Using Geany as a Typst Editor

Last Updated: 3 Aug. 2026

Contents

Abstract

The general consensus is that the official Typst web app and/or VSCode (VSCodium) plus the Tinymist extension offer the smoothest writing experience in Typst.[1] However, if you have a local Typst installation, you may use other editors if you wish. One alternative is vim with the Typst plugin. I will discuss another option in this tutorial: Geany in conjunction with the Typst CLI (Command-Line Interface).

Geany is a popular, versatile text editor and IDE (Integrated Development Environment) based on the GTK3 toolkit. Geany does not have native support for Typst files at the moment.[2] Nevertheless, the procedure described below offers a writing environment with less distractions; some users might find that appealing.

This method uses Geany along with the Typst CLI, and a separate PDF viewer (as the external Preview app), to edit your local Typst files. The preview is asynchronous, i.e., not simultaneous but after you make an edit in Geany and save your file, the PDF will auto-update.

If you make a syntax error in your Typst source file, the PDF will not compile. You can then check output in the Terminal (where the Typst CLI is running) for detailed error messages about the problem and how to fix it.

1. Prerequisites

- A local installation of Typst with its pre-built binary. (You should download and install the most recent version, 0.15.1, released on July 17, 2026.) This tutorial explains how to install Typst locally on your PC, without needing to rely on Typst's web app.
- Geany[3]
- A PDF viewer (such as Document Viewer [aka Xreader], Evince, Atril, etc.)

[1] VSCodium with the Tinymist extension bundles everything you need, and it is updated frequently. It includes instant preview, monitors your source file for changes and a jump-to-location feature (which allows you to jump between the Typst source file and the corresponding location in the Preview pane).

[2] User `elextr` on GitHub explained that file types for new programming languages are not added to Geany by plugins; rather, they are built in. Although it is possible to customize a file type if an existing markup is "close enough" to the syntax, but for a markup language with its own syntax — like Typst — such customization is difficult and probably unlikely.

[3] I have been using ver. 2.0, installed via Mint's Software Manager. A Flatpak version (2.1.0) is also available but I have not experimented with it.

Tip:

This procedure was tested within Mint 22.3 ("Zena") Cinnamon. However, it should work identically with the Xfce and MATE editions.

2. Steps to Follow

- Generate a PDF file by opening your Typst source file in a Terminal, and issuing the command:

```
$ typst compile your-filename.typ
```

- Leave the Terminal open after the PDF is compiled.
- Open the Typst-generated PDF with your favorite PDF Viewer (Document Viewer, Evince, Atril).
- Leave the PDF Viewer open on your desktop.
- Open your Typst source file in Geany.

Optional: If you wish to add a bit of color syntax highlighting to the displayed source file, you can set the document's filetype to **`Markdown source file`**. The syntax highlighting is minimal compared to the detailed highlighting that you see in VSCodium; however, it adds some welcome visual contrast to the Typst source file (*personal opinion*).

To do this, from the `Document` menu > choose `Set Filetype` > then select `Markup Languages` > and click the radio button beside `Markdown source file`.

(You may also wish to enable line wrapping by checking/ticking the " `Line Wrapping` " option under Geany's `Document` menu.)

- Go back to your open Terminal and issue the command:

```
$ typst watch your-filename.typ
```

You should see output similar to this:

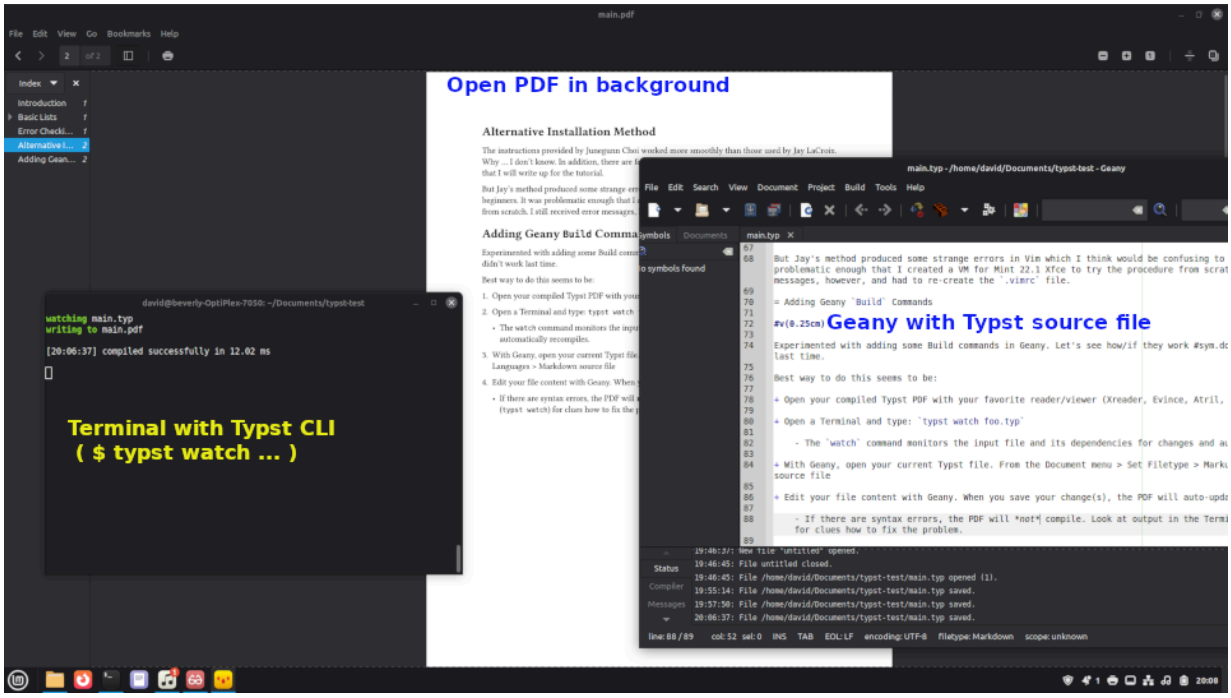
```
watching your-filename.typ
writing to your-filename.pdf

[18:55:54] compiled successfully in 5.59 ms
```

The `watch` command monitors the input file and its dependencies for changes and automatically recompiles to PDF.

- Continue to leave your Terminal open.
- When you make an edit in Geany and save your file, the PDF will auto-update.

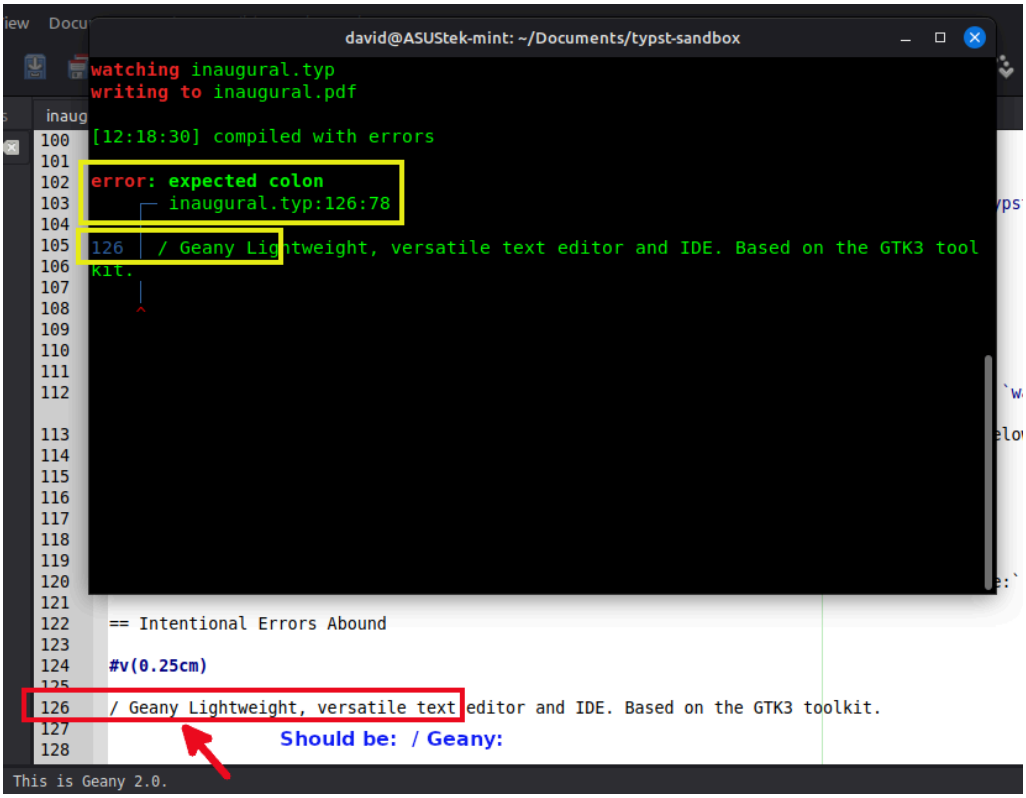
The screenshot below gives you an idea of how this arrangement looks. (To reduce clutter, you can minimize windows as needed.)



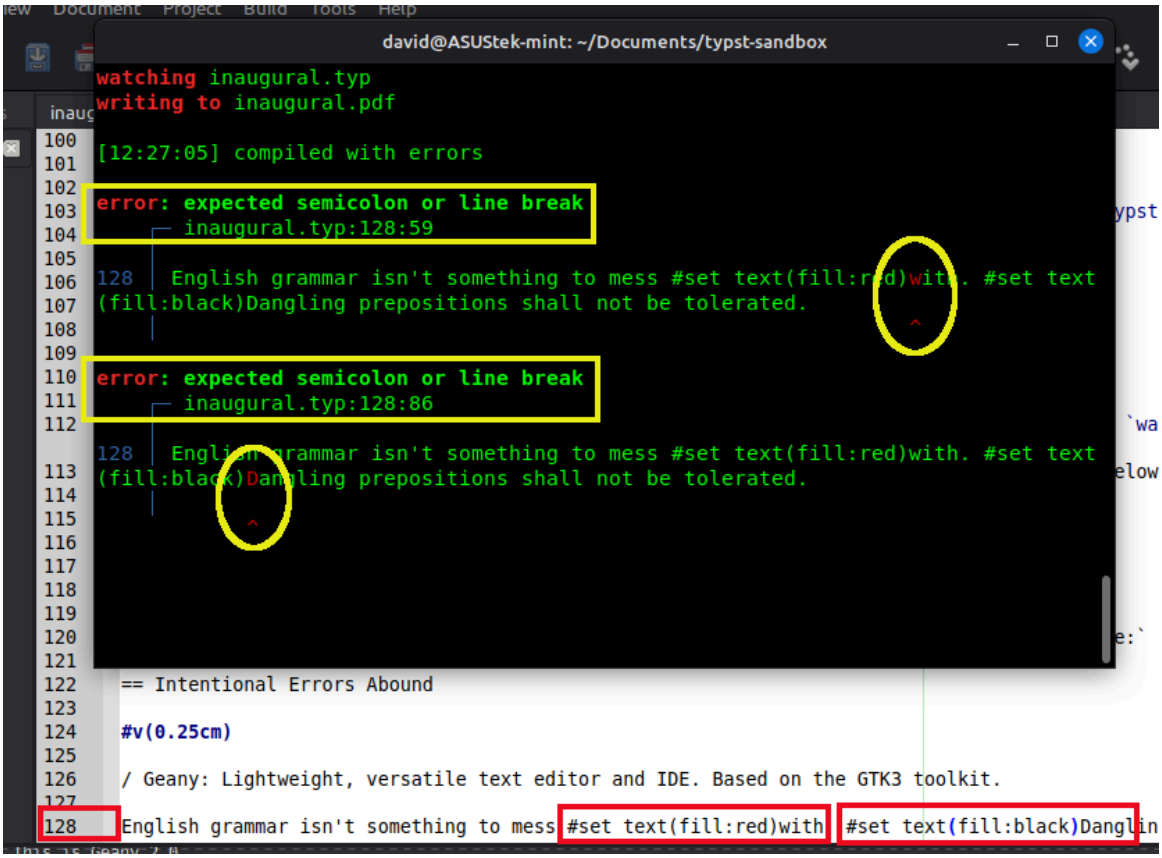
2.1. Dealing with Errors

If you make a syntax error in your Typst source file, the PDF will not compile. You can then check output in the Terminal (where the Typst CLI is running) for detailed error messages about the problem and how to fix it.

In the example below, I intentionally omitted the required colon in the term list. (It should be "`/ Geany:` " rather than "`/ Geany` "). The Terminal output displays an error message "`expected colon`" as well as indicating the line number (126) of the error:

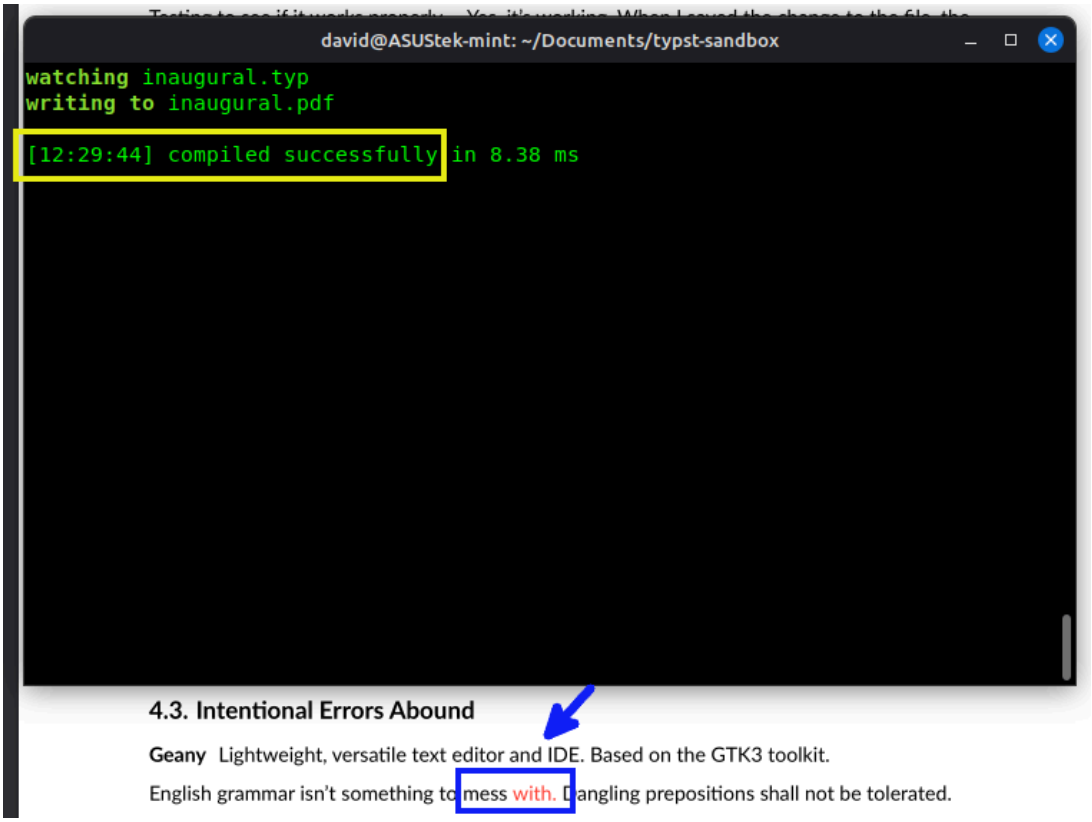


A more complicated example shows that error messages sometimes flag the precise location within a line. The `#set` command to change the text color to red should end with a semicolon in this instance: `#set text(fill:red);with` (instead of the incorrect "`#set text(fill:red)with`").[4] The Terminal output helpfully displays the location of the missing semicolon:



Once you have corrected these two mistakes and saved your change, the Typst source file will compile successfully to PDF:

[4] Another `#set` command switches the text color back to black.



3. Using Geany's `Build` Commands (*Optional*)

If you like this approach well enough to consider using Geany as a Typst editor on a long-term basis, it is possible to configure Geany's `Build` commands for Typst.

Note:

The basic functionality is the same as that described in Section 2. The initial configuration takes some time, but you only need to do it once. Afterwards, you will gain a smidgen of efficiency.

When the configuration is finished and you are using Geany + Typst, you will still have three windows open simultaneously: the Geany editor, a Terminal window, and a PDF viewer.

3.1. Add `typst` Directory to `~/.profile`

Why this step is necessary: My tutorial recommends that you add the Typst binary to your `$PATH` by placing it in your `~/.bashrc` file. However, Geany typically runs commands in a login or non-interactive shell; therefore, environment variables set in `~/.bashrc` do not automatically apply in Geany's terminal.[5]

We must configure Geany so that it can find the Typst environment variable. The first step is to add the `typst` directory to the `~/.profile` file.

- Open a Terminal and issue the command below in your `/home` directory to edit the `.profile` file:

(I chose the nano editor but you may use whatever your preferred text editor is — xed [aka Text Editor], vim, Emacs, etc.)

- Type the command:

[5]The `~/.bashrc` file is only for interactive shells (like in a terminal emulator, i.e., `GNOME-terminal` [as used in Mint Cinnamon], `Xfce4-terminal` or `MATE-terminal`).

```
$ nano ~/.profile
```

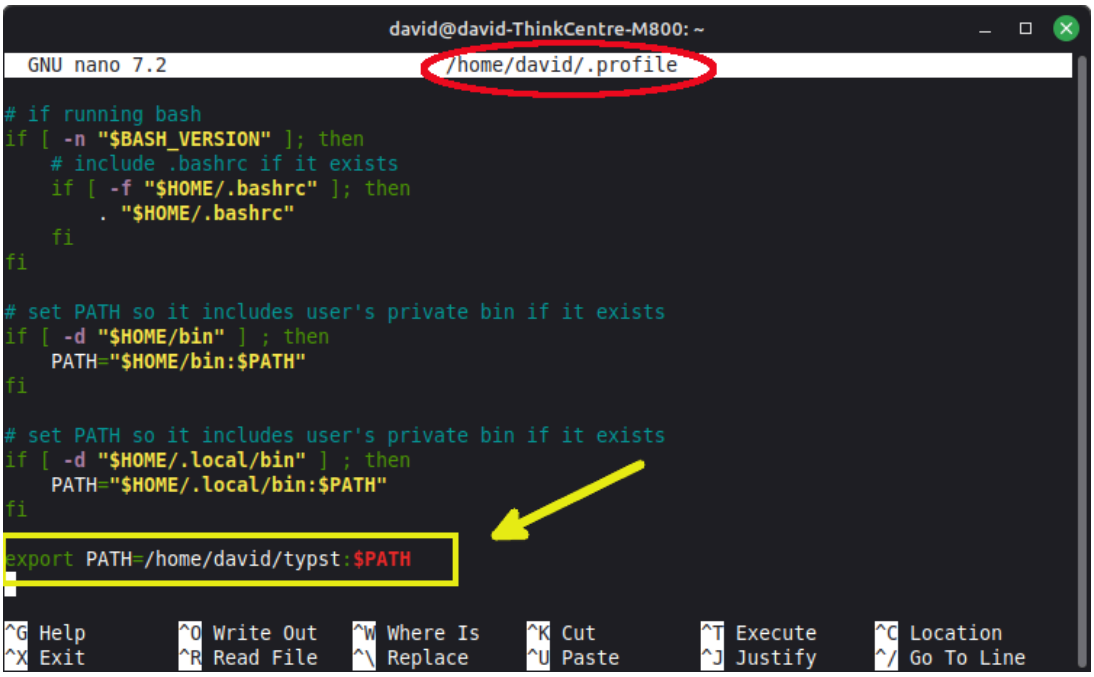
Do not forget to type the initial period (“ . “) before `profile`

- Add the line

```
export PATH=/home/david/typst:$PATH
```

at the bottom of the existing `~/.profile` file:

(Change “`david`” to whatever the username is on your system.)

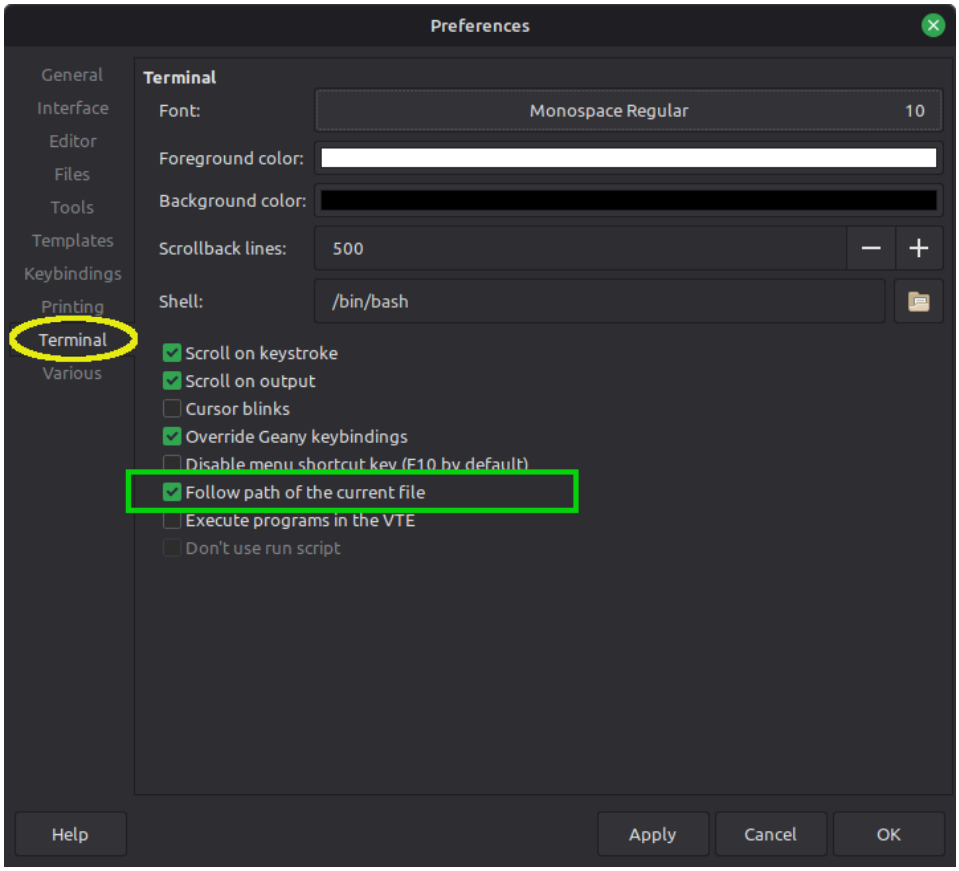


- Save the file.
- I logged out of my current session, then logged back in to make sure the change took effect.[6]

3.2. Geany Setting Re: Terminal

From Geany’s **`Edit`** menu > choose **`Preferences`** > then select **`Terminal`**. Check/tick the option “`Follow path of the current file`” to enable it:

[6] After testing on my personal Mint PCs, I have not experienced any conflicts/glitches due to the Typst environment variable being in two different places, i.e., `~/.bashrc` and `~/.profile`. Typst + VSCodium continues to work normally, as does the `typst.vim` plugin.



3.3. Set `Build` Commands

Next, we need to set up two `Build` commands for Typst. From Geany's **`Build`** menu > choose **`Set Build Commands`**.

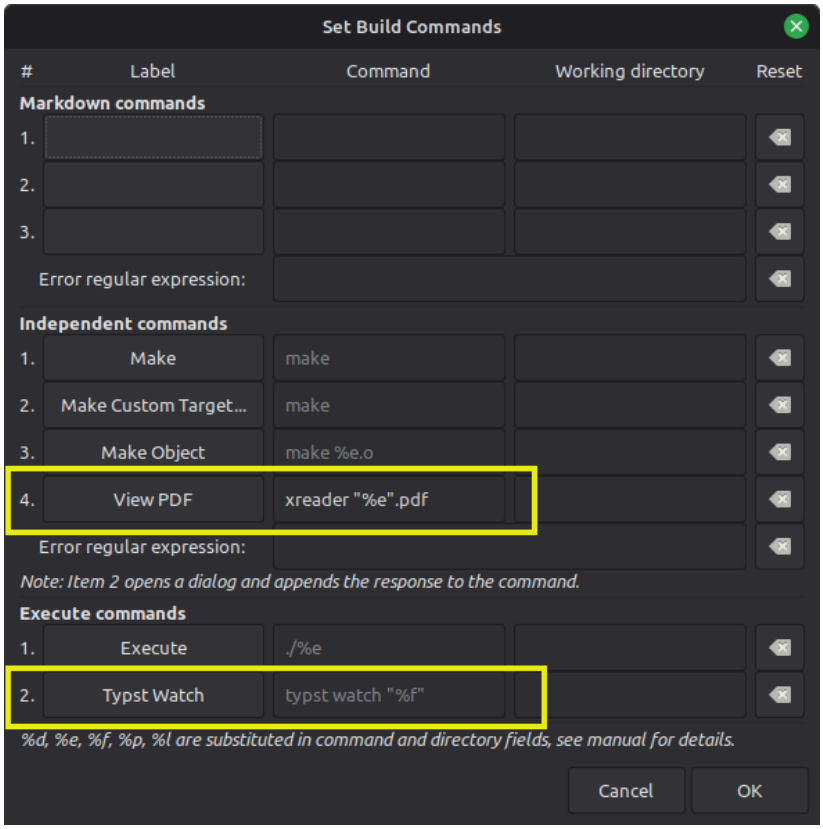
Under the **`Independent commands`** section, add the Label " `View PDF` " and the command

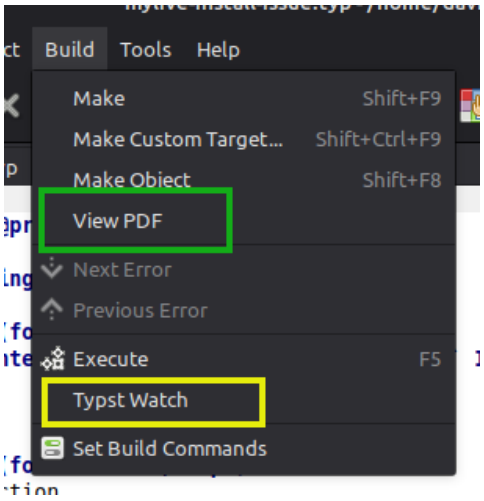
```
xreader "%e".pdf
```

Under the **`Execute commands`** section, add the label " `Typst Watch` " and the command

```
typst watch "%f"
```

The results should look like this:





3.4. Post-Configuration Steps to Follow

- Open your Typst source file in Geany.
- Set the document's filetype to **`Markdown source file`**. To do this, from the **`Document`** menu > choose **`Set Filetype`** > then select **`Markup Languages`** > and click the radio button beside "`Markdown source file`".

(You may also wish to enable line wrapping by checking/ticking the "`Line Wrapping`" option under Geany's **`Document`** menu.)

- From the **`Build`** menu > click on the **`Typst Watch`** command that you just added.

A Terminal will open in a separate window and you should see output similar to this:

```
watching your-filename.typ
writing to your-filename.pdf

[18:55:54] compiled successfully in 5.59 ms
```

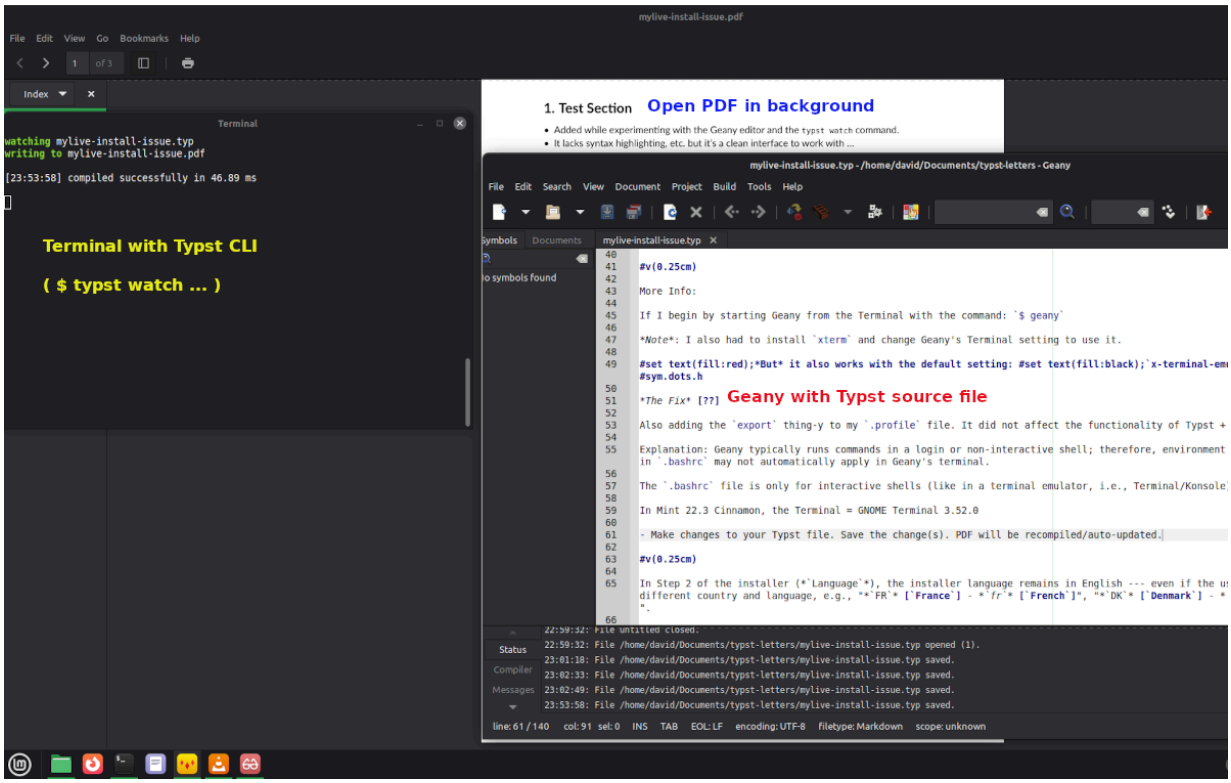
The `watch` command monitors the input file and its dependencies for changes and automatically recompiles to PDF.

- Continue to leave your Terminal open.
- From the **`Build`** menu > click on the **`View PDF`** command that you just added.

The PDF Viewer that you specified in the command will open in a separate window. (This example uses Document Viewer [aka Xreader].)

- Continue to leave your PDF Viewer open.
- When you make an edit in Geany and save your file, the PDF will auto-update.

The screenshot below gives you an idea of how this arrangement looks. (To reduce clutter, you can minimize windows as needed.)



- If you make a syntax error in your Typst file, the PDF will not compile. See the instructions in Section 2.1 for how to proceed.

4. Conclusion

Although editing Typst files with Geany sounds cumbersome and lacks real-time preview, it works well. Some users may find it less distracting than working with VS Code/VSCodium.

This approach has the advantage that no external utilities — such as the Tinymist extension — are needed; in addition, people who feel intimidated by `vi(m)` may appreciate a user-friendly, GUI text editor such as Geany.