`latex.sl`: an enhanced LaTeX mode for Jed

Version 1.3.0
Guido “goccia” Gonzato, Ph. D.
guido.gonzato@univr.it

Università di Verona (Italy)
December 3, 2002

Abstract

The Jed editor provides support for several programming languages and text formats, among which LaTeX. However, the author feels that the default LaTeX mode is too simple, and that it could be improved in many ways.

This document describes an enhanced LaTeX mode. It aims at making the process of writing LaTeX documents an easy and pleasant task. Vaguely inspired by Emacs' AUC TeX, `latex.sl` provides the user with menus, templates, many utilities, and integrated conversion and previewing of LaTeX documents.

Contents

1 Introduction

Many TEXnicians use `emacs`: a great[1] program, but I don't like it very much. `emacs` has very good LATEX support (AUC TEX), but it's also unacceptably bloated and slow.

I use a fast and compact `emacs` clone called Jed, `http://www.jedsoft.org`. It can be customised and extended using S-Lang, an easy-to-learn language similar to C. Jed provides support for a variety of programming modes, with syntax highlighting and other facilities. LATEX support in Jed is provided by `latex.sl` and other S-Lang files included in the distribution.

I feel that the default implementation of `latex.sl` leaves much to be desired. When I write code or documentation, I follow the so-called "Goccia's Rules":

1. the program should be as helpful as possible, but
2. it must not stand in the way; therefore,
3. it should be fully customisable, and
4. it should be usable by geeks and newbies alike.

In my opinion, `latex.sl` breaks Goccia's Rules 1, 3, and 4. It should attempt to be more useful.

Jøergen Larsen, `jl@dirac.ruc.dk`, wrote a much better implementation available from `http://dirac.ruc.dk/~jl/jed/`. I initially added an extensive menu system to his mode, but as its development proceeded I lost sync with it.

[1] or rather, big?

I eventually decided to roll my own LaTeX mode. It was developed on Linux, but it also works and other Unix systems and—woops—Windows. It was tested on a RedHat 7.3 GNU/Linux machine with teTeX, and under Windows 98 with MiKTeX.

The latest version of `latex.sl` is 1.3.0, it works with Jed 0.99.15 upwards, and is available from http://profs.sci.univr.it/~gonzato/jed.

2 Installation

I assume that you have a working installation of Jed. First of all, make a backup copy `JED_ROOT/lib/latex.sl`. In the unlikely event you don't like the new LaTeX mode, you'll restore the original file.

Copy `latex.sl` and `latex.hlp` to `JED_ROOT/lib`, then add these lines to your `.jedrc`:

```
add_mode_for_extension ("latex", "tex");
enable_dfa_syntax_for_mode ("LaTeX");
```

For a system-wide installation, edit the lines above in the file `JED_ROOT/-lib/jed.rc`.

You'll also want to create the DFA [2] cache table. As root, add `latex.sl` to the list in the file `preparse.sl`, then run the command:

```
jed -batch -n -l preparse
```

Caveat: by default, LaTeX mode is incompatible with folding mode because of clashing `Ctrl-Cf` key binding. Unless you're prepared to change the key bindings in `folding.sl`, you can solve the problem setting this variable in your `.jedrc`:

```
variable LaTeX_Font_Key = "n";
```

which will make all font operations start with `Ctrl-Cn` instead of `Ctrl-Cf`.

From now on, I shall assume that you run Jed in Emacs emulation mode. All key bindings will start in `Ctrl-C`; users who prefer IDE mode will use `Ctrl-Z` instead. I also remind you that all operations can be interrupted with `Ctrl-G`.

[2] the regular expression-based highlighting scheme.

2.1 Customisation

You customise `latex.sl` changing the value of its variables. Insert lines like the following in your `.jedrc`. The values shown are the default:

```
variable LaTeX_Default_Output =
               "dvi"; % or: "ps", "dvipdf", "pdf"
variable LaTeX_Indent = 2;
variable LaTeX_Article_Default_Options = "a4paper,12pt";
variable LaTeX_Book_Default_Options = "twoside,11pt";
variable LaTeX_Letter_Default_Options = "a4paper,12pt";
variable LaTeX_Report_Default_Options = "twoside,12pt";
variable LaTeX_Slides_Default_Options = "a4paper,landscape";
variable LaTeX_Prosper_Default_Options =
               "pdf,slideColor,colorBG,azure";
variable LaTeX_Default_Language = "italian,english" % for Babel
#ifdef WIN32
variable LaTeX_View_Dvi_Command = "yap";
variable LaTeX_View_Ps_Command = "gsview32";
variable LaTeX_View_Pdf_Command = "gsview32";
variable LaTeX_Print_Command = "gsview32";
#else
variable LaTeX_View_Dvi_Command = "xdvi";
variable LaTeX_View_Ps_Command = "gv -watch";
variable LaTeX_View_Pdf_Command = "acroread";
variable LaTeX_Print_Command = "lpr";
#endif
```

Windows users will have to make sure that all helper programs are in the PATH.

You can further customise LaTeX mode adding a `latex_mode_hook` function in `.jedrc`. For example, I want accented letters to self insert as the right TeX sequence:

```
define latex_mode_hook ()
{
  set_abbrev_mode (1);
  if ( () = abbrev_table_p ("LaTeX") )
    use_abbrev_table ("LaTeX");
#ifdef WIN32
  % prevent clash with movement keys
  undefinekey ("àà", "LaTeX-Mode");
```

```
  definekey (" \\‘a", "àà", "LaTeX-Mode");
#else
  local_setkey (" \\‘a",     "à");
#endif
  local_setkey (" \\’e",     "é");
  local_setkey (" \\‘e",     "è");
  local_setkey (" \\‘\\i{}", "ì");
  local_setkey (" \\‘o",     "ò");
  local_setkey (" \\‘u",     "ù");
}
```

3 Editing LaTeX Documents

3.1 Getting Started

Start a new document, say `newfile.tex`. The `.tex` extension will automatically activate LaTeX mode, which can also be started on any buffer typing `Alt-X latex_mode`. See also the Buffers/Select Mode menu.

The Mode menu contains many entries and sub-menus, and it also indicates the key bindings when available. You'll want to browse through the menus, especially if you're not a LaTeX expert.

Now select Mode/Templates/Article to create an article template. Fill it with some text: add a couple of sections, some environments, change the fonts, and so on. Let `latex.sl` help you: use the menus, but try and memorise the key bindings. Note the syntax highlighting of keywords and other LaTeX elements.

When you're finished, convert your document using Mode/Compose. `latex` will be run on your document, and if no errors occur the file `newfile.dvi` will be created. If errors do occur, please read Section 3.10.

Now select Mode/View to preview your document. Voilà, all done, without ever using the command line!

3.2 Main Features of LaTeX Mode

As of version 1.3.0, `latex.sl` has the following features:

- thoroughly menu-driven
- syntax highlighting
- full integration with external programs

- templates
- document outline
- integrated debugging
- symbol completion
- many commands to write LaTeX sources in less time.

Available key bindings are shown in the Mode menu entries, and were defined in a hopefully intuitive and consistent manner. Some examples are `Ctrl-Css` for Mode/Sections/\section, `Ctrl-Cec` for the `center` environment, and so on.

Thanks to this arrangement, this guide does not contain tables of key bindings. It's much simpler to read the menus.

There is another important feature you'll want to use. Most commands are *region or word aware*. For example, if a region is defined and you select Mode/Environments/center, that region will be included in a `center` environment. If the cursor is positioned on a word and you select Mode/Font/\emph, the word will be included in a `\emph` command.

3.3 Useful Keys

In addition to all key bindings listed in the Mode menus, the following keys are also useful:

- `Ctrl-C` followed by one of the special characters `& $ # % _ ^ ~` \ will insert `\& \$ \# \% \_ \textasciicircum \textasciitilde \textbackslash`
- `Ctrl-C` followed by { will insert a couple of curly braces {}. This command is region or word aware.
- the single and double quote characters will self insert as smart quotes. That is, type `"word"` to obtain ` ``word'' `
- three dots in a row will insert `\ldots`.

3.4 Symbol Completion

This feature was borrowed by the old `latex.sl`. If you start typing a LaTeX keyword and type `Alt-TAB`, the keyword will self complete. Type `Alt-TAB` again to cycle through possible completions. For example, if you type `\bib` you'll cycle through `\bibitem[]{}`, `\bibliograpy{}`, and `\bibliograpy-style{}`.

Under X11, the window manager may reserve `Alt-TAB` for itself. MS Windows uses this key binding to switch between tasks. In that case, use `ESC-TAB`.

3.5 Indentation

`latex.sl` uses a sort of 'preventive indentation' scheme. Normally, a new line starts at the same column as the line above, but lines within environments are indented. For example, start a `center` environment with `Ctrl-Cec`: the text will be indented by the amount specified by the variable `LaTeX_Indent`. The default is 2 columns, and can be changed as seen in Section 2.1.

If you wish to change the amount of indentation on the fly, select System/S-Lang Command and type at the prompt:

```
S-Lang> LaTeX_Indent = 4;
```

3.6 Environments

Environments can be entered via the Mode/Environments menu. This method will provide the right indentation for both environment and the text it surrounds.

Another way to start a new environment is typing `Ctrl-C[`. You'll be prompted for the environment name, and the `\begin` line will be inserted. When you're done, close the environment with `Ctrl-C]`.

3.6.1 Itemize

In `itemize` or `enumerate` environments you can use `Ctrl-C<return>` to start a new `\item` line.

3.6.2 Tables

`latex.sl` makes it easier to write tables. When you start a `table` or `tabular` environment, you'll be prompted for the number of columns, then a table template will be inserted.

The Mode/Environments/table row menu will insert a line containing the right number of & delimiters, and terminated by \\. Just fill in the blanks.

3.7 Templates

Templates for standard LaTeX document classes: `article`, `book`, `letter`, `report`, `slides`, are provided. In addition, templates for notices and for Prosper presentations (http://prosper.sourceforge.net) have been added for your convenience. The latter lets you write a presentation in minutes!

The most commonly used packages are also listed in the Mode/Templates/-Packages sub-menu.

3.8 Using a Master File

When you work on complex documents, you can set a buffer as the 'master file'. That means that although you may be working on several LaTeX files at the same time, all operations of conversion, previewing etc. will be performed on the master file. This is useful, for example, when you have a main file that includes several parts.

If you disable the master file, all operations will be performed on the current buffer.

3.9 Document Outline

Use this feature to navigate through complex documents. The `*Outline*` buffer shows the document structure, listing all `\parts`, `\sections`, etc. and the line where they appear.

Press <`enter`> or double click on a section to move to the relative line in the LaTeX buffer, or 'q' to quit.

3.10 Composing and Debugging

LaTeX documents are converted using four output profiles: dvi, ps, pdf, and dvipdf. The final output is, respectively: a `.dvi` file, a `.ps` file created with `dvips`, a `.pdf` file created with `dvipdf`, and a `.pdf` file created with `pdflatex`. The default output profile is dvi.

Select Mode/Compose to convert the buffer using the current output profile. A 'beep' will notify you of warnings, e.g.

```
Font Warning.
```

If errors are detected, an error message appears:

```
Error(s) found! Press 'L' to switch to the log.
```

Type 'l' to inspect the LaTeX log, or any other key to continue.

If you choose to examine the log, a new window pops up and the line that explains the problem is highlighted. Now type 'g' (go) to move to the line in the LaTeX source that caused the error, or `Ctrl-C'` to move to the next error, or 'q' to quit the log.

If the conversion succeeded, the following Mode/View command will start the appropriate viewer.

3.11 Maths

It's virtually impossible to list all mathematical symbols supported by LaTeX in a menu. That said, most symbols are available under Mode/Math.

Greek letters are obtained typing `Ctrl-Cm` and a letter; the equivalent Greek letter, if available, will be inserted. For instance, `Ctrl-Cma` inserts `\alpha`, and `Ctrl-CmG` inserts `\Gamma`.

All available symbols are provided by the file `ltx-math.sl`, included in the Jed distribution. In addition, `Ctrl-C` followed by an arrow key produces ono of these: `\leftarrow` `\rightarrow` `\uparrow` `\downarrow`.

3.12 Notes on BiBTeX

LaTeX mode doesn't provide any support for writing BiBTeX files. Jed has an excellent BiBTeX mode already. Edit a file with the `.bib` extension to turn BiBTeX mode on.

4 Known Problems

The current implementation of syntax highlighting could be improved. The most apparent problem is that text enclosed in curly braces isn't always highlighted correctly. This is a limitation of the DFA code in Jed.

The documentation (this guide) is probably a bit too concise.

5 The End

I use `latex.sl` every day, and I think it's very complete. However, I'll be glad to receive suggestions and requests from you. If you find a bug or would like to see a new feature added, please feel free to contact me.

Enjoy! `=8-)`