PreTeXt Publisher’s Guide

PreTeXt Publisher's Guide

Robert A. Beezer
University of Puget Sound

David Farmer
American Institute of Mathematics

DRAFT November 26, 2018 DRAFT

Preface

Once you have *authored* your document, then it becomes time to *publish* it. You could engage a publisher to help you, likely at some cost, such as surrendering your copyright. Or perhaps you intend to retain ownership of your copyright and use it to provide an open license. Then you are likely both the author and the publisher. In this case, conversions from PreTeXt to various output formats will help you with many of the tasks a publisher might help you with. But there is still more to know and do. This guide will help you.

Contents

Chapter 1

(*) Producing a Book

You have finished using PreTeXt to write a textbook, research monograph, laboratory manual, writer's handbook, or some other scholarly work, and you want to share it openly with students and other readers. Good! That's why we created PreTeXt, and why we include this chapter. There is more to do.

This chapter will try to distill our personal experience, along with many conversations with other authors confronting these same decisions. But recognize there is a lot of *advice* here, and a bit of an attitude, simply for your consideration. The decisions are yours.

Open Source. PreTeXt is open source *software*, but that places no restrictions on how you use output that PreTeXt creates (see Item B.0.1:8). You are welcome to sell your copyright to a commercial publisher or university press. But just once, here and now, we will encourage you to consider an open license (see Chapter 2) as a way of paying it forward.

1.1 (*) Front Matter

TODO

1.2 (*) Landing Page

TODO

1.3 (*) Source

TODO

1.4 (*) Marketing

Didn't think about that one? Write it, and they will come? Not necessarily. I have seen nice projects where authors make little extra effort to get the word out, and it shows. With the Internet, effective promotion can be accomplished without much effort or expense. And very soon your project can rise very high in search engine rankings.

1.4.1 (*) Social Media

TODO

1.4.2 (*) Reference Sites

TODO

1.4.3 (*) Analytics

TODO

1.4.4 (*) Discussion Groups

TODO

Chapter 2

Copyright and Licensing

The legal issues described here are based on the authors' experiences and study, which necessarily reflect the laws of the United States. But the Berne Convention, which dates to 1886, has 172 parties, so law and practice are very similar the world over. The United States acceded to the convention in 1988.

2.1 Copyright

Copyright is a monopoly granted by the government. It gives the author control over reproductions, translations, adaptations, performances, communications, etc. of their work for a fixed time. Since 1978, it has not been necessary to register a copyright—it is automatic. So for example, every web page, no matter how simple or unrefined, is copyrighted by its author.

The phrase "All Rights Reserved" is used to assert that the copyright holder intends to exercise all the rights granted by copyright. It is not required to mark a document with the copyright symbol (©) but in case of disputes, it can be helpful.

So in PreTeXt you can go

```
<frontmatter>
  <colophon>
    <copyright>
      <holder>Thomas Jefferson</holder>
      <year>1776</year>
      <minilicense>All Rights Reserved</minilicense>
    </copyright>
  </colophon>
</frontmatter>
```

to assert a "traditional" copyright and the recommended information will then appear on the page after the title page.

In academic publishing, authors have usually transferred, or sold, their copyright to a publisher in return for distribution of their work, or for the promise of financial gain.

2.2 Open Licenses

Copyright allows an author to place a license on their work, granting others greater freedoms, sometimes along with certain specific obligations. So it is important to understand that copyright allows an author or publisher to be very restrictive, and it also allows an author or publisher to be less restrictive. A license makes these less restrictive terms explicit, and the ability to control these terms is made possible by copyright.

Generally an open license allows unlimited copying. It often allows the creation of derivative works, and the mixing of material from a variety of openly licensed documents. A **viral license** obliges the author of a derivative work to grant the same license to the derivative work, rather than asserting more restrictive terms. The licenses are usually perpetual, so they do not expire at a fixed term.

It is easy to get distracted by legal jargon, obtuse arguments, and misunderstandings. We view an open license as statement of intent. The work is free to use forever. It will not go out of print. If you send the author a correction, suggestion, or contribution, it can be incorporated and enjoyed by others freely. And should an author lose interest in a project, or become unable to continue working on it, another individual may take it up and continue to maintain it.

2.3 Creative Commons Licenses

We describe the Creative Commons (CC) licenses first, since they have various options, which are a convenient way to compartmentalize and describe the features of other open licenses. They are known by abbreviations, so a license might be shortened to something like just `CC BY-SA`.

Creative Commons, `CC` All Creative Commons licenses allows unlimited copying, forever. Unless restricted, derivative works are allowed with no obligation.

Attribution, `BY` A derivative work must make clear the contribution of the original author.

Share Alike, `SA` A derivative work must be licensed with the same license as the original.

Non-Commercial, `NC` Bans commercial uses. (Without this clause an open license would allow the sale of copies.)

No Derivatives, `ND` Derivative works are not allowed.

List 2.3.1: Creative Commons Options

Source versus Output. Creative Commons licenses are designed for a variety of media, and so are very popular. For example, images are often licensed with a Creative Commons license.

However, consider the case of a document distributed as a PDF which has been created from source, such as PreTeXt or LaTeX or Markdown. An author may put a `CC BY-SA` license on the PDF while retaining traditional copyright on the source file(s). While the license allows modification, how practical is it to modify a PDF? Worse, we have seen this situation for projects described as "open source." This explains our use sometimes of the term "openly licensed content."

If you are serious about your project being open source, and you want to send that signal to your readers, then mark your output with a CC license *and* mark your source files as also having the same CC license. This is usually accomplished by a statement at the top of each source file asserting copyright and then stating the license. Or this statement can point to a top-level text file, often named `COPYING` or `LICENSE`, with more precise license information. See the PreTeXt distribution for examples using the GPL software license.

Commercial Consideration. It is natural to consider that you have devoted considerable time and effort to your project, and with an open license you will now be donating it to the world to read for free. So you might think, "Why should somebody else profit?" Thus, the `NC` option is alluring. But think about it for a minute.

Suppose a commercial publisher hires an experienced copy editor to go through your book, correcting errors and tightening the prose. Then the publisher prints the improved version of your book in a nice hardback version, and sells the book for $250. If you simply put the `BY-SA` options on your license, that publisher is obligated to (a) give you credit for authoring the original version of the book, and (b) make the copy-edited version available again with a `CC BY-SA` license. Now there is a business opportunity for anybody in the world: make a print-on-demand version of the copy-edited version, and sell it for $200. Then it is a race to the bottom. Eventually the price will come down to the cost of manufacturing, plus some small compensation for managing the relationship with a print-on-demand service Chapter 13. Just how much profit potential is there really available for others?

We assume you have written a book because you have something to say, and chosen an open license because you want to be read. If you do place an NC option on your CC license, then you have made it impossible for anybody else to help you distribute your book in physical forms. A CC BY-SA license already makes your work unattractive to a commercial publisher who wants to charge an unreasonable price, and adding an NC clause simply chokes off some distribution options, limiting the reach of your work.

This discussion is meant to explain why we call it the "CC-NC mistake."

Derivative Works. Will your monograph on the reptiles of the Mongolian steppes ever become a screenplay? As exciting as the topic is to you, we think Hollywood feels otherwise. An ND option probably makes little sense for scholarly works. An conversely, if you get hit by a pie truck tomorrow, it will be a lot easier for an enthusiastic reader to take over custody and maintenance of your project, and they will be *required* to continue with the same license if you have employed the SA option.

So in PreTeXt you can go

```
<frontmatter>
  <colophon>
    <copyright>
      <holder>Steve Jobs</holder>
      <year>1984</year>
      <minilicense>Creative Commons BY-SA</minilicense>
    </copyright>
  </colophon>
</frontmatter>
```

for a CC license with the Attribution and Share-Alike options. If it was not obvious already, this is our recommendation for scholarly work if you choose to use a Creative Commons license. This is an example of what is known as a **copyleft** license.

2.4 GNU Free Documentation License

The GNU Free Documentation License (GFDL) is a license designed for documentation of open source computer programs licensed by the GNU Public License (GPL). However, it explicitly mentions textbooks as one possible use. (GNU is a recursive acronym for "GNU's Not Unix" and is the software project that originally built all the utilities which complement the Linux Kernel to make up an operating system.)

The GFDL is similar to a CC BY-SA license. It allows unlimited copying, forever. Modified versions that are distributed must acknowledge the original contributions and must also carry a GFDL license. So this is a viral license, always. And another example of copyleft.

However, the GFDL does not employ options like a Creative Commons license. More importantly, the GFDL is very explicit about source ("transparent" copies) and derived output ("opaque" copies), and the license applies to both versions. So the GFDL would say PreTeXt source is transparent, and a resulting PDF is opaque, and the license largely treats them identically.

Because the GFDL has the essential characteristics of CC BY-SA, and is so explicit about simultaneously licensing document source authored with a markup language along with output formats, it is our choice for textbook projects.

So in PreTeXt we might go something like

```
<frontmatter>
  <colophon>
    <copyright>
      <holder>Richard Stallman</holder>
      <year>1985</year>
      <minilicense>GNU Free Documentation License</minilicense>
    </copyright>
  </colophon>
</frontmatter>
```

The GFDL is also explicit about including the complete license with your document. You can find various places a version formatted for inclusion in a PreTeXt project, including as part of the source for this document.

2.5 Public Domain

Stating that your work is in the **public domain** basically means you relinquish all the rights you receive via your automatic copyright. So it is a very different situation from using copyright to provide an open license. Creative Commons uses CC0, “No Rights Reserved” to indicate this choice.

2.6 Remixing and License Compatibility

It is popular to describe the potential of remixing parts of open educational resources. For example, a literature professor might collect a variety of openly licensed poems into a reader for students in a course. When the licenses are viral, and different, there arises the problem of what license to put on the collection. Worse, one license might prohibit commercial uses, and another allow it, meaning the licenses are incompatible.

So some thought should go into the choice of a license when the work has the very real potential to be included in another, such as would be the case with a photograph. One solution is to provide more than one license (nothing about copyright prohibits this). Another solution is to avoid licenses with overly restrictive terms, such as restricting commercial use or derivative works.

Remember too, that in addition to multiple licenses, as the copyright holder you may offer your work to another project on different terms. So another author might ask if a chapter of your work may be included in their project, which might use a different license than yours (more or less restrictive), and yu can grant permission for that use under that license. Now there are two versions of your chapter, which could diverge over time if derivatives are allowed, available to others on different terms.

For mathematics books, we do not concern ourselves too much with the potential for remixing. Notation and dependencies make it hard to collect parts of various textbooks and assemble them into something new (and coherent).

Creative Commons maintains a matrix showing compatibility between their own licenses at their FAQ question, Can I combine material under different Creative Commons licenses in my work?, illustrating just how complicated this can become.

Finally, recognize that you can begin with a restrictive license and as you become more comfortable with the idea, change it to a more liberal license that applies to the work at the time of the change. Further, you can always change your license to a more restrictive version, but invariably, you cannot revoke an open license once granted. You could even stop offering an open license all together, and return to traditional copyright as you continue to improve your document. But the version that existed when you made that decision is still available for anyone to use, and possibly improve, independent of your own closed version.

After a while, you realize that openly licensing your writing project gives it an independence and freedom all of its own. It seems to be owned by everybody, and by nobody—at the same time.

2.7 A Final Comment

These discussions remind us of a letter by Thomas Jefferson, the principal author of the United States' *Declaration of Independence*, writing about another monopoly granted by governments—patents.

> If nature has made any one thing less susceptible than all others of exclusive property, it is the action of the thinking power called an idea, which an individual may exclusively possess as long as he keeps it to himself; but the moment it is divulged, it forces itself into the possession of every one, and the receiver cannot dispossess himself of it. Its peculiar character, too, is that no one possesses the less, because every other possesses the whole of it. He who receives an idea from me, receives instruction himself without lessening mine; as he who lights his taper at mine, receives light without darkening me. That ideas should freely spread from one to another over the globe, for the moral and mutual instruction of man, and improvement of his condition, seems to have been peculiarly and benevolently designed by nature, when she made them, like fire, expansible over all space, without lessening their density in any point, and like the air in which we breathe, move, and have our physical being, incapable of confinement or exclusive appropriation.
>
> —Thomas Jefferson
> Letter to Isaac McPherson
> August 13, 1813

Chapter 3

Conversions, Generally

A main goal of PreTeXt is to provide a language for describing a scholarly document by its structure, with contained content, and with no description of the presentation. It then becomes possible to use software to produce different formats, where the presentation takes advantage of that format and enhances the meaning of the content through the expression of the structure.

But different output formats have different capabilities. For example, a conversion to HTML can take advantage of knowls to organize smaller chunks of content, while a conversion to PDF can take advantage of page numbers for cross-references. And in these two examples, the capability of the one output format is mostly impossible or silly in the other. Look here for notes that are independent of the PreTeXt vocabulary, and specific to the format produced by a conversion.

3.1 Processing Parameters

Many of the options for different conversions are accomplished by command-line options, which are fed to `xsltproc` by what that program calls **string parameters**. Here are two examples:

```
xsltproc -stringparam html.knowl.example no mathbook-html.xsl fauna.xml
xsltproc -stringparam latex.font.size 14pt mathbook-latex.xsl fauna.xml
```

The first will make `<example>` content render on a page, rather than hidden in a knowl, which is the default for a conversion to HTML. And the second will use a larger overall font size, while trying to preserve the same number of characters per line (with consequent smaller margins).

Notice that these two examples do nothing to change the *words* that a conversion produces, which is why you do not specify them in your source, and why they are documented here in the *Publisher's Guide*. They are conversion options that do not affect your content in significant ways, and are related to the particular output format.

Chapter 4

(*) Conversion to Online HTML

Similar to the case for LaTeX conversion (Chapter 5) there is a variety of command-line processing parameters available (Section 3.1). Until we are ready to document these carefully, open `xsl/mathbook-html.xsl` and poke around to see what is available.

4.1 Knowled Content

We knowl examples, proofs, and inline exercises by default, mostly so a new author knows that presentation is an option. The knowlization of items can be controlled by a variety of command-line processing parameters (Section 3.1). While these are in flux, we do not list them here, but instead suggest you open `xsl/mathbook-html.xsl` and search on `html.knowl` to see what is available.

Note that some of these switches are for broad categories of items, for example, `html.knowl.theorem` will also knowl `<lemma>`s, `<corollary>`s, `<fact>`s, and more. This choice applies document-wide, there is no plan to support electing this on a case-by-case basis.

4.2 Lists

On a description list (`<dl>`), only `@width` values of `narrow` and `medium` are implemented. (2018-03-28)

Lists with several columns are rendered in row-major order, as of 2018-02-28. In other words, the first list items (`<li>`) in your source will populate the first row.

Chapter 5

Conversion to PDF and Print

The stylesheet `mathbook-latex.xsl` can produce two very similar outputs. Each is a file in LaTeX syntax, which can be converted to a PDF with a LaTeX executable ("engine"). However, there are two purposes for such a PDF. The first is a document which is meant to be read on a screen. We call this an **electronic PDF**. The second is meant to be printed as a physical book, so it would be the file you provide to a copy shop, campus bookstore, or print-on-demand service (see Chapter 13). We call this a **print PDF**.

As an illustration of the difference, an electronic PDF will contain cross-references that are active, and colored to be obvious to the reader. For the print PDF the same cross-reference will be black and inactive.

The `latex.print` switch controls the selection of the two output modes of this single stylesheet. The default is `no` for the electronic version, and `yes` will change to the print version. In this chapter, we describe both of these two, slightly dissimilar, conversions together.

Text Justification and Alignment. The `text.alignment` switch will control how text is aligned (such as in paragraphs). The value `justify` is the default, and `raggedright` will cause alignment on the left, with an uneven right edge, employing less hypenation.

One-Sided or Two-Sided. An electronic PDF will default to page layout appropriate for a document printed single-sided, which makes the most sense for a document that may not be printed, or which might be printed on a personal printer.

A print PDF will default to page layout appropriate for a document printed two-sided, which makes the most sense for a document that may be sent to a print-on-demand service or printed on a printer that will print on both sides of a sheet of paper.

These defaults may be overridden with the `latex.sided` switch with values `one` and `two`.

One-sided layout will default to symmetric left/right margins, and page headers with the page numbers *always* placed in the upper-right corner as part of default page headers. There will be no blank pages between chapters of a book.

Two-sided layout will have asymmetric margins with the ratio of inner (adjacent to spine) to outer at `2:3`. You need some extra space at the spine to compensate for the binding, but when a book is open, the two pages are separated by two inner margins, so these do not need to be as wide as the outer margin to give some distance between the pages. And readers will want more space to write in the outer margins, perhaps providing simple proofs of important results. (This ratio may be changed with the `hmarginratio` key of the `geometry` package.) Page headings will have page numbers on the outside of the page, with odd numbers on the right-side page. Chapters will possibly have a blank page between them, so they begin on a right/odd page. Behavior is similar in the front matter and back matter.

Page Shape. The `latex.geometry` switch can be used to supply a single string composed of any options to the `\geometry{}` command of the LaTeX `geometry` package, and these will override any defaults supplied by PreTeXt. Resist the temptation to pack in as much text on the page as you can. PreTeXt varies the width of the text in reaction to the font size and is already very close to the maximum number of characters per line for comfortable reading by humans.

An electronic PDF may be printed on physical paper, but perhaps you want to make a version that works well on a portable device that naturally supports a portrait orientation, such as an Android tablet, an iPad, a Kindle (device or application), smart phone, Sony Digital Paper, or a ReMarkable tablet. Aspect ratios vary across these devices, but once you settle on a target ratio, we have had good luck with the following algorithm and parameters:

1. Specify 10 point text
2. Text width of about 4.5 inches
3. Add quarter-inch left/right margins to compute text width
4. Use aspect ratio to compute an overall height (about 6.5 inches)
5. Subtract quarter-inch top and bottom margins to obtain text height

Then you can provide the `geometry` package the overall size as the `papersize` and the text width and text height as the `total` size of the body, resulting in equal (tight) margins all around, and good use of limited screen real estate. These parameters create a PDF that is very legible on a larger smart phone, and for fine detail, rotating the device to landscape works well. Really.

Nested Lists. LaTeX can fail if lists are nested too deeply. Maximums may be up to four nested ordered lists, and up to six overall (mixing in unordered lists). If you hit these limits, ask yourself if your situation is really that complicated, or ask us to consider a feature request adding a technical fix.

Multi-Column List Order. Lists with several columns are rendered in column-major order, as of 2018-02-28. In other words, the first list items (`<li>`) in your source will populate the first column.

Watermarks. You can add a watermark to your PDF which will render as large, light grey text, at an angle across teh page. Use `latex.watermark` to set the text (avoid obscure characters or symbols, and do not use any XML markup). Use `latex.watermark.scale` to adjust for long or short text. The LaTeX-only "\\" will survive in the text to create multiple lines of text. Submit a feature request if this is not flexible enough to meet a special need.

Cover Images. The `docinfo/covers` element may be used to specify the filenames (with relative paths) of a front-cover image and a back-cover image via the `@front` and `@back` attributes. The image must be a single-page PDF, and it will be scaled to fit an entire page. So it is your responsibility to supply an image which has the correct aspect ratio and sufficient resolution. These are only supported for the LaTeX conversion.

This is meant to help you create a professional electronic PDF. A print-on-demand service (Chapter 13) will likely want a standalone image (possibly with the front and back, plus a spine, all rolled into one **wrap** image). So build your *real* cover images (Chapter 12), and then modify them for this use.

Fill-Ins. The `<fillin>` element can be rendered as a line near the baseline of the current line of text, or as a complete rectangle ("box"). Set `latex.fillin.style` to the values `underline` or `box`.

Experimentation. Two hooks are provided to allow for arbitrary additions to the LaTeX preamble. It should go without saying that this is for experimenting with new features and is no way supported, including, but not limited to, interactions with current LaTeX packages in use, or those added in the future.

Arbitrary text (LaTeX) used as the values of `latex.preamble.early` and `latex.preamble.late` will be added to the very beginning, or very end (respectively), of the generated preamble.

Styling. Note that some of these switches are about *style*. There are many more ways to influence the style of the LaTeX output, see Chapter 11.

Switch Reference. The following switches can be provided on the command-line, or perhaps more conveniently in a simple XSL file. See Section 3.1 for more detail.

Switch	Default	Values
`latex.print`	`no`	`yes`, `no`
`latex.sides`	Varies	`one`, `two`
`latex.font.size`	`10pt`	`8pt` to `20pt` (8 steps)
`latex.geometry`		Any options to `\geometry{}`
`text.alignment`	`justify`	`raggedright`, `justify`
`latex.watermark`		Simple text, no markup
`latex.watermark.scale`	`2.0`	Positive rational number
`latex.fillin.style`	`underline`	`underline`, `box`
`latex.preamble.early`		Arbitrary LaTeX
`latex.preamble.late`		Arbitrary LaTeX

Table 5.0.1: Options for LaTeX Conversion

LaTeX File. The LaTeX file created by PreTeXt will contain the majority of your content in a form that you could use it in a new standalone LaTeX document, in accordance with Principle B.0.1:9. However some constructions which are not natural in LaTeX, such as a `<sidebyside>`, may be cumbersome to reuse. We continue to improve and refine these situations, though.

Our philosophy is to create and use many new LaTeX environments, allowing styling and fine-tuning to occur in the preamble. This makes the body look more like simple LaTeX and allows for much greater flexiblity in styling, along with greater reliability for successful LaTeX compilation.

The existence, variety, and quality of LaTeX packages changes continuously. We can, and will, swap out some packages for replacements, as needed or desirable. This is to your advantage, as you are absolved of the need to evaluate competing packages, and to insure that they do not clash with each other. So resist the temptation to modify the LaTeX output significantly prior to compilation, as it will inevitably lead to frustration. The LaTeX file is a means to an end—it allows us to create a PDF with excellent typography, and especially for the demands of technical discplines, such as STEM and music. It is meant to ephemeral, not archival.

If some other variety of LaTeX (or TeX) file is desired, a new conversion could be created. Many of the more complicated aspects of any conversion are purposely isolated in the `mathbook-common.xsl` file so that they can be easily re-purposed and there is consistency across output formats.

Chapter 6

(*) Conversion to EPUB

TODO

Chapter 7

(*) Conversion to Jupyter Notebooks

TODO

Chapter 8

Instructor's Version

Once your content is in place, you can begin thinking about various useful derivative versions. A natural example for a textbook is an "Instructor's Version", enhanced with additional material to help an instructor understand your organization and intent, or to provide advice and counsel about teaching the material.

8.1 Solutions

Philosophies about the purpose and use of exercises varies among authors and instructors. Some think hints, answers, and/or solutions, should be universally available to students to use responsibly. Others like to assign exercises to be graded as part of a course grade. Some are resigned to solutions that are distributed in a limited fashion eventually becoming public, or that some groups of students will distribute their own solutions, possibly not uniformly. Wherever you place yourself in this debate, distributing solutions to only instructors is one approach, and some instructors may find this a very helpful aid when they teach material new to them.

There is flexibility in which of `<hint>`, `<answer>`, and `<solution>` can be included or excluded in your text, which can be included or excluded in an Instructor's Version, and which can be included or excluded in a Solution Manual (see Section 9.1). You can choose to author these or not, and then decide which to include in the student version, and which to include in an independently-produced Instructor Version, and which to include in a Solution Manual.

Read Section 9.2 for some practical advice about limiting distribution of solutions.

8.2 Notes and Commentary

The `<commentary>` element is designed primarily for the purpose of adding material to a document to make an enhanced version. It is similar in many ways to a `<paragraphs>` in that it can be placed within any division and must be titled. The main difference is that it is not displayed by default, so you must set a processing switch to enable its appearance:

```
xsltproc -stringparam commentary yes ...
```

Other distinctions are:

- Since it is elective, you need to be careful about cross-references to and from a `<commentary>`. It is highly likely that you will want to make cross-references *within* a `<commentary>` *pointing to* other portions of your text, and this is always a good idea. You will want to avoid making cross-references *to* a `<commentary>` from other parts of the text, with the exception of a cross-reference that originates *within* some `<commentary>`.

- Numbered items are prohibited within a `<commentary>`, such as a `<figure>` or a `<theorem>`. Doing so would disrupt consecutive numbering in different versions, with or without, `<commentary>` included. Numbered equations are not prohibited in the schema, but should definitely be avoided anyway.

8.3 Adding or Removing Divisions

For an Instructor's Version you might wish to add additional material into the front matter (a specialized `<preface>` perhaps), or remove some material from the back matter (an `<appendix>` with solutions that duplicates solutions now placed within the exercises themselves). There may also be parts of each chapter you do not find necessary to include.

Modularizing your source files would allow for a different "master" XML source file to include different portions of the `<frontmatter>` or `<backmatter>`, perhaps just making a different title page. See the Author's Guide for more on modularization.

Additional, minimal, XSLT stylesheets can be used to selectively "kill" portions of your source, such as every "Additional Reading" at the end of each `<chapter>` residing in a `<references>`. Consistent use of elements, leading strings in `@xml:id`, and/or leading strings in `<title>`, can make it a single-line exercise to selectively remove multiple portions of your source without removing other portions. see the Author's Guide for more about additional XSLT stylesheets.

Think carefully about the effect of removals and additions on numbering. In HTML output all numbering is hard-coded and will be based on counts of the entire XML source file. So selectively killing content will not change numbering, but cross-references may point to divisions for which there is no content to serve as the target. Using a different "master" file can impact numbering throughout. Significant portions of the LaTeX output rely on LaTeX's automatic numbering via mechanisms like `\label{}` and `\ref{}`. So if portions of the text are killed, then the `\label{}` of a cross-reference may never be defined. A technical solution would be to provide an option to hard-code all numbering in LaTeX output.

Generally, removing portions of each division will have the least ill-effects on numbering if the portions removed are at the end of a division and no cross-references point there. So, for example, a `<references>` at the end of each `<chapter>` can be safely killed with no ill-effects if there are no cross-references elsewhere to the particular `<biblio>` contained in that `<references>`.

When a division is killed through the use of additional XSLT, knowls and index entries will still be generated as usual for that division as part of the conversion to HTML. Thus, some care may need to be taken if certain knowls should not be uploaded to a server. Using a consistent scheme for the values of the `@xml:id` might make this easy to script. The `<idx>` elements could be killed in a manner similar to the division with a use of the `ancestor` axis in a filter. Of course, the conversion to LaTeX will not create knowls, and the index-creation process does not suffer from the shortcomings of the creation process for HTML.

8.4 Instructor's Notes

The `<commentary>` element, as described above, can be used effectively by an individual instructor to customize a personal version of a book. This is not as fine-grained as highlights or annotations, so is not meant as a replacement for tools that support more localized personal additions.

The ideal way to do this would be with a text having source distributed as a git repository, and with notes managed by git. Here is a rough outline, assuming a solid understanding of git.

1. Clone the author's repository to a local, personal, location.

2. Make a long-lived `notes` branch off the author's `master` branch.

3. Add `<commentary>` with commits on the `notes` branch.

4. Regularly pull `master` from the original repository to receive updates and fixes from the author.

5. Regularly merge `master` into `notes` so the enhanced version gets the author's changes without changing `master`.

6. Produce personalized output from the `notes` branch via PreTeXt as normal, with the switch enabling display of the `<commentary>`.

Chapter 9

Ancillaries

Similar to an Instructor's Version, for a textbook, or other work, a publisher might wish to provide ancillary documents with additional, or repackaged, material.

9.1 Solution Manual

An author may include a `<hint>`, `<answer>`, and/or `<solution>` as part of each `<exercise>` or `<project>`. Some of these may be designed for the reader, while some may designed for the instructor. A separate conversion is available to make a PDF containing just these items.

In the usual way, execute

```
xsltproc pretext-solution-manual-latex.xsl fauna.xml
```

with suitable paths in front of the stylesheet and the source file. The result will be a file like `fauna.tex`, which can be processed with a LaTeX engine such as `pdflatex`. By default, the result will include *all* `<statement>`, `<hint>`, `<answer>`, and `<solution>` for *every* `<exercise>` or `<project>` anywhere in your `<book>`. Note that this conversion is explicitly designed only for `<book>`, so send a request for support for an `<article>`. Division headings (`<chapter>`, `<section>`, ...) will be present, if and only if they have content. Page headers will help locate chapters and sections. Exercise numbers will be complete, to make it easier to locate individual problems. In other words, Exercise 3 from Exercises 5.6 will be labeled as 5.6.3, not just 3, as in the original text.

Any cross-reference that exists inside a solution will be honored and displayed faithfully. By that, we mean that if the author includes an `<xref>` as part of a solution to, say, Theorem 10.6, then that cross-reference will be rendered visibly as “Theorem 10.6” in the solution manual. However, it will not be live (clickable) since the target (the `<theorem>` itself) is not part of the solution manual. We have not, and do not expect to, determine if a cross-reference points to part of another exercise which is visible in the remainder of the solution manual and then elect to make it live/clickable. In other words, all cross-references are static, even if there is the possibility to be more dynamic for a select few. If this is a severe shortcoming, consider producing an Instructor's Version (Chapter 8), enhanced with additional solutions, where all cross-references are live, and targets are more likely to be available.

A set of switches allows a publisher to control including `<statement>`, `<hint>`, `<answer>`, or `<solution>`–for inline exercises, divisional exercises, worksheet exercises, reading questions, projects, and tasks within a project. So there is a total of $4 \times 5 = 20$ yes/no switches, for $2^{20} = 1\,048\,576$ supported scenarios. Find these switches documented as comments early on in the stylesheet. The LaTeX preamble is the same as for the full document, so besides being excessive, it should support any of the LaTeX styling options. You will likely want to place these switches, and any customizations, into a thin XSL file (read about customizations in the Author's Guide).

In practice, you will discover that the conversion will reproduce all of your `<frontmatter>` and `<backmatter>` *exactly* as if it was part of the entire text itself. The reason for this is that you may actually want a little bit of front matter, perhaps some back matter, and maybe a new title page that makes it clear that you have created a solution manual. Best practice would be to have each of your chapters in its own file, and incorporated into your document via a master file using the `<xi:include>` mechanism. (Read about modularity in the Author's Guide if this is new to you.) So make a new "master" file for the solution manual, maybe with a new title, a new preface, and acknowledgements of any help creating exercises and solutions. Be sure to include *all* of the main matter, even if you know some parts may not have any `<exercise>`. You want numbering to be correct for your cross-references and this means having all the content available to be counted, even if it is not visible in the end product. This new master file is really the only overhead involved in getting a quality, reliable solution manual together.

The philosophy behind this conversion is that a publisher may wish to create a *different* range of solutions for instructors, for limited distribution in ways that students are unlikely to find. Thus, we have tried to produce a *functional* document by default, without too much attention to making a *beautiful* document. Of course, improvements and suggestions are always welcome, though here the priority will be ease-of-use.

Digression. This conversion illustrates some advantages of including all the content of your project in one source document, and then selecting a subset of that content for different audiences. The advantage of authoring `<hint>`, `<answer>`, and `<solution>` in close proximity to the `<statement>` should mean a higher probability that changes to one part of an exercise will be reflected in the other parts. And with standard processing tools, and provided switches, an author and publisher can easily decide which parts to show, and when.

By including all of your project's content in one monolithic source document, it is possible to confidently reference supporting parts of the main text via cross-references from solutions. When a new edition is released, any variations in numbering will automatically be reflected in a new solution manual, created with no additional editing or proofreading.

9.2 Private Solutions

Suppose an author distributes a textbook with an open license, and so makes the PreTeXt source available publicly (perhaps as a condition of the license). Perhaps the author also intends `<hint>` provided with `<exercise>` to assist students, and having them available as knowls in HTML output is a great way to make them easily available, but not immediately visible. But the author has also written some, or many, `<solution>` for the `<exercise>`, but these are only meant for instructors, and not for students. See the discussion at Section 8.1 for more background.

One approach is to distribute an Instructor Version or Solution Manual only on request, and only as a PDF. The ability to provide a watermark on every page (via switches in the LaTeX conversion) allows you to include a personalized message such as

```
Issued to Charles Darwin.  Do Not Copy.
```

It would be a trivial technical exercise to remove this, but perhaps the moral imperative (in an extra `<preface>` as well?) would dissuade most from distributing further?

But what about publicly available source code? After several unsatisfactory experiments, we have arrived at the following *solution*. Again it involves an intermediate understanding of the revision control software, git. And again, this is an outline.

- Create a private repository for authors, and other trusted contributors. In other words, if shared, read access is controlled via passwords or something similar.
- Create a branch off of `master` called `solutions`.
- Do all editing of private material, and only editing of private material, as commits to this branch. So a typical commit might just be `<solution>` elements inside existing `<exercise>`. Any script or master file for producing a solution manual might also be part of this branch.
- Do all authoring in this private repository, mostly as commits on `master`.

- Periodically, while the `solutions` branch is checked out, merge `master` to bring in new changes to the main content.

- Never ever merge `solutions` into `master`. In other words, `solutions` is a long-lived branch which never dies and is never merged into another branch. (Never rebase this branch if you have collaborators sharing the private repository.)

- Push and pull both `master` and `solutions` to and from the private repository by setting up tracking branches.

- Create a public repository which is a strict duplicate of the `master` branch. Periodically push the `master` branch of the private repository to the `master` branch of the public repository. Only. Its only purpose is for the next item. Use commands or a setup which makes it impossible to accidentally push `solutions` to this public repository.

- The commits in the public repository will be *identical* to those on `master` in the private repository. So anyone can clone or fork this repository and make pull requests, which authors can apply and mange via the private repository. But solutions will never be part of the interaction with this repository.

Chapter 10

Hosting Your Online Version

You have HTML output, and now where do you put it? A fundamental design decision is that you only need to simply upload your HTML files to a hosting service and since all the links are relative, readers should be able to read your whole book with no more effort than that from you. By design, no extraordinary configuration or privileges are necessary on the server.

For the choice of a **hosting service** you may have a fundamental decision to make. Mostly this applies to authors who are employees of an institution, yet have the freedom to control the copyright on their scholarly work. But there is information here for independent scholars and for other employees.

- You love your institution, and plan to stay for a good long time. They have implicitly (or explicitly) supported your project with time and/or money. A URL with the insitution's domain name on a freely-accessible project is good advertising for the institution. Bandwidth is huge, IT is super reliable and helpful, all this is no-cost to you. Read the next scenario, but you have a good situation, so you might as well use it.

- You are not really attached to your institution, and five years from now you may be somewhere else. Consider hosting your project externally, so it is not tied to your institution.

  Or maybe policy on faculty web pages, or crummy content management systems, make it difficult or impossible to host your project. Or it is buried five levels deep with an impossible URL. Point out the situation to your Provost or Dean, with examples of how *other institutions* do it right. Remember that your colleagues may be writing monographs and textbooks for commercial publishers, likely with institutional support, and selling their copyright. Your institution should be *proud* to host your project prominently. If a reasoned, rational approach does not improve the situation, then consider hosting your work elsewhere.

If you are hosting at your institution, that is a great outcome. There is no cost to you, and everybody is happy. Lobby for a great URL, like `platypus.mammal-institute.org` and the rest should take care of itself. The rest of this section is about the second situation.

To arrange hosting yourself,

1. Purchase a domain name, it should not be a real big annual expense. Choose something professional, rather than just your name (though your name does have a natural appeal). And maybe something general enough that you can host your next book under that same domain name. The idea here is to *own* the domain name, so your book can move anywhere, but that domain name will always point to the book. This name should be *owned and controlled by you*, not your institution, not GitHub, not `5GBFree.com`.

2. Sign up for, and perhaps pay for, a hosting service that lets you point your domain name at the site.

   - Oscar Levin explains that GitHub Pages (`pages.github.com`) is free, super-easy to use if you already use `git`, and makes using your domain name ("custom URL") nearly trivial. (2017-09-08)

- Mitch Keller likes the “Swift” plan at A2 Hosting (`a2hosting.com`) at about $60 annually. (2017-07-05)

Now you are set, and control distribution of your scholarly publication. If you are bothered by the thought of having expenses while you make your work freely available to the world, then consider generating some modest income. For example, sell Google ads against your pages. (Why should *this* disturb anybody? I don’t get it.) Or roll a small royalty into the print-on-demand version, see Chapter 13.

Chapter 11

LaTeX Styles

Print and PDF output is created when PreTeXt outputs a LaTeX file, which can subsequently be easily converted to PDF with engines like `pdflatex` and `xelatex`. There are many ways to influence the style (look, appearance) of this output, enhancing the content, but without changing it. This chapter is directed at publishers who desire to create an attractive and consistent design for their books or articles.

Please read this chapter thorougly once before embarking. Certain important points are made in certain contexts, yet are universally applicable. If you cherry-pick, you will miss them. Note also that some simple changes, like font size, are described in Chapter 5.

11.1 Preparation

We do not encourage authors to make small adjustments in style, especially if they have few skills in book design. Instead, they should choose a design built by others that will fit their needs and desires. We do encourage publishers with design skills to create complete and harmonious designs, and to donate these back to PreTeXt with an open license, for use by all authors. This chapter assumes you are such a publisher. Further, it assumes you have certain technical skills. Specifically

- Good familiarity with basic TeX and LaTeX.
- The ability to debug LaTeX compilations gone bad.
- Willingness to study several LaTeX packages that may be new to you.
- Willingess to mimic and experiment with basic eXtensible Stylsheet Language (XSL).

Fortunately, it is easy to start small, get good results, and expand your skills further.

Begin by creating a file that is a new XSL stylesheet. You can likely safely copy a mature one from the `xsl/latex` directory. Be certain to keep the first few declarations. The `<xsl:import>` is critical, since it will "pull in" all the basic code for the PreTeXt conversion to LaTeX. You will be overriding and appending to that code (which PreTeXt has made straightforward). You can start with an absolute path from your filesystem root, but once public a relative path will be necessary. Remove all of the `<xsl:template>` elements, leaving a hollow shell to begin working with.

What we are doing here is similar to the discussion of "thin XSL stylesheets" in the Author's Guide, only thicker. String parameters are also described in this guide, at Section 3.1.

11.2 Overview

Some changes in style are effected by setting string parameters that exist for use at the command line. However, the more flexible features come from the selection by PreTeXt of certain LaTeX packages. These have been chosen for their flexibility, maturity, and stability. They should be part of a full LaTeX installation, especially one based on TeXLive. We presume each author has a similar installation. Please let us know of any exceptions. Please try to avoid requiring new packages as part of your style, and if necessary, be sure they are mainstream ones. Start a discussion on the development forum if you think it is warranted or necessary. It may be difficult and error-prone for you to employ and integrate an obscure package, and it will cause problems for authors who want to use your style.

This is an incomplete list of the primary packages we employ, and their general purpose. They, and their documentation, can be easily found at the Comprehensive TEX Archive Network, aka CTAN.

`geometry`	Specification of the sizes of paper, margins, headers and footers
`titleps`	Headers and footers (part of `titlesec`)
`titlesec`	Titles of divisions
`tcolorbox`	Boxes, colors, etc. for `<example>`, `<remark>`, etc.

Table 11.2.1: Principal LATEX packages used for styles

11.3 Page Shape

Various dimensions of a printed page, including the page itself, may be adjusted using the `geometry` package, so study the documentation of this package to explore possibilities. The options of this package may be also set on the command line. To make options part of your style, place the `<xsl:param>` declaration in your stylesheet as follows (note the two sets of quotes):

```
<xsl:param name="latex.geometry" select="'foo'"/>
```

This will have the effect of placing the following line into the preamble of the resulting LATEX output file, in the right place:

```
\geometry{foo}
```

Of course, you will want to use something meaningful, such as

```
<xsl:param name="latex.geometry" select="'a4paper,total={16cm,25cm}'"/>
```

to produce in the LATEX output

```
\geometry{a4paper,total={16cm,25cm}}
```

This is typical and illustrates two important universal points. First, PreTeXt puts surrounding infrastructure in place. In this case the `geometry` package is loaded, and in an order that does not cause conflicts, plus the `\geometry{}` command itself is placed and output by PreTeXt. As a style writer, you simply provide the package options you desire. Second, “garbage in, garbage out.” It is very easy to make a typo in your style, and have the LATEX compilation fail. This is why we assume you are comfortable with LATEX compilation and debugging, and not every author should be a style writer.

Set the document font size first, for this will influence later choices. The string parameter, `latex.font.size` (which has `pt` as part of the value) will set an optimal line width. This line width should translate to about 75 characters per line, at the upper end of recommendations for an optimal width. It will also match closely (but not exactly) to line lengths in HTML output. You can look into teh LATEX output before you experiment to see the value used when the stock `\geometry{}` command is issued. You are certainly able to override this width, but read the next paragraph carefully.

This raises two important more universal points. You should expect to repeatedly examine the LATEX output as you develop a style. And most important—are you tempted to use a small font, and *increase* the line width so as to cram more material onto the page, so your book is shorter, and sells via print-on-demand for $7 rather than $8? Then you misunderstood that there are time-tested recommendations for the optimal number of characters per line for human readers, and we just counseled you that PreTeXt is *already at the high end of these recommendations.* You have a certain freedom as a style writer. Use it responsibly. Enhance the content provided by authors, don’t degrade it.

This is the place to think about headers and footers on the page, since you will want to make room for them, and with spacing away from the primary content. See Section 11.4.

11.4 Headers and Footers

The `titleps` package cooperates with "traditional" LaTeX divisions, such as `\chapter` and `\section`, and the `titlesec` package, to pick up the titles of divisions automatically and migrate them to headers and footers on a page. The `ps` is short for "page style", and the documentation is a PDF file *within* the distribution for the `titlesec` package. Primarily, we let LaTeX manage the selection of its page styles: `empty`, `plain`, and `headings`. (The LaTeX `myheadings` page style is ignored.) PreTeXt does some management in the front matter. As a style writer it is not your concern where these styles are employed, but you do influence what information they contain and were it is placed on a page.

Add an `<xsl:template>` to your stylesheet that begins with

```
<xsl:template match="article" mode="titleps-plain">
```

This would then set a style for a `plain` page within an `<article>`. Perhaps just a page number centered in the footer? The text produced by this template (see Section 11.7) will then be placed immediately after a `titlesec` command in the preamble to renew the page style. More precisely, immediately after

```
\renewpagestyle{plain}
```

Note that this means the first and last characters of the text produced are a TEX group ({,}).

The definition of this template will override (replace) the definition given in any imported stylesheet. You can replace the value of the `@match` attribute with `book` to make your style apply to a PreTeXt `<book>`. If your style will be used for both books and articles, and you want the `plain` style to be identical for both, you can expand the `@match` attribute to have the value `article|book`. To have different `plain` styles for a book versus an article, make two separate templates.

The `titleps` package allows at least twelve options per page style: even-numbered page versus odd-numbered page with two-sided printing; left, center, right; header or footer. There are semi-automatic customizable rules, variable widths allowing hanging styles, choices of marks (division at page-start versus division at page-end versus new-division-mid-page, including combinations at the same time), and more. Note that the LaTeX system of `\markleft` and `\markboth` has been abandonded. (Did I hear you say, "Good riddance!"?)

2019-09-30: we have not tested one-sided versus two-sided very rigoursly. Please report problems.

11.5 Titles of Divisions

The `titlesec` package cooperates with "traditional" LaTeX divisions, such as `\chapter` and `\section`, to style the start of each division, containing its PreTeXt `<title>` and in most instances, its number. If a division is credited to (multiple) `<author>`, then that information can be styled, and there are plans (2018-09-30) for epigraphs.

PreTeXt manages numbered versus unnumbered divisions, the correct level for one-off divisions like a `<preface>` or `<appendix>`, and the specialized divisions such as `<exercises>` and `<references>`. A style writer creates two styles at each level of the hierarchy, for a numbered variant, and an unnumbered variant. `titlesec` uses a `numberless` key to indicate the latter. A named template, such as

```
<xsl:template name="titlesec-section-style">
```

would produce text containing complete `titlesec` commands `\titleformat` and `\titlespacing` in both numbered and unnumbered variants.

There are five LaTeX macros created by PreTeXt at the start of each division. For example, `\authorsptx` is a comma-separated list of the content of all the `<author>` elements for the division, in the order given. Look in the LaTeXoutput to find the others nearby. A robust style will include this information, even if the first use of the style may not have any divisions credited to others.

The table of contents and the index are created by a single LaTeX macro. This creates a small technical challenge, since PreTeXt never has a chance to write the contents of the heading and must take what it is given. The upshot is that the `\titleptx` macro will be wrong or empty. So instead, use the `titlesec` device of using the macro parameter `#1` for the title in the correct argument of `\titleformat`. We have enabled this possibility through the package's explicit option. We believe the numberless variant of a <chapter> of a <book>, and the numberless variant of a <section> of an <article>, are the only places this is necessary. 2019-09-30: we will contemplate if this should be the rule and the \titleptx macro will go away. Advise if you see a good answer, either way.

2019-09-30: these templates are highly likely to break into two modes (format and spacing), with a @match that can react to <chapter>, <section>, etc., perhaps differently for books versus articles. The change will only imply some minor editing to achieve the same end result, so don't hold back waiting.

11.6 Environments and Blocks

Objects like <example> and <remark> are almost invariably children of a division, numbered, and work best with a <title>. tcolorbox is a massive package, that we have taken to as a solution to many under-the-bonnet technical problems, such as a hanging indent for numbers of <exercise> or laying out the panels of a <sidebyside>. But it also allows an incredible variety of styling options for these intermediate chunks of text. Think of variable placements of numbers and titles, borders and boxes, and colored backgrounds. With freedom comes responsibility! With a light touch, you can *help* your reader navigate the inherent structure of your PreTeXt source.

Whatever you call them: environments, blocks, or information objects, almost every one can be styled separately (2019-09-30: not captioned items yet, such as <figure>). For example,

```
<xsl:template match="example" mode="tcb-style">
    <xsl:text>colback=pink,</xsl:text>
</xsl:template>
```

would cause the background of every <example> to be light red in color. To make every PreTeXt element that is a variant of an <example> look identical, use one of the **entities** defined in xsl/entities.ent,

```
<xsl:template match="&EXAMPLE-LIKE;" mode="tcb-style">
    <xsl:text>colback=pink,</xsl:text>
</xsl:template>
```

. This would affect <example>, <question>, and <problem>, and future-proof your style when there is a demand for <illustration> as a new kind of "example-like."

There is only one mode, but it can handle a variety of PreTeXt elements in the @match. The text produced by the template will be supplied in a named tcolorbox style via a \tcbset{} command. It is marginally more complicated than that. PreTeXt will manage certain aspects of creating a tcolorbox, such as forming the tcolorbox title to be a string like "Example 4.5 The Chain Rule.", or just "Example 4.6" when an author does not include a PreTeXt <title>. As a style writer, you can change the font, color and placement of that string, but not the use of the type-name, the number, the title, or their order. The text of your template is additive, meaning it is in addition to what the stock PreTeXt conversion provides. Your options come last, so will be new and effective, or will replace what the base conversion to LaTeX does. So in the example above, the base conversion to LaTeX has colback=white (rather than the default gray!), and this will be overridden since pink will come later.

11.7 XSL 101

If you have read this far, and read carefully, you have been exposed to several key principles of writing XSL. Basically you are creating templates which the base LaTeX conversion will "call" in exactly the right place. In a procedural language these might be called **hooks**. Besides a smooth integration with the rest of PreTeXt there are several advantages:

- A template using a @match can apply narrowly or broadly. Witness the example above for <example> versus &EXAMPLE-LIKE;. This attribute is similar to a this pointer in an object-oriented language and the modal template is not dissimilar to a method.

- PreTeXt has a discovery phase when it constructs the LATEX preamble. If your PreTeXt source has no `<example>` in it, then there will be no associated `tcolorbox` style added to the preamble, and the LATEX `example` environment will not be defined. So the preamble is exactly what your document needs, and no more (mostly). The `@match` attribute makes this possible.
- You have seen named templates, which will feel familiar if you know procedural languages. While perhaps comfortable, they are way less powerful, and we noted that we may get rid of them.
- The `<xsl:import>` mechanism allows us to keep base definitions and override others. So as you develop your style, you do not need to start from scratch.

Be aware that *every single character* that you put inside the `<xsl:text>` element will get copied literally into the preamble of your LATEX output, including newlines and spaces you use to indent in your XSL. Conversely, any *whitespace* inside your template, but *between* the `<xsl:text>` elements is ignored. So we like to use multiple `<xsl:text>` elements (except we were a bit lazy getting this out the door initially—do as we say, not as we do) and explicitly create newlines with the `&#xa;` character. Your goal is to have a LATEX preamble with no blank lines and no unintended indentation. We often create comment lines (with two leading `%` characters to aid with readability.

See the PreTeXt website for recommendations for books on XSL if you are encouraged to learn more.

11.8 Testing and Debugging

The sample article tries to have one of everything, plus a few torture tests. It is a good place to test initially, especially with the `tcb-style` templates. The sample book is less haphazard, but does have most of the structure a typical book would have. So if you are designing for a book it is a good place to test page styles, headers and footers, and division titles. (2019-09-30: it is possible `<part>` has not been tested thorougly enough yet.)

When things go bad, such as a non-obvious LATEX compilation, it can help to stop working with PreTeXt source, and instead edit the generated LATEX until the problem is understood.

We do not expect to provide great support for this process. First, because new code and basic support already keeps us busy, and second, because you have the freedom to really make a mess, and thus you should take responsibility for the problem. When you are certain that PreTeXt has done something wrong or inadvisable, please, please do post in the development forum with a careful explanation and a (minimal) example. It will happen. Just don't use the forums as a replacement for this documentation or a bit of sleuthing through the LATEX that *you* are now creating. With freedom comes responsibilty. Thanks.

11.9 Planned Additions

Some items to include, as of 2019-09-30:

- Color schemes, so an author can use a style and just change the colors with a very simple stylesheet.
- Font control, so a style writer has simple instructions that navigate LATEX's procedures.
- Flexibility with the building blocks of a title page.
- Cover design and placement.

Chapter 12

(*) Cover Design

Notes: Rationale (promotion). Procedures for print-on-demand (generally). Tools (Illustrator, GIMP, Inkscape). ISBN placement. Capable students can do design for you.

Covers can be modified for use in an electronic PDF produced from the LaTeX conversion, see Paragraphs .

Chapter 13

Print-On-Demand

If you are both author and publisher, you may wish to make your book available in a physical form, but may be reluctant to purchase and store thousands of copies, or to take orders and arrange shipments. Then **print-on-demand** might be the solution for you.

A print-on-demand service is a manufacturer and distributor of printed books, which are typically only printed once ordered, or in extremely small quantities. They can provide many of the manufacturing and fulfillment services a traditional supplies. Some provide services you pay for that will produce a cover, provide editorial services, or assist with marketing.

We list three such services below, but first describe some commonalities, pro and con.

Updates Generally, you provide a PDF of your text, and we have tried, with the `latex.print` option, to make output that is amenable to this situation. A real advantage of print-on-demand is that you can usually update this PDF at any time, without much trouble. You will need to decide how to indicate versions (or printings?) of your work. Perhaps we will have tools and advice about this soon.

Covers You may need to provide a cover, typically as a PDF meeting some exact specifications. Though you may be able to choose a fairly generic look through a template or wizard. Or pay to have one created for you. See also Paragraphs and Chapter 12.

Price You may choose to sell at your cost, or you may wish to make a profit on each sale. (Note: as copyright-holder you can do this, no matter what license you have chosen, review Chapter 2). A 450-page hardcover book might be sold by a print-on-demand manufacturer to an online bookstore, including some profit for the manufacturer, for $23. If you, as author, want $5 profit, and the online bookstore wants $7 for fulfillment, shipping, and profit, the cost to your reader is now $35. In order for the online bookstore to give the appearance of discounting your book to $35, you may need to declare a suggested retail price of $49.95. So pricing takes a bit of thought. Or guesswork, since the discounting algorithm is not public.

Note in the above scenario, the print-on-demand manufacturer may sell you, the publisher, small quantities at a better price, such as ten copies for $170, shipping included.

ISBN An **International Standard Book Number** is a unique identifier of books and necessary for others to distibute and sell your book. See details for each manufacturer below. Much like a domain name for your book's website (see Chapter 10), this may be something you wish to control and own, foregoing the convenience of someody else providing and owning it for you.

In order of increasing professionalism and decreasing convenience, we describe three print-on-demand manufacturers we are familiar with, plus two others. Additions, corrections, updates, and alternatives are all welcome.

Lulu.com This site caters to people making photo books for relatives, in addition to more serious projects. Account setup may be trivial, an ISBN number may not even be needed, and you may have options for distribution beyond readers simply ordering direct from the site. This might be a good choice for drafts you will use in your own classes, if having your university bookstore print copies is not a good alternative. (2017-11-25)

Create Space This company is owned by Amazon.com. They manufacture and distribute serious books, in addition to music and film. Distribution through Amazon is nearly automatic. There is also "Expanded Distribution", which starts to look more like Ingram (next). (2017-11-25)

CreateSpace attempts to make sure you have the rights to your content. So if they find your book freely available on the Internet, their "Content Validation Request Team" becomes suspicious and investigates. This has caused a few authors a few headaches and delays in making their book available for sale, though all have been successful eventually. (2018-03-06)

Ingram Spark IngramSpark (formerly Lightning Source) is a division of Ingram, which is a very large printer, also providing services to major publishers. Creating an account is not trivial, and you need to provide your own ISBN number. In return, your book is available at Amazon.com and many other online bookstores automatically, and is in many ways indistinguishable from offerings of large commercial publishers. There are also options for international distribution. You can also control settings for discounts and returns. (2018-03-06)

Blurb Blurb specializes in photo books, and uses Ingram for printing. Sizes are limited, and costs are more that the other services. Direct experiences would be a welcome addition. (2018-03-06)

Nook Press Nook Press is a service of Barnes & Noble, and books appear only through their online store. We have no additional information, so direct experiences would be a welcome addition. (2018-03-06)

We currently have no good information about distributing EPUB or Kindle electronic versions for profit. (2017-11-25).

Appendix A

FAQ: Frequently Asked Questions

This is a list of answers to frequent questions, in no particular order.

Why does the conversion to HTML use a fixed width for the text? There is an optimal number of characters per line for human readers, based on research and centuries of book design. So we set a fixed width such that the default font comes close to achieving this optimal value. We also use responsive design to accomodate the constraints of a small screen as best as possible. A reader will not want to have to carefully resize a browser window to achieve the optimal width, nor should a line of text spread to many, many characters across a very expansive screen. See Principle 4.

I do not want my examples in knowls. You can change that! See Section 4.1.

Why are my knowls empty? When viewing the HTML version on your laptop or local computer as files, do not expect knowls to render properly. This is a known bug/feature, and there is nothing to be done about it, unless you run a web server on your own machine, which fortunately is a very easy thing to do (see the section of the Author's Guide about testing HTML output locally). Think of it this way: the knowl content comes from a server, but on your laptop there is no web server. You are just looking at files.

A possible added confusion is that some knowls, such as proofs, will appear. That is because their content is embedded in the page, not taken from a file.

How can I change the colors in the HTML version? There are several colors schemes available. They (currently) have a name in the form `mathbook-X.css`, where $X \in \{0, 1, 2, 3, 4, 5, 6, 7, \text{ups}, \text{manitoba}, \text{bucknell}\}$.

The options for colors are currently being revised, and soon it will be easier to create your own color scheme.

To use one of the above color schemes, you can use a `stringparam` on the command line, like this:

```
--stringparam html.css.file "mathbook-X.css"
```

where X is chosen from the list above. Or in your thin XSL stylesheet you can add the line

```
<xsl:param name="html.css.file" select="'mathbook-X.css'" />
```

Something looks wrong in the HTML output. How can I customize the layout of the HTML version? If there are some anomalies in the HTML version of your book, probably that was just an oversight and can be fixed easily. Send a message to `pretext-support@googlegroups.com` describing the problem and *including a live link to the page showing the error.* Do not make a minimal example. (CSS issues are handled in a completely different way than other software issues.)

The long-term plan is to have a variety of different layout options, which can be chosen as easily as choosing a color scheme. See Principle 7. The first step is to rewrite the current CSS so that it is easy to develop alternate layouts. That should be done by the end of Spring, 2018. Then people can develop new styles! Until that happens, either suffer with the current style, or hack away at your own peril. (Note that the PreTeXt support groups will not provide any help with hacking the layout, but a lot of help will be available when it is time to develop alternate layouts.)

Why does the HTML output load so many external resources? The subtext perhaps being, "Why shouldn't I host these on my own server?" A main goal for PreTeXt is to spare authors the headaches of learning new technologies just so they can get their content in front of readers. That knowledge should be built into software, so an author can work at a higher level, explaining the intricacies of their discipline. So we only assume an author can place locally-built HTML output onto some public server they have permission to use. Any extra enabling techology we do not want to create ourselves gets pulled from other public servers. MathJax, both code and fonts, is a good example, as one of the enabling projects. Perhaps it is *the* enabling project.

This way,

- Authors can concentrate on their writing, not updating services on their server.
- Servers that are hostile to ad-hoc configurations (think "learning management systems") are not an impediment to hosting projects.
- For the most part, updates to external resources happen automatically. This allows authors and PreTeXt developers to concentrate on other aspects of their work.

We get MathJax from a **content delivery network** (CDN). Once we have that dependency, then fonts and search from Google, CSS and Javascript from the American Institute of Mathematics, and other components, all have the same dependency: a decent internet connection. Our experience over several years is that these resources have good uptimes and good bandwidth, and so are not a source of problems. A good offline version, with resources packaged via a script, would be a good long-term project.

Finally, we do not load minor resources indiscriminately. Something in your source should suggest they are necessary and we perform those checks, document-wide. However, since a cross-reference is usually implemented as a knowl, and we cannot be sure what a knowl might contain, we do tend to load resources on *every* page, even if only needed once. We hope to improve this situation. And you are enouraged to help if you have technical skills in these areas.

Searching my PDF output is broken. PreTeXt goes to great lengths to make a high-quality PDF, but if you manipulate it by adding in new pages, or adjust the intermediate LaTeX to use other fonts, you run the risk of breaking some of the features.

A ligature is a combination of two characters into one, like a lower-case "f" followed closely by a lower-case "i" without a dot. These can confuse a search. Verbatim text sometimes ends up with "smart" quotes, where left and right versions are inverted. This frustrates copying source code into an actual program. And so on. If you see problems like this with un-customized PDF output, we would like to hear about it.

Appendix B

PreTeXt Principles

The development of PreTeXt is guided by a list of principles, which appear early in the Author's Guide. We duplicate them here so we can reference them as motivation for various decisions described here.

1. PreTeXt is a markup language that captures the structure of textbooks and research papers.
2. PreTeXt is human-readable and human-writable.
3. PreTeXt documents serve as a single source which can be easily converted to multiple other formats, current and future.
4. PreTeXt respects the good design practices which have been developed over the past centuries.
5. PreTeXt makes it easy for authors to implement features which are both common and reasonable.
6. PreTeXt supports online documents which make use of the full capabilities of the Web.
7. PreTeXt output is styled by selecting from a list of available templates, relieving the author of the burden involved in micromanaging the output format.
8. PreTeXt is free: the software is available at no cost, with an open license. The use of PreTeXt does not impose any constraints on documents prepared with the system.
9. PreTeXt is not a closed system: documents can be converted to LaTeX and then developed using standard LaTeX tools.
10. PreTeXt recognizes that scholarly documents involve the interaction of authors, publishers, scholars, instructors, students, and readers, with each group having its own needs and goals.

List B.0.1: PreTeXt Principles

Appendix C

GNU Free Documentation License

Version 1.3, 3 November 2008

0. PREAMBLE. The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS. This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover

Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING. You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY. If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS. You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties — for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS. You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS. You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS. A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION. Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION. You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE. The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING. "Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents. To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  YEAR  YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with… Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.