

# Slides and Course Notes for Jacobs University\*

Michael Kohlhase  
Jacobs University, Bremen  
<http://kwarc.info/kohlhase>

November 6, 2010

## Abstract

We present a document class from which we can generate both course slides and course notes in a transparent way. Furthermore, we present a set of  $\LaTeX$ XML bindings for these, so that we can also generate OMDoc-based course materials, e.g. for inclusion in the `ACTIVEMATH` system.

## Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Introduction</b>                        | <b>2</b> |
| <b>2</b> | <b>The User Interface</b>                  | <b>2</b> |
| 2.1      | Package Options . . . . .                  | 2        |
| <b>3</b> | <b>Limitations</b>                         | <b>2</b> |
| <b>4</b> | <b>The Implementation</b>                  | <b>2</b> |
| 4.1      | Initialization and Class Options . . . . . | 2        |
| 4.2      | Notes and Slides . . . . .                 | 4        |
| 4.3      | Header and Footer Lines . . . . .          | 6        |
| 4.4      | Colors and Highlighting . . . . .          | 6        |
| 4.5      | Front Matter, Titles, etc . . . . .        | 7        |
| 4.6      | Miscellaneous . . . . .                    | 8        |
| 4.7      | Finale . . . . .                           | 10       |

---

\*Version ? (last revised ?)

# 1 Introduction

This Document class is derived from `beamer.cls`, specializes it with Jacobs stuff and adds a notes version that is more suited to printing than the one supplied by `beamer.cls`.

## 2 The User Interface

### 2.1 Package Options

`showmeta` The `mikoslides` class takes a single option: `showmeta`. If this is set, then the metadata keys are shown (see [Koh10] for details and customization options).<sup>1</sup>

EdNote(1)

## 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 `STEXTRAC` [Ste].

1. none reported yet

## 4 The Implementation

The `mikoslides` package generates two files: the `LATEX` package (all the code between `<*package>` and `</package>`) and the `LATEX`ML bindings (between `<*ltxml>` and `</ltxml>`). We keep the corresponding code fragments together, since the documentation applies to both of them and to prevent them from getting out of sync.

### 4.1 Initialization and Class Options

For the `LATEX`ML bindings, we make sure the right perl packages are loaded.

```
1 <*ltxml>
2 # -*- PERL -*-
3 package LaTeXML::Package::Pool;
4 use strict;
5 use LaTeXML::Package;
6 </ltxml>
```

For `LATEX` we define some Package Options and switches for the `mikoslides` class and activate them by passing them on to `beamer.cls` the appropriate packages.

```
7 <*cls>
8 \DeclareOption{showmeta}{\PassOptionsToPackage{\CurrentOption}{metakeys}}
9 \newif\ifnotes\notesfalse
```

---

<sup>1</sup>EDNOTE: document the functionality

```

10 \newif\ifproblems\problemstrue
11 \DeclareOption{notes}{\notestruer}
12 \DeclareOption{slides}{\notesfalse}
13 \DeclareOption{nopproblems}{\problemsfalse}
14 \ifnotes\else\DeclareOption*{\PassOptionsToClass{\CurrentOption}{beamer}}\fi
15 \ProcessOptions
16 </cls>
17 <*txml>
18 RawTeX('\newif\ifnotes\notesfalse');
19 RawTeX('\newif\ifproblems\problemsfalse');
20 </txml>

```

Depending on the options, we either load the `article`-based `omdoc` or the `beamer` class. In the first case, 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  $\text{\TeX}$  packages.

```

21 <*cls>
22 \ifnotes
23 \LoadClass{omdoc}
24 \RequirePackage{a4wide}
25 \RequirePackage{marginnote}
26 \RequirePackage[notheorems,noamsthm]{beamerarticle}
27 \else
28 \LoadClass[notheorems,noamsthm,10pt]{beamer}
29 \newcounter{Item}
30 \newcounter{paragraph}
31 \newcounter{subparagraph}
32 \newcounter{Hfootnote}
33 \usetheme{Jacobs}
34 \fi
35 </cls>
36 <*txml>
37 LoadClass('omdoc');
38 DefConstructor('\usetheme{','}');
39 </txml>

```

now, we load the remaining packages for both versions.

```

40 <*cls>
41 \RequirePackage{stex}
42 \RequirePackage{latxml}
43 \RequirePackage{amssymb}
44 \RequirePackage{tikz}
45 \usepgflibrary{shapes}\usetikzlibrary{arrows}
46 \RequirePackage{url}
47 \RequirePackage{amsmath}
48 \RequirePackage{comment}
49 </cls>
50 <*txml>
51 RequirePackage('stex');

```

BegOP(2)

EndOP(2)

```
52 RequirePackage('latexml');
53 RequirePackage('amssymb');
54 RequirePackage('graphicx');
55 RequirePackage('tikz');
56 RequirePackage('amsmath');
57 </ltxml>
```

The `etex.sty` package is needed in the L<sup>A</sup>T<sub>E</sub>X, since it makes the extended the number of internal registers of the `etex` program available for processing. These tend to run out with S<sup>T</sup>E<sub>X</sub> otherwise.

## 4.2 Notes and Slides

We define the sizes of slides in the notes. Somehow, we cannot get by with the same here.

```
58 <*cls>
59 \newcounter{slide}
60 \newlength{\slidewidth}\setlength{\slidewidth}{13.5cm}
61 \newlength{\slideheight}\setlength{\slideheight}{9cm}
62 </cls>
63 <*ltxml>
64 DefRegister('\slidewidth'    => Dimension('13.5cm'));
65 DefRegister('\slideheight'  => Dimension('9cm'));
66 </ltxml>
```

For course notes, we define the `note` environment to be a no-operation otherwise we declare the `note` environment as a comment via the `comment` package.

**note**

```
67 <*cls>
68 \ifnotes\renewenvironment{note}{}{}\else\excludecomment{note}\fi
69 </cls>
70 <*ltxml>
71 DefEnvironment('{note}', '#body');
72 </ltxml>
```

the next step is to set up the slide boxes in `article` mode. We set up sizes and provide a box register for the frames and a counter for the slides.

```
73 <*cls>
74 \ifnotes
75 \newlength{\slideframewidth}\setlength{\slideframewidth}{2pt}
76 \newsavebox{\myframebox}
77 </cls>
```

**frame** For the `frame` environment we construct a `lrbox` in the `\myframebox` register that we can later put into an `\fbox` so that it looks like a slide. Furthermore, we redefine the `itenize` environment so that it looks more like the one in `beamer` with `Jacobs` theme.

---

<sup>2</sup>OLD PART: we do not seem to load this any more, what to do there?

```

78 <*cls>
79 \renewenvironment{frame}[1] []%
80 {\stepcounter{slide}
81 \def\itemize@level{outer}
82 \def\itemize@outer{outer}
83 \def\itemize@inner{inner}
84 \renewcommand\metakeys@show@keys[2]{\marginnote{\scriptsize ##2}}
85 \renewenvironment{itemize}
86 {\ifx\itemize@level\itemize@outer\def\itemize@label{\rhd}\fi
87 \ifx\itemize@level\itemize@inner\def\itemize@label{\scriptstyle\rhd}\fi
88 \begin{list}
89   {\itemize@label}
90   {\setlength{\labelsep}{.3em}\setlength{\labelwidth}{.5em}\setlength{\leftmargin}{1.5em}}
91   \edef\itemize@level{\itemize@inner}}
92 {\end{list}}
93 \noindent\hfill\begin{lrbox}{\myframebox}
94   \begin{minipage}{\slidewidth}\sf%
95   {\miko@slidelabel\end{minipage}\end{lrbox}%
96 \begin{center}\fbox{\usebox\myframebox}\end{center}\hfill}
97 </cls>
98 <*txml>
99 DefEnvironment('{frame}[]',
100   "<omdoc:omgroup layout='slide' ?#locator(stex:srcref='#locator')()>"
101   . "#body\n"
102   . "</omdoc:omgroup>\n\n",
103   afterDigestBegin=>sub {
104     $_[1]->setProperty(theory=>LookupValue('current_module')); });
105 </txml>#\$

```

the next step is to set up the slide boxes in `article` mode.

```

106 <*cls>
107 \renewcommand{\frametitle}[1]{\Large\bf\sf\color{blue}{#1}}
108 \fi
109 \makeindex
110 </cls>
111 <*txml>
112 DefConstructor('\frametitle{',
113   "\n<omdoc:metadata ?#locator(stex:srcref='#locator')()><dc:title>#1</dc:title></omdoc:metadat
114 </txml>

```

We start by giving the L<sup>A</sup>T<sub>E</sub>X XML binding for the `frame` environment from the `beamer` class. The `note` environment is used to blend out text in the `slides` mode. It does not have a counterpart in OMDoc.

```

115 <*cls>
116 \ifproblems\newenvironment{problems}{}{}\else\excludcomment{problems}\fi
117 </cls>
118 <*txml>
119 DefEnvironment('{problems}', '#body');
120 </txml>

```

### 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.

```
121 <*cls>
122 \newlength{\slidelogoheight}
123 \ifnotes\setlength{\slidelogoheight}{.4cm}\else\setlength{\slidelogoheight}{1cm}\fi
124 \newsavebox{\slidelogo}\sbox{\slidelogo}{\includegraphics[height=\slidelogoheight]{jacobs-logo}}
```

Now, we set up the copyright and licensing, the copyright remains with the author, but we use the Creative Commons Attribution-ShareAlike license to strengthen den public domain. Here the problem is that we want a hyperref on the CC logo, if hyperref is loaded, and otherwise not. As hyperref is always loaded, we have to find out at the beginning of the document whether it is, set up a switch, and later in the footer line decide what to do.

```
125 \def\source{Michael Kohlhase}% customize locally
126 \def\copyrightnotice{\footnotesize\copyright:\hspace{.3ex}{\source}}
127 \newsavebox{\cclogo}\sbox{\cclogo}{\includegraphics[height=\slidelogoheight]{cc_somerights}}
128 \newif\ifcchref\cchreffalse
129 \AtBeginDocument{\ifpackageloaded{hyperref}{\cchreftrue}{\cchreffalse}}
130 \def\licensing{\ifcchref\href{http://creativecommons.org/licenses/by-sa/2.5/}{\usebox{\cclogo}}
```

Now, we set up the slide label for the article mode<sup>4</sup>

`\slidelabel`

```
131 \newcommand{\miko@slidelabel}%
132 {\vbox to \slidelogoheight{\vss\hbox to \slidewidth%
133 {\licensing\hfill\copyrightnotice\hfill\arabic{slide}\hfill\usebox{\slidelogo}}}}
134 </cls>
```

### 4.4 Colors and Highlighting

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

```
135 <*cls>
136 \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`.

---

<sup>3</sup>EDNOTE: subtitle is difficult to model in DC metadata. I guess that we want to collect the subtitle into dc:title

<sup>4</sup>EDNOTE: see that we can use the themes for the slides some day. This is all fake.

```

137 % \def\STpresent#1{\textcolor{blue}{#1}}
138 \def\defemph#1{\textcolor{magenta}{#1}}
139 \def\notemph#1{\textcolor{magenta}{#1}}
140 \def\stDMemph#1{\textcolor{blue}{#1}}
141 \def\@@lec#1{\textcolor{green}{#1}}
142 \</cls>
143 \<*txml>
144 #DefMacro('defemph{ }', '\textcolor{magenta}{#1}');
145 #DefMacro('notemph{ }', '\textcolor{magenta}{#1}');
146 \</txml>

```

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

```

147 \<*cls>
148 \def\textwarning{\raisebox{-0.05cm}{\includegraphics[width=1.2em]{dangerous-bend}}\xspace}
149 \</cls>
150 \<*txml>
151 DefMacro('textwarning', "");
152 \</txml>

```

## 4.5 Front Matter, Titles, etc

We need to redefine the frontmatter macros inherited from the `beamer` class, since there they take an optional argument.

```

153 \<*txml>
154 DefMacro('title[]{}', '\@add@frontmatter{ltx:title}{#1}');
155 DefMacro('date[]{}', '\@add@frontmatter{ltx:date}[role=creation]{#1}');
156 DefMacro('author[]{}', sub { andSplit(T_CS('@author'),$_[1]); });#$
\</txml>

```

Now, we specialize the slide environment that we have implemented above or inherited from `seminar.cls` for some abbreviations, e.g. separator slides and title slides.

```

157 \<*cls>
158 \newcommand\titleframe{\begin{frame}\titlepage\end{frame}}
159 \newenvironment{titleframewith}{\begin{frame}\titlepage}{\end{frame}}
160 \newenvironment{ttitle}{\begin{center}\LARGE\begin{tabular}{|c|}\hline}%
161 {\hline\end{tabular}\end{center}}\vspace{1ex minus 1ex}}
162 \newenvironment{ttitlejoint}[1]%
163 {\newbox\boxwith\setbox\boxwith\hbox{\begin{tabular}{c}{\em joint work with}\#1\end{tabular}}%
164 \begin{center}\LARGE\begin{tabular}{c}\color{red}}%
165 {\box\boxwith\end{tabular}\end{center}}%
166 \vspace{1ex minus 1ex}}
167 \</cls>
168 \<*txml>
169 DefConstructor('titleframe', "<omdoc:ignore>titleframe elided here</omdoc:ignore>");
170 DefEnvironment('titleframewith',
171               "<omdoc:ignore>begin elided titleframe</omdoc:ignore>"
172               . "#body"
173               . "<omdoc:ignore>end elided titleframe</omdoc:ignore>");
174 DefEnvironment('titleslide', "");

```

```

175 DefEnvironment('{titleslide}', "<omdoc:omgroup ?#locator(stex:srcref='#locator')()>#body</omdoc:
176 DefEnvironment('{ttitle}', "\n<dc:title>#body</dc:title>");
177 </ltxml>

178 %      Must be first command on slide to make positioning work.
179 <*cls>
180 \newcommand{\putgraphicsat}[3]{%
181 \begin{picture}(0,0)\put(#1){\includegraphics[#2]{#3}}\end{picture}}
182 \newcommand{\putat}[2]{\begin{picture}(0,0)\put(#1){#2}\end{picture}}
183 </cls>
184 <*ltxml>
185 </ltxml>

```

## 4.6 Miscellaneous

Some shortcuts that use math symbols but are not mathematical at all; in particular, they should not be translated by L<sup>A</sup>T<sub>E</sub>X<sub>M</sub>L.

```

186 <*cls>
187 \newcommand\hateq{\ensuremath{\hat{=}}}
188 \newcommand\textleadsto{\ensuremath{\leadsto}}
189 </cls>
190 <*ltxml>
191 DefConstructor('\hateq', "\x{03C2}");
192 DefConstructor('\textleadsto', "\x{219D}");
193 </ltxml>

```

We need to disregard the columns macros introduced by the `beamer` class

```

194 <*ltxml>
195 DefEnvironment('{columns}', '#body');
196 DefEnvironment('{column}{-}', '#body');
197 </ltxml>

```

We also need to deal with overlay specifications introduced by the `beamer` class.<sup>5</sup>

```

198 <*ltxml>
199 DefConstructor('\uncover', '#1');
200 #Define a Beamer Overlay Parameter type
201 DefParameterType('BeamerOverlay', sub {
202   my ($gullet) = @_;
203   my $tok = $gullet->readXToken;
204   if (ref $tok && ToString($tok) eq '<') {
205     $gullet->readUntil(T_OTHER('>'));
206   } else {
207     $gullet->unread($tok) if ref $tok;
208     undef; }},
209   reversion=> sub {

```

<sup>5</sup>EDNOTE: this is just to keep latexml quiet, no real functionality here.

<sup>6</sup>EDNOTE: Deyan: We reuse the CMP itemizations defined in the `omdoc.cls.ltxml` binding, adjusting the parameters to be overlay-sensitive



```

210 (T_OTHER('<'), $_[0]->revert, T_OTHER('>'));
211     });
212
213 #Take the "from" field of the overlay range
214 sub overlayFrom {
215     return "" unless defined $_[0];
216     my $overlay=ToString($_[0]); $overlay =~ /^(\\d+)/; $1;}
217
218 #Reuse the CMP itemizations, only adjust the \item constructors.
219 DefMacro('\beamer@group@item[] OptionalBeamerOverlay IfBeginFollows', sub {
220     my($gullet,$tag,$overlay,$needwrapper)=@_;
221     $overlay=$overlay||T_OTHER("");
222     ( T_CS('\group@item@maybe@unwrap'),
223       ($needwrapper ? (Invocation(T_CS('\beamer@group@item@wrap'),$tag,$overlay)->unlist) : ()))
224 DefConstructor('\beamer@group@item@wrap {} OptionalBeamerOverlay',
225     "<omdoc:omtext ?#locator(stex:srcref='#locator')() ?#2(overlay='&overlayFrom(#2)')()>"
226     . "?#1(<dc:title>#1</dc:title>())"
227     . "<omdoc:CMP ?#locator(stex:srcref='#locator')()><omdoc:p ?#locator(stex:srcref="
228     beforeDigest=>sub {
229     Let('\group@item@maybe@unwrap','\group@item@unwrap');
230     #$_[0]->bgroup;
231     useCMPItemizations();
232     return; },
233     properties=>sub{ RefStepItemCounter(); });
234
235
236 DefConstructor('\beamer@itemize@item[] OptionalBeamerOverlay',
237     "<omdoc:li ?#locator(stex:srcref='#locator')() ?#2(overlay='&overlayFrom(#2)')() >"
238     . "?#1(<dc:title ?#locator(stex:srcref='#locator')()>#1</dc:title>())",
239     properties=>sub{ RefStepItemCounter(); });
240 DefConstructor('\beamer@enumerate@item[] OptionalBeamerOverlay',
241     "<omdoc:li ?#locator(stex:srcref='#locator')() ?#2(overlay='&overlayFrom(#2)')() >"
242     . "?#1(<dc:title ?#locator(stex:srcref='#locator')()>#1</dc:title>())",
243     properties=>sub{ RefStepItemCounter(); });
244 DefConstructor('\beamer@description@item[] OptionalBeamerOverlay',
245     "<omdoc:di ?#locator(stex:srcref='#locator')() ?#2(overlay='&overlayFrom(#2)')() >"
246     . "?#1(<omdoc:dt ?#locator(stex:srcref='#locator')()>#1</omdoc:dt>())<omdoc:dd>", # trus
247     properties=>sub{ RefStepItemCounter(); });
248
249 #We ALWAYS use the beamer itemizations for the slides:
250 Let('\CMP@itemize@item'=>'\beamer@itemize@item');
251 Let('\CMP@enumerate@item'=> '\beamer@enumerate@item');
252 Let('\CMP@description@item'=> '\beamer@description@item');
253 Let('\group@item' =>'\beamer@group@item');
254 Let('\itemize@item'=>'\beamer@group@item');
255 Let('\enumerate@item'=>'\beamer@group@item');
256 Let('\description@item'=>'\beamer@group@item');
257 Let('\only'=>'\beamer@group@item');
258 </ltxml>#&

```

Now, some things that are imported from the `pgf` and `beamer` packages:

```
259 <*ltxml>
260 DefMacro('putgraphicsat{}{}{}', 'mygraphics[#2]{#3}');
261 DefMacro('putat{}{}', '#2');
262 </ltxml>
```

## 4.7 Finale

Finally, we set the slide body font to the sans serife, and we terminate the  $\LaTeX$ ML bindings file with a success mark for perl.

```
263 <cls>\ifnotes\else\sf\fi
264 <ltxml>1;
```

## Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in **roman** refer to the code lines where the entry is used.

showmeta= `\subitem *+\showmeta+, \usage{2}`

## References

- [Koh10] Michael Kohlhase. *metakeys.sty: A generic framework for extensible Metadata in L<sup>A</sup>T<sub>E</sub>X*. Self-documenting L<sup>A</sup>T<sub>E</sub>X package. Comprehensive T<sub>E</sub>X Archive Network (CTAN), 2010. URL: <http://www.ctan.org/tex-archive/macros/latex/contrib/stex/metakeys/metakeys.pdf>.
- [Ste] *Semantic Markup for LaTeX*. Project Homepage. URL: <http://trac.kwarc.info/sTeX/> (visited on 12/02/2009).