Tutorial: Curating BibTeX for Local LaTeX Writing

Boris Kiefer
Department of Physics
New Mexico State University

September 11, 2025

1 Introduction

A few days ago, I revisited my approach to writing papers on my own or as part of a collaborative effort. Being used to formulas I often turn to LaTeX as the default choice. For collaborative writing I like Overleaf and I started to think how to access my Paperpile reference library directly from Overleaf (yes, this is possible by obtaining a link in Paperpile and set it as the location of the bibliography in Overleaf). I tried this strategy and it works very well. In particular, I like that there are no intermittent files, just straight-up integration. Then, I thought what do I do if I do not have internet or cloud access to Overleaf and I wanted to emulate the auto-synced Paperpile to Overleaf integration but with TexStudio instead of Overleaf. Sounds simple enough until...

In this tutorial I will explain how to:

- Generate a local *.bib file from Google Drive for offline use.

2 Disclaimer

My proposed integration strategy relies on Google Drive Desktop's "Available offline" feature and I only tested the strategy for Paperpile's synced *.bib export to Google Drive. If these products change export/access in the future, the strategy will need to be adjusted, stay vigilant. This tutorial is based on practical experience with Paperpile, Google Drive, and WSL2. It is not affiliated with or endorsed by Paperpile or Google.

3 Prerequisites

- A *.bib file on Google Drive. In my case, I exported an auto-synced *.bib file from Paperpile. The origin of this file is not crucial as long as it is stored on Google Drive.
- Google Drive for Desktop (install on your computer).
- WSL2 (install on your computer).
- Powershell (install/use on your computer)
- TexStudio, but any LaTeX editor should do.

4 Step-by-Step Instructions

1. I installed **Google Drive for Desktop (GDD)** on my computer. After installation, I determined that Google Drive was mounted as **G:** drive and I made sure that I could access **My Drive** with **Windows Explorer (WE)**.
2. Using WE I navigated to the *.bib file on Google Drive, right clicked and selected **Available Offline**.
3. Now we need to tell GDD what to do with the files you marked. In GDD you have two choices: **streaming mode** and **mirror mode**. In my version of GDD, **mirror mode** creates a synced physical mirror of my entire Google Drive on my computer. Since I only need one file, and I wanted to safe disk space, I opted for **streaming mode** which will store my "Available Offline" marked *.bib file in a synced virtual drive. This distinction of local storage mode will become important in later steps.
4. In WE navigate to *.bib, right-click the file, select properties, and you should see the path to the file. I opened my *.tex file in TexStudio and added the path to my document with `\bibliography{path}`. I cited a few references, compiled and in many cases, I found "?" instead of the reference. The reason: Paperpile exported everything, including abstract, and a lot of other fields that I usually do not need in my references. The danger is that **bibtex** may break, due to unknown escape sequences, special characters, incomprehensible language fields,...

This defeats the purpose of having a synced local reference database. Thus, a curation step is needed.

5. One option is to curate *.bib in TexStudio every time I compile. For large reference databases this step is not practical. So, I decided to do the curation on the synced *.bib file on my computer and to store a physical copy of the curated *.bib in the **Documents** folder.

6. I started up my **WSL2** and installed **bibtools**. Now I tried to locate my synced *.bib file and I found that the "G:" drive unlike the "C:" drive was not mounted in "/mnt/*". So, I realized that a workable solution consists of using the discoverable PC Path from step (4) and run bibtool in WSL2. The rest of this tutorial shows you how to set up bibtool for synced *.bib curation, and how to store the cleaned version physically in a local folder. Notes: 1) all filenames for the curation process and the cleaned *.bib file start with a common string and will appear as a group in the directory listings; 2) choosing the **Documents** folder also has the advantage that I back it up regularly.

5 Bibtool Setup

After some trial and error I realized that bibtool requires a configuration file that details which fields to delete. I created a file named `paperpile_strip.rsc` in Notepad with the following content:

```
% Keep citation keys as-is
preserve.key.case = on

% Pretty printing
print.align       = on        % align the '=' signs
print.line.length = 80        % wrap lines at ~80 chars

% Keep comments
pass.comments     = on

% -------- Remove noisy/sync fields (add more if needed)
    --------
delete.field = {abstract}
delete.field = {keywords}
delete.field = {language}
delete.field = {file}
delete.field = {urldate}
```

```
delete.field = {archivePrefix}
delete.field = {primaryClass}
delete.field = {eprintclass}
delete.field = {eprinttype}
delete.field = {pdf}
delete.field = {note}
delete.field = {annote}
% delete.field = {isbn}
% delete.field = {issn}
% delete.field = {shorttitle}
```

I placed this file in the **Documents** folder, and of course you can edit/change to meet your needs.

6 PowerShell Wrapper

The PowerShell script **paperpile_clean.ps1** calls `bibtool` inside WSL2 to process the synced file and write the result to a user specified location (path's must be adjusted):

```
# Paths
$src = "G:\My Drive\paperpile.bib"   # Paperpile's
   synced file on the virtual G: drive
$rsc = "/mnt/c/Users/boris/Documents/paperpile_strip.rsc
   "
$dst = "C:\Users\boris\Documents\paperpile_cleaned.bib"

# Basic checks
if (-not (Test-Path $src)) {
        Write-Error "Source not found: $src"
        exit 1
}

# Clean: Windows to WSL (bibtool) UTF-8
Get-Content -Raw $src -Encoding UTF8 `
| wsl bibtool -r $rsc `
| Out-File -FilePath $dst -Encoding utf8

Write-Host "Cleaned, $dst"
```

As you can se from the file listing, we define three path variables: **$src** points to the synced *.bib, **$rsc** points to the *.rsc, and **$dst** points to the

destination folder of the cleaned local physical copy of *.bib. The following block makes sure that we can access the synced *.bib file. The following block does the heavy lifting: From within a **PowerShell** you can execute this command/file and it will pipe the synced *.bib file into bibtool executed on WSL2 and write the final result to the **$dst** folder. Specifying utf8 solves encoding differences between PC and WSL2. If something goes wrong you will find an error log for trouble shooting

7 One-Push Curation

It is perfectly fine to execute the script in PowerShell, but it requires the extra step of starting up PowerShell. To streamline this process, we can wrap the PowerShell script in a ***.bat** file, that can be executed by double-click.

```
@echo off
powershell -NoProfile -ExecutionPolicy Bypass -File "C:\
   Users\boris\Documents\paperpile_clean.ps1"
pause
```

Note, the **pause** command on the last line is optional, it makes sure that any errors that may have occurred are visible before the process finishes.

8 Final TexStudio Integration

Since I have now a local physical copy, I can point to it in the *.tex file as usual:

```
\bibliography{C:/Users/boris/Documents/paperpile_cleaned
   }
```

If you use biblatex, replace with `\addbibresource{path}` and compile with biber instead of BibTeX.

9 Final Thoughts

My Paperpile library has several thousand entries and yet the curation process takes only a few seconds, creating low latency. I tested the cleaned *.bib file with a few references and they appeared in my compiled pdf without question marks, suggesting that the curation process was successful. As

pointed out before, using the same starting string for the filenames ensures that they appear as a group in folder listings, and the use of a common folder ensures (in my case) that the cleaned database and helper snippets are regularly backed up.

My original goal was to create a synced *.bib integration for local use on my PC. While I did not reach this goal, I came very close and now I can obtain a clean local copy of my synced *.bib from Google Drive, with a single double-click.

I hope this tutorial was useful and saved you some headaches. Good luck!

10 Screenshots for Sanity-Checking

```
% ============================================
% paperpile_strip.rsc  –  BibTool resource file
% Clean Paperpile .bib: remove noisy fields, align
% ============================================

% Keep citation keys as-is
preserve.key.case = on

% Pretty printing
print.align       = on        % align the '=' signs
print.line.length = 80        % wrap lines at ~80 chars

% Keep comments
pass.comments     = on

% -------- Remove noisy/sync fields (add more if needed) --------
delete.field = {abstract}
delete.field = {keywords}
delete.field = {language}
delete.field = {file}
delete.field = {urldate}
delete.field = {archivePrefix}
delete.field = {primaryClass}
delete.field = {eprintclass}
delete.field = {eprinttype}
delete.field = {pdf}
delete.field = {note}
delete.field = {annote}
% delete.field = {isbn}
% delete.field = {issn}
% delete.field = {shorttitle}
% ---------------------------------------------------------------
```

Figure 1: bibtool configuration file **paperpile_strip.rsc**.

```powershell
# Paths
$src = "G:\My Drive\paperpile.bib"   # Paperpile's synced file on the virtual G: drive
$rsc = "/mnt/c/Users/boris/Documents/paperpile_strip.rsc"
$dst = "C:\Users\boris\Documents\paperpile_cleaned.bib"

# Basic checks
if (-not (Test-Path $src)) {
  Write-Error "Source not found: $src"
  exit 1
}

# Clean: Windows → WSL (bibtool) UTF-8
Get-Content -Raw $src -Encoding UTF8 `
| wsl bibtool -r $rsc `
| Out-File -FilePath $dst -Encoding utf8

Write-Host "Cleaned, $dst"
```

Figure 2: Standalone PowerShell curation script **paperpile_clean.ps1**.

```bat
@echo off
powershell -NoProfile -ExecutionPolicy Bypass -File "C:\Users\boris\Documents\paperpile_clean.ps1"
pause
```

Figure 3: Double-click PowerShell curation script **paperpile_clean_run.bat**.