LaTeX Bundle

Documentation

1 · Installing LaTeX

To use Typeset & View (PDF) and other commands from the LaTeX bundle you need to install a TeX distribution. We recommend that you use MacTeX as it comes with a standard "no questions asked" installer.

MacTeX also takes care of updating your `PATH` variable. If you install another distribution you may need to setup the path manually.

2 · Building a LaTeX File

2.1 Standard Typesetting

Most of the time you just want to typeset the currently selected file. You can accomplish this with the command Typeset & View (PDF), bound to ⌘ + R. TextMate shows you, in its HTML output window, progress on the compile, as well as any errors that may occur.

Depending on the setting of the Keep log window open preference, the HTML output window may stay open even after the typesetting is done. You can click on any of the errors messages in the log window. This will take you to the location of the error in the LaTeX file. Keep in mind, that LaTeX occasionally reports errors very far from where the actual problem occurs. So compile often, then you have less new text to worry about when looking for errors.

2.2 Typesetting Multiple Passes

Because LaTeX processes files in a single pass, you often need to compile more than once to resolve all references. If you use citations, a glossary or other advanced LaTeX features, then you also need to use other commands such as `bibtex` and `makeindex` between runs of LaTeX. You can re-run LaTeX on the same file by clicking on the Run LaTeX button at the bottom of the log window that appears after you invoked Typeset & View (PDF). You will also find buttons in this window that allow you to run BibTeX or MakeIndex for the current file.

Since the process of running different typesetting programs multiple times in order to get the final document is a rather dull one we also support latexmk:

> Latexmk completely automates the process of generating a LaTeX document.

In order to tell TextMate to use `latexmk` when compiling, you have to check Use Latexmk inside the bundle's preference window. After that you only need to invoke Typeset & View (PDF) once and TextMate will automatically run all necessary commands to translate your document.

2.3 Using a Master File

If you work on a large project, you probably want to use TextMate's support for projects. In a larger LaTeX project, such as a book or a thesis you usually split your project in smaller files that contain one part of your document. You then include

these parts in a so-called master file via the commands `\input` or `\include`.

Most of the commands of the LaTeX bundle rely on the fact that they know the location of the master file. If you do not specify the master file explicitly, then the bundle assumes that the active document is the master file. This works perfectly fine, as long as you only use one `.tex` file.

On the other hand, if you call a command from an included file and do not specify the master file, then the LaTeX bundle only knows about the content of the current file. This means that completion commands will only take the current file into account. E.g. Insert ⟩ Label Based on Current Word will only list the labels in the current file. Insert ⟩ Citation Based on Current Word might not work at all, since it only scans the current file for included bibliographies. To fix this behaviour you need to tell TextMate the location of the master file. Currently there a two possibilities to do that:

1. Use the environment variable `TM_LATEX_MASTER`

2. Add the `%!TEX root` directive at the top of the included file

If you specified the master file via one of these methods, then all LaTeX commands use the master file as their basis. In particular, the Typeset & View (PDF) and Watch Document commands will typeset the master file, instead of whatever the currently active file is. So you can freely edit whatever chapter you are working on, and when you want to see the result you just press ⌘ + R without having to worry about switching to the master file.

2.3.1   Using the TM_LATEX_MASTER Environment Variable

You can set the environment variable `TM_LATEX_MASTER` via a `.tm_properties` file located at the top level folder of your project. Just create a new text file with the name `.tm_properties` and the following content:

```
TM_LATEX_MASTER = absolute_path_to_master_file
```

You probably do not want to specify the absolute path to your master file directly, since this would mean that the location of your master file will be invalid if you move your project folder. For this purpose you can use the special variable `$CWD`, which contains the location of the folder where the `.tm_properties` file is located. For example, if your master file is called `Thesis.tex` and resides in the top level folder of your project, then you can use the following line to specify the location of your master file:

```
TM_LATEX_MASTER = "$CWD/Thesis.tex"
```

2.3.2   Using the %!TEX root Directive

The LaTeX bundle supports `%!TEX root` directives. You can specify the location of your master file by adding the line `%!TEX root = path_to_master_file` at the beginning of an included file. `path_to_master_file` can be either absolute or relative.

For example, if your included file `Chapter 1.tex` is located in the folder `Chapter` inside your project folder, and your master file `Thesis.tex` is located in the top level folder of your project, then you can set the master file for `Chapter 1.tex` by including the following line at the top of the file `Chapter 1.tex`:

```
%!TEX root = ../Thesis.tex
```

You can also use the command Set Master File to specify the master document for the current `.tex` file.

2.4   Watching a Document

2.4.1   Introduction

When you watch a LaTeX document, it is continually monitored for changes: Each time you change the content of your `.tex` file and save it, TextMate typesets it and updates the preview.

2.4.2   Usage

Press ctrl + ⌘ + W to start watching a document. If the document is already being watched, you will instead be given the option to stop watching it. You can watch several documents simultaneously.

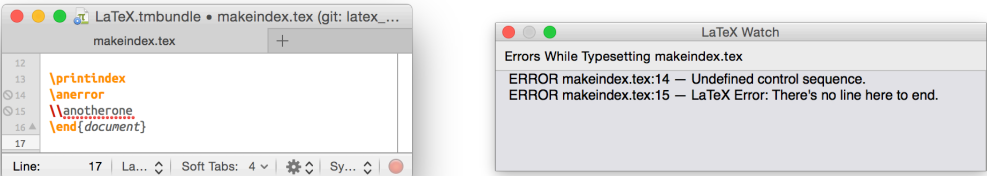When you close the previewer, the associated watcher will automatically quit.

Figure 1: Watch Document shows errors it detects in a floating window

One minor, but interesting difference between the behaviour of Stop Watching and closing the previewer is, that in the latter case temporary files produced by LaTeX will be deleted. If you want to keep your directory clean, then you should just close the previewer after you are done. If you do not care about the auxiliary files – but instead want Watch Document to reuse old temporary files and therefore start faster the next time, then just use ctrl + ⌘ + W to stop monitoring the file for changes.

3 · Previewing a LaTeX File

The Typeset & View (PDF) command has a second component, the "View" one. After a successful build, TextMate proceeds to show you the produced PDF file. This behaviour can be changed by toggling the checkbox Show PDF automatically inside the bundle's preferences. If the preference item is not checked, then you can still view the file on demand by clicking the View button at the bottom of the Typeset & View window.

3.1 Default Preview

In OS X, HTML windows can display PDF files. This is the standard behavior of the LaTeX bundle's `Typeset & View (PDF)` command. After TextMate finished the typesetting process, the window that was used to show you the building progress turns into a view of your PDF file, provided there were no errors.

3.2 External Previewers

You can also setup an external previewer for showing the PDF output. Focus will then switch to that previewer after TextMate finished the typesetting process. You can set the preview application inside the bundle's preferences. We recommend you use Skim, which supports SyncTeX.

3.3 Preview Options

Preview options are somewhat complicated depending on the viewer you choose. There are really two main cases:

1. If you chose TextMate as previewer, then the `Keep log window open` preference has the following effect:

   - If there are no errors or warnings, the `Typeset & View` window will immediately switch to showing you the PDF file.
   - If there are no errors but some warnings then – assuming the `Keep log window open` preference is checked – you will see the warning messages. To display the PDF click on `View in TextMate`. If the `Keep log window open` preference is not checked, then the warning messages will be ignored and TextMate will display the PDF directly.

2. If you use an external viewer, then the `Typeset & View` window will automatically close if there are no errors or warnings, unless the `Keep log window open` preference is checked.

3.4 Refreshing the Viewer

The `Typeset & View (PDF)` command uses a short Applescript to tell Skim, or the TeXShop to reload the PDF file once the typesetting is complete. This is more efficient than enabling the auto-refresh feature in the viewers, because it often takes more than one run of LaTeX before the document is really ready to view. In that case most viewers would try to reload the PDF multiple times.

4 · SyncTeX

SyncTeX allows you to easily hop back and forth between document and the generated PDF, granted you use an external previewer which supports SyncTeX. You need to perform the following steps to enable synchronization:

1. Set your viewer to `Skim`. This enables you to use the `Jump to Current Line in Viewer` command bound to `ctrl` + `⌥` + `⌘` + `O` by default.

2. In Skim go to `Preferences...`. There, choose the Preset `TextMate` under the option `Sync`.

   This assumes that you installed `mate` (see `Preferences...` ⟩ `Terminal` in TextMate). After this is done, shift-command-clicking (`⇧`+`⌘` at a location in the PDF file (as shown in Skim) takes you to the corresponding location in TextMate.

Note: The granularity of the synchronization data is generally "per paragraph", so going from TextMate to Skim or back will generally locate just the paragraph nearest your desired position.

5 · Working With LaTeX

This section describes the various LaTeX tasks, and how they can be accomplished with the commands provided by the bundle. Some of the commands whose behavior is clear from their name (like `Format` ⟩ `Bold` and friends) are not included here, so you will need to traverse the submenus of the LaTeX bundle to discover them.

5.1 Automated Typing

Writing LaTeX often requires typing some amount of standard commands and environments. TextMate makes that a lot easier with a set of commands in the LaTeX bundle, that we'll discuss in this section.

5.1.1 Inserting Commands and Environments

The LaTeX bundle contains two commands that, if you type a lot of LaTeX, will become your best friends. They are `Insert` ⟩ `Environment Based on Current Word`, bound by default to `⌘`+`<`, and `Insert` ⟩ `Command Based on Current Word`, bound by default to `⌘`+`>`. They create an environment/command based on the current word, or a default editable text in the absence of a current word. They are smart enough to understand a number of "standard" shortcuts, also called triggers. For instance, typing `thm` and then calling the `Insert` ⟩ `Environment Based on Current Word` command creates:

```
\begin{theorem}
    | ← insertion point
\end{theorem}
```

Similarly, typing `fig` followed by calling the `Insert` ⟩ `Environment Based on Current Word` command creates a lot of the standard text required in a figure environment. You can further customize these commands.

These two commands understand a series of shortcuts, and use the current word if they do not recognize it as a shortcut. You can customize what these shortcuts are by editing the *LaTeX Configuration File*. This file is originally kept in the LaTeX bundle. When you first use the `Edit Configuration File` command, this file is copied to `~` ▸ `Library` ▸ `Preferences` ▸ `com.macromates.textmate.latex_config.plist`. You can then edit this file whenever you want by executing this command, or delete it to return to the default settings.

This file follows the Property List Format. It consists of a top-level dictionary with six entries:

- `commands`, `environments`, `labeled_environments`

- `commands_beamer`, `environments_beamer`, `labeled_environments_beamer`

The versions with the word `beamer` added are the *extra* shortcuts/words that TextMate recognizes in LaTeX Beamer files, *in addition to the non-beamer ones*. The `commands` and `commands_beamer` entries are dictionaries consisting of pairs, where the key is the shortcut, and the value is the text that ⟨Insert⟩ ⟩ Command Based on Current Word inserts when it recognizes the shortcut. All inserted text, for both commands and environments, is interpreted as Snippet.

The four `environment` dictionaries are a bit different. They contain key-value pairs, where the key is the *name* of the environment, i.e. the text inside the braces in `\begin{}`. The value is itself a dictionary with two entries:

triggers is the list of shortcuts/words that will cause
Insert ⟩ Environment Based on Current Word to insert the environment

content is the text that TextMate inserts into the environment. If the environment is in one of the two labeled environments groups, then this text is inserted right after the closing brace in `\begin{env}`. Otherwise, it is inserted at the beginning of the next line.

Another useful command is Insert ⟩ Environment Closer, which is by default bound to ⌥ + ⌘ + . . This command locates the innermost `\begin{env}` that hasn't been closed by a corresponding `\end{env}` and inserts the closing part. Of course, if you usually use Insert ⟩ Environment Based on Current Word, then you probably don't need this command that often.

Finally, there is a command to quickly insert commands for the various symbols, called Insert ⟩ Symbol Based on Current Word. It is bound by default to ⌘ + \ . It works in two stages:

- First, you write down a few letters recognized by the command. The rules are basically as follows:

  - Single letters are converted to greek letters

  - Two letter combinations are converted to the various commands starting with those two letters (for instance pressing `in` would trigger commands like `\int`, `\inf`, `\infty` etc) with a few exceptions, like `sk` for `\smallskip`.

  - Three letter combinations are converted to arrows, where the three digits signify the kind of arrow, for instance `lar` would stand for left arrows.

  - There's a couple of exceptions to these rules. You can look at and alter the shortcuts in the LaTeX configuration file. They are located under the `symbols` key.

- After you triggered the command once, pressing it again cycles through the various options. For instance, if you started with **e**, then the command cycles between \epsilon and \varepsilon. This is accomplished by this set of entries in the configuration file:

```
"e" = "\\epsilon";
"epsilon" = "varepsilon";
"varepsilon" = "epsilon";
```

When you create your own additions to this list, keep in mind these two simple principles: For the item that is the initial trigger, like the **"e"** above, the text must start with two backslashes. For the items used for cycling through options, it must not.

5.1.2 List Environments

A list of the the most commonly used LaTeX environments would certainly include the three itemize environments `itemize`, `enumerate` and `description`. The LaTeX bundle provides a few shortcuts to work with these environments. To insert one of the environments either:

1. type one of the words `it`, `en` or `desc` and invoke the command ⟨Insert⟩ ⟩Environment Based on Current Word⟨, or

2. type one of the words `item`, `enum` or `desc` and press →⃒.

The first \item is automatically inserted for you. To add a new item press ⤒ and TextMate creates a new entry by inserting the command \item in the next line. This functionality is common among most languages in TextMate that support some sort of list.

There is also a ⟨Lists⟩⟩Itemize Lines in Selection⟨ command, bound to ⌘+⇧+L. It wraps the selected group of lines inside a list environment (`enumerate`, `itemize`, or `description`). Each non-blank line becomes an \item in the environment. If the first 20 characters of every selected non-empty line contain a **:**, then the command creates a description environment. All the characters up to the **:** are placed inside brackets.

5.1.3 Table Environments

The command ⟨Table⟩⟩Create Table⟨ allows you to quickly inserts a table into your document. The command works both with or without a selection.

If you do not select any text before you invoke the command, then TextMate displays the pop-up window shown in Figure 2. From there you can choose how many rows and columns the inserted table should have. If you check ⟨Tabular Only⟩ – accessible via ⌥+⌘+T – then the command creates only a **\tabular** environment. Otherwise the command surrounds the **\tabular** environment with a **\table** environment. After you hit ⟨Insert⟩ the command places the table at the current caret position. You can then fill in the various attributes. To switch between them use the keys →⃒ and ⇧+→⃒.
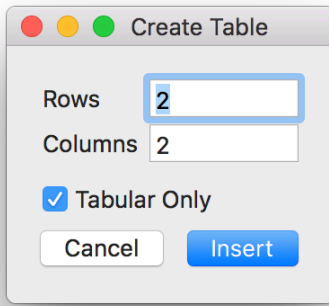
Figure 2: Graphical User Interface for the command "Create Table"

If you select text before you invoke the command, then use the following format:

```
[rows] columns [t]
```

The format specification above states that you typically write down the number of rows, then the number of columns and then optionally add the character t, if you only want to create a tabular environment. The number of rows is optional. If you do not specify it, then TextMate will create a table with two rows. We provide two examples, which should help you understand the interface. The text above the example shows the selection. After that we show the inserted LaTeX code, followed by the rendering of the translated code.

Selected Text: `2 3`

```latex
\begin{table}[htb!]
  \caption{Caption}
  \label{table:label}
  \centering
  \begin{tabular}{ccc}
    \toprule
    \textbf{Header 1} & \textbf{Header 2} & \textbf{Header 3}\\
    \midrule
             r2c1 &              r2c2 &              r2c3\\
    \bottomrule
  \end{tabular}
\end{table}
```

Table 1: Caption

Header 1	Header 2	Header 3
r2c1	r2c2	r2c3

Selected Text: `3 4 t`

```
\begin{tabular}{cccc}
  r1c1 & r1c2 & r1c3 & r1c4\\
  r2c1 & r2c2 & r2c3 & r2c4\\
  r3c1 & r3c2 & r3c3 & r3c4\\
\end{tabular}
```

```
r1c1   r1c2   r1c3   r1c4
r2c1   r2c2   r2c3   r2c4
r3c1   r3c2   r3c3   r3c4
```

5.1.4 Wrapping Text

Often one wants to enclose the currently selected text in an environment, or a command. The LaTeX bundle offers a list of Wrap Selection In commands for all tastes. Here they are:

Wrap Selection in ⟫ **Command**   `ctrl`+`⇧`+`W`

This command wraps the selected text in the command `\emph{}`. To change the environment press the tab key once and write down the name of the command you want. If you did not select any text before invoking the command, then you can press tab once again to insert a text into the empty environment you just created.

Wrap Selection in ⟫ **Environment**   `ctrl`+`⇧`+`⌘`+`W`

This command wraps the selected text in an environment. It also works without a selection.

Wrap Selection in ⟫ **Double Quotes**   `ctrl`+`` ` ``

Wraps the currently selected text in LaTeX double quotes, i.e. ` ``selection here'' `.

Wrap Selection in ⟫ **left...right**   `ctrl`+`⇧`+`L`

Wraps the currently selected text in the `\left`–`\right` pair. I.e. if you select `(text here)` and invoke the command, then TextMate replaces it with `\left(text here\right)`.

Wrap Selection in ⟫ **Display Math**   `ctrl`+`⇧`+`M`

This command wraps the selection between the text `\[` and `\]`.

Wrap Selection in ⟫ **Math Mode**   `ctrl`+`⇧`+`M`

This command inserts `\(` before, and `\)` after the selection.

5.2 Completion

The LaTeX bundle adds the following words to the list of completions (accessed through ⌥): corollary, definition, description, enumerate, equation, itemize, lemma, proof, proposition and verbatim.

5.2.1 Commands

We already talked about how you can quickly create a command with [Insert⟩ ⟩Command Based on Current Word]. You can also complete a partial typed command with [Insert ⟩ Command Completion]. Just write down the first characters of your word, e.g. `\table`, and press ⌥+⌘. TextMate will then display a menu with possible completions consisting of predefined commands and commands from your current master file.
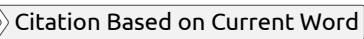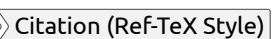
5.2.2 Citations and References

The LaTeX bundle overrides the standard completion behavior when the caret is inside a `\cite{}` or `\ref{}` block. It also overrides the completion behaviour for other equivalent commands like `eqref`, `vref`, `prettyref`, `citeauthor` etc.

If the LaTeX command expects a bibliography key, then pressing ⌘ inside the braces offers completion with respect to all cite keys. This is accomplished by scanning all bib files included in the current master file. For instance, if the caret is right before the closing brace in `\cite{Ka}`, then pressing ⌘ will cycle trough all bibliography keys starting with `Ka`.

If the LaTeX command expects a label key, pressing ⌘ inside the braces of the command will cycle trough all labels of the current master document. You can narrow the search down by writing part of the label key before you hit ⌘. This is especially useful if you use a prefix that specifies the type of the reference in your labels. For example, if you prefix each label of a theorem with `thm:` — i.e. all theorem labels have the form `thm:labelname`, then pressing ⌘ when the caret is right before the closing brace in `\ref{thm:}` will offer as completion options all labels referencing theorems.

If there are many matching completions, it is often more convenient to use the pull-down-list versions of the commands:

- [Insert ⟩ Label Based on Current Word]
- [Insert ⟩ Citation Based on Current Word]
- [Insert ⟩ Citation (Ref-TeX Style)]

You can trigger them by pressing ⌥+⌘.
Both the label and bibliography completion commands also work outside of a `\ref{}` or `\cite{}` block. They will insert the `\ref` or `\cite` for you, if you call them outside of the aforementioned blocks. Here are a two examples of the usual use cases for the commands:

Figure 3: |Insert ⟩ Label Based on Current Word| in action

- You want to insert a citation after the word `example` and you do not know anything about the cite key, author or title. Just insert a ~ after the word `example`, press |⌥|+|⌦| and select |Citation Based on Current Word...|. TextMate will show you a list of all citations. Just select the one you want via the keyboard or mouse. If we assume that the item you chose has the citekey `theKey`, then TextMate will replace `example~` with `example~\cite{theKey}`.

- You want to insert a citation after the word `example~` and you know that the title of the document you want to cite contains the word `important`. Just add `important` after `example~`, press |⌥|+|⌦| and select |Citation Based on Current Word...|. TextMate will now show you a list of all citations that contain the word `important` somewhere in their title, citekey, or authors tag. If we again assume that the item you chose has the citekey `theKey`, then TextMate will replace `example~important` with `example~\cite{theKey}`.

Deltron3030 ✎ Deltron 3030 📕 Battlesong	1
hada2013my ✎ Hada, Erika 📕 My Little Pony	2
pony2010friendship ✎ Pony, My Little 📕 Friendship Is Magic	3
robertson2013ponies ✎ Robertson, Venetia Laura Delano 📕 Of ponies and men My Little Pony: Friendship is magic and the Brony fandom	4

Figure 4: |Insert ⟩ Citation Based on Current Word| in action

One minor but interesting feature of the completion commands is, that they interpret the current word/selection as regular expression. For example, if you selected the text `foo|bar` and press |⌥|+|⌦| then the completion commands will narrow down the completion list to labels/citations that contain one of the words `foo` or `bar`.

5.3   Advanced Tasks

5.3.1   LaTeX Beamer

The LaTeX bundle contains special supports for the Beamer document class: |Insert Environment| and |Insert Command| understand more shortcuts, and also behave intelligently with respect to other shortcuts, adding overlay specifications where appropriate. The same goes for the |Lists ⟩ New Item| command.

In order for all this to work, make sure that the language for the document is set to |LaTeX Beamer| (|ctrl|+|⇧|+|⌥|+|B|).

5.3.2   Drag and Drop

There are three key drag and drop commands in LaTeX:

- You can drag an image file in to the document and have it surrounded by a complete figure environment. Using the modifier key ⌥ while dragging inserts the image surrounded by a `center` environment. If you use the modifier ⇧, then TextMate only inserts the `\includegraphics` line.

- You can drag in another `.tex` file into your document. TextMate will then insert a `\include` command that references the dropped file. To use a `\input` command instead use the modifier key ⌥.

- If you drop a source code file into your document, then TextMate creates a new `listing` command referencing the dropped file. Said command is part of the minted package. To successfully translate the resulting document include `minted` and install the various package dependencies. Do not forget to specify the option `−−shell−escape` (see section "Preferences"), otherwise the translation will fail.

  If you want to include dropped code via the command `lstinputlisting`, use the modifier key ⇧. To translate the resulting document you need the `listings` package. Make sure that you include it via **\usepackage**{listings}.

5.3.3 Templates
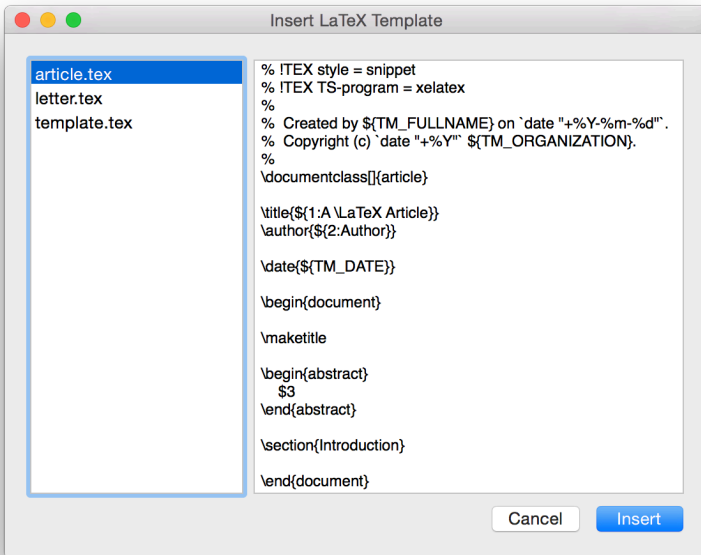
Figure 5: Insert ⟩ LaTeX Template lets you insert LaTeX code into your document

The LaTeX bundle offers a simple template system. We provides a few sample files. If you want to use your own templates, then create the directory ~▸Library ▸Application Support▸LaTeX▸Templates. In there you can put any `.tex` files you like. We also support snippet syntax in template files. To tell TextMate that it should insert the code using snippet syntax add the following line at the top of the template:

```
%!TEX style = snippet
```

To insert a template use the command $\boxed{\text{Insert}} \rangle \boxed{\text{LaTeX Template}}$. The command will show you a window containing a list of all available templates. Select one of these files via the button $\boxed{\text{Insert}}$. TextMate will paste the contents of that file at the cursor's location.

A typical workflow for the template system looks like this:

- Create new file – $\boxed{\mathcal{H}}$ + $\boxed{\text{N}}$

- Set Language to LaTeX – $\boxed{⇧}$ + $\boxed{\text{ctrl}}$ + $\boxed{\mathcal{H}}$ + $\boxed{\text{L}}$

- Type `temp` and press $\boxed{→}$

- Use arrows/keys/mouse to select the template file and press $\boxed{↵}$

- Start working on your document

6 · Preferences

6.1   Global Preferences

To show the Preferences panel for LaTeX select the command $\boxed{\text{Preferences...}}$ or press $\boxed{⌥}$ + $\boxed{\mathcal{H}}$ + $\boxed{,}$ in any LaTeX document. The Preferences window allows you to set options concerning typesetting and viewing.

Figure 6: The settings window of the LaTeX bundle

The following typesetting options are available:

Default Engine   Choose your preferred typesetting engine from the drop-down list. You can override the default engine by using the `%!TEX TS-program` = directive in your source file (see "Local Preferences").

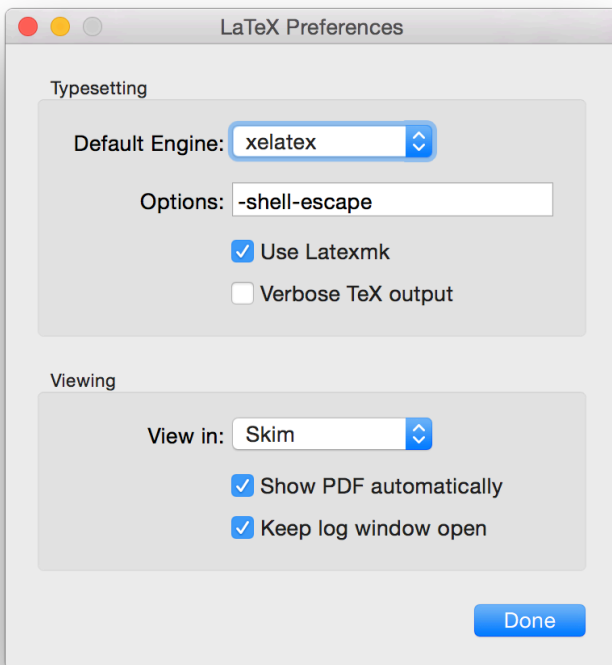> **Note:** `Typeset & View (PDF)` will override the typesetting engine if it detects a package in your source file that is not compatible with the default engine. This only happens if you do not specify the engine via `%!TEX TS-program`.

Options   If you have specific options that you want to pass to the engine, then you can set them here. If you have options that are specific to a single file, then use the `%!TEX TS-options` = directive at the top of your source file.

> **Note:** `Watch Document` currently does not support the `%!TEX TS-options` directive.

Use Latexmk   TextMate supports the popular `latexmk` script which automatically runs `latex`, `bibtex`, `makeindex` and other commands as many times as needed to resolve all cross references and citations in your file. To use `latexmk` just check this box.

> **Note:** `Watch Document` always uses `latexmk` to translate a LaTeX document, regardless of this setting.

Verbose TeX output   If you want to see the raw log file produced by LaTeX in the `Run & View (PDF)` window, then check this box.

You can set the following viewing options:

View in   Select one of the supported viewers. We recommend that you install and use Skim.

Show PDF automatically   If you want that the viewer starts automatically after the typesetting is done, check this box.

Keep log window open   If you want the log window to stay open – so you can check error and warning messages – check this box.

If you use TextMate as viewer – instead of an external viewer like Skim, then you should keep the following in mind: `Show PDF automatically` will not show the PDF file if there are any errors or warnings if `Keep log window open` is checked. If the option is not checked, then the PDF automatically replaces the log assuming there are no errors.

6.2 Local Preferences

There are three options that you can set on a per file basis. As mentioned above these options will override the preferences that you set using the preferences interface.

You can set these "local options" via the %!TEX directives `root`, `TS-program` and `TS-options`:

root   This option allows you to set a master file. More information about this directive is available in the section "Using the %!TEX root Directive".

TS-program To override the typesetting engine for a particular file you can use this directive. For example, to use the engine `xelatex` for a specific document, add the line %!TEX `TS-program = xelatex` at the beginning of the master file.

TS-options You can add file specific (engine) options via this directive.

> **Note:** Whatever options you choose they will be used in the addition to the default options `-interaction=nonstopmode` and `-file-line-error-style`.

6.3 Environment Variables

The LaTeX bundle supports environment variables that let you modify the behaviour of certain commands. You can set this variables either via a .tm_properties file or in the Preferences of TextMater under the tab | Variables |.
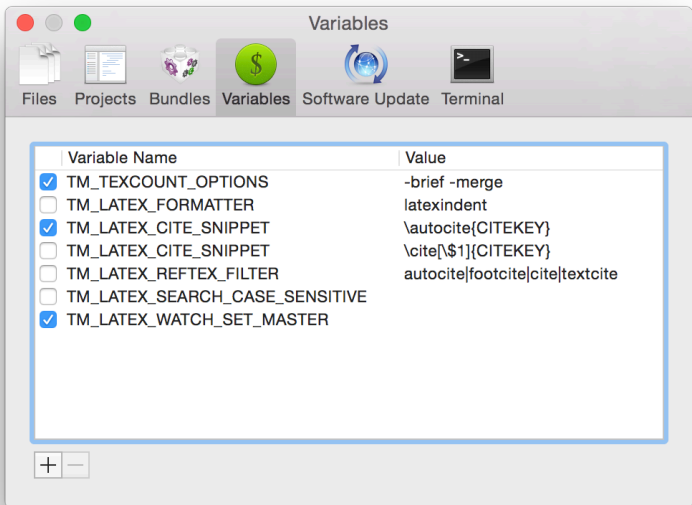

Figure 7: Set environment variables under | TextMate 〉 Preferences... 〉 Variables |

TM_LATEX_CITE_SNIPPET

You can change the default substitution text used by | Insert 〉 〉 Citation Based on Current Word | — \cite{CITEKEY} — via this variable. Every occurrence of CITKEY specified via this variable will be replaced with the chosen citekey. This variable support snippet syntax.

TM_LATEX_FORMATTER

This variable lets you specify the program used in the command Tidy. Currently the only option besides the default formatter LaTeX_Tidy is latexindent.

TM_LATEX_REFTEX_FILTER

This is a regular expression that lets you filter the entries displayed by Insert ⟩ Citation (Ref-TeX Style). Figure 8 shows you the output of the command without this variable on the left, and with the variable set to

```
autocite|footcite|cite|textcite
```

on the right.

Figure 8: TM_LATEX_REFTEX_FILTER lets you filter the menu entries of Insert ⟩ Citation (Ref-TeX Style)

TM_LATEX_SEARCH_CASE_SENSITIVE

The commands:

- Insert ⟩ Citation Based on Current Word
- Insert ⟩ Citation (Ref-TeX Style)
- Insert ⟩ Label Based on Current Word

interpret the current word/selection as case-insensitive regular expression by default. They use this expression to filter the possible entries they display. To interpret the selection instead as case sensitive expression set this variable (to any value you like).

TM_LATEX_WATCH_SET_MASTER

If you set this variable — you can use any value you like — then Watch Document sets TM_LATEX_MASTER to the path of the watched file. For this purpose the command creates a `.tm_properties` file in the folder of the watched file.

This works only if there is no such file in the folder already. If your project folder already contains a `.tm_properties` file, then you can of course still set the master file yourself. Just define the `TM_LATEX_MASTER` variable. For a description of how to do that, please just consult the section "Using the TM_LATEX_MASTER Environment Variable".

TM_TEXCOUNT_OPTIONS

This environment variable specifies additional options for the program `texcount`. The LaTeX bundle uses this tool to print statistics about the word count of the current master file. Figure 9 shows the output with the default options `–nocol –utf8 –inc` on the left, and the *additional* options `–brief –merge` on the right.

```
File: makeglossaries.tex
Words in text: 18
Words in headers: 3
Words outside text (captions, etc.): 0
Number of headers: 2
Number of floats/tables/figures: 0
Number of math inlines: 0
Number of math displayed: 0
Subcounts:
  text+headers+captions (#headers/#floats/#inlines/#displayed)
  0+2+0 (1/0/0/0) Chapter: Adventure Time
  18+1+0 (1/0/0/0) Section: Overview
```

```
18+3+0 (2/0/0/0) File: makeglossaries.tex
```

Figure 9: `TM_TEXCOUNT_OPTIONS` lets you change the output of Project Management ⟩ ⟩ TeXcount

7 · Credits

There were at least two or possibly three versions of a LaTeX bundle floating around in the early days of TextMate by: Normand Mousseau, Gaetan Le Guelvouit and Andrew Ellis. At some point – January 2005 – Eric Hsu pulled together the threads into one package. From then on there have been contributions by Sune Foldager, Brad Miller, Allan Odgaard, Jeroen van der Ham, Robin Houston, Haris Skiadas and many other contributors.

Happy LaTeXing!