LaTeX to eBook 2021

The Book About Itself

Dominic Widdows

Your Cover Will Look Better Here!

Contents

1 Introduction to LaTeX and eBooks

Considering writing a book? Good for you! Publishing a book is easier and cheaper today than it ever has been.

If you know what you want to write about, the next step (and sometimes the hardest step) is getting started. The most important part is the material itself: you can write paragraphs in any number of text and document editors, and figure out how to format the material as a book later. But sooner or later, the question arises of how to turn the material into a book format. That could mean a paper book, which requires physical resources for printing and distributing. Or it could mean an eBook, to be read on an eReader such as an Amazon Kindle or Kobo Clara, or using an app on a phone, tablet, or laptop.

So how do you turn your document into an eBook? This is a very short book demonstrating one way to do this, and if you're reading this on an eReader, that shows that it already worked at least once — for this book.

To start with, I chose to use the LaTeX typesetting system. The original TeX was created by the famous computer scientist Donald Knuth (Knuth and Bibby, 1984), and added to by Leslie Lamport to make LaTeX (Lamport, 1985). LaTeX is used throughout scientific fields to write papers and books. There are many reasons for preferring LaTeX to one of the more 'point, click, and type' text editors: it handles figures, tables, chapters, sections, headings, cross-references, bibliographic references, mathematical equations, and many more things that can be notoriously irritating and time consuming to get right. And it's possible to use different typesetting programs and commands to create all sorts of output from LaTeX input, including eBooks.

If TeX and LaTeX are entirely unfamiliar to you, even this automation and flexibility may not make it the best choice for you to write a book, because there is quite a technical learning-curve for LaTeX. It's not just writing text, it's writing commands telling the typesetting program how to 'compile' the output document — in many ways, LaTeX feels much more like programming than writing a Google or Microsoft Word document. But if LaTeX is something you've already used to write a dissertation or paper, you're probably well aware of its benefits (and its hassles).

In summary, if you've used LaTeX to write papers, now you want to write an eBook, and you need to figure out how to do this, then this little book and the open source template that it comes from might be ideal. There are other ways to make eBooks, and other ways to use LaTeX to make eBooks — this book isn't comprehensive, but it might just enable you to make an eBook end-to-end quickly, cheaply, and easily.

1.1 How to Use this Book

This is a book about itself — it's about how it was written using the templates and tools described in the next few chapters. These are the main ways you can use this material:

- You can read it. It shouldn't take long, it outlines the process used to make this eBook end to end, and then you can decide if this is something you want to try.

- You can use it as an instruction manual, learning and following some of the procedures step-by-step to make your own book.

- You can use it as a template. All of the source files used to make this book are freely available in GitHub at `https://github.com/dwiddows/ebookbook` and Overleaf. The source files are laid out in a way that should make it easy to clone the project and adapt it for your own book.

It follows that you could recreate this eBook for yourself, following just the process described in the book: which is basically to clone the GitHub project as a template, build the project, send the output HTML document through an ePub converter, and send this to your eReader device.

So why would anyone buy a book if it's free? Because anyone who reads the steps above and with enough familiarity to think "git clone ... check dependencies ... build.sh ... check dependencies again ... build.sh ...... yeah, alright" will expect it to take more than a few minutes, hopefully less than an hour, and price their own time realistically. If you want to read this book for free on your eReader, compiling from source is the way to go about it. Or if you want to just click 'buy' now and send the author most of the $2.99 price tag, please go ahead, and thank you!

Either way, if you're not put off by LaTeX and `git` commands, keep reading. I hope the book is useful to you, and wish you all the best of luck and persistence writing your book!

2 End to End: From Source to eReader

We'll start with the end-to-end process for making a book you can read on your eReader device. Once you can do this, change the title, add or remove chapters, and see the changes you've made reflected on your eReader, that will hopefully be the best encouragement to take the parts that are most useful and get started with your own book. The basic outline of the process is shown in Figure 2. The rest of this chapter walks through the steps in more detail.

2.1 Install Dependencies

Typically when you want to use a new piece of software, it has dependencies — other libraries and packages it needs for some of its functionality. For end-user applications, these are usually bundled together so that everything works 'out of the box'. For programming tools, there's normally some system to check which dependencies you already have, and to install only the new ones that you need. How this works depends partly on what platform you're using. Here we'll just list the dependencies you'll need to find and install.

These include:

- A LaTeX system, including a program called `make4ht` or the older `htlatex`. This is crucial for making HTML output, as described in Section 2.3.

- Some eBook converter software. The one recommended below is `ebook-convert`, from Calibre (see `https://manual.calibre-eBook.com/ebook-convert.html`, or just search the web).

- Some eReader or previewer software, such as a Kindle device or app, or another eBook app such as Calibre (see above). You'll want this for seeing how your book looks 'for real' (though of course, not all eBook apps look exactly alike, and users have different settings).

- (Recommended) A shell program that runs bash scripts so that you can run the `build.sh` command. If you don't already have a terminal where you can run bash or a close equivalent, it's probably worth installing and using, and if that instruction sounds difficult, the rest of the process may be hard.

Every modern operating system — basically Windows, Linux, MacOS, and similar variants — has a variety of package installation tools. Some of the dependencies above
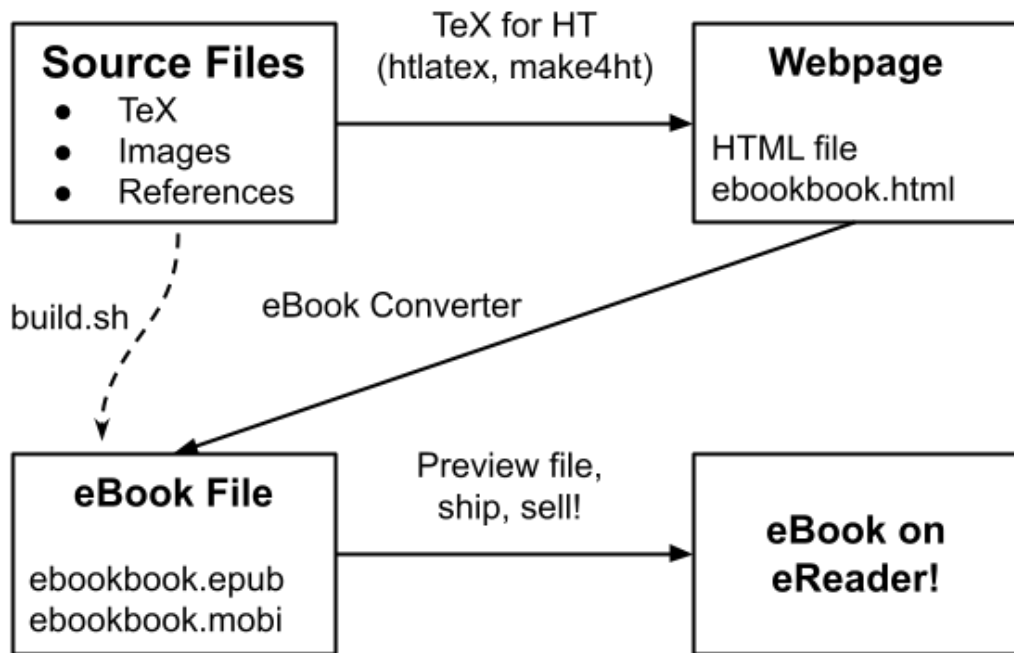
Figure 2.1: Steps in the pipeline for shipping this eBook

can be done at the command-line, sometimes it's easier to download and install something directly from a provider's website.

The best place to keep links to such dependencies and recommended options is the github project wiki at `https://github.com/dwiddows/ebookbook/wiki`.

2.2 Access and Build the LaTeX Source

Next, you'll want to get a copy of the source code for this book. (LaTeX sources count as 'code' for these purposes. It gives instructions to machines, it's easy to make mistakes that show up as error messages, and it's reviewed and stored in source control — so it's like code in these practical senses.)

If you already have `git` installed, this will be something like running `git clone https://github.com/dwiddows/eb` at a command prompt, or downloading the code using some visual client software.

List the contents of this directory and check that you can see the files `ebookbook.tex` and `build.sh`. The first of these is the main TeX document that lays out how to combine the other files into an eBook. The second is a build script — on platforms with bash or a compatible shell installed, you should be able to run just `./build.sh` and get most of the book built using one command. Or at least, to begin with, you should get error messages telling you if anything is missing and needs to be installed. If you're not running a compatible shell, the `build.sh` file at least lists the commands you'll need to run some other way.

So the next step is to run `./build.sh` and ideally it should typeset a copy of this book. If the `ebook-convert` command is available it should even make the eBook files described in the next section.

If this doesn't work, you may need to find and install some `pdflatex` and most importantly `make4ht` or `htlatex` programs that work on your machine. (See the Dependencies section above.)

So long as you can run "`make4ht ebookbook`" and create a file `ebookbook.html`, you can go on to the next step.

2.3 From HTML to ePub ... or Mobi

The most important output from the previous step is a file called `ebookbook.html`. This is formatted for display as a webpage in a browser. This is different from the more common use of TeX to make documents such as academic papers, which are nowadays normally created as PDF files. An HTML file is a collection of content (for example, words and images to display), and typesetting suggestions (for example, make this text a heading, and make this image 40% of the screen width). By contrast, a PDF file has precise instructions about how big to make each character and which page to put it on. So it makes sense that HTML is more like an eBook: instead of saying what text will appear on which page, it gives directives about what text should be bigger and smaller, and this combines with the user's device settings to decide which page it should appear on.

So the `ebookbook.html` file (rather than the corresponding PDF file or any other page-layout format) will be used to create your eBook format.

You can turn your HTML file from a webpage into an eBook by installing and using a converter such as Calibre `ebook-convert` or Amazon's *Kindle Previewer* tool. Like saving an image as a `.jpg`, `.gif` or `.png`, you'll need to select a format to convert to. Options include:

- `.epub` is a cross-platform format that it supposed to be used for any electronic book.

- `.mobi` is an old Amazon Kindle format — and it happens to be the one you can use for e-mailing files to your Kindle.

The `build.sh` script that comes with this book uses `ebook-convert` and creates both `.epub` and `.mobi` files as output. This also takes command-line arguments so that you can specify metadata like the author name and cover image:

```
ebook-convert ebookbook.html ebookbook.epub -cover images/cover.jpg -authors "YOUR
NAME" -language English
```

2.4 Previewing Your Book on an eReader

Hopefully by now you have an eBook. Or at least, a file called something like `ebookbook.epub`. So how do you *read* your book?

For this you'll need an eReader — perhaps an app on your computer or phone, or an eReader device such as a Kindle.

Assuming that most of your writing will be on a computer, that's probably the first place you'll want to see your work. For example, I usually open the Calibre app and load the `.epub` file, or do both at once with the command `calibre ebookbook.epub`. Once the book is imported and loaded, this gives the result shown in Figure 2.2.

To view on an eReader device or phone, you often have to send it to the device or load it on in some way — again, there are a range of methods. For Amazon Kindles, you need to find / set up an email address for the device itself, and send the book to that email address. As an extra complexity, this method only works using the older `.mobi` files. Also Kindle preloads like this do not (at the moment) display the cover page of the book in your library or menu page, because Amazon retrieves this information for *published* books from elsewhere. So the preview in the library view is somewhat disappointing, but it works (Figure 2.3).

Another way is to sign up for a publishing account and start to submit your book for publication. On the Kindle publishing site, for example, this process includes opportunities to preview your book as it would appear on a phone and a Kindle device, before you set up pricing options and get closer to hitting 'publish'. For this book at least, these online previewers gave a more accurate rendering of how the book will look to readers than emailing a `.mobi` file to my Kindle device.
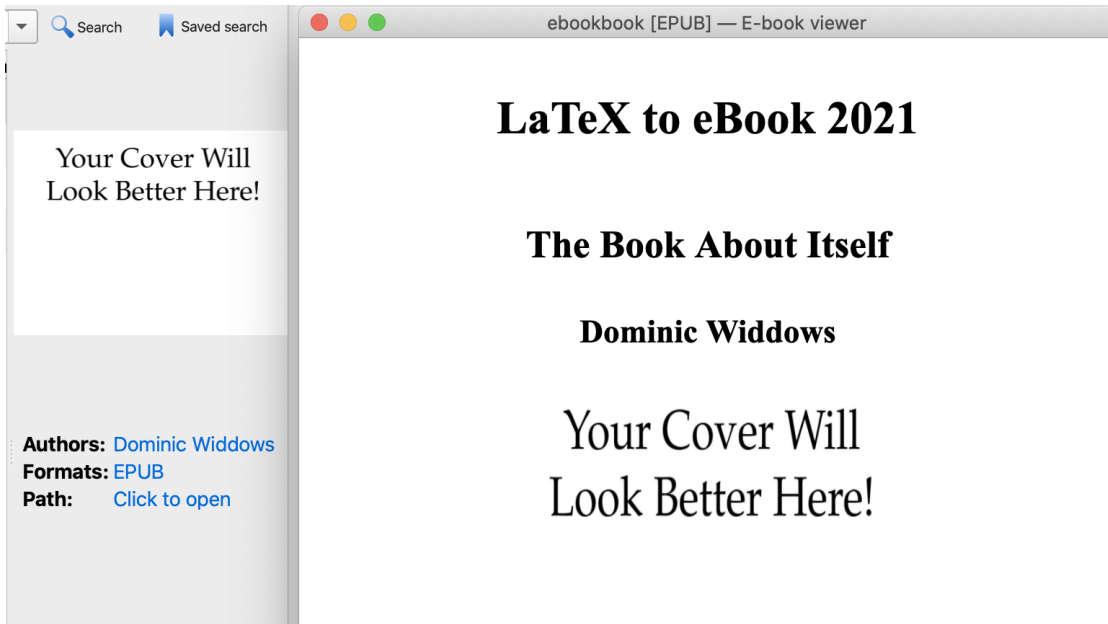
Figure 2.2: Thumbnail image and cover page in Calibre

One way or another, try to make sure you can make an eBook file and view it somewhere.

2.5 Make it Your Own!

Hopefully by now you're at the point where there is an eBook file that you can easily read on some eReader device or app.

The next thing you should do is change something. Change the title in the `cover_page.tex` file from 'LaTeX to eBook 2021' to whatever title you want. Rerun the steps above and hopefully you'll see exactly what you intended: a copy of the eBook, but with your title instead.

At this point you're off to the races. That doesn't mean that it's all plain sailing from here: there will likely be glitches and hurdles along the way. But the main thing is that you have a template, examples of several LaTeX constructs that work with eBooks, and you're able to start turning this into your own book. You may want to save your work separately at this point, call the project something else, and if you use source control, start checking in your work in some way that makes it clear that it's a new project, rather than a work-in-progress on the old project that's intended to be merged back in at some point.

Experiment with removing directives to \input different chapters, and watch the book get shorter. Try changing image files to some other graphic and make sure they change appropriately. And try editing the AUTHOR in the `build.sh` file and make sure the
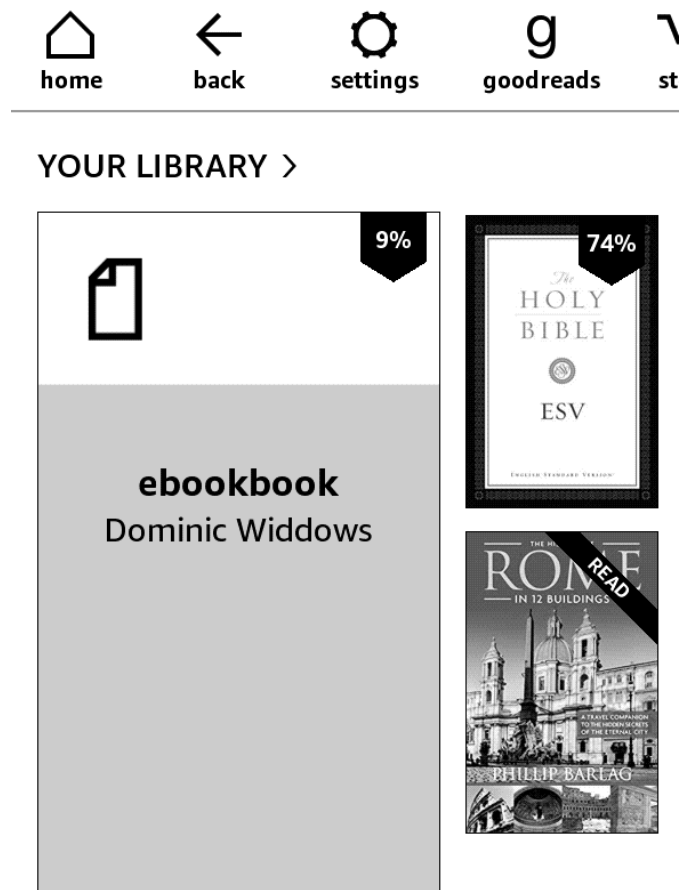
Figure 2.3: Book thumbnail image loaded onto a Kindle

right name shows up when the book is viewed in an eReader.

If this works, you can be pretty confident that your book is on the right track. You'll be able to organize the content into TeX files, image files, etc., and create an eBook!

2.6 Publication

Nearly all of this book is about how to create and preview eBook files, and this is like a software development process — you should be able to run the pipeline over and over again and keep testing that the change you made had the desired effect.

Publishing and marketing your book is a different process: you'll be using an online browser app, signing up for accounts, uploading files, filling in forms, you'll eventually click "submit", and hopefully see your book available for download / purchase after a short while (in the case of this book, a few hours). Look for opportunities to preview your book as part of the submission process.

This process with Amazon Kindle has was straightforward enough form-filling. You can upload your own cover image, or use the online tools to create one. Submission for publication brings up questions like pricing and (related) copyright agreements. To claim a 70% royalty for original work, rather than a 35% royalty, Amazon says you have to price your book between \$2.99 and \$9.99. Hence the \$2.99 price tag for this book.

It takes at least a few hours before your book is live and available — and it may take longer to start appearing in search results.

Note that you can upload new versions of your book any time. Updates (at least minor ones) get automatically processed and released in just the same way as the initial submission. With this book, for example, I submitted two revisions within a few hours, because of course once I was viewing the book 'for real' on my phone, there were a couple of typos and small mistakes I wanted to fix. So in the case of this book, I published it and updated the live version a few times before even telling people it was available.

If you found this book in the Kindle store and can read this as a result, it worked. And if I could do this, you can too!

3 Maintenance and Troubleshooting

We're nearly done with this short template book — at least with the first version. But as with most topics to do with software and electronic information, things will keep changing. Bugs will hopefully get fixed, new bugs will arise, different devices will support different formats, standards may change — for example, it may become possible to load `.epub` files directly onto an Amazon Kindle, rather than having to use the `.mobi` format solely for this purpose. This last chapter will include a few suggestions on how to ask questions and report changes.

3.1 Report Bugs on Github

The open source github repository for this project is `https://github.com/dwiddows/ebookbook/`. If something recommended in this book doesn't work for you, and if you have to change or add extra commands to make it work, please report this as an issue there. The github project wiki can be used for keeping instructions up-to-date and adding links to more resources. This is likely to be much more effective the reporting problems on sites where the book is sold. If you want to review the book itself, post a review. But if you want to report a problem and get help, please use the project github site.

When there are major developments or enough new things to include, I will add these to future editions of this book (if there are any!). But by design, books are meant to be relatively stable, whereas project websites and wikis are designed to have immediate updates. If a new 'edition' comes out too often, we can lose track of what to refer to.

So tl;dr: please leave actual reviews on marketplace sites, but report bugs on github. Not just because I want to avoid a bunch of frustrated bug reports as book reviews, but also because if bugs are reported on the github project, they're more likely to be addressed and fixed!

3.2 That's All For Now

This was always intended to be a short book, just enough to provide a working template, to demonstrate a few typesetting features, and to make the book whose end you're just reaching.

I hope it's given you some confidence to get started, and in particular, I hope the structure works for you. Clone the project, install a few dependencies, try it out, and ideally within a short time you won't be thinking much about this book — it will already be becoming *your* book.

Good luck and enjoy your writing!

Dominic Widdows
September 2021

About the Author

Dominic Widdows is a mathematician, computational linguist, and software engineer. He has worked on differential geometry and Oxford, natural language processing at Stanford, and many projects at MAYA Design, Google, Microsoft, Grab, and LivePerson, where he works particularly on conversational AI and internationalization.

As an author, his work includes the book Geometry and Meaning, and over 50 scientific papers, in areas including pure mathematics, computer science, language processing, bioinformatics, information extraction, logistics, and quantum computing. As a developer, his open source contributions include work on the SkyMap Planetarium App, the SemanticVectors project, and PILMaps for drawing maps.

Enthusiasm for scientific publishing and open source development combined to make this book.

Bibliography

Knuth, D. E. and Bibby, D. (1984). *The TeXbook*, volume 15. Addison-Wesley Reading.

Lamport, L. (1985). *LaTeX: A Document Preparation System*, volume 410. Addison-Wesley Professional; 2nd edition (June 30, 1994).