

Some principles for the user experience of a \LaTeX editor

Sébastien WILMET

First version: August 2012

Last update: March 2022

Introduction

Writing a \LaTeX document can be done in different ways. Some people prefer an application like LyX¹, which hides the \LaTeX code and uses a more sophisticated GUI.

Other people prefer to work directly on the \LaTeX code. GNOME \LaTeX ² has chosen this direction. In this context, let's see, with two examples, some of the principles behind GNOME \LaTeX to offer a good user experience.

1 Example: inserting a figure

To insert a figure, an application like LibreOffice uses a wizard, so the user can choose an image, its size, the caption, etc.

For a \LaTeX application, we could imagine that the corresponding \LaTeX code is generated and inserted in the `*.tex` file. Nice, isn't it?

There is a little problem though: if the user doesn't understand the code, how does he or she modify it afterwards to change an option? A good reaction is to look at the documentation to understand what happens. But a quicker solution is perhaps to re-run the wizard and refill the pieces of information and modify the option.

GNOME \LaTeX prefers to avoid such wizards³. Let's see some options for offering a different user experience.

2 Solutions

For the principles that we follow, a wizard is not a perfect solution. The root of the problem is that looking at the \LaTeX documentation can take some time.

The following sub-sections look at different and complementary solutions.

¹<https://www.lyx.org/>

²<https://wiki.gnome.org/Apps/GNOME-LaTeX>

³A good way to force users to learn \LaTeX .

```
8 \begin{figure}
9   \begin{center}
10  \includegraphics
11  \caption{}
12  \label{fig:}
13  \end{center}
14 \end{figure}
```

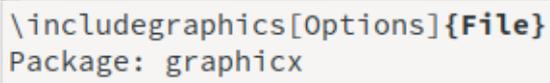


Figure 1: Calltip showing the prototype of a \LaTeX command.

2.1 Good completion

One of the GNOME \LaTeX features is to offer a good completion of \LaTeX commands and their arguments (although there is always room for improvements).

Another related thing is that when the cursor is inside a command argument, and if no completion is available, then a “calltip” is displayed with the prototype of the \LaTeX tag. See Figure 1.

2.2 Inserting an initial snippet of code

An additional solution is a way to easily insert the required commands for doing common actions like creating a figure. Once the snippet of code is inserted, the user can fill in some arguments and other missing pieces of information (such as the path to an image file).

In GNOME \LaTeX there is a toolbar and menus for common actions.

Note that this is different than a wizard, because the user is supposed to understand the initial snippet of code, and thus is able to modify the arguments afterwards, without the need to click again on the toolbar item.

2.3 Jumping to a more complete API reference

When the command completion feature is not sufficient in order to know what to do, then we need to look at some documentation.

The idea is to easily access the relevant and comprehensive documentation related to a \LaTeX command. From the text editor, when the cursor is placed on a \LaTeX command, a keyboard shortcut (or another means) would “jump” to its corresponding documentation (opened in another application or another window).

Note that this is different than showing the documentation of the \LaTeX command in a popup window (via the completion system). Some text editors show the documentation that way, a little like the calltip of Figure 1 but with (much) more information. It’s a question of preference.

This feature is not implemented in GNOME \LaTeX , but some of the “ingredients” are already there: for example with the Devhelp API browser⁴, which permits to easily search and navigate through a reference documentation, and that can be invoked from other applications or integrated into an IDE.

⁴<https://wiki.gnome.org/Apps/Devhelp>

3 Another example: creating a new document

Creating a new document can also be made via a wizard. We follow the step-by-step instructions and we choose:

- The document type (an article, a report, slides, etc).
- The title.
- The author(s).
- And so on.

But there is the same problem as for figures' insertion.

In GNOME \LaTeX , creating a new document is done via a template. There are some default basic templates, and personal templates can be created. This is similar to what is described in section 2.2: the template content is simply copied.

Most users will have their own templates, possibly managed externally with a version control system like Git, and they just make a copy of the file to start a new document.

A recommendation could be to create a big personal template with all the stuff that you could possibly need for a new document. And then, you remove or comment out what you don't need.

Another way is to have re-usable portions (as separate files) that you include with a control sequence like `\input`.

Conclusion

To summarize, the idea of GNOME \LaTeX is to always deal directly with the \LaTeX code, while simplifying as most as possible the writing of this \LaTeX code. The users don't need to be \TeX gurus, but they should understand what happens.

Revisions:

- August 2012: initial version.
- March 2022: the article has been re-worked, with new or improved content. New section: *Jumping to a more complete API reference.*