Abstract

We present a document class from which we can generate both course slides and course notes in a transparent way.

Contents

1 Introduction 2

2 The User Interface 2

2.1 Package Options .............................................. 2
2.2 Notes and Slides .............................................. 2
2.3 Header and Footer Lines ..................................... 4
2.4 Colors and Highlighting ...................................... 4
2.5 Front Matter, Titles, etc ..................................... 4
2.6 Miscellaneous ............................................... 4

3 Limitations 4

4 The Implementation 4

4.1 Class and Package Options ................................. 4
4.2 Notes and Slides .............................................. 6
4.3 Header and Footer Lines ..................................... 9
4.4 Colors and Highlighting ...................................... 10
4.5 Sectioning ..................................................... 11
4.6 Miscellaneous ............................................... 13

*Version ? (last revised ?)
1 Introduction

The mikoslides document class is derived from beamer.cls [Tana], it adds a "notes version" for course notes derived from the omdoc class [Kohlhase:smomdl] that is more suited to printing than the one supplied by beamer.cls.

2 The User Interface

The mikoslides class takes the notion of a slide frame from Till Tantau’s excellent beamer class and adapts its notion of frames for use in the §TeXand OMDoc. To support semantic course notes, it extends the notion of mixing frames and explanatory text, but rather than treating the frames as images (or integrating their contents into the flowing text), the mikoslides package displays the slides as such in the course notes to give students a visual anchor into the slide presentation in the course (and to distinguish the different writing styles in slides and course notes).

In practice we want to generate two documents from the same source: the slides for presentation in the lecture and the course notes as a narrative document for home study. To achieve this, the mikoslides class has two modes: slides mode and notes mode which are determined by the package option.

2.1 Package Options

The mikoslides class takes a variety of class options:\footnote{EdNote: leaving out noproblems for the moment until we decide what to do with it.}

- The options slides and notes switch between slides mode and notes mode (see Section 2.2).
- If the option sectocframes is given, then special frames with section table of contents are produced headers \footnote{EdNote: document the functionality}.
- showmeta. If this is set, then the metadata keys are shown (see Kohl16 for details and customization options).
- If the option frameimages is set, then slide mode also shows the \frameimage- generated frames.
- topsect = \langle sect \rangle can be used to specify the top-level sectioning level; the default for \langle setc \rangle is section.

2.2 Notes and Slides

Slides are represented with the frame just like in the beamer class, see Tanb for details. The mikoslides class adds the note environment for encapsulating the...
course note fragments \footnote{MK: it would be very nice, if we did not need this environment, and this should be possible in principle, but not without intensive LaTeX trickery. Hints to the author are welcome.}

\begin{note}
\begin{frame}
\frametitle{The first slide}
...
\end{frame}
\begin{note}
... and more explanatory text
\end{note}
\begin{frame}
\frametitle{The second slide}
...
\end{frame}

\begin{example}
A typical Course Notes File

By interleaving the \texttt{frame} and \texttt{note} environments, we can build course notes as shown in Figure \footnote{MK: it would be very nice, if we did not need this environment, and this should be possible in principle, but not without intensive LaTeX trickery. Hints to the author are welcome.}

Sometimes, we want to integrate slides as images after all – e.g. because we already have a PowerPoint presentation, to which we want to add STEXnotes. In this case we can use \texttt{\frameimage{}{\langle\text{opt}\rangle}\{\langle\text{path}\rangle\}}\footnote{MK: it would be very nice, if we did not need this environment, and this should be possible in principle, but not without intensive LaTeX trickery. Hints to the author are welcome.}, where \langle\text{opt}\rangle\footnote{MK: it would be very nice, if we did not need this environment, and this should be possible in principle, but not without intensive LaTeX trickery. Hints to the author are welcome.} are the options of \texttt{\includegraphics} from the \texttt{graphicx} package \footnote{MK: it would be very nice, if we did not need this environment, and this should be possible in principle, but not without intensive LaTeX trickery. Hints to the author are welcome.} and \langle\text{path}\rangle\footnote{MK: it would be very nice, if we did not need this environment, and this should be possible in principle, but not without intensive LaTeX trickery. Hints to the author are welcome.} is the file path (extension can be left off like in \texttt{\includegraphics}).

If we want to transclude the contents of a file as a note, we can use the \texttt{\inputref} macro. \texttt{\inputref{foo}}\footnote{MK: it would be very nice, if we did not need this environment, and this should be possible in principle, but not without intensive LaTeX trickery. Hints to the author are welcome.} is equivalent to
\begin{verbatim}
\begin{note}
\inputref{foo}
\end{note}
\end{verbatim}

There are some environments that tend to occur at the top-level of \texttt{note} environments. We make convenience versions of these: e.g. the \texttt{nomtext} environment is just an \texttt{omtext} inside a \texttt{note} environment (but looks nicer in the source, since it avoids one level of source indenting). Similarly, we have the \texttt{nomgroup} environment.
2.3 Header and Footer Lines

2.4 Colors and Highlighting

\textwarning The \textwarning macro generates a warning sign: △

2.5 Front Matter, Titles, etc

2.6 Miscellaneous

3 Limitations

In this section we document known limitations. If you want to help alleviate them, please feel free to contact the package author. Some of them are currently discussed in the \TeXGitHub repository [\TeX].

1. when option book which uses \texttt{pagestyle\{headings\}} is given and semantic macros are given in the \texttt{omgroup} titles, then they sometimes are not defined by the time the heading is formatted. Need to look into how the headings are made. This is a problem of the underlying \texttt{omdoc} package.

4 The Implementation

4.1 Class and Package Options

We define some Package Options and switches for the \texttt{mikoslides} class and activate them by passing them on to \texttt{beamer.cls} and \texttt{omdoc.cls} and the \texttt{mikoslides} package. We pass the nontheorem option to the \texttt{statements} package when we are not in notes mode, since the \texttt{beamer} package has its own (overlay-aware) theorem environments.

\begin{verbatim}
1 ⟨∗cls⟩
2 \RequirePackage{kvoptions}
3 \RequirePackage{etoolbox}
4 \SetupKeyvalOptions{family=mks@cls,prefix=mks@cls@}
5 \DeclareStringOption[article]{class}
6 \AddToKeyvalOption*{class}{{\PassOptionsToClass{class=\mks@cls@class}{omdoc}}
7 \ifdefstring{\mks@cls@class}{book}{{\PassOptionsToPackage{topsect=part}{mikoslides}}}{}
8 \ifdefstring{\mks@cls@class}{report}{{\PassOptionsToPackage{topsect=part}{mikoslides}}}{}
9 \DeclareBoolOption{notes}
10 \DeclareComplementaryOption{slides}{notes}
11 \DeclareDefaultOption{{\PassOptionsToClass{\CurrentOption}{omdoc}}
12 \PassOptionsToClass{\CurrentOption}{beamer}
13 \PassOptionsToPackage{\CurrentOption}{mikoslides}}
14 \ProcessKeyvalOptions{mks@cls}
15 ⟨/cls⟩
\end{verbatim}

now we do the same for the \texttt{mikoslides} package.

\begin{verbatim}
16 ⟨/package⟩
\end{verbatim}

4
Depending on the options, we either load the article-based omdoc or the beamer class (and set some counters).

In notes mode, we also have to make the beamer-specific things available to article via the beamerarticle package. We use options to avoid loading theorem-like environments, since we want to use our own from the \texttt{STEX} packages. The first batch of packages we want are loaded on \texttt{mikoslides.sty}. These are the general ones, we will load the \texttt{STEX}-specific ones after we have done some work (e.g. defined the counters \texttt{m*}). Only the \texttt{stex-logo} package is already needed now for the default theme.
finally, we require the metakeys package from \LaTeX{}, so that we can use the \texttt{\textbackslash addmetakey} mechanism.

\section*{4.2 Notes and Slides}

For the lecture notes cases, we also provide the \texttt{\textbackslash usetheme} macro that would otherwise from the the \texttt{beamer} class. While the latter loads \texttt{beamertheme(theme).sty}, the notes version loads \texttt{beamernotetheme(theme).sty}.^3

\[3\] EdNote: MK: This is not ideal, but I am not sure that I want to be able to provide the full theme functionality there.

EdN:3

\begin{itemize}
\item We define the sizes of slides in the notes. Somehow, we cannot get by with the same here.
\item The \texttt{\textit{note}} environment is used to leave out text in the \texttt{slides} mode. It does not have a counterpart in OMDoc. So for course notes, we define the \texttt{n\textit{ote}} environment to be a no-operation otherwise we declare the \texttt{\textit{note}} environment as a comment via the \texttt{\textbackslash comment} package.
\end{itemize}

\[3\] EdNote: MK: This is not ideal, but I am not sure that I want to be able to provide the full theme functionality there.
We first set up the slide boxes in \texttt{article} mode. We set up sizes and provide a box register for the frames and a counter for the slides.

\begin{itemize}
\item\texttt{frame}
\item\texttt{allowframebreaks}
\item\texttt{allowdisplaybreaks}
\item\texttt{fragile}
\item\texttt{shrink}
\item\texttt{squeeze}
\item\texttt{t}
\end{itemize}

We define the keys.

We define the environment, read them, and construct the slide number and label.

We redefine the \texttt{itemize} environment so that it looks more like the one in \texttt{beamer}.

\begin{itemize}
\item\texttt{outer}
\item\texttt{inner}
\end{itemize}

We redefine the \texttt{itemize} environment so that it looks more like the one in \texttt{beamer}.

\begin{itemize}
\item\texttt{outer}
\item\texttt{inner}
\end{itemize}
We create the box with the `mdframed` environment from the equinymous package.

```latex
\begin{mdframed}[linewidth=\slideframewidth,skipabove=1ex,skipbelow=1ex,userdefinedwidth=\slidewidth,align=center]\sf
```

Now, we need to redefine the frametitle (we are still in course notes mode).

```latex
\frametitle
```

```latex
\renewcommand{\frametitle}[1]{\Large\bf\sf\color{blue}{#1}}\medskip
```

`\frameimage` We have to make sure that the width is overwritten, for that we check the `\Gin@ewidth` macro from the `graphicx` package

```latex
\newrobustcmd{\frameimage}[2][]{%
\stepcounter{slide}%
\ifmks@sty@frameimages%
\def\Gin@ewidth{}
\setkeys{Gin}{#1}%
\ifmks@sty@notes\else\vfill\fi%
\ifx\Gin@ewidth\@empty%
\mycgraphics[width=\slidewidth,#1]{#2}%;
\else\mycgraphics[#1]{#2}%;
\fi%
\par\strut\hfill{\footnotesize Slide \arabic{slide}}%
\ifmks@sty@notes\else\vfill\fi%
\fi%
}% ifframeimages
```

```latex
\pause
```

```latex
\ifmks@sty@notes\newcommand{\pause}{}\fi
```

`nomtext`

```latex
\ifmks@sty@notes\newenvironment{nomtext}[1][]{\begin{omtext}[#1]}{\end{omtext}}%
```

`nomgroup`

```latex
\ifmks@sty@notes\newenvironment{nomgroup}[2][]{\begin{omgroup}[#1]{#2}}{\end{omgroup}}%
```

\footnote{\textbf{EdNote}: MK@DG; we need to do that in the LaTeXML binding as well!}

\footnote{\textbf{EdNote}: MK: fake it in notes mode for now}
### 4.3 Header and Footer Lines

Now, we set up the infrastructure for the footer line of the slides, we use boxes for the logos, so that they are only loaded once, that considerably speeds up processing.

\[\text{\texttt{\textbackslash setslidelogofn}}\]

The default logo is the logo of Jacobs University. Customization can be done by \texttt{\textbackslash setslidelogofn{⟨logo name⟩}}.

\[\text{\texttt{\textbackslash newlength\{slidelogoheight\}}}\]

149 \texttt{\textbackslash ifmks@sty@notes\%
150 \texttt{\textbackslash setlength\{slidelogoheight\}{.4cm}\%
151 \texttt{\textbackslash else\%
152 \texttt{\textbackslash setlength\{slidelogoheight\}{1cm}\%
153 \texttt{\textbackslash fi\%
154 \texttt{\textbackslash newsavebox\{slidelogo\}\%
155 \texttt{\textbackslash sbox\{slidelogo\}{\TeX}\%
156 \texttt{\textbackslash newrobustcmd\{\textbackslash setslidelogo\}[1]{%
157 \texttt{\textbackslash sbox\{slidelogo\}\{\includegraphics[height=\slidelogoheight]{#1}\}}%
158 \texttt{\textbackslash \}}%
\]

\[\text{\texttt{\textbackslash setsource\}}\]

\texttt{\textbackslash source} stores the writer’s name. By default it is \textit{Michael Kohlhase} since he is the main user and designer of this package. \texttt{\textbackslash setsource\{⟨name⟩\}} can change the writer’s name.

\[\text{\texttt{\textbackslash \def\source{Michael Kohlhase}\%
160 \texttt{\textbackslash newrobustcmd\{\textbackslash setsource\}[1]{\def\source{#1}}}\%}
\]

\[\text{\texttt{\setlicensing\}}\]

Now, we set up the copyright and licensing. By default we use the Creative Commons Attribution-ShareAlike license to strengthen the public domain. If package \texttt{hyperref} is loaded, then we can attach a hyperlink to the license logo. \texttt{\setlicensing\{⟨url⟩\}{⟨logo name⟩}} is used for customization, where \texttt{⟨url⟩} is optional.

\[\text{\texttt{\textbackslash \def\copyrightnotice\{\footnotesize copyright:\hspace{.3ex}{\textbackslash source}\}}\%}
162 \texttt{\textbackslash newsavebox\{cclogo\}\%
163 \texttt{\textbackslash sbox\{cclogo\}\{\includegraphics[height=\slidelogoheight]{cc_somerights}\}}%
164 \texttt{\textbackslash newif\textbackslash ifchref\{ccreffalse\}%
165 \texttt{\textbackslash AtBeginDocument\{%
166 \texttt{\textbackslash @ifpackageloaded\{hyperref\}{\usebox\{cclogo\}\{\textbackslash chreffalse\}}\%
167 \texttt{\textbackslash \}}%
168 \texttt{\textbackslash \}\%
169 \texttt{\textbackslash \}\%
170 \texttt{\textbackslash \}\%
171 \texttt{\textbackslash \}\%
172 \texttt{\textbackslash \}}%
173 \texttt{\textbackslash \}}%
174 \texttt{\textbackslash \}}%
175 \texttt{\textbackslash \}}%
176 \texttt{\textbackslash newrobustcmd\{\setlicensing\}[2]{}\%
177 \texttt{\textbackslash \textbackslash \textbackslash \}\%
178 \texttt{\textbackslash sbox\{cclogo\}\{\includegraphics[height=\slidelogoheight]{#2}\}}%
179 \texttt{\textbackslash \textbackslash \textbackslash \}
4.4 Colors and Highlighting

We first specify sans serif fonts as the default.

Now, we set up an infrastructure for highlighting phrases in slides. Note that we use content-oriented macros for highlighting rather than directly using color markup. The first thing to do is to adapt the green so that it is dark enough for most beamers.

\AtBeginDocument{% 
\definecolor{green}{rgb}{0,.5,0}%
\definecolor{purple}{cmyk}{.3,1,0,.17}%
}%

We customize the \defemph, \notemph, and \stDMemph macros with colors for the use in the statements package. Furthermore we customize the \@@lec macro for the appearance of line end comments in \lec.

\STpresent#1\textcolor{blue}{#1}
\defemph#1\textcolor{magenta}{#1}
\notemph#1\textcolor{magenta}{#1}
\stDMemph#1\textcolor{blue}{#1}
\@@lec#1\textcolor{green}{#1}

I like to use the dangerous bend symbol for warnings, so we provide it here.

\textwarning as the macro can be used quite often we put it into a box register, so that it is only loaded once.

\pgfdeclareimage[width=.8em]{miko@small@dbend}{dangerous-bend}

\textbf{EdNote:} see that we can use the themes for the slides some day. This is all fake.
4.5 Sectioning

If the `sectocframes` option is set, then we make section frames. We first define a set of counters.

```latex
\ifmks@sty@sectocframes%
\ifdefstring\mks@sty@topsect{part}{% 
\newcounter{mpart}\newcounter{mchapter}\newcounter{msection}[mchapter]}% 
\ifdefstring\mks@sty@topsect{chapter}{% 
\newcounter{mchapter}\newcounter{msection}[mchapter]}% 
\ifdefstring\mks@sty@topsect{section}{% 
\newcounter{msection}}% 
\ifdefstring\mks@sty@topsect{subsection}{% 
\newcounter{msubsection}[msection]}% 
\ifdefstring\mks@sty@topsect{subsubsection}{% 
\newcounter{msubsubsection}[msubsection]}% 
\ifdefstring\mks@sty@topsect{subsubsubsection}{% 
\newcounter{msubsubsubsection}[msubsubsection]}% 
\fi% ifsectocframes
```

Now that we have defined the counters, we can load the \TeX-specific packages (in particular statements that needs these counters).

```latex
\RequirePackage{stex} \RequirePackage{smglom} \RequirePackage{tikzinput}
```

Finally, we set the `\section@level` macro that governs sectioning.

```latex
\section@level=2 \ifdefstring\mks@sty@topsect{part}{\section@level=0}{}% 
\ifdefstring\mks@sty@topsect{chapter}{\section@level=1}{}% 
```

\textit{EdNote:} I forget: why not use the counters from beamer/article? –¿ document this.
Now \LaTeX is loaded, we redefine the \texttt{omgroup} environment to produce section toc frames (if the option \texttt{sectocframes} is specified.)

\begin{verbatim}
\ifmks@sty@notes\else% only in slides
\def\part@prefix{\ifdefstring\mks@stysection\part{\arabic{mchapter}.}{}}
\renewenvironment{omgroup}\[2\][\]{%
  \metasetkeys{omgroup}{#1}\sref@target%
  \advance\section@level by 1\%
  \ifmks@sty@sectocframes%
  \stepcounter{slide}
  \begin{frame}[noframenumbering]%
  \vfill\Large\centering%
  \red{%
    \ifcase\section@level\or
    \stepcounter{mpart}\
    \def\@0label{Part \Roman{mpart}}%
    \def\currentsectionlevel{part} \or
    \stepcounter{mchapter}\
    \def\@0label{Chapter \arabic{mchapter}}\%
    \def\currentsectionlevel{chapter} \or
    \stepcounter{msection}\
    \def\@0label{\part@prefix\arabic{msection}}\%
    \def\currentsectionlevel{section} \or
    \stepcounter{msubsection}\
    \def\@0label{\part@prefix\arabic{msection}.:\arabic{msubsection}}%%
    \def\currentsectionlevel{subsection} \or
    \stepcounter{msubsubsection}\
    \def\@0label{\part@prefix\arabic{msection}.:\arabic{msubsection}.:\arabic{msubsubsection}}%end ifcase
    \@0label\sref@label@id\@0label
    \quad #2%%
  \vfill%
  \end{frame}%%
  \fi %ifmks@sty@sectocframes
}\fi%
\end{verbatim}

\textbf{EdNote: MK: we should probably just redefine omgroup@num and omgroup@nonum, since they do the actual work so that we can add the sectocframes behavior here without having to copy the internals. Then there is less material that can get out of sync. Additionally, we should have a hook in the original code of those so that we can increment the slides counter in notes node (to keep slides in sync)
4.6 Miscellaneous

We set up a \texttt{beamer} template for theorems like ams style, but without a block environment.

\begin{verbatim}
\def\inserttheorembodyfont{\normalfont}
\defbeamertemplate{theorem begin}{miko}{\inserttheoremheadfont\inserttheoremname\inserttheoremnumber\ifx\inserttheoremaddition\@empty\else (\inserttheoremaddition)\fi%\inserttheorempunctuation\inserttheorembodyfont\xspace}
\defbeamertemplate{theorem end}{miko}{ }
\end{verbatim}

and we set it as the default one.

\texttt{\setbeamertemplate{theorems}[miko]}

The following fixes an error I do not understand, this has something to do with \texttt{beamer} compatibility, which has similar definitions but only up to 1.

\begin{verbatim}
\expandafter\def\csname Parent2\endcsname{}
\end{verbatim}

We need to disregard the columns macros introduced by the \texttt{beamer} class in the notes.

\begin{verbatim}
\ifmks@sty@notes%
\renewenvironment{columns}[1][]{
\par\noindent%
\begin{minipage}{\slidewidth\centering\leavevmode}%
\begin{lrbox}{\columnbox}\begin{minipage}{#2}%
\end{minipage}\end{lrbox}\usebox{\columnbox}
\end{minipage}\par\noindent%
\end{verbatim}

\begin{verbatim}
\ifmks@sty@noproblems%
\newenvironment{problems}{}{}
\else%
\excludecomment{problems}
\fi%

⟨/package⟩
Change History

v0.1
General: Initial Version ........... 1

v0.2
General: course notes back on seminar ................. 1

v0.3
General: changing to Jacobs logo 1

v0.4
General: moving line-end-comment to omdoc.dtx .............. 1
re-basing the whole thing on beamer ................. 1

v0.5
General: eliminating mytwocolumns, this is better done by beamer.cls .............. 1

v0.9
General: basic options handling for

v1.0
the frame environment in notes mode ....................... 1
numbered sectocframes .............. 1
this is almost done .............. 1

v1.1
General: adding \frameimage ... 1

v1.2
General: moving MathHub support out to separate package .... 1
reinterpreting omgroup .............. 1
Removing the old title macros (use the regular ones instead). 1

General: changed to keyval class/package options, allowed arbitrary classes .............. 1

References


[sTeX] KWARC/sTeX. URL: https://github.com/KWARC/sTeX (visited on 05/15/2015).

[Tana] Till Tantau. beamer – A \LaTeX{} class for producing presentations and slides. URL: http://ctan.org/pkg/beamer (visited on 01/07/2014).