LaTeX4Jed: an enhanced LaTeX mode for Jed

Version 1.4.4
Guido ‘goccia’ Gonzato, Ph.D.
guido,dot,gonzato,at,univr,dot,it

Università di Verona (Italy)
Facoltà di Scienze MM. FF. NN.
February 20, 2004

Abstract

The Jed editor supports several programming languages and text formats, including LaTeX. However, the author of this document feels that the default library file `latex.sl` is too simple, and that it could be improved in many ways.

This document describes an enhanced `latex.sl`, which aims at making the process of writing LaTeX documents an easy and pleasant task. Inspired by Emacs’ AUC-TeX, the new `latex.sl` provides the user with menus, templates, many utilities, and integrated conversion and previewing of LaTeX documents.

Contents

List of Tables

1 Introduction

Many TeXnicians use `emacs` or `vim`. Both are great programs with excellent LaTeX support: AUC-TeX.

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

I feel that the default implementation of `latex.sl` is far behind AUC-TeX, and that it could be improved in many ways. When I write code or documentation, I discipline myself to 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 experts and newbies alike.

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

Jørgen Larsen <jl@dirac.ruc.dk> wrote a much better implementation, available from http://dirac.ruc.dk/~jl/jed/. I initially added menus to his mode, but as its development proceeded I lost sync with it. Many ideas of my implementation were taken from Jørgen's; I suggest that you check it out.

I eventually decided to roll my own `latex.sl`: from now on, LaTeX4Jed. It was developed on Linux, but it also works and other Unix systems and even Windows. It was tested on a RedHat 7.3 GNU/Linux box with teTeX, and under Windows 98 with MiKTeX.

The latest version of LaTeX4Jed is 1.4.4, it works with Jed 0.99.15 upwards, and is available from CTAN mirrors; for example, http://www.ctan.org/tex-archive/support/jed.

2 Installing LaTeX4Jed

In the following, I assume that you have a working installation of Jed.

2.1 As Superuser

First of all, make a backup copy of `$JED_ROOT/lib/latex.sl`. In the event you don't like LaTeX4Jed[1], you'll be able to restore the original file. However, I'm pretty sure you will not roll back...

Copy `latex.sl` and `latex.hlp` to `$JED_ROOT/lib`. Optionally, create `$JED_ROOT/lib/latex` and copy the modules `prosper.sl` and `notice.sl` there. More details on modules are given in Section 3.10.

Finally, 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`.

[1] unlikely :-)

I *recommend* that you create the DFA[2] cache table. If you don't, the DFA table will be slowly created each time you start Jed on a LaTeX document. It could take *minutes*!

As root, add `latex.sl` to the list in the file `preparse.sl`, then run the command:

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

This step may be required each time you update LaTeX4Jed to a new version.

2.2 As Normal User

If you don't have one, create a personal Jed library directory; for example,

```
$ mkdir /home/myself/myjedlib
```

then copy `latex.sl` to this directory and add these lines to your `.jedrc`:

```
variable Jed_Home_Library = "/home/myself/myjedlib";
set_jed_library_path (Jed_Home_Library + "," +
                      get_jed_library_path());
Jed_Highlight_Cache_Dir = Jed_Home_Library;
add_mode_for_extension ("latex", "tex");
enable_dfa_syntax_for_mode ("LaTeX");
```

Jed will now find and use the new `latex.sl` instead of the default library file. The highlight cache will be built automatically the first time you use LaTeX4Jed; a Jed update will not affect your private installation.

2.3 Note for Windows Users

LaTeX4Jed can be used with Jed for the DOS console, but it gives its best with `wjed`. However, as of Jed 0.99.16 `wjed` will not use standard menus unless you modify `$JED_ROOT/lib/os.sl`. Lines 33–34 read:

```
%. "menus" evalfile pop          %  Uncomment to enable text menus
. "wmenu.sl" evalfile pop      %  Uncomment to enable GUI menus
```

It should be the opposite: uncomment out line 33 to enable text menus, and comment out line 34. Wjed will now show the Mode menu.

[2] the regular expression-based highlighting scheme.

2.4 Caveat

By default, LaTeX4Jed 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`.

2.5 Colours

Obviously, LaTeX4Jed looks better with `xjed` and `wjed`. Three nice colour schemes are provided: `elegant.sl`, `modern.sl`, and `night.sl`. If you wish, copy them to `JED_ROOT/lib/colors/Xjed/`. If you want or need to use plain console `jed`, add this line to your favourite colour scheme (e.g. `JED_ROOT/lib/colors/blue2.sl`):

```
set_color ("keyword2", "brightgreen", $2); % other keywords
```

I suggest that you insert these lines in `.jedrc`:

```
#ifdef WINGUI
set_color_scheme ("Xjed/night");
#elifdef XWINDOWS
set_color_scheme ("Xjed/modern");
#else
set_color_scheme ("blue2");
#endif
```

Hereafter, 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`.

2.6 Customisation

The user can customise LaTeX4Jed changing the value of its variables. Insert lines like the following in your `.jedrc`. Default values are shown:

```
variable LaTeX_Default_Output =
               "dvi"; % or: "ps", "eps", "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_Default_Language = "italian,english" % for Babel
custom_variable ("LaTeX_Rerun", "y"); % for xrefs
#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";
variable LaTeX_Clearup_Cmd, "del *.out *.aux *.lo? *.to?";
variable LaTeX_Modules_Dir, JED_ROOT + "\\lib\\latex\\";
#else
variable LaTeX_View_Dvi_Command = "xdvi";
variable LaTeX_View_Ps_Command = "gv -watch";
variable LaTeX_View_Pdf_Command = "xpdf -z width";
variable LaTeX_Print_Command = "lpr";
variable LaTeX_Clearup_Cmd = "/bin/rm -f *.out *.aux *.lo? *.to?";
variable LaTeX_Modules_Dir = JED_ROOT + "/lib/latex/";
#endif
```

Windows users must make sure that all auxiliary programs are in the PATH.

The name and purpose of these variables should be self-explanatory. For example, `LaTeX_Rerun` specifies whether (`pdf`)`latex` should be rerun to resolve all cross references. If you don't want to delete temporary files after the `latex` run(s), set `LaTeX_Clearup_Cmd` to “” (empty string).

You can further customise LaTeX4Jed adding a `latex_mode_hook` function to `.jedrc`. For instance, I want to type accented letters on my Italian keyboard, and get the right TeX sequence. So I added this function:

```
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");
}
```

Sometimes, the defaults you set in `.jedrc` are not appropriate for the current editing session. In that case, you can temporarily change the values of the variables using the Mode/Customise Defaults menu entry.

3 Editing LaTeX Documents

3.1 Main Features of LaTeX4Jed

As of version 1.4.4, LaTeX4Jed has the following features:

- thoroughly menu-driven
- advanced DFA syntax highlighting
- full integration with external programs
- templates
- modules
- document structure
- 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` (or `Ctrl-C Ctrl-S Ctrl-S`) for Mode/Sections/\section, `Ctrl-Cec` for the `center` environment, and so on.

Thanks to this arrangement, this guide contains only one table of key bindings: Table 1. In most cases, 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.

Keybinding	**Action**
Ctrl-C⟨Left⟩	go to previous paragraph
Ctrl-C⟨Right⟩	go to next paragraph
Ctrl-C&	\&
Ctrl-C$	\$
Ctrl-C#	\#
Ctrl-C%	\%
Ctrl-C(	\{
Ctrl-C)	\}
Ctrl-C_	_
Ctrl-C<	\textless{}
Ctrl-C>	\textgreater{}
Ctrl-C^	\textasciicircum{}
Ctrl-C~	\textasciitilde{}
Ctrl-C\	\textbackslash{}
Ctrl-C|	\textbar{}
Ctrl-C{	{}, region or word aware
' or "	smart quotes
Ctrl-C' or Ctrl-C"	smart quotes, region or word aware
...	{\ldots}
<-	{\leftarrow}
<=	{\Leftarrow}
< --	{\longleftarrow}
< ==	{\Longleftarrow}
< ->	{\leftrightarrow}
< =>	{\Leftrightarrow}
->	{\rightarrow}
=>	{\Rightarrow}
-->	{\longrightarrow}
==>	{\Longrightarrow}
|->	{\mapsto}
|-->	{\longmapsto}
Ctrl-Ce⟨TAB⟩	insert environment, with completion
Ctrl-Cf⟨TAB⟩	insert font, with completion
Ctrl-Ch	info help on word
Alt-⟨TAB⟩ or Alt-V	symbol completion
TAB	indent line
ESC 1⟨TAB⟩	unindent line
Ctrl-C[	start environment
Ctrl-C]	close environment
ESC 1⟨cmd⟩	command, long form
Ctrl-C⟨RET⟩	new \item line
F8	preview document
F9	run (pdf)latex

Table 1: Key bindings not available in menu entries.

3.2 Getting Started

Start a new document, say `newfile.tex`. The `.tex` extension will automatically invoke LaTeX4Jed, 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 LaTeX4Jed 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, typeset 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 4.

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

3.3 Directory Independence

You don't have to start Jed in the same directory where your LaTeX files are kept. When you begin editing a file, LaTeX4Jed takes note of its parent directory. All subsequent operations will be performed from there. As a result, if your file includes external files of figures, everything will work as expected.

3.4 Editing Compressed Files

Jed has the ability to edit compressed files. To enable this feature, insert this line in `.jedrc` or `jed.rc`:

```
auto_compression_mode ();
```

LaTeX4Jed works on compressed files with no problems, but obviously it cannot deal with compressed external figures or `\include`d parts. Editing compressed files may be useful on low-resource machines.

3.5 LaTeX Info Help

On Linux and other Unix-like systems, the LaTeX Info pages are surely installed. Check out typing `info latex` at the shell prompt.

LaTeX4Jed integrates with the Info pages. If you wish to get some help on a LaTeX command or keyword, place the cursor on that word and press `Ctrl-Ch`. If the word is covered in the Info pages, the relevant info page will be loaded. Press ⟨`q`⟩ to quit the Info page.

3.6 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, after typing `\bib` you'll cycle through `\bibitem[]{}`, `\bibliograpy{}`, and `\bibliograpystyle{}`.

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

3.7 Indentation

LaTeX4Jed uses a kind of 'preventive indentation' scheme: it tries to save you from hitting the ⟨`TAB`⟩ key to indent the line.

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, but it can be changed as seen in Section 2.6.

If you want to force an indented line, press the ⟨`TAB`⟩ key. This will simply indent the line by `LaTeX_Indent`, regardless of the current environment. `ESC 1`⟨`TAB`⟩ will unindent a line, provided that the line starts with the adequate number of spaces.

Finally, if you're editing messy LaTeX sources[3], you can indent environments with `Ctrl-CeD`.

3.8 Environments

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

[3] not written with Jed, obviously ;-)

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]`.

Environments can be renamed via the Mode/Environments/reName menu entry or its corresponding key binding, provided that the cursor is within an environment.

3.8.1 Using a Prefix

Some environments and commands use optional parameters. By default, menu entries and key bindings produce the command without the optional part. Goccia's rule 1 is obeyed, but what about rule 2?

Using a *prefix* (shortly, pressing `ESC 1` before any other key binding) will insert a more complete form. For example, if you press `Ctrl-Cpl` you will get `\includegraphics{}`, while pressing `ESC 1 Ctrl-Cpl` you will get `\includegraphics[scale=|width=|height=]{}`.

3.8.2 Itemize

In `itemize` or `enumerate` environments you can use `Ctrl-C⟨RET⟩` to start a new `\item` line.

3.8.3 Tables

LaTeX4Jed 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 entry will insert a line containing the right number of `&` delimiters, and terminated by `\\`. Just fill in the blanks.

3.9 Templates

Templates for standard LaTeX document classes: `article`, `book`, `letter`, `report`, and `slides`, are provided. In addition, templates for notices and for Prosper presentations (`http://prosper.sourceforge.net`) were added as modules (see below) 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. The listed packages were chosen after some statistics on several LaTeX documents.

3.10 Modules

LaTeX4Jed has a modular structure, so that its functionality can be enhanced as needed. Additional modules can be written by the user as S-Lang files and must reside in `LaTeX_Modules_Dir`. This directory is `$JED_ROOT/lib/latex` by default, but the user can choose his or her directory in `.jedrc`:

```
variable LaTeX_Modules_Dir = "/home/guido/.jedmodules/";
```

Writing modules is very simple, even if you are not an S-Lang expert. There you are a template:

```
% mymodule.sl
define mymodule_fun1 ()
{
  % your code here
}
define mymodule_fun2 ()
{
  % your code here
}
% ... more code...
$1 = "LaTeX-Mode";
$2 = "Global.M&ode.Modules.mymodule";
% optional keybindings
definekey_reserved ("mymodule_fun1", "t1", $1);
definekey_reserved ("mymodule_fun2", "t2", $1);
% optional menus
menu_append_item ($2, "mymodule fun&1", "mymodule_fun1");
menu_append_item ($2, "mymodule fun&2", "mymodule_fun2");
% end of file mymodule.sl
```

Version 1.4.4 of LaTeX4Jed ships with these modules:

notice.sl for writing single-page notices;

prosper.sl for Prosper presentations;

textpos.sl for poster-like material.

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`. An alternative way is toggling Math mode, after which you type `` ` ``-letter (backquote-letter) to get the Greek letter. All available symbols are provided by the file `ltx-math.sl`, included in the Jed distribution.

Arrows are obtained as shown in Table 1.

3.12 Document Structure

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

Press ⟨`RET`⟩ or double click on a section to move to the relative line in the LaTeX buffer, or ⟨`q`⟩ to quit.

4 Composing and Debugging

LaTeX4Jed typesets your documents using five output profiles: dvi, ps, eps, pdf, and dvipdf. The final output is, respectively: a `.dvi` file, a `.ps` or `.eps` file made with `dvips`, a `.pdf` file made with `dvipdf`, and a `.pdf` file made with `pdflatex`. The default output profile is dvi.

Select Mode/Compose (or press `F9`) to typeset the buffer using the current output profile. A 'beep' will notify you of warnings, e.g.

```
Rerun to get cross-references right.
```

If errors are detected, the cursor will be moved to the offending line. Press `Ctrl-C'` to move to the next error. The `latex` error log can be inspected at any moment in the the `*LaTeX log*` buffer.

If the conversion succeeded, the following Mode/View command will start the appropriate viewer. If you try to run a viewer but the document has not been typeset yet, a single `latex` run will be performed and the viewer will be launched.

4.1 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.

5 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 `.bib` extension to turn `bibtex` mode on.

6 Known Issues

- 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. The same problem affects strings that span more than two lines. This is a limitation of the DFA code in Jed.
- Auxiliary programs are not launched as separate processes. In other words, you have to wait for a `latex` run to complete before you can edit the text. Would asyncronous compilation be a better choice?
- There is no support for AMS-TEX symbols. I'll gladly accept a user-contributed module, which should be straightforward to write—perhaps just a bit lengthy.
- The documentation (this guide) is probably too concise. Please let me know if you think it is.

7 The End

LATEX4JED was written and is copyrighted © by Guido Gonzato, Ph.D. `guido,dot,gonzato,at,univr,dot,it`.

So, in actual fact this program is *copylefted* by me.

Many thanks to Günter Milde and Jörg Sommer for their valuable contributions, and to all users who helped me with suggestions and bug reports. I'm especially indebted with Jørgen Larsen for showing me how to write good S-Lang code. Wait—can I forget to thank John E. Davis for his great editor?

I use LaTeX4Jed every day, and I think it's very useful and complete. However, I'll be glad to receive suggestions and requests from you. If you find a bug or would like me to add new features, please feel free to contact me. If you want to send me a module, you're more than welcome!

Enjoy,

Guido =8-)