



# Jalapeno

## How This Book Works

---

Allegan County GIS  
[www.allegancounty.org/gis](http://www.allegancounty.org/gis)

August 6, 2018



# Contents

0.0.1	How Jalapeño Works . . . . .	1
	General Notes: . . . . .	1
	Project file structure: . . . . .	1
	Using the glossary . . . . .	1
	Glossary requirements: . . . . .	1
	Creating a new glossary entry . . . . .	2
	Rebuilding the glossary . . . . .	2
	Using glossary terms in a subdocument: . . . . .	2
	Using the bibliography(References) . . . . .	2
	Bibliography requirements: . . . . .	2
	Creating a new bibliography entry . . . . .	3
	Rebuilding the bibliography . . . . .	3
	To cite a bibliography source in a subdocument: . . . . .	3
	Using the Index . . . . .	3
	Index requirements: . . . . .	3
	Creating a new index entry . . . . .	3
	Rebuilding the index . . . . .	3
	Using index terms in a subdocument: . . . . .	4
	Using the Appendices . . . . .	4



## 0.0.1 How Jalapeño Works

### General Notes:

- jalapeno folder is a git package.  
<https://github.com/nbesteman/jalapeno>
- Project is coded with relative paths and jalapeno can be located anywhere.

### Project file structure:

...\jalapeno\..

folder	description
documentation	resources used in Jalapeño
processing	.tex documents and build folders
source	common image files

...\jalapeno\documentation\..

folder or file	description
moduleTemplates	.tex templates
packageDocs	L <sup>A</sup> T <sub>E</sub> X documentation
references	reference and appendix resources
unsorted	catch all for unsorted documentation
BookStructureMM.mm	A mindmap of jalapeno

...\jalapeno\processing\..

folder or file	description
...Part	folders of book <i>parts</i>
build	L <sup>A</sup> T <sub>E</sub> X workspace and location of .pdf output and referenceEntries.bib*
commonTitle.tex	code for all title pages
fullCompile.sh	shell script to compile GISDocumentation.tex
GISDocumentation.tex	master document code
glossaryEntries.tex	entries that appear in glossary
indexEntries.tex	entries that appear in the index
preamble.tex	preamble code for all documents

**\*Note about referenceEntries.bib** Any reference entries built here can be cited in any .tex document in the project.

## Using the glossary

**Glossary requirements:** Glossary commands require a Perl interpreter. Activeperl is a free Perl interpreter and can be downloaded from:

<https://www.activestate.com/activeperl/downloads> (A typical installation adds Perl to your

`path`). Compiling the glossary requires running the `makeglossaries` command either in a  $\text{\LaTeX}$  IDE or in command line as described here. PDFLatex must be run first to create a `.aux` file that is used by `makeglossaries` to create an `.gls` file. After the `.gls` file is created, PDFLatex must be run again to insert the glossary at the `\printglossaries` location.

**Creating a new glossary entry** To create a new glossary entry: Add an entry to `glossaryEntries.tex`. Save it there and then use the `makeglossaries` command to recompile the `.gls` file.

**Rebuilding the glossary** To Recompile the `.gls`. In the (main document)build folder:

- Launch command prompt
- enter command: `makeglossaries GISDocumentation*`

**\*Note:** This command reads the `.aux` file and creates the `.gls` file. The `.aux` file is created by compiling with PDFLatex. If there is no `.aux` file the command will fail.

**Using glossary terms in a subdocument:** In the subdocument you must add code to input the `glossaryEntries` file. For example:  
After the line:

```
\input{../../../../preamble}
```

Add the line:

```
\input{../../../../glossaryEntries}
```

**To use a glossary term in the subdocument:**

In place of the term, use code referencing the key (in the `glossaryEntries` file):

- `\gls{key}`

**To add the glossary to the subdocument:**

- Add the line `\makeglossaries` to the preamble of the subdocument.
- Add the line `\printglossaries` to the subdocument.
- Run `makeglossaries` in command line on the subdocument similar to how is described above.

## Using the bibliography(References)

**Bibliography requirements:** Compiling the bibliography requires running `bibtex` either in a  $\text{\LaTeX}$  IDE or in command line as described here. PDFLatex must be run first to create a `.aux` file that is used by `bibtex` to create a `.bbl` file. After the `.bbl` file is created, PDFLatex must be run again to insert the bibliography at the `\bibliography` location.

For example, the command: `...\bibliography{referenceEntries}` ...places the bibliography called `referenceEntries.bib` which must be in the same folder as the project `.aux` file.

**Creating a new bibliography entry** To create a new bibliography entry: Add an entry to `referenceEntries.bib`. Save it there and then use `bibtex` to recompile the `.bbl` file.

**Rebuilding the bibliography** To Recompile the `.bbl`. In the (main document)build folder:

- Launch command prompt
- enter command: **`bibtex GISDocumentation`**

**\*Note:** This command reads the `.aux` file and creates the `.bbl` file. The `.aux` file is created by compiling with PDFLatex. If there is no `.aux` file the command will fail.

**To cite a bibliography source in a subdocument:**

In the place that you want the citation:

- `~\cite[pg.#]{key}`

**To add the bibliography to the subdocument:**

- Similar to adding to the master document but not documented here.

## Using the Index

**Index requirements:** Compiling the index requires running the `makeindex` command either in a  $\text{\LaTeX}$  IDE or in command line as described here. PDFLatex must be run first to create a `.aux` file that is used by `makeindex` to create an `.idx` file. After the `.idx` file is created, PDFLatex must be run again to insert the index at the `\printindex` location.

**Creating a new index entry** To create a new index entry: Add an entry to `indexEntries.tex`. Save it there and then use the `makeindex` command to recompile the `.idx` file.

**Rebuilding the index**

**To Recompile the `.idx`**In the (main document)build folder:

- Launch command prompt
- enter command: **`makeindex GISDocumentation*`**

**\*Note:** This command reads the .aux file and creates the .idx file. The .aux file is created by compiling with PDFLatex. If there is no .aux file the command will fail. Run PDFLatex first

**Using index terms in a subdocument:** In the subdocument you must add code to input the indexEntries file. For example:  
After the line:

```
\input{../../../preamble}
```

Add the line:

```
\input{../../../indexEntries}
```

**To use a index term in the subdocument:**

In place of the term, use code referencing the key (in the indexEntries file):

- `\index {key}`

**To add the index to the subdocument:**

- Add the line `\makeindex` to the preamble of the subdocument.
- Add the line `\printindex` to the subdocument.
- Run `makeindex` in command line on the subdocument similar to how is described above.

## Using the Appendices