The HiWi's Guide to the Booklet Builder

maja.lecher@uni-bayreuth.de

Version 4, February 2025

Welcome to the Booklet Builder Guide! It is here to make your life easier.

1 What is LaTeX and why bother using it?

This booklet builder is written in LaTeX. LaTeX is a high-quality typesetting system that is routinely used for scientific documents, as well as almost any form of publishing.

Its power lies not only in its ability to consistently output files in exactly the format you envisioned (no more of that figures-emigrating-to-Hawaii-when-adding-a-single-letter-somewhere!), but also in its capacity to automatise tedious, repetitive tasks like building a species list.

If you're not used to the modular nature of LaTeX, the many output files with obscure endings may look a bit daunting at first — but don't worry, it's easier than it seems and you'll get the hang of it in no time.

If all goes well, you will never have to touch the source code for the booklet builder beyond changing the occasional file path.

2 What do I need on my computer?

1. A **LaTeX editor**
   LaTeX is a markup language, so you could technically write your source code in any text editor, but in order for your code to compile and turn into a shiny PDF document, you need a dedicated LaTeX editor.

   - For smaller files (with, say, 50 different species), it should be sufficient to use an online LaTeX editor like Overleaf (`https://www.overleaf.com/`). Overleaf has the advantage of being free (with some limitations) and collaborative, allowing multiple people to work on the same project at once, and you don't need to install anything. The downside is that once your file exceeds a certain size or you want more than two collaborators to work on the project simultaneously, you would need to upgrade to a paid version. The University of Bayreuth does not, to my knowledge, provide free premium accounts.

   - If the file becomes too large or you want to work on your project locally, you can move to one of the many editors capable of compiling LaTeX code. I personally recommend TeXMaker, which is a dedicated LaTeX editor. TeXMaker is open source and cross-platform, meaning that Windows, Linux, and MaC OS users alike should be able to work with it. If you're already working with programming languages like R or Python and already have a preferred integrated development environment (like Visual Studio or Sublime), you can also use that. This guide will include screenshots of both TeXMaker and Overleaf as examples. In order for your editor to interpret LaTeX code, you also need to install TeX Live. This isn't complicated, but unfortunately takes a while (up to four hours) due to its large size. You can find a list of installation/good instruction links below:
     - Step-by-step instructions (by Uni Regensburg): `https://www.uni-regensburg.de/assets/physik/fakultaet/Studium/LaTeX/Anleitung_Installation_LaTeX_01.pdf`
     - TeX Live: `https://www.tug.org/texlive/`
     - TeXMaker: `https://www.xm1math.net/texmaker/download.html`

       .

2. **Logos** To make the booklet pretty, the title page includes logos of the University of Bayreuth and of the faculty. These files are stored in a folder called `Logos`. It contains the following files:

   - `biogeo.jpg`
   - `dist_eco.jpg`
   - `GCE-Logo.jpg`
   - `Uni_Bayreuth.jpg`

3. **Files**
   LaTeX is modular, meaning that different tasks (like making the title page or listing all species) can be dealt with in separate files. This approach makes it easy to only change what needs to be changed. The master file, typically titled `main.tex`, generally both contains the main body of your document and deals with arranging the content from other files with specialised tasks. These files are referenced by their file path and file-name, therefore it is important to make sure that you **do not accidentally change the file names**, and that **all files remain in the same folder**.

   > **NOTE:** A word on `.csv` files and the importance of checking raw text files. The ending `.csv` is short for *comma-separated value*. These types of files can be opened using spreadsheet programs like Excel or LibreOffice Calc for easy viewing, but internally, they are only simple lines of text without any formatting. Spreadsheet programs recognise certain characters in these text files, usually commas, as cell boundaries, and display them as such — for example, the line
   > ```
   > column A, column B, column C
   > ```
   > would be displayed as
   > ```
   > | column A | column B | column C |
   > ```
   > in a spreadsheet program. While commas are typically the default setting for boundaries, it is possible to use a different character like a semicolon. This is the case for the Booklet Builder in order to make it possible to add notes that include commas in sentences. The LaTeX files for the Booklet Builder are likewise explicitly written to interpret semicolons, not commas, as entry separators.
   > This means that **if you choose to open any of the content files with a .csv ending using a spreadsheet program** for easy viewing and editing, please be aware that when you open or close it, the program may ask you about "separator options". **Make sure you choose "separated by semicolon"**, otherwise any commas in the text will be re-interpreted as cell boundaries by the program, the spreadsheet may re-encode all semicolons as commas, and as a result, the LaTeX files will not be able to properly interpret your input files.
   > It is an easy mistake to undo, but can be a hassle. If you want to circumvent this issue altogether, troubleshoot, or double check, you can also choose to open .csv files in simple text editors like Microsoft Notepad or Kate, although it's less pretty than viewing it as a spreadsheet.
   > Spreadsheets may also have trouble displaying special characters like German Umlaute. If this is the case, this should only be a problem when viewing in the spreadsheet program itself. As long as you make sure the character encoding is "Western Europe (ISO-8859-15/EURO)", it will not be an issue in the final document output.

- **Content files**
  These are the files meant to be changed by you.

  - `booklet_title.csv`
    This file includes the title of your booklet, location, version, date, author, sources, etc. (**Note the file ending! The content file is a** `.csv` **file. If the file ending is** `.tex`**, you are looking at the typesetting file discussed below**).
  - `species_list.csv`
    This file is the heart and soul of the booklet! It includes a list of the species, family, author, synonyms, local names, comments, the file paths for the identification pictures, and the option to change the width of these pictures.
  - `special_pages.csv`
    Using this file is optional. If you have any information or pictures that you would like to include at the end of your booklet, this is where you can put it. You can find more info about the format of these special pages in section 3.4.
  - `species_register.csv`
    This file is used to create an alphabetical species register at the very end of your booklet. Content-wise, it is a duplicate of `species_list.csv`, only the sorting is modified. You can find more info about this in section 3.5.

- **Typesetting files**
  These files compile and typeset all the information stored in the content files. They are written such that you should not need to touch them, unless you specifically want to change the typesetting, the file path of the content files, or need to debug something.

  - `booklet_title.tex`
    Formats the title page of the booklet.
  - `main.tex`
    Formats the main body of the text and outputs the final PDF file.

4. Empty folder named `Images`
   Each species in the booklet typically includes two identification images. These are often sourced from Flora Helvetica (https://www.infoflora.ch/en/) and Rothmaler. You will need to download these yourself (depending on the content of the booklet), and make sure you store them in a folder called "Images", as the filepath in the .tex files specifically references this folder.

And you're all set to start working! The step-by-step workflow is explained in the next section.

3 Workflow

3.1 Changing the booklet title

- Open `booklet_title.csv`. If you open it using a spreadsheet program like Excel or LibreOffice Calc, it should look something like fig. 1a. If, instead, you're using a plain text editor, it should look like fig. 1b. If it doesn't, please refer to the note about `.csv` files in section 2.

(a) Spreadsheet

(b) Plain text

Figure 1: booklet_title.csv

- Change the content in column B (or, equivalently, anything after the semicolon in plain text) as needed. All fields are optional, if a line isn't necessary, just leave it blank. **Do not delete any rows!**

- Save and close the file. If asked, **make sure you save it as a `.csv` file with "semicolon" as separator, and as a `.csv` file. Don't let your spreadsheet program change the file type to `.ods` or `.xls`!**. Also make sure the file is encoded in as "Western Europe (ISO-8859-15/EURO)", otherwise German Umlaute (ä, ö, ü, ß) will not compile properly. The encoding option typically pops up in the same window as the separator options when you open or save the file.

3.2   Creating a species list

- Open `species_list.csv`. Same `.csv` caveats as outlined in section 3.1 apply.

- For each species, you can now add the name, author, synonyms, family, its local name, and, if you wish, comments and locations. Locations (like "GRA") will be shown in the outer margin, and you can specify the colour (good options are "red", "blue", "Forest-Green" (note the capitalisation!), and "orange". If you leave the field LocationColour blank, the text will be black).
  If you want to get fancy and have multiple locations with multiple different colours for the same species, you can do this by using a LaTeX command directly in the Location column. The syntax is \textcolor{<colour>}{<location>} , for example

  $$\text{\textcolor{red}{NE}, \textcolor{blue}{SW}}$$

  This will show up as

  $$\text{NE, SW}$$

  If you do this, just leave the field LocationColour blank.

- The columns FigOneWidth and FigTwoWidth give you the option to change the image size if necessary. Their exact use is specified in section 3.3.

All fields are optional and can be left blank.

Some important notes:

- Make sure not to change the column names! The LaTeX script explicitly refers to these names.

- If you want to include the character "&" (e.g. for multiple authors), **add a backslash in front of the** "**&**", i.e. "\&". LaTeX will otherwise interpret it as an internal command and output gibberish. The same is true for "_". **Make sure it's actually a backslash and not a forward slash!** Debugging that obscure error might set you back hours that you will never get back.

- The columns `FileNamePhoto` and `FileNameID` are explained in section 3.3

- Sort the species list according to how you wish it to be displayed in the booklet (i.e. if you want them sorted alphabetically by name, select that column and sort accordingly. If you want to also sort by Family, sort by that column.) Note: the sorting of the table of contents (= "species register") at the beginning of the booklet is independent of this! See how you can adjust this in section 3.5.

3.3   Downloading ID images

Each species is typically shown using two images: one photo, and one drawing. Each booklet entry therefore has space for two figures.

- Download your ID pictures. This can be done, for example, via screenshots of the digital Rothmaler, or simply by saving photos from the Flora Helvetica webpage. **Make sure the images are of sufficient quality.** LaTeX doesn't compress images unless specifically asked to do so, however it cannot magically make an image better than it is. Aim for at least 300 dpi for a clear image. **Think also about the image format – the same length-width ratio as DIN A4 will fit best, but it's not strictly necessary (see below).**

- Save all pictures in the `Images` folder. **Make sure to name each file clearly and without spaces**, e.g. `Acer_pseudoplatanus_photo.jpg` and `Acer_pseudoplatanus_ID.jpg` for the photo and the ID picture, respectively.

- Add the file names to the `species_list.csv` file (see also fig. 3). **It is not necessary to include the full file path as long as each image is stored in the `Images` folder.**

- Specify FigOneWidth and FigTwoWidth, which give you the option to change the image size if necessary. By default, each image will occupy half the page (of half the page, that is). If you have an image that you would like to extend beyond that, you can do this by specifying `FigOneWidth` and `FigTwoWidth`. By default, i.e. if you leave these fields blank, both are set to 0.5. Have a look at fig. 2 to see what the page division of a typical booklet page looks like and to get a feeling for how the figure width changes with this specification. Ideally, `FigOneWidth` and `FigTwoWidth` should add up to 1, but you have some wiggle room here if you need it.
  Note also that if you want to, for example, add three images for a species, or if you have a very specific layout in mind, you can easily implement this within the existing framework by creating a single compilation image with your desired layout (Canva is great for this, but feel free to use whatever), calling this image with FileNamePhoto, and setting the figure widths accordingly (i.e. FigOneWidth = 1, FigTwoWidth = 0).
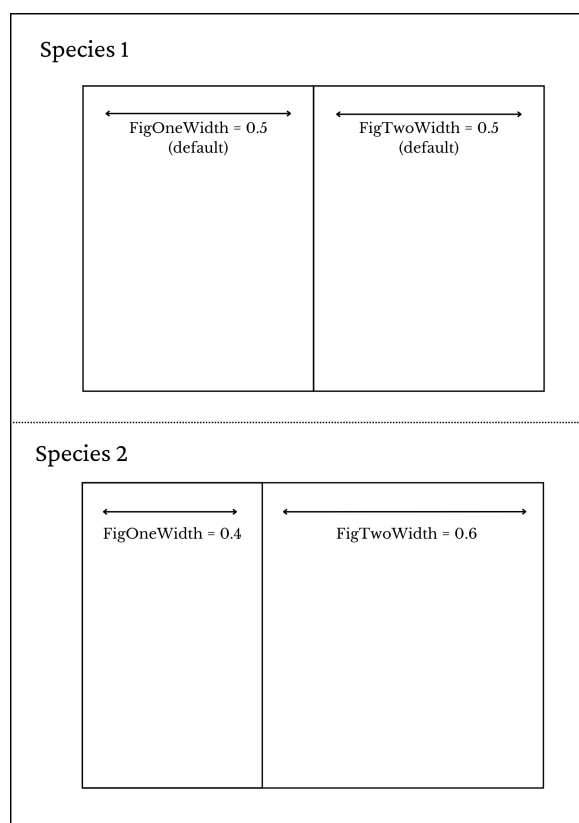
Figure 2: Page division and figure widths

- Save and close the `species_list.csv` file.

The finished species list should look something like fig. 3

	Family	Species	Author	Synonyms	LocalName	FileNamePhoto	FileNameID	Comments	FigOneWidth	FigTwoWidth
1	Family	Species	Author	Synonyms	LocalName	FileNamePhoto	FileNameID	Comments	FigOneWidth	FigTwoWidth
2	Amaryllidaceae	Allium lusitanicum	Lam.	Allium senescens	Berg-Lauch	Allium_lusitanicum_1.jpg	Allium_lusitanicum_2.jpg			
3	Amaryllidaceae	Allium oleraceum	L.		Gemüse-Lauch	Allium_oleraceum_1.jpg	Allium_oleraceum_2.jpg			
4	Apiaceae	Bupleurum falcatum	L.		Sichelblättriges Hasenohr	Bupleurum_falcatum_1.jpg	Bupleurum_falcatum_2.jpg	Subsp. faclatum for image 1		
5	Apiaceae	Pimpinella saxifraga	L.		Gewöhnliche Kleine Bibernelle	pimpinella_saxifraga_photo.jpg	pimpinella_saxifraga_ID.png		0.4	0.6
6	Apiaceae	Seseli libanotis	(L.) W. D▸	Libanotis pyrenaica	Berg-Heilwurz	Seseli_libanotis_1.jpg	Seseli_libanotis_2.jpg			

Figure 3: species_list.csv

3.4 Special pages

After the standard species section you have the option to add some more information. You can do this in the form of images that you include in the file `special_pages.csv` the same way you added the ID images. If you would like to include text, you can type it out, screenshot it, and turn it into an image.

Unlike the main section of the booklet, here you have the flexibility to split the page into as many sections as you want to using the `MaxHeight` and `MaxWidth` columns. This will give LaTeX a factor by which it will multiply the maximum width and height of the image.

- `MaxHeight` lets you set the maximum height of your image. By default, this is the page height (= 1), i.e. your image will occupy at most the full page.
  If you wanted to add three images of the same size to a single page, you could, for example, set the height for all three to 0.3 (or 0.2, 0.5, 0.3 – or whatever you fancy. As long as the values add up to roughly 1).

- `MaxWidth` is similar to `MaxHeight`, but gives you the option to restrict the width of your image. If you want to leave some space around your images, for example, you can set a large `MaxHeight` and restrict your image size itself using `MaxWidth`.

Note: `MaxHeight` and `MaxWidth` will give you the **maximum bounds** of your image, but will not change the ratio of the image height:width itself.
Make sure that you use a point, not a comma!

3.5 Species register

It is useful to have a species register with alphabetically sorted species while keeping the main booklet sorted by family. To get this register, you simply

- copy the contents of your finished `species_list.csv` into `species_register.csv`

- select the column "Species" in full

- **unselect the very first row, i.e. the column name "Species"**

- go to `Data` > `Sort ascending` > `Extend selection`
  Note: you can, of course, adapt the sorting to your requirements. If you want to sort it by species name or author instead, you can do that.

- close the file, making sure that the encoding is correct and that the column names are still in the first row.

3.6 Compile the booklet

As mentioned in section 2, it is up to you whether you want to use something like Overleaf or work locally. The former may be easier to deal with at the beginning.

3.6.1 If you're working with Overleaf

- Compress all Booklet Builder files into a `.zip` file. On windows, this is easily done by selecting all files in your file manager, right-clicking it, and selecting `Send to →` `Compressed (zipped) folder`. On MacOS and Linux, just select `Compress`. Your compressed folder should include

    - The `Images` folder containing your ID pictures
    - The `Logos` folder containing the university logos
    - The typesetting file `main.tex`
    - The typesetting file `booklet_title.tex`
    - The content file `booklet_title.csv`
    - The content file `species_list.csv`
    - The (re-sorted) content file `species_register.csv`.

- Go to https://www.overleaf.com/ and set up an account.

- In Overleaf: open a new project using the big green `New Project` button at the top left. Select `Upload Project`.

- Select or drag your newly created `.zip` folder into the popup window.

And that's it! Your project should automatically compile and output a PDF. Your file will include a table of contents at the beginning, which will update automatically if you change anything in your species list. You can download your PDF using the `Download PDF` icon next to the green `Recompile` button on the upper right.

3.6.2 If you're working locally

- Open the file `main.tex` in TeXMaker (either by starting up TeXMaker and selecting File > Open > [your filepath], or by right-clicking `main.tex` in your file browser, and selecting "Open with TeXMaker")

- In TeXMaker: Press the button next to "Quick Build", wait a few seconds for the compilation to finish, and then press the button next to "View PDF".
  If TeXMaker tells you that there is no file, wait a bit longer after the first button before you press the second. Most likely, the compilation just didn't finish in time. You can generally ignore any warnings and errors that pop up, they are harmless and will not stop the file from compiling. If it still doesn't compile for some reason, **go over your csv file and make sure you didn't make any mistakes there**. See also section 4 for common mistakes.
  Note: make sure that `main.tex` is located in the same folder as the remaining files and folders!
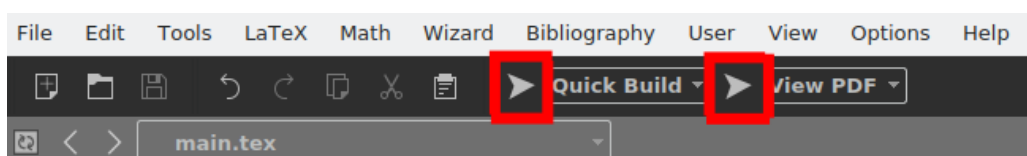

Figure 4: TeXMaker

And that's it! The folder should now automatically contain a pdf file named `main.pdf`. This is your finished file, feel free to re-name it.

Note: Your booklet includes a table of contents at the beginning. If the page numbers show up as "**??**", re-compile the file (i.e. press both buttons again). Sometimes TeXMaker is a little temperamental for no discernable reason, it should be fine after the second (or third, if you have a very large file) build attempt.

3.7 Workarounds

Now all the tedious leg work is done and you have the backbone of your booklet, you may want to tweak little things, insert exceptions, or add a special page that only your project needs. While this can't be done using the BookletBuilder directly (imagine how bloated everything would get if every eventuality was coded in), there are workarounds that should allow you to do everything you need. As a general solution: small tweaks are easiest to accomplish by importing the booklet into Canva (`https://www.canva.com/`). I have found this to be a very smooth experience even for large files, it's free, quality does not suffer, and you should be able to move all elements independently. The only caveat I have found is that the table of contents at the beginning doesn't format properly. If something isn't listed below: get creative!

3.7.1 I can't find an ID image and need to describe my species using words

This is easy: write your description in a text editor of your choice, take a screenshot, and just use that picture instead of the ID photo.

3.7.2 A particular page is being weird/I want one page to look different from the others

The BookletBuilder can only take away the brunt of the work by automatising repetitive tasks with identical settings. If you need something special outside the standard framework, or a particular page just doesn't want to behave, you can, however, create the page you want using e.g. Word (or PowerPoint), convert it to a PDF, and splice it into the finished booklet. There are numerous services that can do this for you, for example `https://www.ilovepdf.com/` or `https://smallpdf.com/pdf-tools`. Just don't forget to first add a dummy page with the desired title into its correct spot in the booklet! This way, your merged-in-after-the-fact Frankenstein page will still show up in the table of contents. Don't forget to remove the dummy page at the end, of course.

If you want everything to look cohesive, it is best to use the same font that is used in the Booklet Builder. In LaTeX, this font is called `charter`. In Word, this corresponds to the font `Bitstream Charter`, which you might need to download first.

WORK IN PROGRESS

4   Troubleshooting

Please note that if something doesn't compile, **don't look at the LaTeX code but at your csv files!** The code is tested and it works. Go through these steps one by one to troubleshoot:

- STEP 1: Open your .csv files (if necessary in a plain text editor) and check for

    - the spelling of the column names
    - any trailing commas
    - accidental forward/back slashes, underscores and ampersands
    - mis-spelled filenames
    - filenames with spaces (get rid of the spaces!)
    - encoding.

    Pay particular attention to anything mentioned in a box, printed in bold, or emphasised in red.

- STEP 2: If you have scrutinised every line and can't find a mistake, it's time to turn to the **images**. If for some reason an image is corrupt (e.g. lacking metadata), your file will not compile.
    To test whether this is the case, you can do the following:

    - re-name your current `Images` folder (just add a letter or so)
    - create a new empty folder called `Images`
    - re-compile and see if it still refuses to output a PDF, or whether you now get a PDF with no images. In case of the latter, hurray: you have found the culprit and can now follow the steps below. In case of the former: the images are not the problem. Proceed to STEP 3.

    Unfortunately, images often look fine through the preview file browser, i.e. you may not be able to tell at a first glance which of the files are the culprit. Instead, you may need to just open the first image and...

    - click your way through until you hit an error message. If that happens:
    - delete the offending file name in your `species_list.csv` file
    - re-compile and see if the error persists. If it does:
    - continue until you find the next faulty image

    Repeat these steps until the compilation is successful

- STEP 3: If all else fails, congratulations: you have created your very own unique error! Unfortunately this is where it gets tedious, because you need to find the line(s) in the csv causing the trouble. There is no other way of doing this but following a process of elimination. Begin by doing the following:

    - create a backup of `species_list.csv` and store it safely
    - temporarily delete all but the first line of the original `species_list.csv`
    - re-compile and see if the error persists. If it does: go and get the newest version of `species_list.csv` from git. Make sure the encoding is correct. If it doesn't:

- re-insert the first 5-10 lines of your own species list and re-compile. If the error persists: double check each line. If it compiles:
- progressively insert lines (doesn't have to be individually, I typically start with blocks of 50 lines and refine from there) of your species list until you hit an error, then go look for a typo.

Happy hunting!