Beamer by example

Andrew Mertz, William Slough
Department of Mathematics and Computer Science
Eastern Illinois University
Charleston, IL 61920
cfaem@eiu.edu, cfwas@eiu.edu

Abstract

There are a variety of LaTeX classes which can be used to produce "overhead slides" for presentations. One of these, beamer, provides flexible and powerful environments which can be used to create slides and PDF-based documents suitable for presentations. Although the class is extensively documented, first-time users may prefer learning about this class using a collection of graduated examples. The examples presented here cover a wide spectrum of use, from the simplest static slides to those with dynamic effects.

Introduction

LaTeX users in search of a way to produce "overhead slides" for use with an LCD projector have many choices today — perhaps too many! For example, Michael Wiedmann has a web site [6] that lists 12 different tools, all LaTeX-based, capable of producing PDF output.

For first-time users, it can be difficult to decide which of these many approaches best matches their needs. In our experience, we have found the beamer class [5] to be easy to use, flexible, and well documented.

The user's guide for beamer is, of course, the ultimate authority for its use. However, at 203 pages, a potential user might be frightened off before having a chance to experience its capabilities. Our intention here is to provide a sampling of beamer's capabilities by displaying a variety of examples.

A first example

A beamer document consists of a sequence of *frames*. In the simplest case, all of a frame's content is displayed at once. A frame of this type is the electronic equivalent of an overhead transparency.

Figure 1 gives a complete example of a beamer presentation, stored in a file named `talk.tex`. In the preamble, familiar LaTeX commands appear. The body of the document specifies a title page is to appear, followed by two frames. Each `frame` environment specifies the desired frame title and the contents to appear on that frame.

Processing the `talk.tex` source with pdfLaTeX yields `talk.pdf`, a PDF file suitable for presentations. Figure 2 shows the resulting output.

```
\documentclass{beamer}

\title{A Tiny Example}
\author{Andrew Mertz and William Slough}
\date{June 15, 2005}

\begin{document}

\maketitle

\begin{frame}
 \frametitle{First Slide}
 Contents of the first slide
\end{frame}

\begin{frame}
 \frametitle{Second Slide}
 Contents of the second slide
\end{frame}

\end{document}
```

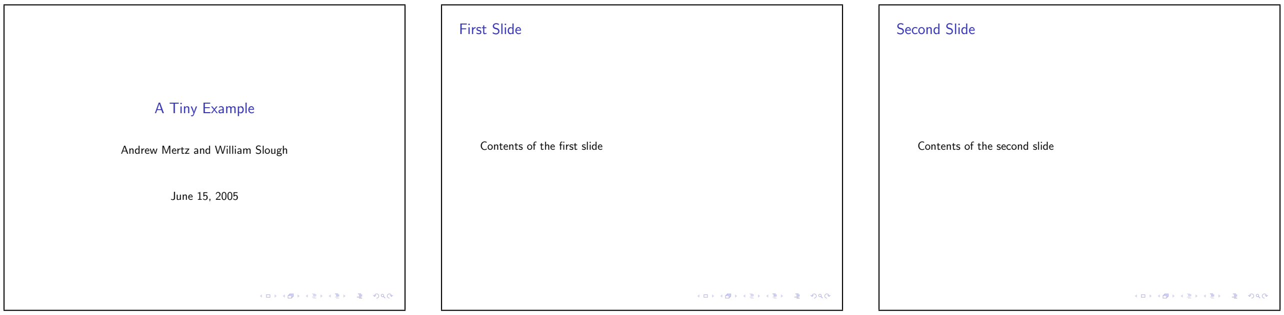
Figure 1: Contents of our initial `talk.tex`.

Using this simple example as a template, a newcomer to beamer can produce a wide variety of presentations — in effect by learning about just one new environment, the `frame`.

Frame content

A frame can be subdivided into the following basic components, many of which are optional:

- Head line and foot line
- Left and right sidebars
- Navigation bars
- Logo



A Tiny Example

Andrew Mertz and William Slough

June 15, 2005

First Slide

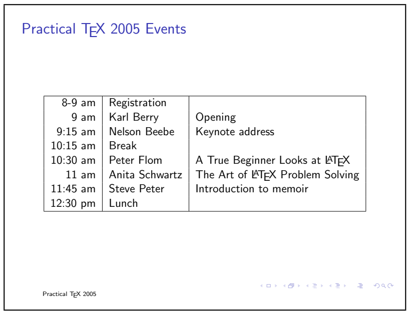
Contents of the first slide

Second Slide

Contents of the second slide

Figure 2: The three output frames of `talk.pdf`.

```
\begin{frame}
\frametitle{Practical \TeX\ 2005 Events}
\begin{center}
 \begin{tabular}{|r|l|l|}\hline
  8-9 am   & Registration   & \\
  9 am     & Karl Berry     & Opening \\
  9:15 am  & Nelson Beebe   & Keynote address \\
  10:15 am & Break          & \\
  10:30 am & Peter Flom     & A True Beginner Looks at \LaTeX  \\
  ... etc ...
 \end{tabular}
\end{center}
\end{frame}
```



Practical TEX 2005 Events

8-9 am	Registration	
9 am	Karl Berry	Opening
9:15 am	Nelson Beebe	Keynote address
10:15 am	Break	
10:30 am	Peter Flom	A True Beginner Looks at LATEX
11 am	Anita Schwartz	The Art of LATEX Problem Solving
11:45 am	Steve Peter	Introduction to memoir
12:30 pm	Lunch	

Practical TEX 2005

Figure 3: Source and output for a frame with a centered table.

- Frame title
- Background
- Content

In our examples, we use relatively few of these components, choosing to emphasize the content over the more decorative elements which are possible. Our first example does include a navigation bar located at the lower right hand corner of the frame, which is present by default.

As far as frame content is concerned, most of the LaTeX environments and commands work within beamer documents in the expected ways. So, for example, enumerated and itemized lists, mathematics, and tables can all be expressed in ways familiar to LaTeX users. The only thing different is that these commands must appear within beamer's `frame` environment. To illustrate, Figure 3 shows how a table can be centered on a frame, along with its corresponding output.

One special case is worth noting, however. If verbatim text is to appear within a frame, the frame must be marked as *fragile*. This is accomplished as follows:

```
\begin{frame}[fragile]
 % ... frame contents ...
\end{frame}
```

To include graphics within a frame, facilities of the graphicx package may be used. Several points are worth noting related to graphics. To begin with, beamer automatically loads the graphicx package, so no explicit `\usepackage` statement is needed. For properly sizing graphics within a frame, it helps to know that beamer formats its output to a size of 128 millimeters by 96 millimeters, or 5.04 inches by 3.78 inches. The native graphics formats supported by pdfLaTeX are JPEG, PNG, PDF, and MetaPost output.

Figure 4 illustrates how the graphics file named `p2005.png` can be placed within a frame.

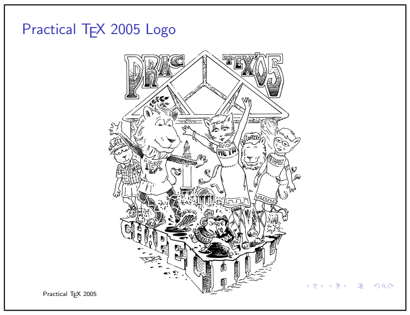
Frames with color

Emphasis in presentations may be obtained by changes in color, in addition to the more traditional font changes. Since beamer automatically loads the xcolor package [2], colors can be specified using the syntax of xcolor. In particular, the "named" color model can be combined with a percentage using the `xcolor !` specifier. For example,

```
{\color{BlueViolet!30} A B C}
```

will typeset the text "A B C" using the `BlueViolet` color, at 30% intensity.

```
\begin{frame}
  \frametitle{Practical \TeX\ 2005 Logo}
  \begin{center}
    \includegraphics[height=3.25in]{p2005}
  \end{center}
\end{frame}
```

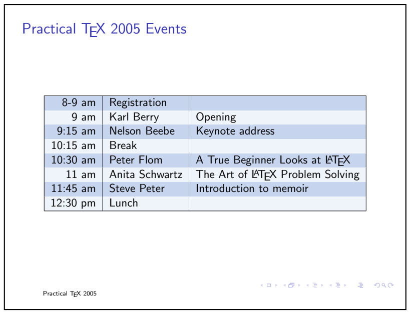


Practical TEX 2005 Logo

Practical TEX 2005

Figure 4: Source and output for a frame with included graphics.

```
\begin{frame}
\frametitle{Practical \TeX\ 2005 Events}
\begin{center}
  \rowcolors{1}{\RoyalBlue!20}{\RoyalBlue!5}
  \begin{tabular}{|r|l|l|}\hline
    8-9 am   & Registration   & \\
    ... etc ...
  \end{tabular}
\end{center}
\end{frame}
```



Practical TEX 2005 Events

8-9 am	Registration	
9 am	Karl Berry	Opening
9:15 am	Nelson Beebe	Keynote address
10:15 am	Break	
10:30 am	Peter Flom	A True Beginner Looks at LATEX
11 am	Anita Schwartz	The Art of LATEX Problem Solving
11:45 am	Steve Peter	Introduction to memoir
12:30 pm	Lunch	

Practical TEX 2005

Figure 5: Source and output for a frame containing a table with alternating colors.

A particularly effective use of color can be applied to a tabular environment, via the rowcolors command from xcolor. In this command, a starting row number and two colors are specified. These two colors are used to alternately shade the rows of a table, beginning with the given row. For example,

```
  \rowcolors{1}{RoyalBlue!20}
               {RoyalBlue!5}
```

requests two shades of RoyalBlue are to be applied to a table, beginning with its first row. This command should immediately precede the tabular environment, as shown in Figure 5.

To use color specifications like these within a beamer document, some additional options must be given within the documentclass. Ordinarily, these color capabilities would be obtained with an appropriate \usepackage command. However, as mentioned earlier, beamer automatically loads xcolor (among others), thus making the usual command, \usepackage{xcolor}, both unnecessary and illegal. What this means is that options we wish to specify for these packages must be given in a different way. This is the reason for the beamer options. For example,

```
  \documentclass[xcolor=pdftex,dvipsnames,
                        table]{beamer}
```

specifies three different options to be used with the xcolor package. The first option, pdftex, provides information about the correct color driver to use. The option dvipsnames allows a set of predefined color names, such as RoyalBlue, to be used. (These named colors are sometimes referred to as the "Crayola" colors.) Finally, the table option informs xcolor that the colortbl package needs to be loaded. It is this last option that defines the \rowcolors command used in Figure 5.

Frames with two columns

Since frames have a landscape orientation, it can be helpful to be able to subdivide a frame into two columns. The columns environment of beamer is designed to meet this need.

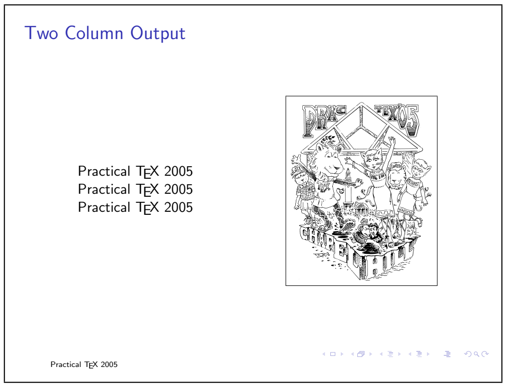
Figure 6 provides an example of how a frame can be subdivided into two columns. The columns environment allows an alignment option which specifies whether columns are to be aligned along their top line, bottom line, or centered. In this example, the c option causes the two columns to be aligned along their vertical centers. Within the columns environment two columns appear, as specified with the two \column commands.

```
\begin{frame}
\frametitle{Two Column Output}

\begin{columns}[c]
\column{1.5in}
Practical \TeX\ 2005\\
Practical \TeX\ 2005\\
Practical \TeX\ 2005

\column{1.5in}
\framebox{\includegraphics[width=1.5in]{p2005}}
\end{columns}

\end{frame}
```


Two Column Output

Practical TeX 2005
Practical TeX 2005
Practical TeX 2005

Practical TeX 2005

Figure 6: Source and output for a double-column frame.

Frames with overlays

Up to this point, the frames we have considered consisted of a single *overlay*. When the frame is displayed, everything on that frame appears at once.

Alternatively, a frame can consist of a sequence of overlays, which can be used to support incremental display. Overlays can be used to "hold back" information during a presentation or to produce certain kinds of animated effects.

As a matter of taste, some people feel that a single overlay is preferable, since information is not hidden from the audience. However, we feel that there are situations where multiple overlays are appropriate, especially when used judiciously.

The beamer class provides numerous ways to specify frames with multiple overlays. We illustrate three techniques:

- Using the `\pause` command
- Using overlay specifications
- Including multiple graphics files

Using the `\pause` command is a simple way to produce overlays. All text from the beginning of the frame to the place where a `\pause` command appears is formatted and placed into an overlay. In this way, the example in Figure 7 creates a frame with three overlays. When the resulting PDF file is viewed, the three lines of output are incrementally revealed. The command

```
\setbeamercovered{dynamic}
```

is in effect, so overlays not yet revealed will faintly appear. This allows the speaker to focus on the current overlay, yet not entirely hide information from the audience.

Our second example of incremental display involves overlay specifications. In beamer, every overlay within a frame is assigned a number, starting with one, reflecting the order in which they are displayed.

```
\begin{frame}
\frametitle{Overlays with {\tt pause}}

\setbeamercovered{dynamic}

Practical \TeX\ 2005\\  \pause
Practical \TeX\ 2005\\  \pause
Practical \TeX\ 2005
\end{frame}
```

Figure 7: Specifying multiple overlays with `pause`.

Figure 8 illustrates how a game of tic-tac-toe can be displayed. In this example, there are ten overlays — one for the grid, and one for each successive play in the game. The syntax

`\onslide<`*m*`->{`*text*`}`

indicates that the specified *text* is to appear on every overlay from m onwards.

Initially, only the grid should appear, so no X or O appears on the first overlay. In this example, the first play is an X in the upper right-hand corner, so this X appears on overlay 2 and every successive overlay. O counters by playing in the center, so this O appears on overlay 3 and every successive overlay. The rest of the example follows in a similar way. In this situation, we do want to hide from view overlays not yet revealed, so the command

```
\setbeamercovered{invisible}
```

is appropriate.

For our third example of incremental effects, we use the facilities of the package xmpmulti and a drawing tool which supports multiple layers. We use xfig, but many other choices are possible.

```
\begin{frame}
\frametitle{Tic-Tac-Toe via {\tt tabular}}

\setbeamercovered{invisible}
{\Huge
\begin{center}
  \begin{tabular}{c|c|c}
   \onslide<9->{O}  & \onslide<8->{X} & \onslide<2->{X} \\ \hline
   \onslide<6->{X}  & \onslide<3->{O} & \onslide<5->{O} \\ \hline
   \onslide<10->{X} & \onslide<7->{O} & \onslide<4->{X}
  \end{tabular}
\end{center}
}
\end{frame}
```

Figure 8: Using overlay specifications.

```
\begin{frame}
\frametitle{Tic-Tac-Toe via Graphics Files}

\setbeamercovered{invisible}
\begin{center}
\multiinclude[format=pdf,width=3in]{game}
\end{center}

\end{frame}
```

Figure 9: Specifying multiple overlays with graphics files. The files to be included are named game-0.pdf, game-1.pdf, ..., game-9.pdf.

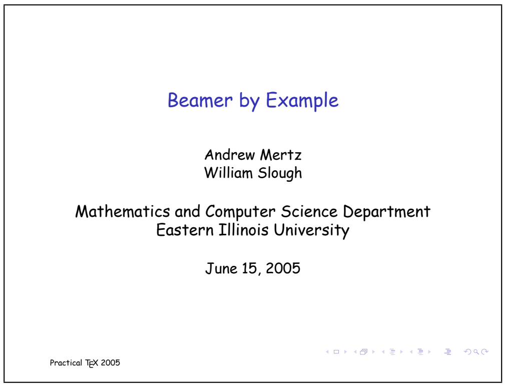
Revisiting the game of tic-tac-toe, a grid can be drawn on layer 0, followed by the first move on layer 1, the second move on layer 2, and so forth. After all layers of the drawing are complete, each layer is exported to a PDF file with a suffix which matches the layer number. For example, layer 0 is exported to game-0.pdf, layer 1 is exported to game-1.pdf and so forth. The \multiinclude command of the xmpmulti package, illustrated in Figure 9, causes the graphics files to appear as overlays.

It is worth noting that beamer overlays are numbered beginning with 1, but xmpmulti considers the first overlay to begin with 0.

Ornamental aspects

It is possible to "dress up" beamer presentations in a variety of ways. For example, one could use the Microsoft Comic Sans font, which can be made available with the comicsans package [4]. Once this font has been established within the TEX system, adding the following two lines to the preamble of the beamer document will make it the default font:

```
\usepackage{comicsans}
\renewcommand{\sfdefault}{comic}
```



Beamer by Example

Andrew Mertz
William Slough

Mathematics and Computer Science Department
Eastern Illinois University

June 15, 2005

Figure 10: Title frame composed with the Microsoft Comic Sans font.

Figure 10 shows a resulting frame.

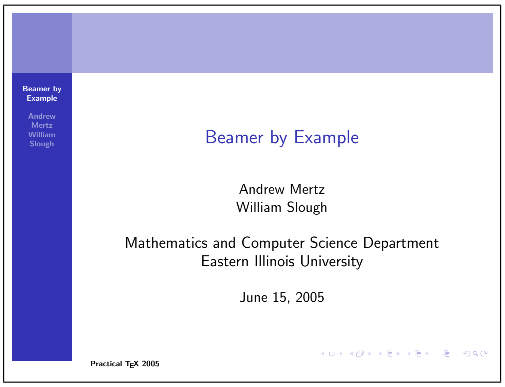
Another possibility is to choose from among the many different beamer *themes*. As an example, adding the following lines to the preamble of our beamer presentation gives the title frame shown in Figure 11.

```
\usepackage{beamerthemesplit}
\usetheme{Berkeley}
\usecolortheme{dolphin}
```

In addition to the visual appeal of themes, additional navigation tools are incorporated in the frames based on the LATEX sections and subsections present in the beamer presentation.

Producing N-up output

It is often desirable to produce a printed document which mirrors the content of the beamer presentation. There are two steps needed to accomplish this; first, create an overlay-free version of the presentation, and second, produce an N-up version of the presentation.



Beamer by Example

Andrew Mertz
William Slough

Beamer by Example

Andrew Mertz
William Slough

Mathematics and Computer Science Department
Eastern Illinois University

June 15, 2005

Practical TEX 2005

Figure 11: Title frame composed with the Berkeley theme.

Removing overlays from a beamer presentation is easily done within the preamble, by adding the handout option:

```
\documentclass[handout,
                   xcolor=pdftex,dvipsnames,
                           table]{beamer}
```

Processing a beamer document with this option causes all overlays for a given frame to be collapsed into a single frame.

Once overlays have been removed, putting multiple frames onto a single sheet of paper is a separate problem related to PDF files. The pdfpages package [3], for example, solves this problem. For an automated approach based on the same package, pdfjam [1], a Unix shell script, can be used.

Conclusion

The contribution of many individuals in the LaTeX community have made it possible to produce overhead slides using typesetting standards of the highest quality. We are especially indebted to the work of Till Tantau and the other package designers cited earlier.

References

[1] David Firth. PDFjam. http://www2.warwick.ac.uk/fac/sci/statistics/staff/academic/firth/software/pdfjam.

[2] Uwe Kern. *Extending LaTeX's color facilities: The* xcolor *package*. http://www.ctan.org/tex-archive/macros/latex/contrib/xcolor.

[3] Andreas Matthias. *The* pdfpages *package*. http://www.ctan.org/tex-archive/macros/latex/contrib/pdfpages/.

[4] Scott Pakin. *The* comicsans *package*. http://www.ctan.org/tex-archive/macros/latex/contrib/comicsans.

[5] Till Tantau. *User's Guide to the* Beamer *Class, Version 3.01*. http://latex-beamer.sourceforge.net.

[6] Michael Wiedmann. *Tools for Creating Screen or Online Presentations*. http://www.miwie.org/presentations/presentations.html.