LaTeX Template Documentation

A comprehensive guide to use this template

Karl Voit*

December 10, 2011

Contents

1 How to use this LaTeX document template 2
- 1.1 Modify this template for your requirements . . . 2
- 1.2 How to compile this document . . . 3
- 1.3 How to get rid of the template documentation . . . 3
- 1.4 What about modifying the template? . . . 3

2 `preamble.tex` — Main preamble file 4
- 2.1 UTF8 as input charset . . . 4
- 2.2 Language settings . . . 4
- 2.3 Headers and footers . . . 4
- 2.4 References . . . 5
  - 2.4.1 Example citation commands . . . 5
  - 2.4.2 Manual compiling with `Biber` . . . 6
  - 2.4.3 Using this template with BibTeX . . . 6
- 2.5 Miscellaneous packages . . . 7

3 `mycommands.tex` — various definitions 7
- 3.1 `myfig` — including graphics made easy . . . 8
- 3.2 `myclone` — repeat things! . . . 8
- 3.3 `myquote` — correct quotation marks . . . 8

*http://LaTeX.TUGraz.at

4 typographic_settings.tex — Typographic finetuning 8
4.1 References related to typographic settings . . . 9
4.2 French spacing . . . 9
4.3 Text figures . . . 9
4.4 Abbrevations using SMALL CAPS . . . 10
4.5 Colorized headings and links . . . 10

5 pdf_settings.tex — Settings related to PDF output 10

1 How to use this LaTeX document template

This LaTeX document template from LaTeX@TUG[1] is based on KOMA script[2]. It provides an easy to use and easy to modify template. All settings are documented and many references to additional information sources are given.

1.1 Modify this template for your requirements

1. Put your desired PDF file name in the second line of file `Makefile`
   - replace »Projectname« with your filename
   - do not use any file extension like `.tex` or `.pdf`
2. OPTIONAL: Modify files in the folder `preamble` if necessary
   - following sections should give you an idea what to do and why
3. Modify `userdata.tex`:
   - `\myauthor`, `\mytitle`, and so forth
4. Modify `main.tex`:
   - your desired general document structure
5. replace file or fill your content in `introduction.tex`
   - you can rename `introduction.tex` but then you have to modify its include command in `main.tex` too
6. OPTIONAL: create further tex-files (similar to `introduction.tex`) for each chapter of your document

[1] http://LaTeX.TUGraz.at
[2] http://komascript.de/

- include them according to the example of `introduction` in `main.tex`

7. generate your document
   - with a LaTeX editor:
     - select `main.tex` as the »main project file« or make sure to compile/run only `main.tex` (and not `introduction.tex` or similar)
   - OR with GNU make: run `make pdf`

1.2 How to compile this document

If your system provides GNU make[3], it is very easy to compile this document:

```
make pdf
```

You can get a list of all other commands provided by the Makefile by invoking `make help`.

If you do not have GNU make, please stick to your usual compile method and notice the changes for `Biber` in section 2.4.

1.3 How to get rid of the template documentation

Simply remove the files `Template_Documentation.pdf` and `Template_Documentation.tex` (if it exists) in the main folder of this template.

1.4 What about modifying the template?

This template provides an easy to start LaTeX document template with sound default settings. You can modify each setting any time. It is recommended that you are familiar with the documentation of the command whose settings you want to modify.

The following sections describe the settings and commands of this template and gives a short overview of its features.

[3] https://secure.wikimedia.org/wikipedia/en/wiki/Make_%28software%29

2 preamble.tex — Main preamble file

In file preamble/preamble.tex you will find the basic definitions related to your document. This template uses the KOMA script extension package of LaTeX.

There are comments added to the \documentclass{} definitions. Please refer to the great documentation of KOMA[4] for further details.

What should I do with this file? For standard purposes you might use the default values it provides. You must not remove its include command in main.tex since it contains important definitions. This file contains settings which are documented well an can be modified according to your needs. It is recommended that you fully understand each setting you modify in order to get a good document result.

2.1 UTF8 as input charset

You are able and should use UTF8 character settings for writing these TeX-files.

2.2 Language settings

The default setting of the language is American. Please change settings for additional or alternative languages used.

2.3 Headers and footers

Since this template is based on KOMA script it uses its great scrpage2 package for defining header and footer information. Please refer to the KOMA script documentation how to use this package.

[4] scrguide.pdf for German users

2.4 References

This template is using `biblatex` and `Biber` instead of BibTeX. This has following advantages:

- better documentation
- Unicode-support like German umlauts (ö, ä, ü, ß) for references
- flexible definition of citation styles
- multiple bibliographies e. g. for printed and online resources
- cleaner reference definition e. g. inheriting information from `Proceedings` to all related `InProceeings`
- modern implementation

In short, `biblatex` is able to handle your `bib`-files and offers additional features. To get the most out of `biblatex`, you should read the very good package documentation. Be warned: you'll probably never want to change back to BibTeX again.

Take a look at the files `references-bibtex.bib` and `references-biblatex.bib`: they contain the three references `tagstore`, `Voit2009`, and `Voit2011`. The second file is optimized for `biblatex` and takes advantage of some features that are not possible with BibTeX.

This template is ready to use `biblatex` with `Biber` as reference compiler. You should make sure that you have installed an up to date binary of `Biber` from its homepage[5].

2.4.1 Example citation commands

- cite [**Voit2011**] and cite [**Voit2009**; **Voit2011**].
- citet **Voit2011** and citet **Voit2009**; **Voit2011**
- autocite [**Voit2011**] and autocite [**Voit2009**; **Voit2011**].
- autocites [**Voit2011**] and autocites [**Voit2009**; **Voit2011**].
- citeauthor **Voit2011** and citeauthor **Voit2009**; **Voit2011**
- citetitle **Voit2011** and citetitle **Voit2009**; **Voit2011**
- citeyear **Voit2011** and citeyear **Voit2009**; **Voit2011**

[5] http://biblatex-biber.sourceforge.net/

- textcite **Voit2011** and textcite **Voit2009**; **Voit2011**
- smartcite[6] and smartcite[7].
- footcite[8] and footcite[9].
- footcite with page[10] and footcite with page[11].
- fullcite **Voit2011** and fullcite **Voit2009**; **Voit2011**

Please note that the citation style as well as the bibliography style can be changed very easily.

2.4.2 Manual compiling with `Biber`

To generate a document using `Biber`, you can stick to following example:

```
pdflatex main.tex
biber main
pdflatex main.tex
pdflatex main.tex
```

Using the `Makefile` which comes with this template, you simply have to call `make pdf` in order to get the resulting document.

In the file `preamble/preamble.tex` you can modify the template defaults for references such as changing the citation style or adding your own reference file(s). Look out for `\usepackage[...]{biblatex}` to modify its options.

2.4.3 Using this template with BibTeX

If you do not want to use `Biber` and `biblatex`, you have to change several things:

- in `preamble/preamble.texw`
  - remove the usepackage command of `biblatex`
  - remove the `\addbibresource{...}` command

[6] **Voit2011**.
[7] **Voit2009**; **Voit2011**.
[8] **Voit2011**.
[9] **Voit2009**; **Voit2011**.
[10] **Voit2011**.
[11] **Voit2011**.

- in `main.tex`
  - replace `\printbibliography` with the usual `\bibliographystyle{yourstyle}` and `\bibliography{yourbibfile}`
- if you are using GNU `make`: modify `Makefile`
  - replace `BIBTEX_CMD = biber` with `BIBTEX_CMD = bibtex`
- Use the reference file `references-bibtex.bib` instead of `references-biblatex.bib`

2.5 Miscellaneous packages

There are several packages included by default. You might want to activate or deactivate them according to your requirements:

`pifont` For additional special characters available by `\ding{}`

`ifthen` For using if/then/else statements for example in macros

`eurosym` Using the character for Euro with `\officialeuro{}`

`xspace` This package is required for intelligent spacing after commands

`color` This package defines basic colors

`blindtext` This package is used to generate blind text for demonstration purposes.

3 `mycommands.tex` — various definitions

In file `preamble/mycommands.tex` many useful commands are being defined.

What should I do with this file? Please take a look at its content to get the most out of your document.

One of the best advantages of LaTeX compared to WYSIWYG software products is the possibility to define and use macros within text. This empowers the user to a great extend. Many things can be defined using `\newcommand{}` and automates repeating tasks. It its recommended to use macros not only for repetitive tasks but also for separating form from content such as CSS does for XHTML. Think of including graphics in your document: after writing your book, you might want to change all captions to the upper side of each figure. In this case you either have to modify all `includegraphics` commands or you were clever enough to define something like

\myfig[12]. Using a macro for including graphics enables you to modify the position caption on only *one* place: at the definition of the macro.

Following section describes some macros that came with this document template from LaTeX@TUG and you are welcome to modify or extend them or to create your own macros!

3.1 myfig — including graphics made easy

The classic: you can easily add graphics to you document with \myfig:

```
\myfig{flower}%% filename w/o extension in directory "figures"
      {0.7\textwidth}%% maximum width/height, aspect ratio will be kept
      {This flower was photographed at my home town in 2010}%% caption
      {fig:flower}%% label
```

3.2 myclone — repeat things!

Using \myclone[42]{foobar} results the text 'foobar' printed 42 times. But you can not only repeat text output with myclone. Default argument for the optional parameter 'number of times' (like '42' in the example above) is set to two.

3.3 myquote — correct quotation marks

You should *never* use quotation marks found on your keyboard since they end up in strange characters or false quotation marks.

In LaTeX you have to use different quotation marks. With \myquote{foobar} you can get correct quotation marks around 'foobar'. Please do check the definition of this function in order to modify its settings according to your language and quotation marks.

4 typographic_settings.tex — Typographic finetuning

The settings of file preamble/typographic_settings.tex contain typographic finetuning related to things mentioned in literature. The settings in this file relates to personal taste and most of all typographic experience.

[12] See below for a detailed description

What should I do with this file? You might as well skip the whole file by excluding the `\input{preamble/typographic_settings.tex}` command in `main.tex`. For standard usage it is recommended to stay with the default settings.

4.1 References related to typographic settings

4.2 French spacing

Why? [**Bringhurst1993**]: '2.1.4 Use a single word space between sentences.'

How? [**Eijkhout2008**]:
`\frenchspacing %% Macro to switch off extra space after punctuation.`
Note: This setting might be default for KOMA script.

4.3 Text figures

...also called old style numbers. (German: »Mediävalziffern« [13])

Why? [**Bringhurst1993**]:

> '3.2.1 If the font includes both text figures and titling figures, use titling figures only with full caps, and text figures in all other circumstances.'

How? Quoted from Wikibooks[14]:

> Some fonts do not have text figures built in; the textcomp package attempts to remedy this by effectively generating text figures from the currently-selected font. Put `\usepackage{textcomp}` in your preamble. textcomp also allows you to use decimal points, properly formatted dollar signs, etc. within `\oldstylenums{}`.

...but proposed LaTeX method does not work out well. Instead use:
`\usepackage{hfoldsty}` (enables text figures using additional font) or
`\usepackage[sc,osf]{mathpazo}` (switches to Palatino font with small caps and old style figures enabled).

[13] https://secure.wikimedia.org/wikibooks/de/wiki/LaTeX-W%C3%B6rterbuch:_Medi%C3%A4valziffern

[14] https://secure.wikimedia.org/wikibooks/en/wiki/LaTeX/Formatting#Text_figures_.28.22old_style.22_numerals.29

4.4 Abbrevations using small caps

Why? [**Bringhurst1993**]: '3.2.2 For abbrevations and acronyms in the midst of normal text, use spaced small caps.'

How? Using the predefined macro `\myabk{}` for things like UNO or UNESCO using `\myabk{UNO}` or `\myabk{UNESCO}`.

4.5 Colorized headings and links

This document template is able to generate an output that uses colorized headings, captions, page numbers, and links. The color named 'DispositionColor' used in this document is defined near the definition of package `color` in the preamble (see section 2.5). The changes required for headings, page numbers, and captions are defined here.

Settings for colored links are handled by the definitions of the `hyperref` package (see section 5).

5 `pdf_settings.tex` — Settings related to PDF output

The file `preamble/pdf_settings.tex` basically contains the definitions for the `hyperref` package including the `graphicx` package. Since these settings should be the last things of any LaTeX preamble, they got their own TeX file which is included in `main.tex`.

What should I do with this file? The settings in this file are important for PDF output and including graphics. Do not exclude the related `input` command in `main.tex`. But you might want to modify some settings after you read the documentation of the `hyperref` package.