The rst2beamer manual

Warning
This manual is disgracefully incomplete.

Introduction

In brief

rst2beamer is a docutils script that converts restructured text into Beamer-flavoured LaTeX.

In more detail, Beamer is a LaTeX document class for presentations. rst2beamer[1] provides a Docutils[3] writer that transforms a simple text format called restructured text or ReST[4] into Beamer-flavoured LaTeX. A commandline script is installed that therefore allows ReST to be used to prepare slides and presentations. So a user can write a simple text file that can be converted into LaTeX (and hence PDF or Postscript), in order to quickly produce neat, professional presentations with a neat unified look.

What rst2beamer is (and isn't) for

rst2beamer was developed when one of us (PMA) grew frustrated with the length of time it took to prepare a presentation for teaching, even simple slides with bullet-pointed text and lumps of code. Preparing handouts or downloadable versions was a further problem. rst2beamer was thus developed to quickly throw together a structured presentation, with sections, titles and bullet points.

It is not a replacement for Powerpoint, Keynote or other graphical presentation tools. Nor is it intended to be. While rst2beamer allows the inclusion of images, if you need fancy transitions and navigation or complex layout, you'd be better off with once of the aforementioned programs. rst2beamer is meant to be:

- Quick: a presentation can be rapidly written, without slow and tedious composition of pages and adjusting the position of page elements.
- Simple: the ReST syntax is easy to remember and write. The rst2beamer source files can be easily read in their raw form.
- Portable: as the rst2beamer source files are plain text, they behave the same across platforms. Also, by using the ReST syntax, presentations can be easily converted to RTF, HTML and other formats.

Thus rst2beamer may be useful for teaching or conference presentations and has been used at Python conferences for those purposes.

Conversely, rst2beamer doesn't do everything that Beamer or ReST can. Beamer is a sophisticated system that can make sophisticated presentations. rst2beamer doesn't try to do everything that Beamer can, as this would foil the goal of simplicity. Supported features include:

- Overlay lists (i.e. those that appear point-by-point)
- Beamer themes
- Automatic centering and resizing of figures
- Embedded notes and the output of note slides.
- Arranging slide contents into columns.

Some examples can be found in the `docs/examples` directory of the distribution.

There may also be ReST syntax that doesn't work or looks bad in the beamer output but most ReST will translate without problems.

Installation

rst2beamer can be installed in a number of ways. setuptools[2] is preferred, but a manual installation will suffice.

Prerequisites

rst2beamer requires the installation of the docutils package[3]. Although older versions of docutils (e.g. v0.4) are useable, development focuses on more recent version (e.g. v0.6).

Via setuptools / easy_install

From the commandline call:

```
% easy_install rst2beamer
```

Superuser privileges may be required.

Via setup.py

Download a source tarball, unpack it and call setup.py to install:

```
% tar zxvf rst2beamer.tgz
% cd rst2beamer
% python setup.py install
```

Superuser privileges may be required.

Manual

Download and unpack the tarball as above. Ensure Docutils is available. Copy the script `rst2beamer.py` to a location it can be called from.

Usage

> **Note**
> Depending on your platform, the scripts may be installed as `.py` scripts, or some form of executable, or both.*

rst2beamer is called:

```
rst2beamer [options] [<source> [<destination>]]
```

For example, the simplest and typical way to use rst2beamer would be:

```
rst2beamer infile.txt outfile.tex
```

where `infile.txt` contains the ReST and `outfile.tex` contains the produced Beamer LaTeX.

It supports the usual docutils and LaTeX writer (rst2latex) options, save the `documentclass` option (which is fixed to `beamer`) and hyperref options (which are already set in beamer). It also supports:

--theme=THEME Specify Beamer theme.

--overlaybullets=OVERLAYBULLETS Overlay bulleted items. Put [<+-| alert@+>] at the end of begin{itemize} so that Beamer creats an overlay for each bulleted item and the presentation reveals one bullet at a time

--centerfigs=CENTERFIGS Center figures. All includegraphics statements will be put inside center environments.

--documentoptions=DOCUMENTOPTIONS Specify document options. Multiple options can be given, separated by commas. Default is "10pt,a4paper".

--shownotes=SHOWNOTES Print embedded notes along with the slides. Possible arguments include 'false' (don't show), 'only' (show only notes), 'left', 'right', 'top', 'bottom' (show in relation to the annotated slide).

Of course, rst2beamer only produces the LaTeX source for a presentation. LaTeX hackers will have no difficulty using this, but most others will want to convert immediately to PDF. This can be done easily with `pdflatex`. For example:

```
pdflatex outfile.tex
```

will produce outfile.pdf. pdflatex is included in most TeX distributions.

Getting started

TO BE COMPLETED

Themes

Beamer can be displayed in a variety of themes, changing the sizing, color and layout of page elements. The theme can be controlled by the commandline option `theme`. For example:

```
rst2beamer --theme=Rochester infile.rst
```

will produce a Beamer file styled with the "Rochester" theme. Some other themes include "AnnArbor", "Bergen", "CambridgeUS" and "Warsaw".

> **Note**
>
> The selection of themes available with your LaTeX installation may vary but a typical selection can be seen here[6]. The default Beamer theme (called "default") is very plain and so rst2beamer uses the "Warsaw" theme as its default.

Sections

Beamer supports the idea of grouping slides into sections and subsections. rst2beamer allows this too, but frametitles must be in the lowest section level. A section with no child sections is the lowest. Note that if you are going to use subsections anywhere in the document but your first slide isn't in a subsection, you have to use dummy a section before your first slide:

```
Introduction
------------

dummy
~~~~~


Slide 1
--------

- Point 1
- Point 2
```

The top level title is set as the presentation title while 2nd-level titles are set as slide titles (`frametitles` in Beamer terms). While all other titles are converted as normal, Beamer ignores them. There is some problem in the production of literals. rst2latex converts them to ragged-right, noindent typewriter font in a quote. Under beamer however, this makes them appear as italics. This was solved by overriding literal production with a simpler enviroment, albeit one that occasionally produces buggy output. Options to hyperref are dropped, due to this already being used in beamer.

Lists

TO BE COMPLETED

Images

Images default to being centered and having a height of 0.7textheight (you can turn off the centering with a commandline switch). Thus:

```
Slide Title
-----------

.. image :: image_name.png
```

produces a graph centered in the middle of the slide. Simple.

Columns

Setting slide elements into columns teeters into the realm of page layout. If layout is important, you may be better off using a more sophisticated presentation tool. However, rst2beamer provides a few simple tools for quick and consistent page layout into columns.

The simplest method is the r2b_simplecolumns directive. This will turn every element underneath it into a column, side by side with each other. Individual column width will be divided evenly from the total available, which by default is set to 0.90 of the page width. The directive accepts an optional argument `width` that that can be used to set this total. So this:

```
.. r2b_simplecolumns::
   :width: 0.95

   This is a demonstration of the rst2beamer simple
   column directive.

   Notice that we have used the optional argument,
   "width". It is set to 0.95.

   * A list or image
   * can be
   * a column
```

will produce a slide with three columns, containing the first, then second paragraph, then the list. Their individual width will be 0.95 divided by three.

However, custom r2b directives won't be recognised by any writer other than rst2beamer. It would be lose the ability to turn your presentation into other formats like HTML, etc. Therefore, we allow containers with certain names to act like column sets. Most other writers should recognise these containers and at worst ignore them rather than throw an error. For example:

```
.. container:: r2b_simplecolumns

   The custom r2b directives won't be recognised by any
   writer other than rst2beamer.

   So, any container with the name 'r2b_simplecolumns'
```

```
or 'r2b-simplecolumns' will be handled like the simple
columns directive.
```

Finally, we allow columns to be explicitly set and their width controlled with the r2b_columnset and r2b_column directives. For example:

```
.. r2b_columnset::
   :width: 0.95

   .. r2b_column::
      :width: 0.60

      If you insist on setting columns explicitly, you can,
      grouping multiple elements.

      The width of the column set and individual columns can
      be given. This set and column are 0.95 and 0.60 wide
      respectively.

   .. r2b_column::

      Columns not given a width (like this one) share the
      remainder.
```

A set of columns is indicated with the directive `r2b_columnset`. It takes the optional argument `width`, which indicates the total page width the contained columns will take up. Again, by default it is 0.90.

Only `r2b_column` directives can occur directly within a columnset. These group one or more text elements to appear within a single column. They can take an optional argument `width`, which indicates how wide that column is. Any unsized columns will be given a width from sharing the unallocated width of the columnset. Overallocating width (i.e. columns accounting for more width than available) will cause an error.

In all cases, columns may not be nested within columns more columnsets within columnsets.

Notes

Beamer support the inclusion of notes in a presentation. These are written as text embedded in the main presentation, but usually invisible. When notes are made visible, they are printed in another "slide" alongside the annotated slide page.

rst2beamer supports the inclusion and showing of notes. Their appearance is set with the commandline argument `--shownotes`. For example:

```
rst2beamer --shownotes <option> mypresentation.rst
```

where `option` can be:

false don't show any notes (the default)

true show notes as per `right`

only show only the notes, not the presentation

left, right, top, bottom show the notes in the given position to the presentation

Notes can be included in ReST with the `r2b_note` directive. For example:

```
.. r2b_note::

   This is an example.
```

will inject a note into the current slide. Multiple notes can be included in one slide and will be compiled into a single note. For example:

```
Farnarkling history

.. r2b_note::

   Don't forget to mention the Sorenson brothers.

Greats of the sport

.. r2b_note::

   They would arkle with great authority.
```

will produce a slide with the content:

Farnarkling history

Greats of the sport

and the accompanying note (normally hidden):

Don't forget to mention the Sorenson brothers.

They would arkle with great authority.

If notes are being shown, and a slide has no notes, the note page will simply be empty.

Again, the custom r2b directives won't be recognised by any writer other than rst2beamer, so we allow certain containers to act like notes:

```
.. container:: r2b_note

   Compatibility is important
```

Any container with the name 'r2b_note' or 'r2b-note' will be handled like the notes directive:

```
.. container:: r2b-note

   This will be understood by other ReST writers.
```

Note

See the "notes" input and output example files.

Limitations

Earlier versions of rst2beamer did not work with docutils 0.4, seemingly due to changes in the LaTeX writer. While this has been fixed, most work has been done with docutils snapshots from version 0.5 and up. In balance, users are recommended to update docutils.

More recently, changes in the LaTeX writer in docutils 0.6 broke rst2beamer again. We think all those bugs have been caught.

Not all features of beamer are supported, and some - that deal with with page layout or presentation - may never be. Introducing complex syntax to achieve complex and specific page effects defeats the point of ReST's simple and easy-to-write format. If you need a complex presentation, use Powerpoint or Keynote.

If the content for an individual slide is too large, it will simply overflow the edges of the slide and disappear. Arguably, this is a sign you should put less on each slide.

Miscellaneous tips and tricks

TO BE COMPLETED

Development notes

History & motivation

While preparing a course, one of us (PMA) became frustrated with the length of time it took to prepare a presentation for teaching, even simple slides with bullet-pointed text and lumps of code. Preparing handouts or downloadable versions was a further problem. Given that docutils already has good LaTeX output, PDF production via the Beamer document class was a logical choice. rst2beamer started as a semi-ugly hack of docutil's LaTeX machinery, making as few modifications are possible due to (a) laziness and (b) wanting to leverage as much of an existing robust code base as possible. It wasn't - and isn't - intended to be feature-complete: it worked with the ReST that I prepared and will probably give adequate output for most other simple ReST documents. It was subsequently taken up by the other of us (RK) who whipped the hacky script into shape and added several features to make the produced slides far more presentable. Unbeknownst to either of us, it seems to have been adopted by Pythonistas for use at conferences.

Alternatives

Other output options for ReST were considered and discarded as follows:

- **ReportLab's Pythonpoint** requires a fixed frame size and would need custom XML output. Styling is done through Reportlab stylesheets, which can be complex.
- **Prosper** is another LaTeX solution. On balance, Beamer seemed better although the point is arguable.

- **AxPoint** requires Perl.
- **slides and foil** are old LaTeX solutions that are now somewhat creaky.

A few other restructured text writers for presentation are available:

- **rst2beamer:** A later project of the same name is present in the docutils sandbox. It seems to be a simple wrapper around the standard LaTeX writer (much as this project started as) and may not support more beamer specific syntax. Reports from anyone who has used it would be welcome.
- **s5:** A ReST-to-S5 writer is present in the standard docutils distribution. It produces some very nice presentations with good visual effects, although arguably at the cost of some very specific syntax. Those who find rst2beamer underpowered might do well to look here.

Credits

rst2beamer is developed by Ryan Krauss and Paul-Michael Agapow. Thanks to Dale Hathaway for helping track down the docutils 0.4 bug. Thanks to those who reported and helped us track down bugs: Perttu Laurinen, Mike Pennington, James Haggerty and Dale Hathaway.

References

[1] rst2beamer homepages at agapow.net and cs.siue.edu

[2] Installing setuptools

[3] Docutils homepage

[4] Restructured text

[5] Beamer homepage

[6] Patrick Pletscher Beamer Themes