

Saving and restoring the font cache

Gerald Q. Maguire Jr.

July 2025

This document is a work in progress.

One of the problems when using Lua^AT_EX in Overleaf is timeouts. This occurs because Overleaf has a wall clock time limit. Unfortunately, when using Lua^AT_EX and `babel` with the many fonts used in the template this can lead to timeouts while the font cache is being built for the set of fonts used in the document.

The solution to this comes in several steps:

Step 1 Generate, save, and upload the font files

- Compile `warmup.tex` first; this will load all of the fonts specified in the `examplethesis.tex`, as `warmup.tex` is a subfile that inherits the main document's preamble, hence it loads all the same fonts. Additionally, due to the command `\MySaveFontCache`, `warmup.tex` will also save the font files into a folder (`saved_font_cache`) in the same folder as `output.log`, etc.
- Download all of the “Other logs and files” into a file, such as `/tmp/output.zip`.
- Make a temporary directory, such as `/tmp/c4`, then change to this directory and unzip the `/tmp/output.zip` file.
- Go to the directory `/tmp/c4/saved_font_cache` and delete all of the files with the `lua` extension [these are the raw Lua font files].
- In your Overleaf project, upload the folder `/tmp/c4/saved_font_cache` into the top-level of your project

Step 2 Restoring the save font files - simply compile the `restore_font_cache.tex` file - this will copy the saved font cache files into the working font cache

Now, if you have problems with timeouts due to font cache rebuild taking a long time, you only need to do the last of the above steps.

How much time do you save using the saved fonts?

In my experiments when using all of the enabled fonts in the template, it took about 109 seconds of wall clock time to create the font cache from scratch.

When using the saved files it took ≈ 7 seconds to restore them. This is at the cost of storing the saved fonts.

Why do you have to download and then upload the saved files?

In my experiments, I found that trying to save the files directly into a folder at the project's top level resulted in these files being cleared when the cache was cleared, a new web session was started, or after being idle for some period of time. In contrast, the uploaded files persist across all of these events.

*

What is the difference between lua and luc files?

The file extension `lua` is an uncompiled Lua file, while the file extension `luc` represents a compiled lua file. The compiled files is about 66% of the size of the corresponding Lua file. However, for some of the very large `lua` files (such as the Noto CJK font families where the uncompiled size is more than 10 MB) the reduction can be as much as 56%.

Why upload only the luc files?

If there is a `luc file` in the font cache, the program `luaotfload` will skip loading the `lua` file. Thus saving both time and space. However, any missing fonts that you are now using will be processed and a `lua` file created and then compiled to create a `luc` file.

What if you change the set of fonts you are using?

As noted above, any new fonts you use will be loaded via `luaotfload` and you will have to wait for them to be processed and compiled.

Therefore, if you make a major change in the fonts you are using, simply repeat Step 1 and Step 2 again.