Short guide to Jørgen Larsen's LaTeX mode for the Jed editor

Jørgen Larsen
<jl@dirac.ruc.dk>

17th January 2003

Contents

1 Introduction

The aims of this mode are: to provide tools which are useful when writing and editing LaTeX input files, to manage projects with several LaTeX input files, to provide an interface to the latex/pdflatex engine, to make it easy to jump from an error message to the corresponding line in the input file, and to define simple interfaces to previewer, printer, bibtex, etc. The user is assumed to have a working knowledge of LaTeX.

Composing, previewing and printing can be done by pressing the function keys **F6**, **F8** and **F9** respectively (as in many other texshell programs), and you can step through the error messages using **F4** and **Shift-F4**. These and other operations can be activated from the Mode menu and its submenus; the submenus are also accessed by shortcut key combinations of the form **ResKey** + one more key. Here **ResKey** is the *Reserved Key Prefix* whose value depends on the Jed emulation you are using: with the Emacs emulation **ResKey** is **Ctrl-C** , and with the Borland IDE emulation **ResKey** is **Ctrl-Z** .

This mode uses the idea of a main file (or primary file or master file). All operations such as composing, previewing, printing, bibtex'ing etc. are done on the main file.

The mode has tools for inserting and renaming LaTeX environments, for inserting font commands, and for indenting a region, an environment or the entire buffer. It redefines the 'find matching delimiter' operation so that it also will find LaTeX delimiters.

2 Usage

2.1 Inserting font commands

Font commands such as `\textbf` or `\emph` can be inserted using the Font commands menu (shortcut key **ResKey Ctrl-F**).

The insert operation is region-or-word aware: If a region is defined, the font command will be applied to this region, otherwise it will be applied to the word at the cursor.

2.2 Operations on environments

Operations relating to environments (that is, constructions of the form `\begin{foo}` ... `\end{foo}`) can be performed using the Environments menu (shortcut key **ResKey Ctrl-E**).

When you select Insert new environment, you will be prompted for the name of the new environment. The name is read 'with completion', that is, if you are going to insert a 'known' environment, just type the first few letters of its name and then press the **Tab** key. – If a region is marked, the new environment will be placed around the region.

In list environments you can use **ResKey Ctrl-J** to insert a new `\index` or `\index[ ]`.

A number of standard environments will prompt for supplementary information, see below.

Table 1 Column specifiers in `array` and `tabular` Standard LaTeX. – The array and tabularx have more options.

specifier	effect
`l`	a column of left aligned items
`r`	a column of right aligned items
`c`	a column of centered items
`|`	a vertical line
`@{`*`decl`*`}`	insert *decl* in every row
`p{`*`wd`*`}`	a column with each item typeset in a parbox of width wd
`{`*`num`*`}{`*`cols`*`}`	repeat cols num times

2.2.1 Inserting an `array` or `tabular` environment

When inserting an `array` or `tabular` environment you have to specify a *Column format*, see Table 1, and optionally a *Position* which is one of `c t b` and determines the vertical position of the object (center, top, bottom).

Each row consists of items separated by `&`. Multicolumn items are made using `\multicolumn{num}{col}{item}`. Rows are separated by `\\`

Vertical lines are made by `\vline`, `\hline`, and `\cline` (which takes an argument of the form *col1-col2*).

Notes:

1. The booktabs package has aestethichally more satisfactory horizontal lines (vertical lines are considered bad taste): `\toprule`, `\midrule`, `\cmidrule` and `\bottomrule`
2. The array package has extended versions of `array`, `tabular` and `tabular*`.

2.2.2 Inserting a `figure` or `table` environment

When inserting a `figure` or `table` environment you will be prompted for

1. *Position* which should be a suitable subset of the five characters `tbhp!`. The position parameter influences the way LaTeX places the float object on the page: `t` = top, `b` = bottom, `h` = here, `p` = on a separate floats-only page, `!` = don't be so fuzzy about the typographical rules when placing this float.
2. *Caption*: the figure/table text. The figure/tabel will not be numbered if there is no caption.
3. *Label*: the argument to the `\label` command.

2.2.3 Inserting a `minipage` environment

When inserting a `minipage` you will be prompted for one or more of these paramters:

1. *`Width`*: the width of the minipage. This must be a *length* such as `5cm` or `0.6\textwidth`.
2. *`Position`* (optional): determines the vertical position of the minipage relative to the surrounding text and can be one of `c t b`.
3. *`Height`* (optional): the height of the object; a length.
4. *`Innerpos`*: the vertical position within the box. One of `c t b`.

2.2.4 Inserting a `picture` environment

When inserting a `picture` environment you will be prompted for

1. *`Width`* and *`Height`*: two positive numbers determining the dimensions of the `picture` object.

   The actual width and height will be *`Width`*`\unitlength` and *`Height`*`\unitlength`.

2. *`x_offset`* and *`y_offset`* (optional): integers.

2.3 Inserting sections

If you want to write well-structured LaTeX documents, it is recommanded that you specify sections, subsections ... as environments, i.e. write something like
`\begin{`*`Sectype`*`}[`*`Shorttitle`*`]{`*`Title`*`}`
`some text`
`\end{`*`Sectype`*`}`
where *`Sectype`* is `section` or `subsection` or `subsubsection` or `paragraph` or `subparagraph`.

Chapters, parts and all the starred variants still have to be made the old way, eg. `\chapter{Intro}` or `\section*{FAQs}`.

Section commands are inserted using the Insert section menu (shortcut key ResKey Ctrl-S). Sections etc. will always be inserted as environments if possible (you may turn this feature off by setting the option `SECTION_AS_ENV` to `0`).

When inserting a section environment or command, you will be prompted for a *`Title`*, an optional *`Shorttitle`* and an optional *`Label`*.

2.4 The documentclass declaration

A documentclass declaration is inserted the same way as environments, that is, select the Insert new environment menu and type `document` (actually, the value `document` is the default value if the cursor is located at the beginning of the buffer).

2.5 Matching delimiters

If you place the cursor on a delimiter and press **F11** (or **Ctrl-**); then the cursor jumps to the matching delimiter. If no match is found, an error message is given and the cursor remains at the original position. – The matching delimiter function recognizes the following pairs of delimiters:

```
( )    [ ]    { }

\[ \]    \( \)   \left \right    \begin{foo} \end{foo}
```

2.6 Indentation

The `latex` mode can indent the text in a way that reflects its syntactical structure. Use the Indenting menu (shortcut key **ResKey Ctrl-Q**).

The indentation scheme is aware of *braces* (`{ }`), LaTeX *environments* (i.e. `\begin{foo}` ... `\end{foo}` constructions), *displayed math* using `\[ \]`, *big delimiters* (i.e. `\left \right` constructions), and *items*. The actual amount of indentation is calculated using certain parameters (options), cf. Section 3.1: `BRACE_INDENT`, `ENVIRONMENT_INDENT`, `DISPLAYMATH_INDENT`, `BIGDELIM_INDENT`, and `ITEM_INDENT`.

Note that the indentation mechanism will insert and/or remove space characters, but never newline characters.

2.7 Insert macro; complete symbol

(Functions in this section are taken from Jed's standard `latex` mode.)

ResKey Ctrl-M is the *insert macro* key. Example: If you want to insert the command `\usepackage` in your buffer, then press **ResKey Ctrl-M**, and the minibuffer will prompt you for a name; type in the first few letters and press **Tab** and then **Space** to cycle through the suggested completions.

ResKey Ctrl-I (or **ResKey Tab**) is the *complete symbol* key. Example: If you want to insert the command `\usepackage` in your buffer, then try typing `\us`; then press **ResKey Ctrl-I** one or more times until `\usepackage` appears.

Both these functions use the list of commands in the file `ltx-comp.dat`.

A related Jed function is the very useful `dabbrev` which is bound to **ESC /** and to **Ctrl-A** (except with IDE emulation where it is bound to **Ctrl-V**). Type the first few letters of a word; then press the `dabbrev` key to get the desired completion (`dabbrev` gets its completions from the current buffer, so it only 'knows' words therein). – Note that the LaTeX mode adds the backslash to `dabbrev`'s list of letters.

2.8 Useful keys

2.8.1 The dollar key

The dollar key **$** inserts two $ characters and places the cursor between them; *or* if a region was marked then a $ is placed on either side of the region. If the cursor is placed immediately after a \, then the **$** key inserts a single $.

2.8.2 The quote key

The quote key **'** is the standard 'insert quote'* which inserts either a left quote character or a right quote character (it tries to guess which one from the context).

2.8.3 The backquote key

The key combination **ResKey Ins** or **ResKey ~** toggles the meaning of the backquote key **`** between 'quoted insert' (the default†) and 'math insert'.

The backquote key **`** as 'quoted insert' acts as a special character that undoes the special-ness of other keys. Thus **``** always inserts a ', and **`$** always inserts a $.

The backquote as 'math insert' acts as a special prefix character, thus the key combination **`a** expands to `\alpha`. Note that on some keyboards the backquote key is a dead key, and in that case you will have to type **`** followed by a **Space** instead of just **`**.

Table 2 lists the available backquote combinations.‡

2.8.4 Comments

The mode uses the standard commenting functions. Press **ResKey ;** to comment the region/current line, and press **ResKey :** to uncomment the region/current line. (The current line is the line where the cursor is.)

2.8.5 Braces and brackets

The key combination **ResKey Ctrl-B** or **ResKey {** inserts braces (`{ }`); if a region is marked then the braces are placed on either side of the region, or if the cursor is placed on a word then the braces are placed on either side of the word.

The key combination **ResKey Ctrl-K** inserts brackets (`[ ]`); if a region is marked then the brackets are inserted on either side of the region, or if the cursor is placed on a word then the brackets are inserted on either side of the word.

*from `texcom.sl`.

†You can change the default to 'math insert' by setting the option `MATH_MODE` to `1`.

‡The math backquote combinations come from Jed's standard `LaTeX` mode.

Table 2 Backquote key combinations in math mode

`a	\alpha	`A	\forall		
`b	\beta			`Ctrl-B	\leftarrow
`c	\mathcal			`Ctrl-C	\cos
`d	\delta	`D	\Delta	`Ctrl-D	\det
`e	\epsilon	`E	\exists	`Ctrl-E	\exp
`f	\phi			`Ctrl-F	\rightarrow
`g	\gamma	`G	\Gamma		
`h	\eta				
`i	\in	`I	\infty		
`k	\kappa				
`l	\lambda	`L	\Lambda	`Ctrl-L	\lim
`m	\mu				
`n	\nu	`N	\nabla	`Ctrl-N	\downarrow
`o	\omega	`O	\Omega		
`p	\pi	`P	\Pi	`Ctrl-P	\uparrow
`q	\theta	`Q	\Theta		
`r	\rho				
`s	\sigma	`S	\Sigma	`Ctrl-S	\sin
`t	\tau			`Ctrl-T	\tan
`u	\upsilon	`U	\Upsilon		
`v	\vee	`V	\Phi		
`x	\chi				
`y	\psi	`Y	\Psi		
`z	\zeta				
`(	\langle	`[	\subseteq	`{	\subset
`)	\rangle	`]	\supseteq	`}	\supset
`\	\setminus	`/	\not	`|	\vee
`+	\cup	`-	\cap	`*	\times
`<	\leq	`>	\geq	`.	\cdot
`~	\tilde	`^	\hat	`Ctrl-^	\sup
`!	\neg	`&	\wedge	`Ctrl-_	\inf
`0	\emptyset				

2.8.6 Sub- and superscripts

The key combination **ResKey +** inserts `^{}`; if a region is marked then `^{` is inserted immediately before the region and the `}` immediately after.

The key combination **ResKey −** inserts `_{}`; if a region is marked then `_{` is inserted immediately before the region and the `}` immediately after.

2.9 Calling external programs

The external programs operate on the *main file* (and files derived from it). The main file defaults to the file that made Jed load the `latex` mode. You can change the main file from the Main file menu (shortcut key **ResKey Ctrl-A**); if the current file is different from the main file, then the 'select main file' algorithm suggests the current file as new main file.

To *compose* (or compile or 'run through latex') the main file, press **F6** (or **ResKey Ctrl-C**). If there were errors, a list of error messages and warnings will appear in a separate window*; then press **F4**/**Shift-F4** (or **ResKey ,**/**ResKey '**) to step forward/backward through the list while the other window displays the input line where latex detected the error.† See also Note 1 below.

To *preview* the dvi file corresponding to the main file, press **F8** (or **ResKey Ctrl-V**).

To *print* the dvi file corresponding to the main file, press **F9** (or **ResKey Ctrl-P**); you will be prompted for the first and last page to be printed.

bibtex is called by pressing **F7** (or **ResKey Ctrl-L**).

makeindex is called by pressing **ResKey Ctrl-N** .

Use the dvi/ps/pdf output menu (shortcut key **ResKey Ctrl-O**) to select the desired combination of composer and previewer: latex+dviviewer, latex+PSviewer, and pdflatex+Acrobat Reader.

The actual calls to the various external programs are defined via the following options (with obvious meanings) `LATEX`, `PDFLATEX`, `DVIVIEWER`, `PDFVIEWER`, `PSVIEWER`, `DVIPSPRINT`, `BIBTEX` and `MAKEINDEX` all of which can be redefined at any time; initial values are assigned in the configuration file `latexrc.sl`.

The above options should be assigned string values holding the actual commands; `%s` is used as a placeholder for the name of the mainfile. Example: `DVIVIEWER` could be assigned the value `xdvi %s`.

Notes: If the OS allows several processes to be run simultaneously, please note the following:

*The number of lines of this window is given by the value of the option `COMP_BUF_SIZE`.

†As is well known, the line where latex detects an error is certainly not always the line where the error actually is.

1. The compose process is run as a separate process.[‡] This means that you can use the editor while the compose process is running. When the process has finished, use **Ctrl-F4** to jump the first error.

2. The previewer will be started as a separate process. Thus you can use the editor while the previewer is running.

3. Other external programs (such as bibtex and makeindex) are also run as separate processes in the same way as the previewer. On Win32 systems this means that you will not always be able to run say bibtex when the viewer is running (you will see the Jed error message `A compile process is already running`.)

3 Configuration

3.1 Options

The behaviour of the mode is governed by a number of *options*. Options are labeled by a `String_Type` label. The following options are used:
`ASYNC_COMP`, `BIBTEX`, `BIGDELIM_INDENT`, `BRACE_INDENT`, `COMP_BUF_SIZE`, `DFLT_CLS_NAME`, `DFLT_CLS_OPT`, `DFLT_ENV_NAME`, `DFLT_SEC_NAME`, `DISPLAYMATH_INDENT`, `DVIPSPRINT`, `DVIVIEWER`, `ENVIRONMENT_INDENT`, `FIG_LABEL`, `ITEM_INDENT`, `LATEX`, `MAKEINDEX`, `MATH_MODE`, `OUTPUT`, `PDFLATEX`, `PDFVIEWER`, `PSVIEWER`, `SECTION_AS_ENV`, `SEC_LABEL`, `TAB_LABEL`, `WARNINGS`.

The function `latex_optionslist` prints a list of all currently assigned options and their actual values.

The function `latex_set_option` assigns values: the syntax is `latex_set_option (String_Type label, Any_Type value)`. The function `latex_get_option` returns the value of an option.

3.2 The configuration files

The mode uses two configuration files: it always reads the system-wide configuration file `latexrc.sl` in the Jed library path, and then (if it exists) an optional `latexrc.sl` in the user's home directory (as defined by the environment variable `HOME`).

In the configuration file you can specify the calls to external programs (previewers etc.) and customize the behavior of the `LaTeX` mode. Parts of the costumization is done using the options, see above, but two other kinds of costumization are possible:

1. The mode uses global variables `LaTeX_classes` and `LaTeX_environments` to hold the names of some standard document classes, section types and

[‡]This facility can be turned off by setting the option `ASYNC_COMP` to `0` in the configuration file.

environments. The user can add values to these variables by writing command lines of the form `LaTeX_classes += ",myclass";` in the configuration file.

2. You can add document class templates to the mode's database. This is done in the following way: First define a function with two `String_Type` arguments, the document class options and the document class name, and the add it to the database using the function `latex_add_cls_templ`. As an example we show how the mode itself inserts its `letter` template:

```
static define insert_letter_cls (opt, name)
{
  vinsert ("\\documentclass%s{letter}\n\n", opt);
  insert (strcat ("\\name{}\n\\address{}\n",
                  "\n\\begin{document}\n\\begin{letter}{"));
  push_spot ();
  insert ("}\n\\opening{Hello}\n\n\n" +
          "\\closing{Bye}\n\\end{letter}\n\\end{document}\n");
  pop_spot ();
}

latex_add_cls_templ ("letter", &insert_letter_cls);
```

4 Installation

Place the following files `latex.sl`, `ltxacmpl.sl latexrc.sl` and `ltxabout.hlp` in a directory where Jed looks for `sl`-files (e.g. `jed/lib/`).

Add two lines to your `.jedrc`/`jed.rc` file:

```
add_mode_for_extension ("latex", "tex");
autoload ("latex_mode", "latex");
```

To generate the DFA cache table, add `latex.sl` to the list (in `preparse.sl`) of modes for which DFA tables should be constructed, and do the preparsing, ie. run `jed -batch -n -l preparse`.

(You may also want to add the file `ltxacmpl.sl` to the list (in `bytecomp.sl`) of files which are going to be precompiled.)

Index