

The Beamer SubFrame Package

Reordering frames in the PDF file without reordering the source

Mike Kaufmann

`m.km@gmx.de`

Version 0.1

29.06.2011

Introduction

- ▶ can reorder frames in PDF without reordering the source
- ▶ to embed or append frames with details
- ▶ only one level
- ▶ very first version
- ▶ not well tested
- ▶ propably many flaws
- ▶ therefore **experimental**



Frame Order with Details embedded

Order in PDF with details embedded (same as source):

- ▶ frame on foo
- ▶ frame with details on item foo x
- ▶ frame with more details on item foo x
- ▶ frame on bar
- ▶ frame with details on item bar y
- ▶ frame with more details on item bar y

▶ Show Picture



Frame Order with Details appended

Order in PDF with details appended (different from source):

- ▶ frame on foo
- ▶ frame on bar
- ▶ frame with details on item foo x
- ▶ frame with more details on item foo x
- ▶ frame with details on item bar y
- ▶ frame with more details on item bar y

▶ Show Picture

Dependencies

- ▶ only works with Beamer class
 - ▶ not loaded otherwise
 - ▶ error message
 - ▶ disregarding → additional errors
- ▶ needs verbatim



Usage – Package and Options

```
\usepackage[...]{beamersubframe}
```

Options:

embed subframes shown in order of source,
`\appendsubframes` ignored, default

append subframes removed and appended in place of
`\appendsubframes`

- ▶ options mutually exclusive
- ▶ if both given → last one wins

Usage – Appending Subframes

Add `\appendsubframes` before `\end{document}`

- ▶ no more subframes allowed after `\appendsubframes`
- ▶ `\appendsubframes` only allowed once
- ▶ restrictions not tested by package

Usage – Changes of Source

- ▶ For frames with details to be embedded or appended conditionally, change to subframe environment (same options as frame environment) [▶ Show Source Example](#)
- ▶ Add labels to
 - ▶ frame to skip or return to
 - ▶ first subframe with details[▶ Show Source Example](#)
- ▶ Add conditional links/buttons to
 - ▶ frame with details
 - ▶ last subframe with details[▶ Show Source Example](#)

Usage – Changes of Source

- ▶ For frames with details to be embedded or appended conditionally, change to subframe environment (same options as frame environment) [▶ Show Source Example](#)
- ▶ Add labels to
 - ▶ frame to skip or return to
 - ▶ first subframe with details[▶ Show Source Example](#)
- ▶ Add conditional links/buttons to
 - ▶ frame with details
 - ▶ last subframe with details[▶ Show Source Example](#)

Usage – Changes of Source

- ▶ For frames with details to be embedded or appended conditionally, change to subframe environment (same options as frame environment) [▶ Show Source Example](#)
- ▶ Add labels to
 - ▶ frame to skip or return to
 - ▶ first subframe with details[▶ Show Source Example](#)
- ▶ Add conditional links/buttons to
 - ▶ frame with details
 - ▶ last subframe with details[▶ Show Source Example](#)

Usage – Conditional Execution

Command:

```
\ifappend{... executed if appended ...}%  
    {... executed if embedded ...}
```

- ▶ useful for conditional links or buttons
- ▶ can be used in own commands

Restrictions of Beamer SubFrame

Additional restrictions to those already mentioned:

- ▶ no `\subframe` command
- ▶ no `\againsubframe` command
- ▶ appendix not tested
- ▶ only tested with mode `beamer`
- ▶ after errors `.nav` file has to be erased (sometimes)

Themes used for Testing

- ▶ Dresden
- ▶ Berkeley
- ▶ Montpellier
- ▶ Malmoe
- ▶ AnnArbor
- ▶ no options used

Files generated by Beamer SubFrame

`\jobname.sfr` contains subframes for appending

`\jobname.sfp` contains information about subframes used for
sidebar and miniframe navigation

Navigationbar

- ▶ frames and slides in order of PDF file
- ▶ section and subsection confusing on appended subframes
- ▶ presentation leads to last frame in PDF
- ▶ appendix not tested

Discussion necessary

Sidebar and Miniframes

- ▶ number and order of miniframes always as if subframes embedded
- ▶ highlighting always as if subframes embedded

ToDo

- ▶ proper manual
- ▶ proper package
- ▶ package documentation
- ▶ more testing
- ▶ fix navigationbar
- ▶ enabling beamer modes
- ▶ multiple levels of subframes (?)
- ▶ ...

But first lets talk about this package.



subframe Environment – embedded

- ▶ parses and stores first three arguments (if there)
- ▶ adds `environmet=subframe` to *options*
- ▶ calls `frame` environment with stored arguments

[▶ Goto Source](#)



subframe Environment – appended

- ▶ verbatim environment
- ▶ writes to `.sfr` file
 - ▶ `\beamerrestore...`
 - ▶ `\begin{frame}` with arguments
 - ▶ contents
 - ▶ `\end{frame}`
- ▶ one file for all subframes

[▶ Goto Source](#)



Appending Subframes

- ▶ does nothing in embed mode
- ▶ in append mode
 - ▶ closes `.sfr` file
 - ▶ inputs `.sfr` file

[▶ Goto Source](#)

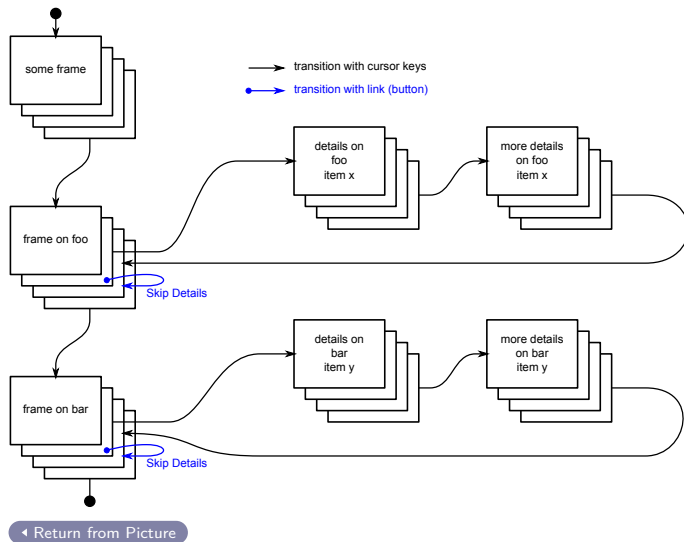
Just a Test Frame

(nothing here, just for testing)

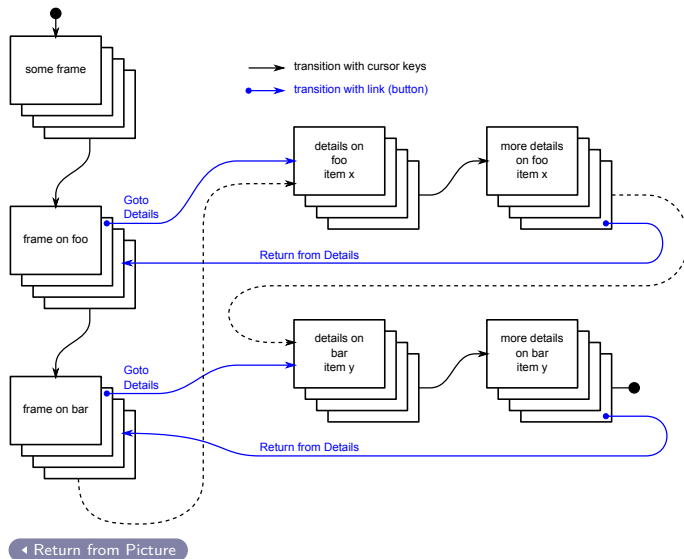
Thanks for reading!

Thanks in advance for testing and feedback!

Frame Order with Details embedded



Frame Order with Details appended



Source Example

Changing Environment

Change from

```
\begin{frame} ... options ...  
... framecontents ...  
\end{frame}
```

to

```
\begin{subframe} ... options ...  
... framecontents ...
```

```
\end{subframe}
```

◀ Return from Source Example

Source Example

Adding Labels

```
\begin{frame}  
...  
  
\begin{subframe}[label=detailsstart]  
...  
  
\begin{subframe}  
...  
  
\begin{frame}[label=skiptohere]  
...  

```

[◀ Return from Source Example](#)

Source Example

Adding Buttons

```

\begin{frame}
...
\ifappend{... link to detailsstart ...}%
    {... link to skipto here ...}
...
\begin{subframe}[label=detailsstart]
...
\begin{subframe}
...
\ifappend{... link to skipto here ...}{ }
...
\begin{frame}[label=skipto here]
...

```

[◀ Return from Source Example](#)



subframe Environment – Source (1)

(no source here, just for testing)



subframe Environment – Source (2)

(no source here, just for testing)

◀ Return from Source

subframe Environment – Source (3)

(no source here, just for testing)

subframe Environment – Source (4)

(no source here, just for testing)

◀ Return from Source

Appending – Source

(no source here, just for testing)

◀ Return from Source