

# How to make presentations with L<sup>A</sup>T<sub>E</sub>X

Ignas Anikevicius

August 24, 2011

## Contents

1	The beamer document class	1
2	Using a Standard Document Class	2
3	Thoughts on a Clever Work-flow	2

## Todo list

■	Add a note about GROUP YOUR SLIDES	2
■	finish about this kind of work-flow	3

This tutorial is for the production of scientific presentations using the L<sup>A</sup>T<sub>E</sub>X typesetting system. There are numerous ways of making a presentation in L<sup>A</sup>T<sub>E</sub>X mostly because there are a lot of views on how presentations should look like and how they should be prepared. The most convenient way is to use common packages such as `beamer`, `prospcr`, `slides`, however, one can also tame the standard document classes, such as `scrartcl` [1].

There is also another source of information [2], which might be a good starting point for using the Beamer document class. Also, only the `beamer` and the `scrartcl` ways of making presentations will be presented here as the first seems to be the most popular and reliable way of making presentations, whereas the second way brings more insights on how you can customize a class to typeset almost any document.

## 1 The beamer document class

As mentioned before, the `beamer` package is the most popular among scientists and it works quite well with PGF/TikZ packages which might make it the best solution out there. It is very straight forward to use it — it defines several new environments (e.g. `\frame`, `\columnnt` etc.) and commands (e.g. `\frametitle`, `\pause` etc.).

The best way to learn anything is by example. Since this is not an exception, there are very good tutorials on the website which describe the use of `beamer` document class. The list is as follows:

**A Beamer tutorial by Amber Smith<sup>1</sup>** This is just a showcase of what `beamer` is capable of. However, the author does not mention the `TikZ` package, which is actually written by the same author and it integrates with `beamer` very well.

**A Beamer tutorial by Charles Batts<sup>2</sup>** This tutorial explains the beamer capabilities very well and gives lots of examples. I do think, that it is very suitable for learning and getting acquainted with `beamer`.

**Scientific Presentations using KOMA script classes<sup>3</sup>** Although this link contains an article about making scientific presentation using a standard `scrartcl` class provided by KOMA script, it actually contains several very good references on how to make your presentations look nice and informative. **Beamer** presentations might look much better with minimal amount of effort and this article becomes very useful as it points out some mistakes, which are made by inexperienced presenters. Reading this will definitely help you to get professional presentations quickly.

**Beamer by Example<sup>4</sup>** This is yet another  $\text{PracTeX}$  article and it is also on teaching the reader how to use `beamer` class to get well looking presentations.

**Dual screen beamer presentations<sup>5</sup>** This article give a very interesting insight on how one can adapt `beamer` for dual screen presentations. Recommended for advanced `beamer` and  $\text{\LaTeX}$  users.

By following the links to the  $\text{PracTeX}$  journal articles one could find example-files as well as sources for the article. Examining those files might prove to be a very valuable experience and it is highly recommended to check out the examples if they are supplied in the archive file.

## 2 Using a Standard Document Class

This is an alternative method to create presentations. It involves using `scrartcl` class which is then customized to a great extent. The decorations of the presentation are done by using the `TikZ` package and since everything is done from scratch, the user will end up having a unique theme for his/her presentations.

As you might understand, with a lot of customizability comes slightly steeper learning curve and one need to spend more time initially to get everything set up. However, then to convert an article to a presentation is much faster, which is mainly the point of this method. Also, you do not need to learn new commands/environments, which might also be considered as an advantage of this method.

There is an excellent example and article on the aforementioned  $\text{PracTeX}$  journal [1]. Please download the zipped sources where you will find everything. You can check the presentation `.tex` file and change various parameters in the preamble and see how the style of presentation is affected.

## 3 Thoughts on a Clever Work-flow

As might know from the mentioned resources, when making presentation one mostly has to divide the content to slides either by some sort of manual page breaks or by encapsulating everything in a `frame` environment which will automatically do that. There is also another issue, that one might spend quite a lot of time while making a slide so that everything would look very good and redoing all the slides for every new presentation or talk one has to give might be too time consuming. These are the reasons why a clever working system is very important and it might decrease the time consumption substantially without sacrificing the quality of the slides.

---

Do not make all your slides in one file as it has several drawbacks:

**File becomes very large** It gets harder and harder to find things when one needs to edit something or do other kind of tinkering.

**Harder to reuse content** It is not very convenient to search in big files for information every time you need to access some information you have already typeset.

**Hard to deduce the length of a part** It is much harder to get an impression of the length of some section of the presentation as the file gets bigger because the line count of the file does not reflect the individual section sizes at all.

A much more clever approach would be to make slides in batches. For example, if one has 4 sections in a presentation, then 4 different files will ease the organization substantially. Some people even tend

Add a note about  
GROUP YOUR  
SLIDES

to have all slides as separate `.tex` files, which get compiled when needed and then the resultant `.pdf` files are used when needed. This has a huge benefit, that every slide will look exactly as intended, however, it also means, that such features as table of contents or the highlighting of the current section must be either hard-coded or not shown at all.

Therefore, a good work-flow would consist of having different `.tex` files for different subtopics and then combining them when needed. Using the `beamer` package would ensure that all text will appear on the slide and that it will be of required aspect-ratio and resolution. What is more, it will give those very handy features like linking and table of contents inside the document.

finish about this  
kind of work-flow

## References

- [1] Markus Kohm Marius Hofert. Scientific presentations with latex. *The PracTeX Journal*, 2010(2).
- [2] Wikibooks. Latex wikibooks: Presentations.