The Sphinx Thesis Resource (sphinxtcr)

Jeff Terrace

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Computer Science
Adviser: Professor John P. Smith

November 2012
Abstract

This project is a collection of extensions and monkey patches to Sphinx to better format a PhD thesis.
Acknowledgements

Thanks to Sphinx for a kickass build system and docutils for the groundwork of multiple format output.
To my parents.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>iii</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>iv</td>
</tr>
<tr>
<td>List of Figures</td>
<td>viii</td>
</tr>
<tr>
<td><strong>1 Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Installation</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Building</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Changes</td>
<td>2</td>
</tr>
<tr>
<td>1.4 Documents Using sphinxtr</td>
<td>3</td>
</tr>
<tr>
<td><strong>2 Typography</strong></td>
<td>4</td>
</tr>
<tr>
<td>2.1 Headings</td>
<td>4</td>
</tr>
<tr>
<td>2.1.1 Third-level</td>
<td>4</td>
</tr>
<tr>
<td>2.2 Text</td>
<td>5</td>
</tr>
<tr>
<td><strong>3 Lists</strong></td>
<td>7</td>
</tr>
<tr>
<td>3.1 Unordered Lists</td>
<td>7</td>
</tr>
<tr>
<td>3.2 Ordered Lists</td>
<td>7</td>
</tr>
<tr>
<td>3.3 Description Lists</td>
<td>8</td>
</tr>
<tr>
<td>3.4 Mixed</td>
<td>8</td>
</tr>
<tr>
<td><strong>4 Math</strong></td>
<td>10</td>
</tr>
</tbody>
</table>
5 References and Citations
   5.1 Subsection ........................................ 11
   5.2 Citations ......................................... 11
   5.3 Footnotes ......................................... 11
   5.4 External Links ................................... 12

6 Figures and Tables
   6.1 Vector SVG Figures ............................... 13
   6.2 Image Figures .................................... 14
   6.3 Subfigures ....................................... 14
   6.4 Table ............................................ 15
   6.5 Text Wrapping Table ........................... 16

Bibliography ............................................. 18
List of Figures

6.1 Sirikata System Overview ............................................. 13
6.2 Open3DHub Browsing Interface ................................. 14
6.3 Example Model Resolutions ................................. 15
6.4 Mean Size of Progressive Format .......................... 15
6.5 Mixed Alignment Table ............................................. 16
6.6 Open3DHub Cassandra Column Families ............... 17
Chapter 1

Introduction

Creating a PhD thesis is typically done using LaTeX. This works really well for producing a PDF, but a giant PDF file is not a great way to put documents on the web. There are solutions that exist to turn latex source files into HTML, but in my experience, they tend to produce poor HTML output.

The Sphinx project is a wonderful tool for creating portable documents, allowing for output to many different formats. Unfortunately, it has many shortcomings when trying to typeset something so advanced as a PhD thesis. The aim of this project is to modify Sphinx to support all of the needs of a thesis writer. Many of the patches are not appropriate for contributing directly to the upstream Sphinx repository, so this is instead a separate project.

This sphinxtr output is available in several formats at: http://jterrace.github.com/sphinxtr

The source code for sphinxtr can be found at: https://github.com/jterrace/sphinxtr
1.1 Installation

Install the required Python packages:

```
pip install -r requirements.txt
```

1.2 Building

You need `make`. The following targets are supported:

- **html**
  Builds HTML format, separated into sections

- **singlehtml**
  Builds HTML format on a single page

- **text**
  Builds text files, separated into sections

- **singletext**
  Builds a single text file

- **latexpdf**
  Builds into latex source files and then compiles into a PDF. Requires latex.

1.3 Changes

The following changes and additions have been made from vanilla Sphinx:

- A cross-format bibtex bibliography based on sphinx-natbib
• Tables that can go inside figures

• Changed table formatting to look pretty, like booktabs

• Improved alignment in table environment

• Added support for short captions that show up in the “list of figures” section

• Changed equation reference formatting from “(1)” to “1”

• Full customization of latex preamble and style file

• Numbered figures

• Numbered section references

• A singletext output that builds into a single text file, similar to singlehtml

• A subfigure environment

1.4 Documents Using sphinxtr

• [Jeff Terrace’s PhD Thesis](#)
Chapter 2

Typography

This chapter has typography stuff. This also shows how to have a toctree inside a chapter.

2.1 Headings

The title of this chapter, “Typography”, is the first heading level. This section, “Headings”, is the second level.

2.1.1 Third-level

The third level heading. You probably shouldn’t go beyond this because it just looks ridiculous, but you can anyway.

Fourth-level

The fourth level heading.
Fifth-level  The fifth level heading.

Rubric Heading  A rubric heading is just a paragraph heading without document structure.

2.2  Text

You can make emphasized text.

You can make bold text.

You can make fixed-width font.

You can make block quotes:

this is a block quote

You can make code blocks:

this is a code block

Some convenience substitutions are defined in the epilog:

- e.g.,
- i.e.,
- et al.
- dash—
- non-breaking space -\rightarrow <-

You can have inline superscript or subscript text.

Big quotes, also known as an epigraph:
Your avatar can look any way you want it to, up to the limitations of your equipment. If you’re ugly, you can make your avatar beautiful. If you’ve just gotten out of bed, your avatar can be wearing beautiful clothes and professionally applied makeup. You can look like a gorilla or a dragon, or a giant talking penis in the Metaverse. Spend five minutes walking down the street, and you will see all of these.

—Neal Stephenson, Snow Crash
Chapter 3

Lists

This is a few examples of different list types.

3.1 Unordered Lists

• item 1

• item 2

– item 3

– item 4

• item 5

3.2 Ordered Lists

1. item 1

2. item 2
3. item 5

3.3 Description Lists

term1

  definition 1

term2

  definition 2

term3

  definition 3

3.4 Mixed

term

  definition
    • list1
    • list2

term2

  something
    1. num1
    2. num2
• list1

• list2

3. num3
Chapter 4

Math

Math uses latex math syntax:

\[ A''_c = \sqrt[3]{\sum \frac{L_c^2}{\sum L^2} \left( \frac{A_c}{\sum A} \right) \left( \frac{A'_c}{\sum A'} \right)} \cdot T \]

Equations can have labels which you can reference [4.1]

\[
\frac{\log(1 + E_{\text{current}})}{\log(1 + E_{\text{max}})}
\]

(4.1)
Chapter 5

References and Citations

You can reference a section by its label. This chapter is Chapter 5.

5.1 Subsection

This subsection is Section 5.1.

5.2 Citations

COLLADA [1] is a cool 3D file format. I wrote a paper about 3D stuff [3]. The website we built is running [2]. The bibliography is in bibtex format.

5.3 Footnotes

Reference a footnote [4].

[1] This is a footnote at the end of the page or document.
5.4 External Links

You can link to a [website](#).
Chapter 6

Figures and Tables

6.1 Vector SVG Figures

Vector figures are nicely supported. You should have a PDF file and an SVG file. The PDF will be used for the latex output and the SVG for the HTML output. The HTML output has a nice zoom feature using Colorbox.

![Diagram of Sirikata metaverse platform architecture](image)

**Figure 6.1:** The Sirikata metaverse platform architecture.
See an example in Figure 6.1. I suggest making figures in something like Inkscape. If you have only a vector PDF, you can use pdf2svg to convert (\texttt{brew install pdf2svg} or \texttt{apt-get install pdf2svg}).

### 6.2 Image Figures

Regular rasterized images work fine too.

A PNG example is shown in Figure 6.2.

### 6.3 Subfigures

The subfigure directives allow you to place multiple figures side-by-side in the document. Here’s an example:

You can reference the entire Figure 6.3 or one of its subfigures, e.g., Figure 6.3a.
Figure 6.3: Example of a teddy bear model at different resolutions of the progressive format (1 draw call) and its original format (16 draw calls). The size in KB assumes downloading progressively, e.g., 6.3e’s size includes lower-resolution textures.

<table>
<thead>
<tr>
<th>Progressive</th>
<th>128</th>
<th>256</th>
<th>512</th>
<th>1024</th>
<th>2048</th>
</tr>
</thead>
<tbody>
<tr>
<td>0%</td>
<td>0.53</td>
<td>0.63</td>
<td>0.81</td>
<td>1.03</td>
<td>1.35</td>
</tr>
<tr>
<td>25%</td>
<td>0.65</td>
<td>0.75</td>
<td>0.97</td>
<td>1.16</td>
<td>1.45</td>
</tr>
<tr>
<td>50%</td>
<td>0.74</td>
<td>0.85</td>
<td>1.02</td>
<td>1.26</td>
<td>1.58</td>
</tr>
<tr>
<td>75%</td>
<td>0.79</td>
<td>0.95</td>
<td>1.11</td>
<td>1.34</td>
<td>1.70</td>
</tr>
<tr>
<td>100%</td>
<td>0.88</td>
<td>0.99</td>
<td>1.20</td>
<td>1.44</td>
<td>1.82</td>
</tr>
</tbody>
</table>

Figure 6.4: Mean size of progressive format as a fraction of the original across all test models, shown as a function of the progressive stream downloaded and texture resolution.

6.4 Table

Tables can be put inside the figtable directive which automatically numbers them, adds a caption, and adds a label.

Table 6.4 has all right-aligned columns.
<table>
<thead>
<tr>
<th>Left Align</th>
<th>Right Align</th>
</tr>
</thead>
<tbody>
<tr>
<td>Some text is left aligned</td>
<td>Followed by right-aligned</td>
</tr>
<tr>
<td>Some more text here</td>
<td>And more text here</td>
</tr>
<tr>
<td>And even more text</td>
<td>Also even more text here</td>
</tr>
</tbody>
</table>

**Figure 6.5:** This table has mixed alignment

Table 6.5 has one column left-aligned and one column right-aligned.

### 6.5 Text Wrapping Table

Text wrapping in tables work if you specify the width and either `raggedleft` or `raggedright`.

A text wrapping table example is shown in Figure 6.6.
<table>
<thead>
<tr>
<th>Column Family</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Users</strong></td>
<td>Stores a list of users who have authenticated with OpenID.</td>
</tr>
<tr>
<td><strong>Names</strong></td>
<td>Stores a list of the 3D models in the database with their associated metadata.</td>
</tr>
<tr>
<td><strong>TempFiles</strong></td>
<td>Temporarily stores the binary file data of uploaded files until they have been processed.</td>
</tr>
<tr>
<td><strong>Files</strong></td>
<td>Stores the binary file data for uploaded and verified files.</td>
</tr>
<tr>
<td><strong>Sessions</strong></td>
<td>Stores HTTP session information used by the Django framework to look up session state associated with a user’s browser cookie.</td>
</tr>
<tr>
<td><strong>OpenIdAssocs,</strong></td>
<td>Stores OpenID authentication information for users.</td>
</tr>
<tr>
<td><strong>OpenIdNonces</strong></td>
<td></td>
</tr>
<tr>
<td><strong>CeleryResults</strong></td>
<td>Stores the result of application processing tasks (see Section something).</td>
</tr>
<tr>
<td><strong>APIConsumers</strong></td>
<td>Stores a list of consumers of the API for use with the OAuth protocol.</td>
</tr>
</tbody>
</table>

Figure 6.6: A list of Open3DHub’s Cassandra column families and their descriptions
Bibliography

