

Basic Book Builder

A Pandoc build process for books, docs and articles.



Basic Book Builder

A Pandoc build process for books, docs and articles.

Author:	John Haverlack
Version:	1.2.3
Date:	2026-02-05
Affiliation:	Alaska Center for Energy and Power
Institution:	University of Alaska Fairbanks
ISBN:	UNSPECIFIED
DOI:	UNSPECIFIED
Maturity:	BETA

Copyright © 2026 Alaska Center for Energy and Power

Licensing

Open Source Edition (PDF and HTML)

Licensed under the Creative Commons Attribution-NoDerivatives 4.0 International License.

- License: [CC BY-ND 4.0](#)

Contents

1	Introduction	1
1.0.1	Workflow	1
2	Getting Started	2
2.1	Clone the Basic Book Builder	2
2.2	Setting Up Build Environment	2
2.2.1	Non deb systems	2
2.3	Testing	3
2.4	Creating a New Project	3
2.5	Building	3
2.5.1	PDF	3
2.5.2	HTML	4
2.5.3	DocX	4
2.5.4	ODT	4
2.5.5	Make All	4
3	Features	5
3.1	Metadata	5
3.2	Front-matter	5
3.2.1	Meta Data Page	5
3.2.2	Cover Illustration	5
3.2.3	Dedication	5
3.3	Table of Contents	6
3.4	Chapters	6
3.5	Appendixes	6
3.6	Glossary	6
3.7	Index	6
3.7.1	Markdown Indexes	6
3.8	Bibliography	6
3.8.1	Zotero Integration	6
3.8.2	Obsidian Citations	6
3.9	Images	7
3.10	<i>LaTeX</i> Equations	7
3.10.1	Left Justified / Inline	7
3.10.2	Centered	7
3.11	Custom Markdown	8
3.11.1	Call-outs	8
3.12	Advanced	11

CONTENTS

3.12.1	Changing Fonts	11
3.12.2	Adding Filters	11
4	Design	12
4.1	Directory Structure	12
4.2	Configuration Files	12
4.2.1	Pandoc Main Config	13
4.3	Build Workflow	14
	Glossary	15
	Index	17
	Bibliography	19

Introduction

This is a **basic book builder** template is based on a Pandoc([Pandoc - Index, n.d.](#)) + Makefile build process in conjunction Obsidian to edit content and Zotero to manage citation, to generate multiple format of the document including:

- PDF
- HTML
- *LaTeX*
- Markdown
- ODT
- DOCX
- Epub

1.0.1 Workflow

1. Create a new project
2. Initialize `metadata.yaml` with title, copyright, author, etc.
3. Customize `input-files:` in `pandoc.yaml`
4. Edit chapters Markdown “content” files using Obsidian
5. Build your document

This **basic book builder**([Jehaverlack/Basic-Book-Builder, n.d.](#)) template has been curated by John Haverlack.([John Haverlack | ACEP, n.d.](#))

Getting Started

2.1 Clone the Basic Book Builder

Clone the repository

```
git clone https://github.com/jehaverlack/basic-book-builder.git
```

Navigate to the repo directory

```
cd basic-book-builder
```

2.2 Setting Up Build Environment

Debian Linux

This script only support Debian based Linux system (e.g Debian, Ubuntu, Zorin). And this script will install several ATP packages on your system.

Warning

This script **has not** been widely tested across many platforms

To initialize your systems build environment run:

Setup Build Environment

```
./scripts/setup-linux-deb.sh
```

2.2.1 Non deb systems

For more information on setting up the build environment for your system see: **TBD: NEEDS DOCUMENTED.**

2.3 Testing

Once you have set up your build environment, you test by building the **basic book builder**.

Testing Build Environment

```
make all
```

2.4 Creating a New Project

Tip

You do not want to edit your new project document in the `basic-book-builder` directory. You will want to create a **new project** for your document based on this template.

Create a new project

```
./scripts/new-project.sh <NEW_PROJECT_DIR>
```

```
cd <NEW_PROJECT_DIR>
```

Update your project configuration:

1. Edit: `metadata.yaml`, for title, author, etc
2. Edit: `templates/pandoc.yaml`
3. Open: `<NEW_PROJECT_DIR>` with Obsidian, and edit `chapters` documents.
4. Build: `make pdf`

Warning

`<NEW_PROJECT_DIR>` should be outside of the **basic-book-builder** directory

2.5 Building

2.5.1 PDF

```
make pdf
```

NOTE: Building the PDF also builds the **LaTeX** version of the book.

2.5.2 HTML

```
make html
```

2.5.3 DocX

```
make docx
```

NOTE: DocX is a work in progress. Frontmatter and formatting need work.

2.5.4 ODT

```
make odt
```

NOTE: ODT is a work in progress. Frontmatter and formatting need work.

2.5.5 Make All

To build all supported formats

```
make all
```

Features

The basic book builder template integrates feature from many systems including Pandoc, MathJax, Obsidian, and Zotero. The following list covers these features from a functional point of view.

3.1 Metadata

Metadata for the project is managed in `metadata.yaml`. Edit this file to make **global changes** to variables for your document. Definitions in `metadata.yaml` will propagate to all format builds.

3.2 Front-matter

Front-Matter documents control the Title Page, Metadata and other pages before the Title of Contents. Front matter content is manged in these files:

- `templates/frontmatter.tex`
- `templates/frontmatter.html`
- `templates/frontmatter-md.html`

Unfortunately when you edit one, you need to edit the respective changes to the other front-matter formats. **### Title Page**

Edit the **Front-matter** files to modify the formatting of the Title Page. You'll need to modify the HTML / LaTeX code.

3.2.1 Meta Data Page

Edit the **Front-matter** files to modify the 2nd page Metadata format and fields to display.

3.2.2 Cover Illustration

Edit the **Front-matter** files to optionally un-comment the Cover Art Acknowledgement.

- This is disabled by default

3.2.3 Dedication

Edit the **Front-matter** files to modify un-comment the Book Dedication.

- This is disabled by default

3.3 Table of Contents

The table of contents is auto generated. But can be disabled in `template\pandoc.yaml` by setting: `-toc: false`

3.4 Chapters

Chapter Markdown source files are located in the `chapters` directory. To edit which chapter files are included the document:

- Edit the `input-file:` section of the `template\pandoc.yaml`

3.5 Appendixes

TBD

3.6 Glossary

TBD

3.7 Index

TBD

3.7.1 Markdown Indexes

TBD

3.8 Bibliography

TBD

3.8.1 Zotero Integration

TBD

3.8.2 Obsidian Citations

TBD

3.9 Images

You can use basic Markdown Images

```
! [ACEP] (lib/img/ACEPLogo.png)
```



Figure 3.1: ACEP

3.10 *LaTeX* Equations

You can embed LaTeX equations directly in your Markdown pages.

3.10.1 Left Justified / Inline

To left justify *LaTeX* surround the code inside single \$:

Left Justified Equation
<code>\$x = \frac{1}{2}\$</code>

$$x = \frac{1}{2}$$

Inline
Or inline, <code>\$x = \frac{1}{2}\$</code> , in a sentence.

Or inline, $x = \frac{1}{2}$, in a sentence.

3.10.2 Centered

To center justify *LaTeX* surround the code inside double \$\$:

Centered Equation

```
$$x = \frac{1}{2}$$
```

$$x = \frac{1}{2}$$

3.11 Custom Markdown

3.11.1 Call-outs

A few call-out box styles have been custom added to easily highlight certain types of content.

Markdown Example

```
[!established] Green Call-out Title  
A green box
```

Will render:

Green Call-out Title

A green box

Markdown Example

```
[!proposed] Blue Call-out Title  
A blue box
```

Will render:

Blue Call-out Title

A blue box

Markdown Example

```
[!speculative] Purple Call-out Title  
A purple box
```

Will render:

Purple Call-out Title
A purple box

Markdown Example
<code>[!caution] Caution Note</code> Beware of this section. AKA a orange box.

Will render:

Caution Note
Beware of this section. AKA a orange box.

Markdown Example
<code>[!warning] Warning Note</code> Same as the Caution Box

Will render:

Warning Note
Same as the Caution Box

Markdown Example
<code>[!danger] Alert</code> A slightly stronger warning. AKA a red box.

Will render:

Alert
A slightly stronger warning. AKA a red box.

Markdown Example

```
[!code] Code Block
'''
if [ true ]; do
  echo "The Cow Jumps over the Moon"
fi
'''
```

Will render:

Code Block

```
if [ true ]; do
  echo "The Cow Jumps over the Moon"
fi
```

Markdown Example

```
[!cli] Command line
A command line example
'''
$ ls
build  chapters  conf  filters  lib  LICENSE.md  Makefile
metadata.yaml >README.md  scripts  templates
'''
```

Will render:

Command line

```
A command line example
$ ls
build  chapters  conf  filters  lib  LICENSE.md  Makefile
metadata.yaml README.md  scripts  templates
```

3.12 Advanced

3.12.1 Changing Fonts

It is possible to change fonts by editing `templates/pandoc.yaml`, and `templates\styles.css`, but you will have to make sure you have the proper font's installed on your system first.

3.12.2 Adding Filters

Custom filters can be created. This is complicated and typically needs done in 3 places for:

- `obsidian`
- *LaTeX*
- `HTML`

Design

4.1 Directory Structure

The **basic-book-builder** directory tree.

```
basic-book-builder
  build
    docx
    html
      conf
      lib
        diag
        img
        mathjax
    latex
    odt
    pdf
  chapters
  conf
  filters
  lib
    citations
    diag
    img
    mathjax
  scripts
  tmp
  templates
```

4.2 Configuration Files

The **basic-book-builder** configuration files, scripts, and build files.

```
../basic-book-builder
  chapters
  conf
  filters
```

```

    callouts-html.lua
    callouts.lua
lib
    diag
    img
    zotero.bib
Makefile
metadata.yaml
scripts
    new-project.sh
    replace-metadata.sh
    setup-linux-deb.sh
    tmp
templates
    docx.yaml
    epub_template.html
    epub.yaml
    frontmatter-docx.md
    frontmatter.html
    frontmatter-md.html
    frontmatter.tex
    header.tex
    html.yaml
    latex.yaml
    markdown.yaml
    odt.yaml
    pandoc.yaml
    pdf.yaml
    style.css
    style_epub.css

```

About conf

NEVER: edit file directly in `conf`. They will be overwritten by the build process. ###
Metadata

Update the `metadata.yaml` file to globally configure your document title and other metadata. Edit this file to control your document title, author, etc.

4.2.1 Pandoc Main Config

Update the `input-files:` section of `templates/pandoc.yaml` to control what chapter source files are in which order in your document.

```
# pandoc.yaml
```

```
input-files:
# - chapters/Preface.md
- chapters/00_Introduction.md
- chapters/01_Getting_Started.md
- chapters/02_Usage.md
- chapters/03_Features.md
- chapters/Appendix.md
- chapters/AppendixA.md
# - chapters/AppendixB.md
- chapters/Glossary.md
- chapters/Index.md
- chapters/Bibliography.md
```

4.3 Build Workflow

For a PDF Build:

```
make pdf
```

will build dependencies on these file:

1. Makefile
2. conf/pandoc.yaml
3. conf/pdf.yaml
4. lib/zotero.bib
5. chapters/*.md

NOTE: the Makefile will copy templates/*.yaml to conf/*.yaml. **Never** edit conf/*directly.

Glossary

Term	Description
AST (Abstract Syntax Tree)	Pandoc’s internal structured representation of a document. Lua filters operate on the AST before rendering to a target format.
Bibliography	A <code>.bib</code> file (e.g., Zotero export) used by Pandoc with <code>citeproc</code> to generate formatted citations and reference lists.
Callout	A styled block (e.g., <code>[!warning]</code>) transformed by Lua filters into format-specific output such as LaTeX environments or HTML <code><div></code> elements.
Defaults File (YAML)	A Pandoc configuration file defining output format, filters, metadata, and rendering options (e.g., <code>pdf.yaml</code> , <code>html.yaml</code>).
DOCX	Microsoft Word document format generated by Pandoc; styling is controlled via a <code>reference.docx</code> file.
Filter (Lua)	A script that modifies Pandoc’s AST before rendering. Used in this workflow for callouts and other structural transformations.
Frontmatter	Format-specific content inserted before the main body (e.g., <code>frontmatter.tex</code> , <code>frontmatter.html</code> , <code>frontmatter-docx.md</code>).
HTML	Web output format generated by Pandoc, typically styled via CSS and supporting interactive features.
LaTeX	Typesetting system used for high-quality PDF generation via XeLaTeX in this workflow.
Lua Filter	A lightweight scripting extension used by Pandoc to customize document transformations programmatically.
Makefile	Automation script orchestrating the build pipeline for PDF, HTML, DOCX, ODT, and Markdown outputs.
Markdown	Plain-text markup language serving as the source format for all outputs in the workflow.
Metadata	Key–value pairs (e.g., <code>metadata.yaml</code>) defining title, author, license, version, and other document variables.
ODT	OpenDocument Text format used by LibreOffice; similar to DOCX but open-standard.
Pandoc	Universal document converter used to transform Markdown into PDF, HTML, DOCX, ODT, EPUB, and more.

CHAPTER 4. DESIGN

Term	Description
PDF	Portable Document Format output produced via Pandoc + XeLaTeX.
Reference Document	A DOCX or ODT file defining styles (fonts, headings, spacing) used when generating Word/LibreOffice outputs.
Resource Path	Directories Pandoc searches for images, diagrams, and other assets during rendering.
Template	A format-specific layout file (e.g., LaTeX template) used to control document structure beyond styling.
TOC (Table of Contents)	Automatically generated navigation structure controlled by <code>toc</code> and <code>toc-depth</code> options.
XeLaTeX	LaTeX engine used to generate PDFs with modern font support (via <code>fontspec</code>).
Zotero	Reference manager used to generate <code>.bib</code> files for citation processing with Pandoc.

Index

Pandoc, [1](#)

Bibliography

Bibliography

Jehaverlack/Basic-Book-Builder: A Pandoc Workflow for Building PDF, HTML, EPUB, and LaTeX Book Format. n.d. <https://github.com/jehaverlack/basic-book-builder>.

John Haverlack | ACEP. n.d. <https://www.uaf.edu/acep/about/our-team/john-haverlack.php>.

Pandoc - Index. n.d. <https://pandoc.org/>.

Bibliography