

On writing L^AT_EX documents for the simuline organization

Dr. Ernst Reißner

May 17, 2023

Contents

1	Introduction	2
2	Tools	2
2.1	Configuration	3
3	Sections	4
4	Miscellaneous	5
4.1	How to deal with ltex	5
4.2	Terminology Management, Glossaries	5
4.3	Relation to other documentation systems	6
	Glossary	6

List of Figures

List of Tables

1 Introduction

This document contains rules how to write documents specifically in \LaTeX . Of course all rules for writing documents apply (TBD: reference) and this documents focuses on aspects specific for latex documents. The rules here are followed for similine organization, but is applicable in a broader context maybe with modifications.

At the current stage, this document is only a torso. Its purpose is to check whether the overall bibliography is usable.

2 Tools

\LaTeX is a compiled language. Thus, one needs an editor, possibly with an according \LaTeX -extension, the proper compiler and a viewer for each output format.

As an editor `emacs` with extension `auctex` is a possible choice, but we chose `visual code` with an according extension. TBD: add a reference to the install script for `code`. The main viewer is `okular` but also `chrome` with markdown extension (TBD: which) is needed. What makes latex difficult from a point of view of tooling is, that there is not a single converter but a universe of related tools, e.g. to make a

bibliography or an index, and also latex itself must be enriched loading appropriate packages.

TBD: add reference to the installation script **zypper** and installation script for visual code extensions, among them also those for latex. TBD: think about how to reference documents in Markdown format.

As **maven** is the general build tool, we bundle all those converters in a single maven plugin which does nothing but invoking converters by need. For our special needs, we developed the **latex-maven-plugin**. Starting from version 1.5, it has a goal **vrs** listing all converters with their applicable versions. It is recommended to have all of them installed. In addition, there are lots of latex-packages required, all of them in the **texlife** distribution which contains also most of the converters, but not all of them. The site of the plugin gives hints on the pieces of software used by the plugin and provides a manual covering the full functionality.

What is important from the point of view of quality management is, that the plugin detects failures and warnings so that if conversely the build process completes without error or warning, one can be sure that the result is without failure also. Another point is that the plugin is for building all documents in one build run but supports also development of a single document. Maybe the most widely used tool for building a single L^AT_EX document is **latexmk** and the plugin is designed to cooperate with **latexmk**, switching to **latexmk** if an error occurs or if a section must be added and then again switching back to the plugin if all documents shall be compiled.

In the long run, all projects, even this one shall use the **latex-maven-plugin**. This is currently not possible because this would introduce a cyclic dependency, because on the one hand, the latex plugin is subject to the rules prescribed in this quality management project and must be referenced in the latex plugin project, and on the other hand, this quality management project needs the plugin to build. This can only be resolved if this one project refers to a lower version of the sibling project. At the beginning, this is not possible. In general, compilation should work with default parameters.

For now, this document is generated locally only using command line tools **lualatex** and **bibtex** (in future we will require **biber** which presupposes the package **biblatex**). Later we require a glossary and an index, glossary with **bib2gls** and an index, where the tooling is not yet so clear. Instead of directly using the combination of these tools, one could use **latexmk** with an appropriate config file **.latexmkrc**.

2.1 Configuration

This section collects configuration of tex documents. A big part of configuration is the set of packages used and the options of the packages. Note that these packages

partially presuppose the abovementioned tools. These tools are mandatory also in other cases for sake of reproducibility.

Some packages must be loaded before declaration of the `documentclass`. This document is just an example.

```
\RequirePackage[12tabu, orthodox]{nag}
```

Then comes the declaration of the `documentclass`. Currently we support the classes `article`, `book` and `beamer`. Since this is a (western) european company, only a4 paper is supported. This shall be provided by the package `geometry`, but the option shall be given not as package option but as documentclass option to make it available to other packages also. Font size of 12pt shall be in general used.

```
\documentclass[a4paper,12pt]{article}
```

It is a requirement to support “document development”. This includes making a development environment as much WYSIWIG¹-like as possible. To that end, one shall activate `synctex` as follows

```
\synctex=1
```

After that, in the long run further packages shall be input from a general header file. In the time to be, we specify here the packages needed and prohibited:

prohibited `inputenc`, `fontenc` because these do not fit `lualatex`.

mandatory `geometry`, `microtype`, `rerunfilecheck`, `listings`, `hyperref`. This listing is incomplete.

future in future `biblatex` is mandatory.

3 Sections

Each document needs a title and an author and an identification of the version of the document. What is not allowed is a date, because this corrupts certain equality checks: A generated PDF file shall change only if this is really intended and not because the date changes (which happens every day). If a document refers to software, as e.g. a manual, the version of the document is that of the software.

Each document requires a table of contents. If there are figures, tables or listings, then also an according table, e.g. table of figures, is required.

All documents shall have a reference, an index and a glossary. Each of them must have an entry in the table of contents.

An introductory section is mandatory and in most cases, it is advisable to have a summary or a miscellaneous section, or a section with open points.

¹What you see is what you get

4 Miscellaneous

A big subject is how to obtain reproducible documents. Also documents which contain computed data, either figure or computed pictures. A discussion on `pythontex` can be found in [Poo15]. We are lucky that `pythontex` allows computations in python but also in octave, i.e. in the matlab language, the two major languages we use besides our developing language java.

Note that there are alternatives to

4.1 How to deal with l_{at}ex

L_{at}ex has many rules but there is one particular, which is difficult to conform with. It is about spacing.

The root of the problem is, that L_{AT}_EX inserts more space after full stop, but not after other dot followed by a space. But how does L_{AT}_EX decide this? If the dot is preceded by a lower case letter, then it assumes a full stop, after an upper case letter it assumes no full stop. This rule applies in most cases but not in all. Here are two examples:

- “Apple Inc. sells phones” Here, L_{AT}_EX thinks the period ends a sentence and wrongly adds an extra space.
- “My school scores were all A. Einstein would be proud of me!” Here, L_{AT}_EX does not realize that the period ends a sentence and does not add the expected space. Note that `chtex` does not find this problem.

The problem is solved by adding an invisible lower case letter by `\@`. Note how spacing improves:

- “Apple Inc. sells phones”
- “My school scores were all A. Einstein would be proud of me!”

4.2 Terminology Management, Glossaries

Terminology management means that basic terms are defined at a single source. It is decided that this places is a file in BIB format read by `bib2bls`. It is conceivable to have more than one of these BIB files but each term must be defined once over all BIB files.

A very good example of how useful and necessary terminology management is, are the terms precision and accuracy referring to numbers versus precision and accuracy referring to measurement.

4.3 Relation to other documentation systems

Here are important: ‘markdown’: how to include markdown in a tex file. jupyter notebooks: ‘jypynotex’ ‘texinfo’.

Glossary

accuracy Accuracy of measurement determines how close the measurement is to the actual value. 5

accuracy Accuracy of a fixed/floating point number is the number of correct digits. For an interval it is the width of the interval 5

precision Precision of measurement is the degree to which it may vary from one to the next. 5, *see also* precision

precision Precision is the number of digits in a number (format). The maximum number p of significant digits that can be represented in a format, or the number of digits to that a result is rounded. IEEE754 5

References

[Poo15] Geoffrey M. Poore. PythonTeX: reproducible documents with L^AT_EX, Python, and more. *Comput. Sci. and Discov.*, 8(1), 7 2015.