The recorder option for latex processors and its applications

Ernst Reißner

Contents

1 Introduction

This document is about the `recorder` option available for all modern LaTeX processors, among those also `lualatex`. If the LaTeX processor is applied to a file `xxx.tex` with this option set, it writes a file `xxx.fls` in the FLS format which is described in Section 2. Essentially, it contains dependency information for that file which can be used to decide on rebuilding the document. This is mostly to spare processing time while still ensuring that the created documents are up-to-date. Among others, `latexmk` uses this information.

Another possible use case for the FLS files is to ensure reprocucibility of the results. The FLS file can be the main contribution to this: Essentially for reproduction all the input files mentioned in the FLS file must be the delivered to reproduce the results.

Part is in the installation so if this is known, only the files mentioned in the FLS file as input which are outside the installation must be delivered. CAUTION: There are packages not part of an installation but installed separately fitting into the folder structure of the installation like the tikz uml package described in [Kie16] at time of this writing.

To get all creates files all files mentioned in the FLS file as output must be delivered.

The problem with the FLS files is, that no official information on the FLS format seems available and so there is no way but reverse engineering.

The risk to have an incomplete specification is always present, also that the format is updated and the reverse engineered specification is rendered invalid. Thus, software based on this specification must be failure tolerant. The goal that the result of running this software is perfect if no warning is displayed implies that, the parser for FLS files emits a warning for small deviations from the expected still tolerating these deviations.

Another problem is, that seemingly the list given by the FLS files is not complete. Some gaps are spotted in Section 3

2 The FLS format

The FLS format seems to be line based and so is the description. The format is described in terms of the possible form of lines.

In addition, the form of a line is quite restricted. It is of the form `type file`, consisting of line type and a file path. The available `type`s are

PWD The working folder. Seemingly this is the folder containing the latex file `xxx.tex` to be processed. This type of line occurs once and is the first line. We refer to it as the *current directory*.

INPUT Intentionally all input files. Among those `xxx.tex`, `xxx.aux`, Possibly graphic files and files related with fonts.

OUTPUT Intentionally all output files. Among those `xxx.log`, `xxx.tex`, `xxx.pdf` but also very strange cache files. Note that `xxx.fls` itself and also the file `xxx.synctex.gz` are not mentioned. Maybe there are more files not mentioned.

The line `PWD` is first and there is a single one specifying the working folder. Paths are not restricted to the local folder but may point also into the latex installation. Paths may be absolute and relative. Relative paths seem to be relative to the working folder. TBD: do research on paths with blanks.

Note that a file may occur as input and output like `xxx.aux`.

The relative paths either start with `./` or with `../`. In the first case, it seems that these paths refer to files in a subdirectory of the current directory, not in the second.

The absolute paths either refer to the installation or to a cache. In my linux box, the cache is at `/.cache/texmf/`, so it is in my home directory.

The installation directory of a TEX Live installation, is given by `kpsewhich latex.ltx`, just eliminating three containment levels from the path.

The rest of the input lines of the fls files are the following on my linux box:

```
INPUT /var/lib/texmf/web2c/luahbtex/lualatex.fmt
INPUT /var/lib/texmf/fonts/map/pdftex/updmap/pdftex.map
INPUT /etc/texmf/web2c/mktex.cnf
```

3 Gaps

The FSL file itself is not among the outputs mentioned, which may be intentional as this information is redundant. Besides this, the only examples without loading packages, the author knows about at time of this writing are the file `xxx.synctex.gz` and related. In general, each package is responsible for keeping track of its file input and output. So to ensure that the input and output files mentioned in the FLS file, it is advisable to restrict oneself to a liminted set of packages.

If FIG files are used then as mentioned in [Rei16], Section 3.2, also pictures can be included. Some of the supported formats allow even inclusion of further files.

Metapost files, described in [Hob14] also offer a way to include other files. Also the processor for metapost, `mpost` offers its own `recorder` option as indicated in [Hob14], Appendix B 2.1. Note that also `luatex` can be used as a metapost processor and it should be checked which of the dependencies within metapost files `luatex` records within its FLS file.

Strange enough, nowhere in the manual [Hob14] is introduced the probably most relevant way of including other metapost files, namely by the command `input`, although Section 13.1 uses this

mechanism to include `TEX.mp`. There is a separate Section 12 on reading and writing files, which mention the commands `readfrom`, `closefrom` and `write…to`.

Besides this, there is a more hidden way to include files, because between `btex` or `verbatimtex` and `etex` can be placed arbitrary TEX material and in particular the `\input` macro to include further TEX files.

For `lualatex`, [Hag15], Section "Helpers" illustrates that lua code can be directly invoked within matapost. Of course, lua can access files, possibly reading or writing, and it is hardly conceivable, that the files seen or touched are recorded.

The author did not check for any of these constructions which may introduce dependencies, which of all these dependencies are taken into account, in the FLS file created by `mpost` with given `recorder` option.

Besides running an external metapost processor, the package `luamplib` available for `lualatex` (see [HHR+23]) can include metapost files or metapost code directly. This offers the chance to include the dependencies not in an "external" FLS file of the metapost file but directly in the FLS file of the processed latex file.

In older versions of `luamplib` even metapost files included with the command `input <file>;` where not recorded in the FLS file.

Maybe there are more files not mentioned.

4 References

[Hag15] H. Hagen. Lua in MetaPost. *MAPS*, 46:101–108, 2015.

[HHR+23] H. Hagen, T. Hoekwater, E. Roux, P. Gesang, and K. Dohyun. *The luamplib package.* `https://ctan.org/pkg/luamplib?lang=en`, v2.24.0 edition, 4 2023.

[Hob14] John D. Hobby. *MetaPost, a user's manual*, 5 2014. for version 2.00, `https://www.tug.org/docs/metapost/mpman.pdf`.

[Kie16] N. Kielbasiewicz. The TikZ-UML package, 3 2016.

[Rei16] E. Reißner. The xfig file format for xfig 3.2. see `http://www.simuline.eu/LatexMavenPlugin/xfig/xfigFormat.pdf`, 12 2016.