dialog --stdout
    --title "$title"
    --shadow
    --msgbox "Please select features permissions": on

print 'Print document (normal quality)' on
modify 'Modify contents of document' on
copy 'Copy text and graphics' on
annotations 'Add/modify annotations' on
forms 'Fill in interactive form fields' on
extract 'Extract text and graphics (accessibility)' on
assemble 'Insert/rotate page, create bookmarks' on
quality 'Print document (high quality)' on


if [ -z "$permissions" ]; then
    permissions=4
fi

# Ask for Owner Password
ownerpassword="
dialog --stdout
    --title "$title"
    --shadow
    --msgbox 'Enter the owner password (mandatory)' on

# Ask for User Password
userpassword="
dialog --stdout
    --title "$title"
    --shadow
    --msgbox 'Enter the user password (optional)' on

# Output command line
echo -n "ps2pdf -OOwnerPassword="$ownerpassword "
echo -n " -OUserPassword="$userpassword "
echo " -dEncryptionR=3 -dKeyLength=128 -dPermissions="$permissions "
echo " in.ps out.pdf"

6.4 Other PDF Features

PDF documents are now the de facto standard for the secure and reliable distribution and exchange of electronic documents and forms around the

This slide contains a few lines of R Jacket using the Verb package environment of package pacsymb!

\node{article}{% only for the article version
  \usepackage{amsmatharticle}
  \usepackage{fullpage}
} \node{presentation}{% only for the presentation version
  \usepackage{beamerthemes}
} To be continued...

Figure 2.4: Including verbatim data in a frame.
4 = Print document
8 = Modify contents of document
16 = Copy text and graphics from document
32 = Add or modify text annotations

To allow printing and copying, but disable modifying the contents and annotations, the value is \(-4 - 8 - 32\) so use \(-dPermissions=-44\). To enable all, use \(-dPermissions=-4\). To disable all, use \(-dPermissions=-64\).

Command line

```
ps2pdf13 -sOwnerPassword=OWNER -sUserPassword=USER \
-dEncryptionR=2 -dKeyLength=40 \
-dPermissions=PERMISSIONS in.ps out.pdf
```

### Revision 3

See [9, Table 3.15, p. 77] for full details of the user access permission values.

4 = Print document (possibly not at the highest quality level, depending on whether 2048 is also set)
8 = Modify contents of document, except as controlled by 32, 256 and 1024
16 = Copy text and graphics from document other than that controlled by 512
32 = Add or modify text annotations, fill in interactive form fields, and if 256 is set, create or modify interactive form fields
256 = Fill in existing interactive form fields, even if 32 is clear
512 = Extract text and graphics (in support of accessibility to disabled users or for other purposes)
1024 = Assemble the document (insert, rotate, or delete pages and create bookmarks or thumbnail images), even if 16 is clear
2048 = Print the document to a representation from which a faithful digital copy of the PDF contents could be generated. When this is clear (and 4 is set), printing is limited to a low-level representation of the appearance, possibly of degraded quality.

To enable all, use \(-dPermissions=-4\). To disable everything apart from viewing, combine the following -4 (base) -4 (print) -8 (modify) -16 (copy) -32 (annotate) -256 (interactive fields) -512 (copy for disability access) -1024 (assemble) -2048 (high quality print), so \(-dPermissions=-3904\).

Command line

```
ps2pdf14 -sOwnerPassword=OWNER -sUserPassword=USER \
-dEncryptionR=3 -dKeyLength=128 \
-dPermissions=PERMISSIONS in.ps out.pdf
```

### Shell Script Automating the Procedure

This script allows you to quickly get the whole list of arguments with a dialog base shell script, as shown on Figure 6.3.

**Requirements:** sh shell, dialog and dc (an arbitrary precision calculator)

**File name:** pdf_security.sh
6.3 PDF Encryption

PDF Encryption is available with Ghostscript’s macros `ps2pdf`. This allows you to restrict the use of the final PDF file. Figure 6.2 shows an example of a few restrictions applied to a document. There are two revisions of the security handler:

Revision 2: Revision 2 does not work with GS 8.14$^2$. It is intended to be used with `KeyLength` equal to 40 bits and produces PDF 1.3 compliant documents; that is, intended to be used with Adobe Acrobat 4.0;

Revision 3: Revision 3 is able to use `KeyLength` up to 128 bits and produces PDF 1.4 compliant documents; that is, intended to be used with Adobe Acrobat 5.0.

### 6.3.1 Common Command Line Parameters

- **-sOwnerPassword** Password allowing full non-restrictive access to the document. Mandatory to use PDF restrictions.
- **-sUserPassword** Password needed for opening the document. This parameter may be left empty.
- **-dEncryptionR** Revision to use. May be either 2 or 3.
- **-dKeyLength** Length of the encryption key. With revision 2, must be equal to 40, with revision 3, should be equal to 128.$^6$
- **-dPermissions** Permissions to apply to the document. See description below.

### 6.3.2 Revision 2

Document Security can be set with the Permissions flag. For `EncryptionR=2`, subtract these values from -4 to disable an access.$^5$

$^5$http://www.ghostscript.com/doc/AFPL/index.htm

$^6$With revision 3 it must actually be a multiple of 8 in the interval [40, 128].

### 3.1 Introduction

Figure 3.1 presents a bird’s-eye view of the structure of a \LaTeX{} file using the `prosper` class. The following code shows a typical usage of the class. Additional references may be found in [4].

```latex
\documentclass[slideColor, pdf, manci] {prosper}
\title{Example Presentation}
\author{Xavier Pere \& egue\`{e}s}
\email{xavier.persorgers@epfl.ch}
\institution{EPFL}\maketitle
\begin{document}
\begin{slide}
Introduction
Nothing more to say!
\end{slide}
\begin{overlays}[3]
\begin{slide}\Dissolve\List displayed step-by-step\end{slide}
\begin{itemstep}
Item Prosper showing
\end{itemstep}
\begin{item}
\textit{a list of items}
\end{item}
\begin{itemstep}
\end{itemstep}
\end{overlays}
\end{document}
```
3.1.1 Options of the Class

To make a presentation using the PROSPER package, you need to specify it in your \documentclass (you can also specify it in a \usepackage command in the preamble). Thus, the first line in the \LaTeX file should be of the form:

\documentclass[[options]]{prosper}

Options:
- final
- draft
- total
- initial
- slideBW
- slideColor
- colorBG
- nocolorBG
- slideBW
- slideColor
- ps
- pdf

Legend:
- Mandatory or default choice
- Optional

Figure 3.1: Structure for a \LaTeX file using prosper

3.1.2 Producing Nice Looking PDF

Let us try to intercept then \( S \) gives no group those substitutions \( w \) of any other variable ground dependencies \( S \in \mathcal{S} \) for \( S \in \mathcal{S} \) does not provide the same sha

Let us try to intercept then \( S \) gives no group those substitutions \( w \) of any other variable ground dependencies \( S \in \mathcal{S} \) for \( S \in \mathcal{S} \), all \( S \) provide the same sha

Figure 6.1: PDF using Type 3 bitmapped fonts (left) and Type 1 scalable fonts (right).

cannot handle those fonts, because there are no font descriptors available. It leaves them embedded in PDF files and renders them very badly, although printing those documents does not make too many differences, if the original resolution was high enough.

There are two solutions to fix this misbehaviour:

1. Include \usepackage{pslatex} in your document preamble. This is a small package that makes \LaTeX default to “standard” PostScript fonts. It is basically a hacked merger of \times.sty and \mathptm.sty. You must have installed standard \LaTeX (\LaTeX 2ε) and PSNFSS PostScript fonts to use this package. The main novel feature is that the pslatex package (unlike \times.sty) tries to compensate for the visual differences between the Adobe fonts by scaling Helvetica by 90%, and “condensing” Courier (i.e. scaling horizontally) by 85%.

2. Use \texttt{dvips -Ppdf} to compile your PostScript file. This will include the PostScript (Type 1) versions of the Computer Modern and \texttt{AMS} fonts, which must be installed. They are not included by default in the \texttt{LaTeX} distribution but they are in the Mik\TeX distribution.\footnote{Go to \texttt{http://www.tex.ac.uk/tex-archive/systems/win32/miktex/1.20} and get the files \texttt{amsfonts.zip} and \texttt{cmps.zip}.} In addition, you may append the \texttt{-G0} (zero) option to \texttt{dvips} that turns off dvips default behaviour (which breaks ligatures in many fonts).

NB: There is a bug in older version of dvips that makes \texttt{-Ppdf} flag incompatible with the \texttt{\usepackage{times}} package. If in doubt, it is safest to use the \texttt{-Pcmz -Pamz} flags to dvips.

6.2.1 Checking Fonts in PDF Files

One way to find out what type of fonts you’ve actually ended up with in a PostScript file is to open the file using Adobe Acrobat Reader. From the “File” drop down menu, select “Document Info→Fonts”. Click on the “List all Fonts” button. Make sure you either scroll the resulting list or expand the popup window sufficiently to see everything that is listed. If you see “Type 3” anywhere in the list, you have done something wrong!
LaTeX and pdfLaTeX behave by default differently concerning the character-spacing. As a result, the same TeX file compiled with LaTeX and pdfLaTeX may show different line-breaks, paragraphs, page-breaks, . . . Fortunately, there is a switch in pdfLaTeX to ensure LaTeX spacing.\footnote{Include \texttt{pdfadjustspacing=1} in the preamble.}

Methods 2 – 4: These are the usual ways to create PDF file.

- \texttt{dvipdf} essentially runs \texttt{dvips} then pumps that output through \texttt{ps2pdf}, a script which uses Ghostscript to generate PDF output. You may add security as if you used \texttt{ps2pdf} (see \S\ 6.3);
- \texttt{dvipdfm} is written from the ground-up to create PDF directly from DVI files and generally does a better job;
- Method 4 allows you to add security to your PDF files (see \S\ 6.3). In fact, the PDF output will be slightly smaller than with \texttt{dvipdf}.

6.1.1 I still want to use pdflatex & pstrick!

pstricks macros cannot be used with pdflatex since pstricks uses PostScript arithmetic, which is not in PDF. As such, a package pdftricks has been written to circumvent this limitation, so that the extensive facilities offered by the powerful pstricks package can be made use of in a pdflatex document. This is brought by making use of the shell escape function available in the web2c TeX compiler,\footnote{The \texttt{pdfTeX} distribution works well.}, while this package is of no use to other commercial implementations.

Shell escape allows this package to suspend a TeX compilation whenever a pstricks block is encountered, to compile the corresponding block into an EPS picture, to convert it to PDF format (needed by pdflatex) and to return to the TeX compilation and finish the job.

http://sarovar.org/projects/pdftricks/

6.2 Producing Nice Looking PDF

A problem that might appear when producing PDF documents is that they are rendered to the screen using ugly bitmapped (type 3) fonts instead of vectorial or outline (type 1) fonts (see Figure 6.1).

"Bitmapped fonts are to typesetting what punch-card machines are to digital storage. They were necessary at one time, when no other viable technology was available, but they have long since been made obsolete. That they are still the default... is at best a sign of laziness and conservatism among the TeX crew, or at worst an inept expression of adoration for Knuth."

Gordon Kindlmann

Explanation: The default installation of dvips uses fonts with a fixed resolution (.pk fonts) encoded as 300dpi (dots per inch) bitmaps. This is unnoticeable for printing; however, the resulting PDF files are barely legible when scaled down to today’s screen resolutions (typically 72dpi). These fonts are embedded in PostScript output as Type 3 fonts. Acrobat Distiller

There are several options that can be specified to the package. You can read about all the options in detail in the documentation that comes with PROSPER. The useful ones are:

- draft compiles a draft version of the presentation, with figures replaced by bounding boxes.
- final compiles a complete version of the presentation with figures and captions in their proper places.
- ps compiles the \LaTeX file to PostScript for printing purposes.
- pdf compiles the \LaTeX file to a PDF format suitable for projectors.

Another important option to specify is which presentation style to use. PROSPER comes with several styles which are described in \S\ 3.5. There are also options to specify slide background colors, slide numbers, etc. In general, unless you require black and white slides (e.g., for printing purposes), you will not need to set any color options in the \texttt{\documentclass}; the style files will manage them for you.

3.2 Preamble

In the section between \texttt{\documentclass} and \texttt{\begin{document}}, you should specify the contents of the title page and some options (such as logos and slide captions) that can be applied to all slides. The normal \LaTeX macros have been redefined to generate the title and associated texts with proper font sizes, etc. Some of the macros available for designing the title slide include:

- \texttt{title} Defines the title of the presentation.
- \texttt{subtitle} Defines the subtitle of the presentation.
- \texttt{author} Defines the author of the presentation.
- \texttt{email} Defines the email of the author.
\institution \ Defines the institution.
\slideCaption \ Puts a caption at the bottom of each slide.
\Logo \ Places a logo on each slide at a specified position:
  \Logo{(-1.2,-1.2)}{\includegraphics{logo-filename}}
\DefaultTransition \ Defines the type of transition that should occur between slides. See also § 5.1, p. 47.

Since the \hyperref package is included by PROSPER, you can use the \href command to include mailto: links or direct hyperlinks to Web pages in the above commands (and, of course, in the rest of your document). As in standard LATEX, the title slide is generated by the \maketitle command in the document body.

3.3 Slides

The “frame” of BEAMER is the “slide” in PROSPER. You still have to inform LATEX the contents to be typeset on each slide. This is easily performed with the environment \slide as shown below:
\begin{slide}
|Title of the slide|
|Contents such as maths, lists|
\end{slide}

3.4 Overlays

Overlays may be used to animate contents of a slide. To create a sequence of elements appearing and disappearing, you have to embed the corresponding slide environment into an \overlays definition:
\begin{overlays}{n}
\begin{slide} \end{slide}
\end{overlays}

The argument \textit{(n)} stands for the number of steps composing the animation. This is a manually computed value.
\fromSlide{(p)}{\textit{(mat)}} \ Puts \textit{(mat)} on slides \textit{(p)} through \textit{(n)}.
\onlySlide{(p)}{\textit{(mat)}} \ Puts \textit{(mat)} on slide \textit{(p)} only.
\untilSlide{(p)}{\textit{(mat)}} \ Puts \textit{(mat)} on slides 1 through \textit{(p)}.

6 Creation of the PDF Files

6.1 To pdflatex or Not?

There exists several ways to produce a PDF file out of a \TeX \ file:
1. \texttt{pdflatex file.tex}
2. \texttt{latex file.tex dvipdf file.dvi}
3. \texttt{latex file.tex dvipdfm file.dvi}
4. \texttt{latex file.tex dvips file.dvi -o ps2pdf file.ps}

Method 1: One of the largest advantages of \texttt{pdfLATEX} (a.k.a. \texttt{pdflatex}) over the conventional \texttt{LATEX} is the fact that \texttt{pdflatex} merges easily pictures in PDF, PNG, JPEG or TIFF format instead of the standard EPS format. However, it cannot deal with typesetting involving the package \texttt{pstricks}. As the other methods support all PDF features and \texttt{pstricks}, you should consider using them (however, package \texttt{Beamer} is not well suited for compiling files with other method than \texttt{pdflatex}).

\footnote{Actually, \texttt{pdfLATEX} is the real name but as you compile your \TeX \ file using \texttt{pdflatex}, we will only refer to this method using the less generic name \texttt{pdfLATEX}.}
### Usage with Package hyperref

**hyperref** provides a command to specify the duration for a given slide of the presentation. This duration \( n \) is expressed in seconds and should be specified right after the start of the corresponding slide in the presentation package you use\(^1\).

```latex
\hypersetup{pdfpageduration=n}
```

#### Advanced Usage

The Dur entry in the page object specifies the page’s **display duration** (also called **advance timing**): the maximum length of time, in seconds, that the page will be displayed before the presentation automatically advances to the next page. (The user can advance the page manually before the specified time has expired.) Example below shows how you may specify a timeout of 4 seconds before automatically advancing to the next slide.

```latex
\special {ps: /pdfmark where {pop} /userdict /pdfmark load put} ifelse
[ {ThisPage} /Dur 4 /PUT pdfmark ]
```

### Performances

Table 5.1 shows the performances of a few Adobe Acrobat Reader versions with automatically advancing a huge document containing only text with a timeout of 0 seconds, which means as fast as possible.

<table>
<thead>
<tr>
<th>Windows XP</th>
<th>Linux 2.4.26</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Version</td>
</tr>
<tr>
<td>4.05</td>
<td>5.05</td>
</tr>
<tr>
<td>5.05</td>
<td>5.1</td>
</tr>
<tr>
<td>5.1</td>
<td>6.0.1</td>
</tr>
<tr>
<td>6.0.1</td>
<td>19.7</td>
</tr>
<tr>
<td>FPS</td>
<td>FPS</td>
</tr>
<tr>
<td>60.3</td>
<td>5.05</td>
</tr>
<tr>
<td>19.7</td>
<td>19.8</td>
</tr>
<tr>
<td>15.4</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.1: Adobe Acrobat Reader’s Performances on a Pentium IV 1.8 GHz, 512 MB RAM

---

\(^1\) **BEAMER**: inside a \texttt{frame} definition but before any contents.

---

### 3.4. Overlays

- **\FromSlide(p)**: All the material after the occurrence of this command will be put on slides \( p \) through \( n \).
- **\OnlySlide(p)**: All the material after the occurrence of this command will be put on slide \( p \) only.
- **\UntilSlide(p)**: All the material after the occurrence of this command will be put on slides 1 through \( p \).

#### 3.4.1 List of Items

If you wish to step through a list of items, you should use the **itemstep** environment instead of the well-known **itemize** environment.

```latex
\begin{itemstep}
\item First item
\item Second item
\item Third item
\end{itemstep}
```

#### 3.4.2 Replacing Contents

The commands **\fromSlide**, **\onlySlide** and **\untilSlide** have the same definition as their unstarred version except that they typeset \( \langle \text{mat} \rangle \) in a zero dimension box, meaning that the position pointer is not moved.

```latex
\begin{onlyInPS}
\includegraphics{example.ep}
\end{onlyInPS}
```

The other useful commands for choosing the contents depending on the chosen mode (PS or PDF) are:

- **\PDFForPS([ifPDF])** ([ifPDF]) Puts contents \( \langle \text{ifPDF} \rangle \) if the chosen mode is PDF, otherwise puts contents \( \langle \text{ifPS} \rangle \).
- **\onlyInPS([contents])** Puts \( \langle \text{contents} \rangle \) only if the chosen mode is PS.
- **\onlyInPDF([contents])** Puts \( \langle \text{contents} \rangle \) only if the chosen mode is PDF.
3.5 Presentation Styles

Prosper already offers several styles tuned for printing slides in both color and black & white, as well as displaying on a screen. Nevertheless, the class has been devised in such a way that it is fairly easy to add your own style if you are dissatisfied with the existing ones. You are strongly encouraged to share Prosper styles you are proud of with other users by sending them to Prosper’s author such that he can add them to the next release.

3.5.1 Available Styles

- **alien glow**
- **autumn**
- **azure blends**
- **capsules contemporain**

5.2 Additional PDF Presentation Features

5.2.1 Presentation Mode

Some PDF viewer application may allow a document to be displayed in the form of a presentation or “slide show”, advancing from one page to the next either automatically or under user control.
\texttt{\textbackslash transsplithorizontal<\textbackslash duration>\textbackslash>\textbackslash>\textbackslash Splits horizontally to the inside.  
\textbackslash (\textbackslash duration)\textbackslash is a number specifying the duration in seconds. Default is 1 sec.

\texttt{\textbackslash transsplitverticalout<\textbackslash duration>\textbackslash>\textbackslash>\textbackslash Splits vertically to the outside.  
\textbackslash (\textbackslash duration)\textbackslash is a number specifying the duration in seconds. Default is 1 sec.

\texttt{\textbackslash transsplitverticalin<\textbackslash duration>\textbackslash>\textbackslash>\textbackslash Splits vertically to the inside.  
\textbackslash (\textbackslash duration)\textbackslash is a number specifying the duration in seconds. Default is 1 sec.

\texttt{\textbackslash transblindshorizontal<\textbackslash duration>\textbackslash>\textbackslash>\textbackslash Horizontal blinds. \textbackslash (\textbackslash duration)\textbackslash is a number specifying the duration in seconds. Default is 1 sec.

\texttt{\textbackslash transblindsvvertical<\textbackslash duration>\textbackslash>\textbackslash>\textbackslash Vertical blinds. \textbackslash (\textbackslash duration)\textbackslash is a number specifying the duration in seconds. Default is 1 sec.

\texttt{\textbackslash transboxout<\textbackslash duration>\textbackslash>\textbackslash>\textbackslash Growing box. \textbackslash (\textbackslash duration)\textbackslash is a number specifying the duration in seconds. Default is 1 sec.

\texttt{\textbackslash transboxin<\textbackslash duration>\textbackslash>\textbackslash>\textbackslash Shrinking box. \textbackslash (\textbackslash duration)\textbackslash is a number specifying the duration in seconds. Default is 1 sec.

\texttt{\textbackslash transwipe[direction=\textbackslash angle\textbackslash]} Wipes from one edge of the page to the facing edge. \textbackslash (\textbackslash angle)\textbackslash is a number between 0 and 360 which specifies the direction (in degrees) in which to wipe.  
According to \[9, \textbackslash 8.3.3\], only the values 0, 90, 180 and 270 are supported.

\texttt{\textbackslash transdissolve<\textbackslash duration>\textbackslash>\textbackslash>\textbackslash Dissolves. \textbackslash (\textbackslash duration)\textbackslash is a number specifying the duration in seconds. Default is 1 sec.

\texttt{\textbackslash transglitter[direction=\textbackslash angle\textbackslash]} Glitters from one edge of the page to the facing edge. \textbackslash (\textbackslash angle)\textbackslash is a number between 0 and 360 giving the direction (in degrees) in which to glitter.  
According to \[9, \textbackslash 8.3.3\], only the values 0, 270 and 315 are supported.

Auto-Advancing

You may use the following command to animate a range of slide at a given period. Example below animates next slides at every 0.5 sec. \textit{See also} \[2.3.5, p. 9.}

\texttt{\textbackslash transduration<2-><0.5>}

5.1.4 Prosper

Default Transition

To specify the default transition to be used for all slides of your presentation, use the command \texttt{\DefaultTransition\{\textbackslash trans\}}.

Transition for a given slide

You may specify a transition to use for a given slide with the parameter \texttt{\textbackslash trans} of the slide’s definition: \texttt{\begin{slide}\{\textbackslash trans\}\{\textbackslash title\}}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{example.pdf}
\caption{Example of slide with Prosper presentation styles.}
\end{figure}
5.1. Adding Slide Transitions

Parameters

D  Optional. The duration of the transition effect, in seconds. Default value: 1.

S  Optional. The transition style to use when moving to this page from another during a presentation. Values are listed in 5.1.1.

Dm  Optional; Split and Blinds transition styles only. The dimension in which the specified transition effect occurs:
   H  Horizontal
   V  Vertical
   Default value: H.

M  Optional; Split and Box transition styles only. The direction of motion for the specified transition effect:
   I  Inward from the edges of the page
   O  Outward from the center of the page
   Default value: I.

Di  Optional; Wipe and Glitter transition styles only. The direction in which the specified transition effect moves, expressed in degrees counterclockwise starting from a left-to-right direction. Only the following values are valid:
   0  Left to right
   90  Bottom to top (Wipe only)
   180  Right to left (Wipe only)
   270  Top to bottom
   315  Top-left to bottom-right (Glitter only)
   Default value: 0.

Figure 5.1 illustrates the relationship between transition duration (D) and display option (Dur, see 5.2.1). Note that the transition duration specified for a page (page 2 in the figure) governs the transition to that page from another page; the transition from the page is governed by the next page’s transition duration.

5.1.3 Beamer

Transition commands in Beamer are inside frame environment.

\transsplithorizontalout\langle\duration\rangle  Splits horizontally to the outside. \(\langle\duration\rangle\) is a number specifying the duration in seconds. Default is 1 sec.
**Box:** a rectangular box sweeps inward from the edges of the page or outward from the center, revealing the new page.

**Wipe:** a single line sweeps across the screen from one edge to the other.

**Dissolve:** the old page dissolves gradually to reveal the new one.

**Glitter:** similar to Dissolve, except the effect sweeps across the page in a wide band moving from one side of the screen to the other in a given direction.

### 5.1.2 Advanced Options

According to [9, § 8.3.3], there exists several options to customize the transitions. If you know what you do, you may include your own PDF tags using the `\special` \LaTeX tag:

```
\special{pdfmark where {pop} {userdict /pdfmark/cleartomark load put} ifelse { {ThisPage} << /Trans << /S/Wipe /D 5 /Di 90 >> >> /PUT pdfmark }
```

### 3.6. How do I...

#### 3.6.1 How can I get a slide in portrait orientation?

If you want to add a ‘slide’ of text between two common slides in portrait orientation, simply enter your text between the end of the previous slide (ended by `\end{slide}`) and the start of the next one (usually indicated by `\begin{slide}`).

#### 3.6.2 How can I incrementally display tables?

To show a table cell-by-cell, you need to use the command `\fromSlide` with the contents of each cell as argument. Do not enclose the column delimiters (§) nor the new line character. Add the comment character (%) at the end of each line to prevent \LaTeX adding unwanted misalignment.

**File name:** `prosper-table-1.tex`

1. `\fromSlide`
2. `\begin{slide}(Table)`
3. `\begin{tabular}{|l|l|l|}`

---

*If you use \TeX, this is achieved with the command `\verbatim`.*
Problem. The whole table is displayed from first slide with empty cells. What should I do?

Solution (preliminary). Create a single table for each step of the animation, allowing us to have a better control on which cell to display or not, and therefore not having empty cells at all.

Cell-by-Cell (preliminary)

File name: prosper-table-2.tex

```latex
\begin{tabular}{|l|l|l|}
\hline
\begin{tabular}{|l|l|l|l|}
\hline
a & b & c & d & e & f & g & h & i & j & k & l & m \\
\hline
\end{tabular} \\
\end{tabular}
```

Problem. Second row display makes cell grow as cell (2,1) is wider than cell (1,1). Problem is similar with cell (2,2) being wider than cell (1,2). Is it possible to get rid of this?

Solution (final). As said before, we need to create different tables for each step of the animation. Environment `tabular` cannot calculate properly the width of cells as it has no information on a second row being displayed later. The solution is hence to force the width of a cell according to the maximal width it will have at the end of the animation. There is a few adjustments to

5 | Presentation Features

Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Adding Slide Transitions</td>
<td>47</td>
</tr>
<tr>
<td>5.1.1 Available Transitions</td>
<td>47</td>
</tr>
<tr>
<td>5.1.2 Advanced Options</td>
<td>48</td>
</tr>
<tr>
<td>5.1.3 Beamer</td>
<td>49</td>
</tr>
<tr>
<td>5.1.4 Prosper</td>
<td>50</td>
</tr>
<tr>
<td>5.1.5 \TeXPower</td>
<td>51</td>
</tr>
<tr>
<td>5.2 Additional PDF Presentation Features</td>
<td>51</td>
</tr>
<tr>
<td>5.2.1 Presentation Mode</td>
<td>51</td>
</tr>
</tbody>
</table>

5.1 Adding Slide Transitions

In the following, \texttt{(trans)} is a reference to the transition style to use when moving to the page from another during a presentation. You might be unable to customize the way the transition is performed, according to the presentation package you use. In this case, you will have to use the \texttt{\special{}} tag as described in \ref{5.1.2} to customize the transition.

5.1.1 Available Transitions

\texttt{R: default option}. The new page simply replaces the old one with no special transition effect.

Split: two lines sweep across the screen, revealing the new page. The lines may be either horizontal or vertical and may move inward from the edges of the page or outward from the center.

Blinds: multiple lines, evenly spaced across the screen, synchronously sweep in the same direction to reveal the new page. The lines may be either horizontal or vertical. Horizontal lines move downward, vertical lines to the right.
We define a custom itemize environment which calls to \mystep:
\newenvironment{myitemize}{\begin{itemize}}{\end{itemize}}
\let\origitem=\item
\let\origmyitem=\myitem
\renewcommand{\origitem}{\item}
\renewcommand{\myitemize}{\begin{itemize}}

4.10 Identified Bugs

You may browse for a list of identified bugs and sometimes workaround solutions on the \TeXPower SourceForge repository:

http://sourceforge.net/tracker/?group_id=60743&atid=495145

3.6. How do I

\begin{slide}{Table}
\begin{tabular}{|l|l|l|}
\hline
a & b & cdefghijklm \\
\hline
\end{tabular}
\end{slide}

be done manually whenever the cell needs to be stretch. The commands at lines 6 and 7 define a box whose width is equal to the largest contents of the first column of the table (line 6) or second column of the table (line 7). You may define other commands for further columns according to your needs.

Cell-by-Cell (final)

File name: prosper-table-3.tex

\begin{slide}
\begin{tabular}{|l|l|l|}
\hline
a & b & cdefghijklm \\
\hline
\end{tabular}
\end{slide}
3.7 Identified Bugs

Black Boxes in Adobe Acrobat Reader

You might encounter strange behaviour with your presentations when using Adobe Acrobat Reader. Sometimes the contents of a slide is replaced by a big black box that disappears when zooming in or out. This seems to be a PROSPER-style related bug and the trick is either to rewrite properly the style or to choose another one.

Other Bugs

You may browse for a list of identified bugs and sometimes workaround solutions on the PROSPER SourceForge repository:

http://sourceforge.net/tracker/?group_id=14812&atid=114812

4.9.2 Incremental Highlight with Permanent Color Change

This second version of the incremental highlight will display non yet active items using a dimmed color, when they become active for the first time, using a highlight color and using another color when they have been activated.

Requirements: package ifthen

File name: texpower-highlight2.tex

Source: transformed from the fulldemo in the \TeX\Power package
4.8.2 How can I incrementally display a table?

Problem: Cells may grow if future contents is wider than the visible contents.

Solution: The most robust way of doing this is to create an empty box with the same dimensions as the text to be hidden. Use the command \boxedsteps to make \step create a blank box the size of (paramcontents) when inactive and put (stepcontents) into a box when active. The dual command is \nonboxedsteps to activate the default behaviour.

4.9 Useful Macros

4.9.1 Incremental Highlight

The incremental highlight is a way to step through an enumeration of items and displaying in another color (or highlighting) each item as it is introduced.

Requirements: package ifthen

File name: texpower-highlight.tex

Source: taken from the fulldemo in the \TeXPower package

4.1 Introduction

The \TeXPower bundle contains style and class files for creating dynamic online presentations with \LaTeX. The heart of the bundle is the package \texttt{texpower.sty} which implements some commands for presentation effects. This includes setting page transitions, color highlighting and displaying pages incrementally.

The package \TeXPower is completely independent of the document class used and the method of PDF creation.

\textbf{Beware:} If you use the \TeX or the Mik\TeX distributions, the class files is not part anymore of them because it has a non-free license from IBM Research Center.

4.1.1 Example

The following code shows a typical usage of the class (see also Figures 4.1, 4.2 and 4.3).
4.8 How do I...

4.8.1 How can I incrementally display a paragraph of text?

Problem: There is a slightly different baseline alignment of the texts displayed incrementally.

Solution: The easiest solution is to use \parstepwise, but if the arguments of \step are long, you’ll get problems with line breaks, as the command \parstepwise forces \step to put its argument in a box. You can use \hidetext like this:

```
\parstepwise[\let\hidestepcontents=\hidetext]
\hidestep{\text{Step one}}
\hidestep{\text{Step two}}
```

But note that \hidetext, being implemented using the soul package, is quite fragile\footnote{Fragile commands, see §1 C.2, p. 73.}.

If you are not using structured backgrounds, \highlightvanished is another alternative which can be used exactly like \hidetext, but is much more robust (note that this will fail whenever your text should appear in front of different background colors, for any reason).

In the argument of \highlightvanished, which uses \textcolor, paragraph breaks are not allowed.
\parstepwise{⟨\colorbox{lightgray}{contents}⟩} Works like \liststepwise, but \boxedsteps
(see §4.8.2) is turned on by default. Use for texts where \steps are to be filled into blank spaces.

Example:
\begin{verbatim}
  \parstepwise{
    \step{A paragraph of text.}
    \step{Another paragraph of text.}
  }
\end{verbatim}

4.7.1 Changing the way \texttt{⟨stepcontents⟩} is displayed

You should think of \texttt{\step} as a command replaced by

\hidestepcontents{⟨stepcontents⟩} When this step is not yet active.

\displaystepcontents{\activatestep{⟨stepcontents⟩}} When this step is activated for the first time.

\displaystepcontents{⟨stepcontents⟩} When this step has already been activated before.

You may redefine these macros to change the behaviour. Simply use an instruction of the form \texttt{\let⟨stepcommand⟩=⟨action⟩}, where \texttt{⟨stepcommand⟩} is one of \hidestepcontents, \displaystepcontents or \activatestep and \texttt{⟨action⟩} may be:

Options for \texttt{\displaystepcontents}

\displayidentical Simply expands to its argument. this is used by default by \activatestep.
\displayboxed Expands to an \texttt{\mbox} containing its argument. This is used by \boxedsteps for interpreting \hidestepcontents
\hideignore Expands to nothing. Used by \nonboxedsteps.
\hidephantom Expands to a \texttt{\phantom} containing its argument. Used by \boxedsteps
\hidevanish In a colored document, makes its argument “vanish” by setting all colors to \texttt{\vanishcolor}.\footnote{It is set to \texttt{\pagecolor} by default.}
\hidetext Produces blank space of the same dimensions as the space that would be taken if its argument would be typeset in the current paragraph. Respects automatic hyphenation and line breaks.
\hidedimmed In a colored document, displays its argument with dimmed colors. This does not make the argument invisible.

4.2 Preamble

In the so-called section \texttt{preamble} between \texttt{\documentclass} and \texttt{\begin{document}}, you should specify the contents of the title page and some options (such as logos and slide captions) that can be applied to all slides. The normal \LaTeX macros have been redefined to generate the title and associated texts with proper font sizes, etc. Some of the macros available for designing the title slide include:

\title Defines the title of the presentation.
\author Defines the author of the presentation.
\MyLogo Puts a logo or whatever you wish at the bottom of each slide, replacing the standard message “Typeset by \TeX.”

\begin{figure}[h]
  \centering
  \includegraphics[width=\textwidth]{example.png}
  \caption{\TeXPower: Title of a presentation.}
  \label{fig:example}
\end{figure}

Color options
whitebackground (default) sets standard colors (see §4.5) to match a white background color.
lightbackground sets standard colors to match a light (but not white) background color.
darkbackground sets standard colors to match a dark (but not dark) background color.
blackbackground sets standard colors to match a black background color.

colorhighlight enables highlight of item (see §4.7.1).
colormath colors all mathematical formulae.
coloremph makes \texttt{\em} and \texttt{\emph} switch colors instead of fonts.
Every page starts with what was on the current page when `\stepwise` started;
- The first page also contains everything in \( \text{(contents)} \) which is not in \( \text{(stepcontents)} \) for any `\step` command;
- The second page additionally contains the \( \text{(stepcontents)} \) for the first \( \text{step} \) command, and so on, until all \( \text{(stepcontents)} \) are displayed;
- When all \( \text{(stepcontents)} \) are displayed, `\stepwise` ends and typesetting is resumed on the current page.

Use of the optional parameters. These should be conditions in the syntax of the `\ifthenelse` (see Listings 4.9.1 and 4.9.2 for examples of use).

\begin{itemize}
\item \( \langle \text{activatefirst} \rangle \) checks whether this `\step` is to be activated for the first time. The default is
\begin{align*}
\text{\value{step}} = \text{\value{stepcommand}}.
\end{align*}
If one uses \( \text{\value{step}} = (n) \), this `\step` can be forced to appear as the \( (n) \)th one.
\item \( \langle \text{whenactive} \rangle \) checks whether this `\step` is to be considered active at all. The default behaviour is to check whether this `\step` has been activated before (this is saved internally for every step).
\end{itemize}

For more information on advanced programming steps, see [11].

\begin{itemize}
\item \( \langle \text{steponce} \rangle \) Like `\step`, but goes inactive again in the subsequent step.
\end{itemize}

\begin{itemize}
\item \( \langle \text{liststepwise} \rangle \) Works exactly like `\stepwise`, but should be used for list environments and aligned equations.
\end{itemize}

\begin{itemize}
\item \( \langle \text{hyperref} \rangle \) is necessary for page transition effects to work (see §5.1). In particular, the `\pageduration` command only works if the version of `\hyperref` loaded is at least v6.7b (where the key `\pdfpageduration` was introduced).
\item `\hidecontent` is necessary for the implementation of the commands `\hidetext` and `\highlighttext`.
\item `\fixseminar` unfortunately, there are some fixes to `\seminar` which can not be applied in the `\TeXPower` packages because they have to be applied
\end{itemize}
Figure 4.4: Panels (left, right, top and bottom) with \TeXPower.

The \texttt{\foothead} command must be used only between paragraphs or at places where ending the current paragraph does not hurt.

\begin{verbatim}
\foothead{A Slide with Pause}

Formula below tells us how we may get energy out of water.

$u = \alpha_2 \ldots$
\end{verbatim}

\texttt{\foothead} may appear before the footer is processed by the output routine, no logo will appear (as if \texttt{\MyLogo{}} were declared). This stays in effect until \texttt{\foothead} is encountered, at which point the contents of \texttt{\MyLogo} are restored. So, for instance, if you do not want the logo to appear at all, you can put the \texttt{\LogoOff} command before the \texttt{\begin{document}} command. If you want the logo only on the title page, then you can put this command after the first occurrence of \texttt{\foilhead}. You can then turn the logo back on by putting the \texttt{\LogoOn} command in a convenient place.

4.3. Toggling the Logo

There is a better method than undefining/redefining \texttt{\MyLogo} for inhibiting a logo from appearing on selected slides or all slides. This feature is implemented with two switches. These macros are \texttt{\LogoOn} and \texttt{\LogoOff} and they do exactly what their names imply. If \texttt{\LogoOff} appears before the footer is processed by the output routine, no logo will appear (as if \texttt{\MyLogo{}} were declared). This stays in effect until \texttt{\LogoOn} is encountered, at which point the contents of \texttt{\MyLogo} are restored. So, for instance, if you do not want the logo to appear at all, you can put the \texttt{\LogoOff} command before the \texttt{\begin{document}} command. If you want the logo only on the title page, then you can put this command after the first occurrence of \texttt{\foilhead}. You can then turn the logo back on by putting the \texttt{\LogoOn} command in a convenient place.

4.4. The Other Three Corners

Since the macros \texttt{\Restriction} and \texttt{\MyLogo} control the bottom left corner of the page, there are other macros for putting text in the other three corners. These are

\begin{verbatim}
\rightheader{(text)}
\leftheader{(text)}
\rightfooter{(text)}
\end{verbatim}

They each take one argument, the text you want to place in the associated corner of the page. These can also be redeclared within the document.
with the appropriate attention paid to the output routine. By default the headers are empty and the lower right footer is just the page number:

- \rightheader{}
- \leftheader{}
- \rightfooter{\quad\textsf{\thepage}}

except on the title page where they are all suppressed. You can easily suppress page numbering by declaring \rightfooter{}.

4.5 Standard Colors

\TeXPower maintains a list of standard colors which are recognized and handled by \TeXPower’s color management. Some commands like \texttt{\dimcolors} affect all standard colors.

```
defineTColor\{\texttt{name}\}\{\texttt{model}\}\{\texttt{def}\}\} Acts like definecolor from the color package, but the color \{name\} is also added to the list of standard colors.
```

```
\addTColor\{\texttt{name}\}\} Adds the existing color \{name\} to the list of standard colors.
```

4.5.1 Slide’s Background Color

As other packages, \TeXPower allows you change the background color. You may choose a single solid color or one of the provided gradient method. The way the background is displayed depends on the color option you set in the package declaration (see § 4.4.2).

```
\texttt{\backgroundstyle\{\texttt{options}\}\{\texttt{style}\}\}} Is the central command for structured page backgrounds. It works like \texttt{\pagestyle} and other commands of this type. This means \{style\} is a symbolic name specifying the general method by which the page background is constructed.
```

The detailed construction is influenced by parameters which can be set in \{options\}, which should be a comma-separated list of \{key\}:=\{value\} pairs.

Using a Gradient Background

\{style\} may have one of the following values\(^1\):

- \texttt{vgradient} Vertical gradient. The page background is constructed using the \texttt{\gradrule} command. If there are panels (, the gradient rule fills the rectangular space left between the specified panels.

\texttt{Options: \{stripes\}, \texttt{startcolor}, \texttt{endcolor}}.

- \texttt{hgradient} Horizontal gradient. The page background is constructed using the \texttt{\gradrule} command.

- \texttt{doublevgradient} Double vertical gradient. The page background is constructed using the \texttt{\dblgradrule} command.

\texttt{Options: \{stripes\}, \texttt{startcolor}, \texttt{midcolor}, \texttt{endcolor}}.

\(^1\)Only a few options are listed here. See [11] for a full specification.

4.6 Panels

Panels can be added to your slides anchored either at left, right, top or bottom, as shown on Figure 4.4.

```
\texttt{\documentclass\{\texttt{a4paper}, landscape\}\{\texttt{foils}\}}
\usepackage\{\texttt{graphicx, color, hyperref, soul, fixseminar}\}
\usepackage\{\texttt{darkbackground, display\}}\{\texttt{texpower}\}
\title\{First Presentation\}
\author\{Xavier Pers\} egers\}
\LogOff
\rightfooter{}
\begin\{document\}
\DeclarePanel\{\texttt{right}\}\% My Panel
\\texttt{\hfill} % if panel is at top/bottom
\\texttt{\vfill} % if panel is at left/right
\\button\{\texttt{Acrobatmenu\{PrevPage\}\{Back\}}\}
\\button\{\texttt{Acrobatmenu\{NextPage\}\{Next\}}\}
\texttt{\backgroundstyle\{\texttt{stripes=50, startcolor=red, endcolor=black}\}\{\texttt{vgradient}\}}
\maketitle
```

\textbf{Beware:} If you use panels and a gradient background, you have to declare the gradient after the panel.

The command \texttt{\button} takes four optional arguments which are left out in the example below. These are \{width\}, \{height\}, \{depth\} and \{alignment\} in that order. If given, \{width\}, \{height\}, \{depth\} set the dimensions of the framed area comprising the button, without the shadow. If specified, the optional parameter \{alignment\} (one of \texttt{l}, \texttt{c}, \texttt{r}) gives the alignment of the text inside the button box. See 6.4.3 for a list of available buttons you may add to your panels.

4.7 Overlays

As there exists no concept of “slide” in \TeXPower as this was the case with BEAMER or PROSPER, the concept of overlays is a bit easier with \TeXPower.

```
\pause Ships out the current page, starts a new page and copies whatever was on the current page onto the new page, where typesetting is resumed. This creates the effect of a pause in the presentation. This
```