

# Org Beamer quick reference card

Fabrice Niessen

June 5, 2014

# Plan



# Citations

# Description

Welcome to **Org Beamer reference card**. It contains the reference documentation that describes how to **write presentations** using:

- Org mode 8 and
- the  $\text{\LaTeX}$  Beamer class.

Those **free** tools allow you to easily produce **high quality PDF files** which are going to look on *every* computer exactly the way they looked on *your* computer.

# Objectives

As they are created **like any other Org mode document**, preparing presentations with Org mode is very different from preparing them with WYSWYG programs such as PowerPoint, Impress or Keynote.

The obvious advantage of this approach is that **you don't have to know L<sup>A</sup>T<sub>E</sub>X** in order to **create Beamer presentations**.

# Requirements

A working  $\text{\LaTeX}$  installation is required for exporting to PDF. If it is not yet installed on your system, install  $\text{\TeX}$  Live (for example).

# Creating a title page

A title page is automatically inserted into the first frame.

By default, it will arrange the following elements on the title page:

- the document **title** (file name, if none specified),
- the **author(s)**'s names (user-full-name, if none specified), and
- a **date** (\today, if none specified).

```
#+TITLE:      Document title  
#+AUTHOR:     Author's name  
#+DATE:       Date
```

The author's email can be included with:

```
#+AUTHOR:      \href{mailto:email@example.com}{Author}  
#+AUTHOR:      \texorpdfstring{Author\newline\url{email@example.com}}{Author} % DOES  
#+BEAMER_HEADER: \author{\texorpdfstring{Author\newline\url{email@example.com}}{Au
```

Other elements:

- the document **subtitle**,
- their affiliation (**institute**), and
- a **title graphic**

can be included with the following commands:



# Global structure

Org mode presentations contain headings at different levels.

By default,

- Headings at the **first** outline level will become **titles** of the different slides (called **frames** in Beamer),
- Deeper levels will be used as structural environments, and
- The “Contents” frame is blank.

# Creating a table of contents

If you set the H option from the `#+OPTIONS:` keyword such as:

```
#+OPTIONS: H:2
```

then:

- Top-level headlines become **sections** listed in the **table of contents** (created by default), and
- **Second**-level headlines become the **frames**.

In many themes, sections and subsections (when H:3) appear in the sidebar or headline.

# Creating a simple frame

```
* Introduction

** A title
   #+BEAMER: \framesubtitle{A subtitle}

Some content.
```

The **subtitle** does not have an Org syntax because it's specific to the Beamer back-end only.

# Create a handout

```
##LATEX_CLASS_OPTIONS: [handout]
```

```
##LaTeX_HEADER: \usepackage{pgfpages}
```

```
##LaTeX_HEADER: \mode<handout>\pgfpagesuselayout{2 on 1}[a4paper,border shrink=5mm]
```



# Present a bibliography

```
#+LATEX_CLASS: beamer  
#+LATEX_CLASS_OPTIONS:
```

Common options:

- 8pt, 9pt, 10pt, **11pt**, 12pt, 14pt, 17pt, 20pt
- draft: no graphics, footlines,...
- handout: no overlays

,*#+LATEX\_CLASS\_options*: [bigger,allowframebreaks]

# L<sup>A</sup>T<sub>E</sub>X preamble

Append any line of code in the L<sup>A</sup>T<sub>E</sub>X preamble with:

```
#+BEAMER_HEADER: \usepackage{...}
```

It will go in the [EXTRA] placeholder of the header associated to the beamer L<sup>A</sup>T<sub>E</sub>X class (see [org-latex-classes](http://org-latex-classes)).

Other possibilities:

```
#+LaTeX_HEADER: \institute[short name]{My institute 1}  
#+LaTeX_HEADER_EXTRA: \institute[short name]{My institute 2}  
#+BEAMER_HEADER: \institute[short name]{My institute 3}
```



# Affiliated keywords

The Beamer back-end reads both

- `#+ATTR_LATEX:` and
- `#+ATTR_BEAMER:`

affiliated keywords.

# Appearance of the presentation

```
#+BEAMER_THEME: Boadilla
```

is equivalent (for Boadilla) to:

```
#+BEAMER_COLOR_THEME: dolphin  
#+BEAMER_FONT_THEME: default  
#+BEAMER_INNER_THEME: [shadow]rounded  
#+BEAMER_OUTER_THEME: infolines
```

# Beamer back-end (for Org export engine)

Type:

```
M-x load-library RET ox-beamer RET
```

to load the Beamer back-end library, and to obtain **extra commands** in the  $\text{\LaTeX}$  export menu:

C-c C-e 1 B Export as  $\text{\LaTeX}$  buffer (Beamer).

C-c C-e 1 b Export as  $\text{\LaTeX}$  file (Beamer).

C-c C-e 1 P **Export as PDF file** (Beamer).

C-c C-e 1 0 Export as PDF file and open (Beamer).

# Structure editing support

Type:

```
M-x org-beamer-mode RET
```

to load the minor mode `org-beamer-mode` **easing the edition** of the **document structure** (through the key binding `C-c C-b`, which offers fast selection of a Beamer environment).

You can also turn it on with:

```
#+STARTUP: beamer
```

in your document.

# For a column view of options and configurations for the individual frames

```
#+COLUMNS: %45ITEM %10BEAMER_env(Env) %10BEAMER_act(Act) %4BEAMER_col(Col) %8BEAMER  
#+COLUMNS: %20ITEM %13BEAMER_env(Env) %6BEAMER_envargs(Args) %4BEAMER_col(Col) %7B
```

## Environment specification (BEAMER\_env property)

- This becomes visible through the B\_frame tag (visual aid only).

- Headlines become frames when their level is equal to `org-beamer-frame-level` (or H value in the OPTIONS line).
- Though, if a headline in the current tree has a `BEAMER_env` property set to either `frame` or `fullframe`, its level overrides the variable, giving you some flexibility in deciding **what is** and what isn't **a frame**. This works in both “directions”: to **add or** to **remove sectioning levels** above the current headline (which becomes a frame)!

- A `fullframe` is a frame with an ignored title (`frametitle` is set to the empty string).



# Blocks

Environment specification ( $\text{BEAMER}_{envproperty}$ )

Use a different **block type for the current “block” environment** (default: block).

structureenv environment

- For highlighting text.
- To help the audience see the structure of your presentation.

Paragraph Heading.

block environment

Answered Questions

How many primes are there?

Open Questions

# Blocks

Environment specification ( $\text{BEAMER}_{envproperty}$ )

Use a different **block type for the current “block” environment** (default: block).

structureenv environment

- For highlighting text.
- To help the audience see the structure of your presentation.

Paragraph Heading.

block environment

Answered Questions

How many primes are there?

Open Questions

# Blocks

## Environment specification ( $\text{BEAMER}_{envproperty}$ )

Use a different **block type for the current “block” environment** (default: block).

## structureenv environment

- For highlighting text.
- To help the audience see the structure of your presentation.

Paragraph Heading.

## block environment

### Answered Questions

How many primes are there?

### Open Questions

## Special cases

You can add an appendix (frames that you do not intend to show during your talk, but which might be useful to answer a question) by inserting such a **level 1 headline** after the last regular slide of your actual presentation:

```
* Appendix material follows                                :B_appendix:
:PROPERTIES:
:BEAMER_env: appendix
:END:

# Backup slides
```

Ignoring page number in backup slides can be achieved by setting the option `noframenumbering` on all “backup” slides.

### noteNH

Note with its title ignored.

### againframe

You can “continue” frames that you previously started somewhere (but where certain details have been suppressed) at a much later point (for example only in the appendix) to show additional slides there.

overprint

> What may not be easy or possible is to use the directive, which is > what I used in my previous response to you.

You can always use the only environment. <https://github.com/suvayu/.emacs.d/blob/master/org-mode-config.el#L215>

That said, I think overlays with only is not as smooth as with simple overlay specifications to regular environments or macros like

, etc.

As for an :overlay specification, I believe it is already supported but only for lists (ox-beamer.el:725). I would love to have that for images too!

Set **overlay** specifications in current block to **create dynamic effects** (*multiple slides*, called *overlays*, for a single frame) = old BEAMER\_envargs property.

Headlines support the BEAMER\_act property:

```
* Headline
:PROPERTIES:
:BEAMER_act: [+~]
:END:

# Diff with [<+>]?

- Item
- Item
```

It is translated as:

- an overlay/action specification, or

XXX <> seem to be added when they aren't present. Copied as is if present.

Dynamic lists are possible on a case by case basis :

```
#+ATTR_BEAMER: :overlay +-
- Item 1
```

Set **overlay** specifications in current block to **create dynamic effects** (*multiple slides*, called *overlays*, for a single frame) = old BEAMER\_envargs property.

Headlines support the BEAMER\_act property:

```
* Headline
:PROPERTIES:
:BEAMER_act: [+~]
:END:

# Diff with [<+~>]?

- Item
- Item
```

It is translated as:

- an overlay/action specification, or
- a default overlay specification when enclosed within square brackets.

XXX <> seem to be added when they aren't present. Copied as is if present.

Dynamic lists are possible on a case by case basis :

```
#+ATTR_BEAMER: :overlay +-
- Item 1
```

# The Queen's old armchair

- Princess Anne
- Prince Charles
- corgis



# Question on ML

```
\begin{figure}  
  \begin{center}  
    \includegraphics<1>[width=.7\textwidth]{figure1}  
    \includegraphics<2>[width=.7\textwidth]{figure2}  
    \includegraphics<3->[width=.7\textwidth]{figure3}  
  \end{center}  
\end{figure}
```

The following works for me:

```
#+beamer: \only<1>{  
  [[file:figure1.png]]  
#+beamer: }\only<2>{  
  [[file:figure2.png]]  
#+beamer: }\only<3->{  
  [[file:figures3.png]]  
#+beamer: }
```

There is the BEAMER\_act property that can be used to apply overlay information on blocks but I don't think it's possible on individual figures. Of course, you could put each figure in a separate block. The following/attached will match what you had originally.

```
#+options: H:1
```

```
* The slide
```

Insert **optional arguments for the current frame environment** using the BEAMER\_OPT property.

XXX or block? See <http://orgmode.org/manual/Beamer-export.html>. I'd still like to see something more like a “for-dummies” explanation of passing options and arguments to L<sup>A</sup>T<sub>E</sub>X entities. I'm not saying the documentation is woefully inadequate (hardly that – Suvayu's page got me rather far, and I got stuck on a couple of details). My experience was: it never would have occurred to me on my own to use the headline text for L<sup>A</sup>T<sub>E</sub>X code, and if there was a hint anywhere in the docs to suggest that this would be the way to go, I didn't find it. That's a conceptual leap that passed me by.

- This is for frames, and for environments within a frame
- It specifies options for the current frame or block, and will automatically be enclosed within square brackets.
- fragile option is added automatically
- You might want to put allowframebreaks=0.9 there



# Splitting a frame into multiple columns

To get multiple columns in a frame:

- 1 Press `C-c C-b | (BMCOL)` on the headlines (inside the frame) which will become columns  
The headline of column environments won't be outputted in the PDF file.
- 2 Specify the **column width** as a **percentage** of `\textwidth`  
**!CAUTION! No absolute width**, such as 4cm, which wouldn't be correctly translated...

Instead of `block`, those structural environments will become `column` (with the width parameter as a factor of `\textwidth`).

Consecutive `column` environments will be put in a `columns` environment.

Two  
lines.

One line (but aligned).

## Multiple columns



column

columns

> How can I put options that would apply to a frame (e.g., > )?

```
#+BI--ND: org-beamer-frame-default-options "allowframebreaks"
```

for allowing frame breaks for the whole document,

\* A very long slide for allowing on a frame by frame basis.

## Frame structure (Explicit page breaking) I

If the text does not fit on a single slide, all you have to do to automatically break up the frame into several frames, is set the option `allowframebreaks`:

```
** A long "frame" with breaks
:PROPERTIES:
:BEAMER_opt: allowframebreaks,label=
:END:
```

Until the Beamer issue #265 is solved, we need to unset the `framelabel` as shown above (`label=`).



# Vertical alignment

You can specify *top* **vertical alignment** globally by the `t` class option:

```
#+LaTeX_CLASS_OPTIONS: [t]
```

For single frames, you can use the same option locally:

```
* Vertically top-aligned
:PROPERTIES:
:BEAMER_opt: t
:END:

...
```

You can add that **special property** by editing the `Opt` column within the “column view” (first press `C-c C-x C-c`).

## Result of an evaluation on two columns

Balancing text in columns.

... a fancy verbatim block ...

- Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed

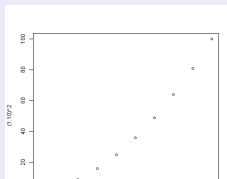
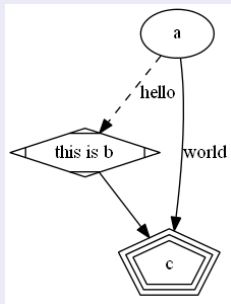
commodo consequat.

- Duis aute irure dolor in reprehenderit in voluptate velit

# Using graphics

How to center pictures horizontally?

## Figures



Skip proof  
nil

# Summary

# For further reading



A. Salomaa.

*Formal Languages.*

Academic Press, 1973.

# For further reading



A. Salomaa.

*Formal Languages.*

Academic Press, 1973.



E. Dijkstra.

Smoothsort, an alternative for sorting in situ.

*Science of Computer Programming*, 1(3):223–233, 1982.

# For further reading



A. Salomaa.

*Formal Languages.*

Academic Press, 1973.



E. Dijkstra.

Smoothsort, an alternative for sorting in situ.

*Science of Computer Programming*, 1(3):223–233, 1982.



E. Feldman and J. Owings, Jr.

A class of universal linear bounded automata.

*Information Sciences*, 6:187–190, 1973.

# For further reading



A. Salomaa.

*Formal Languages.*

Academic Press, 1973.



E. Dijkstra.

Smoothsort, an alternative for sorting in situ.

*Science of Computer Programming*, 1(3):223–233, 1982.



E. Feldman and J. Owings, Jr.

A class of universal linear bounded automata.

*Information Sciences*, 6:187–190, 1973.



P. Jančar, F. Mráz, M. Plátek, and J. Vogel.

Restarting automata.

*FCT Conference 1995*, LNCS 985, pages 282–292. 1995.



# Proof details

Text omitted in main talk.

# More details

Even more additional material.

# Abbreviations