

# Geany $\text{\LaTeX}$ – A $\text{\LaTeX}$ plugin for Geany

Version 0.5

Frank Lanitz

[frank@frank.uvena.de](mailto:frank@frank.uvena.de)

June 12, 2010

**Note:** Please note that this document has been created on June 12, 2010. If you are using a devel version from SVN, please compile and check `doc/geanylatex.tex` from sources. Please check Page 4, Section 4.2 on how to do this.

## Contents

List of Tables	III
List of Figures	III
1 About the plugin	1
2 News & ChangeLog	2
3 Requirements	3
4 Installation	3
4.1 Compiling the plugin . . . . .	3
4.2 Compiling the documentation . . . . .	4
5 Usage	4
6 Features	4
6.1 $\text{\LaTeX}$ -Wizard . . . . .	5
6.1.1 General usage of wizard . . . . .	5
6.1.2 Default templates . . . . .	6
6.1.3 Extending by own templates . . . . .	6
6.2 Inserting References and Labels . . . . .	7
6.3 BibTeX templates for catalogue entries . . . . .	8
6.4 Replacement of special characters . . . . .	8
6.5 Inserting of special character . . . . .	9

6.6	Inserting of Environment . . . . .	9
6.7	Format . . . . .	9
6.8	Autocompletion of <code>\begin</code> and <code>\begingroup</code> . . . . .	10
6.8.1	Usage of feature . . . . .	10
6.9	Inserting <code>\usepackage{}</code> -entry to header . . . . .	10
<b>7</b>	<b>Configuration</b>	<b>11</b>
7.1	GeanyLaTeX's configuration dialog . . . . .	11
7.1.1	Use KOMA script by default . . . . .	11
7.1.2	Show extra toolbar . . . . .	11
7.1.3	Modus of autocompletion . . . . .	12
7.2	Key bindings . . . . .	12
7.3	Hidden preferences . . . . .	14
7.3.1	Deactivate toolbar items if document is a non $\text{\TeX}$ -type . . . . .	14
7.3.2	Size of context for autocompletion . . . . .	14
7.3.3	Apply autocompletion only to $\text{\TeX}$ -like files . . . . .	14
7.3.4	Customized reference strings . . . . .	15
7.3.5	Autocompletion of <code>{}</code> after <code>_</code> and <code>^</code> . . . . .	15
7.3.6	Autoadding of <code>{}</code> after a command . . . . .	16
<b>8</b>	<b>Contribution to the plugin</b>	<b>16</b>
8.1	Extending plugin . . . . .	16
8.1.1	Adding a new translation . . . . .	16
8.1.2	Adding a new feature . . . . .	16
8.2	Testing & bug reporting . . . . .	16
8.3	Packaging . . . . .	17
8.4	Improving and extending of documentation . . . . .	17
8.5	Providing additional data for plugin . . . . .	17
8.6	Propaganda . . . . .	17
<b>9</b>	<b>Development</b>	<b>17</b>
9.1	Development version . . . . .	17
9.1.1	Sending a patch . . . . .	17
<b>10</b>	<b>Known issues</b>	<b>18</b>
<b>11</b>	<b>Recommendations to improve work with <math>\text{\LaTeX}</math> and Geany</b>	<b>18</b>
11.1	Geany's build system . . . . .	18
11.1.1	Document backward search . . . . .	18
11.2	Geany's template system . . . . .	18
11.3	Geany's code snippet function . . . . .	18
11.4	Other useful plugins . . . . .	19
11.4.1	GeanyLipsum . . . . .	19
11.4.2	geanyVC . . . . .	19
11.4.3	Spellcheck . . . . .	19
11.4.4	tasks out of the addons plugins . . . . .	19

<b>12 License</b>	<b>20</b>
<b>13 Bugs, questions, homepage</b>	<b>20</b>

## List of Tables

1	List of available symbols on custom templates . . . . .	6
2	List of available keybindings . . . . .	13

## List of Figures

1	Plugin manager with Geany $\text{\LaTeX}$ 0.5 at Geany 0.19 . . . . .	4
2	$\text{\LaTeX}$ -Wizard of version 0.5 . . . . .	5
3	Insert label dialog on Geany $\text{\LaTeX}$ 0.5 . . . . .	7
4	Insert reference dialog on Geany $\text{\LaTeX}$ 0.5 . . . . .	7
5	Insert environment dialog on Geany $\text{\LaTeX}$ 0.5 . . . . .	9
6	Dialog for inserting <code>\usepackage{}</code> . . . . .	11
7	Plugin toolbar of Geany $\text{\LaTeX}$ 0.5 . . . . .	11
8	general.conf example for contextsize . . . . .	14
9	general.conf example for deactivating file type specific restrictions for autocompletion	15

# 1 About the plugin

Geany $\text{\LaTeX}$  is a little plugin to improve  $\text{\LaTeX}$  support in Geany. It implements a couple of hopefully useful functions:

- A wizard to create new  $\text{\LaTeX}$  documents in a fast and easy way with a bunch of templates available
- A front end for adding labels  $\backslash\text{label}$  and references  $\backslash\text{ref}$  and  $\backslash\text{pageref}$  getting suggestions from the document's aux file
- Insertion of special characters through the menu
- Help on BibTeX entries by templates
- Easy insertion of format patterns like  $\backslash\text{texttt}$  through the menu
- Support of environment insertion by offering a dialog and recognizing selections
- Shortcuts for inserting  $\backslash\text{item}$  and  $\backslash\text{newline}$
- Toolbar with frequently used format options
- A couple of useful auto-completion functions during typing

## 2 News & ChangeLog

### Geany $\text{\LaTeX}$ 0.5 – 2010-06-13

- Introduced custom templates for  $\text{\LaTeX}$ -Wizard
- Added a  $\text{\LaTeX}$ -Wizard icon to the toolbar
- Added shortcuts for inserting common list environments like `enumerate`, `itemize` and `description`
- Some general bugfixes and improvements. As always, see the ChangeLog or svn log.
- Switched to waf for building the plugin
- Moved some  $\text{\LaTeX}$ -specific stuff out of Geany's core into the plugin. This affects features like
  - Autocompletion of `\end{}` and `\endgroup{}`
- Proceeded to Geany Plugin API v184
- Made reference insertion configurable.
- Added an function to insert `\usepackage{}` into header of file
- Automatic adding of `{}` after typing of `_` and `^`
- Added automatic inserting of `{}` after typing a command and hitting return in case of none pair is already present

### Geany $\text{\LaTeX}$ 0.4 – 2009-05-26

- Added a toolbar with frequently used format commands
- Added a configuration dialog to configure basic options of the plugin
- Moved documentation into a  $\text{\TeX}$ -document
- Replace `\u-UTF-8` letters by octal coded chars to avoid dependency on C99.
- Added a function to bulk replace special characters inside marked text by keybinding
- Added a function for special characters substitution during typing

## 3 Requirements

**Please note:** This section of documentation is only valid with standalone distribution of Geany $\text{\LaTeX}$ . If you are planning to use the common geany-plugins project, please check documentation over there as there are some specialties you might like to know.

For compiling the plugin yourself, you will need the GTK ( $\geq 2.8.0$ ) libraries and header files. You will also need its dependency libraries and header files, such as Pango, Glib and ATK. All these files are available at <http://www.gtk.org>.

And obviously, you will need to have Geany with its header files installed (in case you are compiling the plugin on your own). If you installed Geany from the sources, you should be ready to go. If you used a prepared package, e.g. from your distribution, you probably need to install an additional package, probably called geany-dev or geany-devel. Please note that in order to compile and use this plugin, you need Geany 0.19svn or later (Geany Plugin API v184 or higher).

Furthermore you need, of course, a C compiler and python installed. The GNU version of the C compiler is recommended. Furthermore, there should be a working  $\text{\LaTeX}$ -environment on your System.

There is no special need in RAM or CPU, so the plugin should compile and run on all systems Geany is able to run.

## 4 Installation

**Please note:** This section of documentation is only valid for the standalone distribution of Geany $\text{\LaTeX}$ . If you are about to use the common geany-plugins project, please check their documentation as there are some specialties you might like to know.

### 4.1 Compiling the plugin

Compiling and installing the code is done by the following three commands:

```
$ ./waf configure
$ ./waf build
$ ./waf install %$
```

For more configuration details, run `./waf --help`

By default, the plugin is installed into the `lib` subfolder of your Geany installation. So if you have installed Geany to `/usr/local/` the plugin will be installed to `/usr/local/lib/geany/`. Translation files will be installed to `/usr/local/share/locale/` in this case.

If there are any errors during compilation, check your build environment and try to find the error, otherwise contact one of the authors<sup>1</sup>

---

<sup>1</sup>Contact data can be found in Chapter 13, page 20.

## 4.2 Compiling the documentation

Sources of this documentation are available through `doc/geanylatex.tex` inside source tree. To compile the sources, usage of `pdflatex` (should be delivered with your favorite  $\text{\LaTeX}$  distribution) is recommended. For compiling into HTML format you might like to use `htlatex`. The HTML version of this documentation shipped with source tarball has been compiled with

```
htlatex geanylatex.tex xhtml -cvalidate -interaction=batchmode
```

## 5 Usage

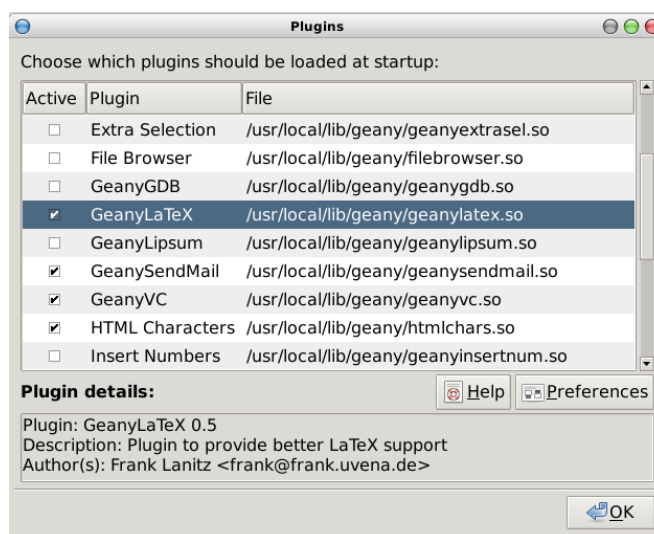


Figure 1: Plugin manager with Geany $\text{\LaTeX}$  0.5 at Geany 0.19

After Geany $\text{\LaTeX}$  has been installed successful the plugin can be loaded through Geany's plugin manager and a new sub menu in the Tools menu will appear as well as new key bindings will be available inside Geany's key binding interface. Inside the sub menu you will find entries for functions supported by this version of the plugin. The main menu entry will be called something like `LaTeX`, depending on your locale.

Also if the option for showing the toolbar is activated on configuration dialog, the toolbar with common used format functions appears on at top of editor widget. This feature is turned off by default.

## 6 Features

Let's go into more detail on some features.

## 6.1 L<sup>A</sup>T<sub>E</sub>X-Wizard

### 6.1.1 General usage of wizard

The L<sup>A</sup>T<sub>E</sub>X-Wizard is implementing a easy way creating a number of default documents.

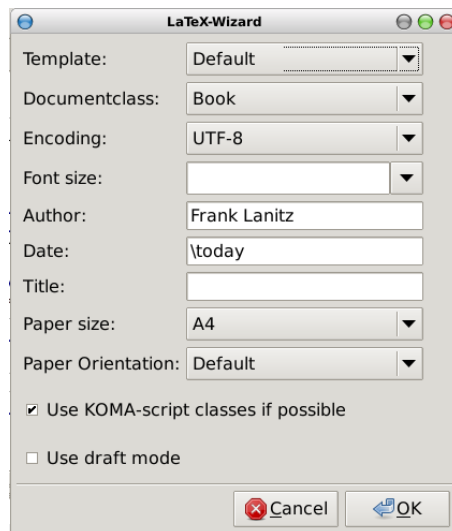


Figure 2: L<sup>A</sup>T<sub>E</sub>X-Wizard of version 0.5

The wizard is offering a chance to choose from a couple of templates with the possibility of adding customer templates. This can be chosen from **Templates** pulldown on top of dialog.

This can be set by choosing the needed entry form **Documentclass** pulldown menu.

**Encoding** is configuring the packages `inputenc` to for example `\usepackage[utf8]{inputenc}` in case of the document encoding should be UTF-8. Also it sets the encoding Geany is using for the newly created document.

**Font size** as well as **Paper size** will set class option for font/paper size of the new created document. **Author**, **Date**, **Title** will be also passed to the corresponding command inside the file header.

Option **Use draft mode** will add `draft` to list of document options which allows some help during debugging of document.

Since KOMA script is quiet popular the option **Use KOMA script if possible** allows to activate the usage of KOMA script. If this options is activated instead of `book`, `scrbook` will be used as document class. Default is activated here. This option is deactivated by default and can be set through GeanyL<sup>A</sup>T<sub>E</sub>X's configuration dialog mentioned earlier in this document.

This wizard can also be called by a shortcut. Please have a look onto Section 2, page 13.



### 6.1.2 Default templates

Document types that are currently supported by the wizard are:

- book
- report
- article
- letter (default letter class)
- presentation (L<sup>A</sup>T<sub>E</sub>X beamer)

### 6.1.3 Extending by own templates

GeanyL<sup>A</sup>T<sub>E</sub>X is offering a way for extending the wizard by user defined templates. This templates will be stored inside the plugin configuration dir with file extension glt. For creating a customized template you will need to create a normal \*.tex file and store it inside the directory. On most Linux systems this should be `~/.config/geany/geanyLaTeX/`.

Inside your template you can refer to wizard's field by using some special strings which are:

Table 1: List of available symbols on custom templates

Symbol	Usage
{CLASSOPTION}	Will be replaced by the classoptions set on the wizard as for example font size or paper size.
{DOCUMENTCLASS}	Will be replaced by the choosen document class based on the pulldown of wizard and whether option for KOMA script has been set.
{DATE}	Will be replaced by the input given on the date field of wizard.
{TITLE}	Will be replaced by the input given on the title field of wizard.
{AUTHOR}	Will be replaced by the input given on the author field of wizard.
{ENCODING}	Will be replace by choosen encoding from pulldown of wizard
{OPENING}	Will be replaced by »Dear Sir or Madame« in local geanyL <sup>A</sup> T <sub>E</sub> X is running with. If you like to overwrite it, please don't use the symbol and hardcode the phrase instead.
{CLOSING}	Will be replaced by »With kind regards« in local geanyL <sup>A</sup> T <sub>E</sub> X is running with. If you like to overwrite it, please don't use the symbol and hardcode the phrase instead.

If you have other than the default templates defined they will be added to templates pulldown. So when creating a template, please keep care to set up a good name for the file, as the filename will be the identifier you can choose from on pulldown.

In future a number of templates should be available also online at <http://frank.uvena.de/files/geany/data/geanyLaTeX/>. Please feel also free to publish templates in case of you have some useful one.

If you need more general templates, you may have a look onto Geany's build in template feature – briefly introduced on Chapter 11.2, page 18.

## 6.2 Inserting References and Labels

An often used feature on writing of documents is adding and referring to labels. Geany $\text{\LaTeX}$  is adding some support here for more comfortable adding new labels and reference offering a GUI.

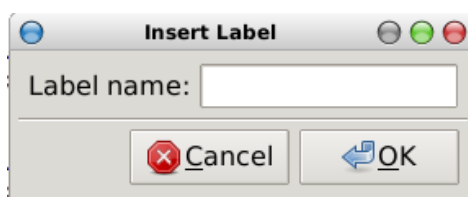


Figure 3: Insert label dialog on Geany $\text{\LaTeX}$  0.5

After an label was added Geany $\text{\LaTeX}$  is offering a dialog for inserting normal references and page references to a label.

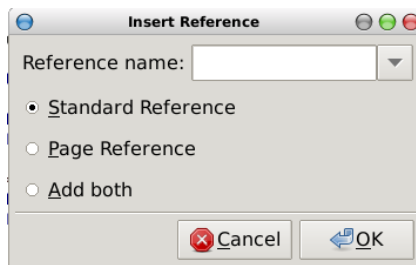


Figure 4: Insert reference dialog on Geany $\text{\LaTeX}$  0.5

The suggestions inside the pull down are based on the aux files creating by processing of \*.tex file located inside directory of current  $\text{\TeX}$ -file. When first step was successful the files are parsed for `\newlabel{\}{\}` and outcome is tried to interpret them properly. The found entries will be inserted into pull down sorted by alphabet.

Both, the inserting labels as well as the inserting reference dialog can be accessed by key binding also. See Chapter 2 here.

### 6.3 BibTeX templates for catalogue entries

Geany<sup>L</sup>A<sub>T</sub>E<sub>X</sub> is offering a number of often used templates for BibTeX catalogue entries. They can be access by the plugin submenu in Geany's tools menu:

- Article
- Book
- Booklet
- Conference
- Inbook
- Incollection
- Inproceedings
- Manual
- Mastersthesis
- Misc
- PhdThesis
- Proceedings
- Techreport
- Unpublished

When choosing an entry from list on menu a template with common used fields will be generated and inserted into the document. The template will be inserted on position of cursor which will no be moved during the process. As an example for a book, this will be inserted to the document:

```
@Book{
Author = {},
Editor = {},
Publisher = {},
Title = {},
Year = {},
}
```

### 6.4 Replacement of special characters

Geany<sup>L</sup>A<sub>T</sub>E<sub>X</sub> is able to replace special characters to their there T<sub>E</sub>X substitute. This can be done in two different ways:

1. **On input:** If this switch is active all special characters will be replaced during typing of text. You can turn the switch on/off at Replacement of special characters submenu inside.
2. **Bulk replace of selected text:** A selected text will be parsed and all known special characters will be replaced by their T<sub>E</sub>X substitute. This can be very useful on importing a large amount of text into your document including characters like ö or ». This function is available through the Replacement of special characters submenu on plugin's submenu of Geany's Tools menu.

For both functions there are also shortcuts available.

## 6.5 Inserting of special character

The plugin is offering a number of special characters with their  $\text{\TeX}$  substitutes to be inserted on easy accessing through the plugin menu.

## 6.6 Inserting of Environment

Geany $\text{\LaTeX}$  is offering a feature for inserting environments into your documents. It can be chosen from a pulldown menu and will be inserted at current position of cursor. If there is a selection active, the selection will be included into environment.

```
\begin{your_environment}  
    ... selected text ...  
\end{your_environment}
```

In case of an empty (= no selection) an empty environment with

```
\begin{your_environment}  
...  
\end{your_environment}
```

will be inserted to the document.

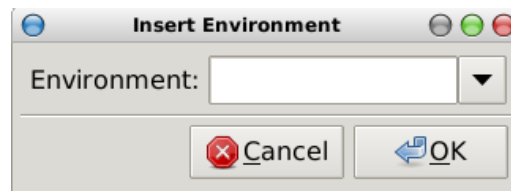


Figure 5: Insert environment dialog on Geany $\text{\LaTeX}$  0.5

## 6.7 Format

Geany $\text{\LaTeX}$  is able to help on formation of text. For doing this its offering you to insert often use format patterns to your document. Patterns that are currently supported are:

- Italic
- Boldfont
- Underline
- Slanted
- Typewriter

- Small Caps
- Emphasis
- Centered
- left-aligned
- right-aligned

Geany $\text{\LaTeX}$  will add the correct format pattern to the document. If there is an selection active, that pattern will be placed around so the selected text will be formatted with this chosen style.

Following items are also accessible using the Geany $\text{\LaTeX}$  toolbar:

- Italic
- Boldfont
- Underline
- Centered
- left-aligned
- right-aligned

## 6.8 Autocompletion of `\begin` and `\begingroup`

Since version 0.5 Geany $\text{\LaTeX}$  is supporting autocompletion for closing `\end` and `\endgroup` for begin commands. Before Geany 0.19 this has been part of the Geany core but has been moved out as it is something  $\text{\LaTeX}$  specific.

### 6.8.1 Usage of feature

After the feature has been enabled (Please check 7.1.3, page 12 here for more detailed information), in every case you enter a `\begin{}` or `\begingroup{}` the plugin will automatically add the fitting `\end{}` or `\endgroup{}` if its not finding a closing tag within the definded context length – by default this means inside following 5 lines. If you like to change this size, please check Chapter 7.3.2, page 14.

This feature is by default file type depending, so it will only work on  $\text{\TeX}$ -like file types.

## 6.9 Inserting `\usepackage{}`-entry to header

From time you need to insert a new package into header of a document, but don't want to change to top of document and scroll back to where you were.

Since version 0.5 Geany $\text{\LaTeX}$  is offering an easy to use dialog which is taking over the package name and possible package options to insert them into header of document. Right now, its placed direct in top of the `\begin{document}` statement if there is any.

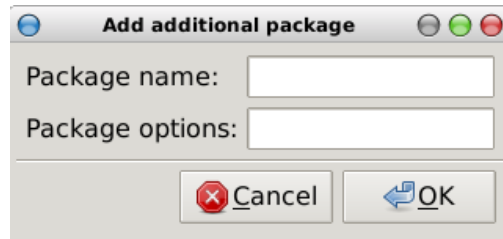


Figure 6: Dialog for inserting `\usepackage{}`

## 7 Configuration

GeanyLaTeX can be configured in two major ways:

1. GeanyLaTeX's configuration dialog
2. Geany's keybindings interface

### 7.1 GeanyLaTeX's configuration dialog

With version 0.4 the configuration dialog is offering two options which can be changed:

#### 7.1.1 Use KOMA script by default

KOMA script bei Markus Kohm is a very popular set of document classes mainly used in Europe. With this option the default setting for e.g.  $\text{\LaTeX}$ -Wizard can be configured<sup>2</sup>. Option is turned off by default.

#### 7.1.2 Show extra toolbar

Decides whether toolbar with some format icons should appear in the top of editor widget. Option is turned off by default. Just give it a try.

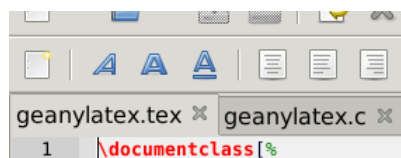


Figure 7: Plugin toolbar of Geany $\text{\LaTeX}$  0.5

---

<sup>2</sup>Currently only position where this option is being used to be honest

### 7.1.3 Modus of autocompletion

Here you can choose, whether the Geany<sup>1</sup>LaTeX should do some autocompletion or not. Values are either

1. Don't care about this inside plugin or
2. Always perform autocompletion on LaTeX

## 7.2 Key bindings

Keybindings which are available:

Table 2: List of available keybindings

Shortcut	Description
Run LaTeX-Wizard	Starts the LaTeX-Wizard for creating a new document
Insert \label	Runs the dialog for inserting a new label into your document.
Insert \ref	Starts an dialog for easy inserting \ref and \pageref into your document. The dialog is supporting easy parsing of aux files so it can suggest a couple of already set labels.
Insert linebreak \\	Inserts a a newline \\ into the document.
Turn input replacement on/off	A shortcut for turning input replacement on or off. When input replacement is activated special characters like ě are replaced by there T <sub>E</sub> X substitute like \v{e}
Replacement of special characters	A selected text will be parsed and all known special characters will be replaced by their T <sub>E</sub> X substitute. This can be very useful on importing a large amount of text into your document including characters like ö or ».
Run insert environment dialog	Runs a dialog for easy inserting an environment. If there is some text selected, the environment will be placed around.
Insert \item	This shortcut will add an simple \item to the document. This can be very useful during writing of lists with a huge number of items.
Format selection in bold font face	Format a selection with bold font face. This is done be adding \textbf{...} around selection.
Format selection in italic font face	Format a selection with italic font face. This is done be adding \textit{...} around selection.
Format selection in typewriter font face	Format a selection with typewriter font face. This is done be adding \texttt{...} around selection.
Format selection centered	Formats selected text centered on page (uses \centering
Format selection left-aligned	Formats selected text left-aligned on page (uses \raggedleft
Format selection right-aligned	Formats selected text right-aligned on page (uses \raggedright
Insert description list	Inserts an description environment as well as a 1 <sup>st</sup> \item element.
Insert itemize list	Inserts an itemize environment as well as a 1 <sup>st</sup> \item element.
Insert enumerate list	Inserts an enumerate environment as well as a 1 <sup>st</sup> \item element.



## 7.3 Hidden preferences

As not all users need to configure everything on there plugin, Geany  $\text{\LaTeX}$  has some hidden preferences which can be set through command line.

### 7.3.1 Deactivate toolbar items if document is a non $\text{\TeX}$ -type

By default, Geany $\text{\LaTeX}$  is deactivating buttons inside toolbar, which don't make much sense to be applied on non- $\text{\TeX}$  file types. As this is not always wished, its possible to turn this feature off via a hidden preferences.

If you want to do so, just add a new section called `toolbar` into your `general.conf` file of Geany $\text{\LaTeX}$  plugin which stats `glatex_deactivate_toolbaritems_with_non_latex=false`. As a result, your config file could look similar to this:

```
[general]
glatex_set_koma_active=false
glatex_set_toolbar_active=true

[toolbar]
glatex_deactivate_toolbaritems_with_non_latex=false
```

Setting this option back to true will go back to default behaviour.

Please ensure, you reload the plugin once this option has been changed.

### 7.3.2 Size of context for autocompletion

Inside configuration file you can add a value to adjust the size of context, which is being searched for auto-completion of `\end` and `\endgroup`. The default value is 5. If you want to reset it, just add a new line to your configuration file with `glatex_set_autocompletion_contextsize` followed by an integer value. An example could look like this:

```
[general]
glatex_set_koma_active=true
glatex_set_toolbar_active=false
glatex_set_autocompletion=true

[autocompletion]
glatex_set_autocompletion_contextsize=2
```

Figure 8: `general.conf` example for `contextsize`

### 7.3.3 Apply autocompletion only to $\text{\TeX}$ -like files

With this option, you can force Geany $\text{\LaTeX}$  to apply all autocompletion functions also to non- $\text{\TeX}$  file types as for example an C-source code file. As this is only in a very low number of cases a really good idea, the option is by default turned on.

```
[general]
glatex_set_koma_active=true
glatex_set_toolbar_active=false
glatex_set_autocompletion=true

[autocompletion]
glatex_autocompletion_only_for_latex=false
```

Figure 9: general.conf example for deactivating file type specific restrictions for autocompletion

### 7.3.4 Customized reference strings

Geany $\text{\LaTeX}$  is able to insert references to a label where its using some default value. As this value is not always optimal, it can be changed using a hidden preference by setting `glatex_reference_page`, `glatex_reference_chapter` or `glatex_reference_all` inside configuration file as shown inside the example configuration snippet.

```
[general]
glatex_set_koma_active=true
glatex_set_toolbar_active=true

[reference]
glatex_reference_page=\textbf{\pageref{{{reference}}}}
glatex_reference_chapter=\textbf{\ref{{{reference}}}}
glatex_reference_all=\textbf{\ref{{{reference}}}, page \pageref{{{reference}}}}
```

Please take care in this case `{{reference}}` will be replace by label name.

Also `\t`, `\r`, `\n` will be handled as known from C so you will need to add a second `\` in front of in such cases. Even this seems to be annoying on the first hand, it allows you to insert some more complicated constructs over here which might require a new line inside.

### 7.3.5 Autocompletion of `{}` after `_` and `^`

Geany $\text{\LaTeX}$  is able to autocomplete `{}` after typing `_` and `^`. This might be useful on typing mathematic text and formula. However, as this option is turn on by default and it might get annoying you can deactivate it by setting `glatex_set_autobraces` inside `[autocompletion]` section of configuration file. An example which is turning off the feature might can look like this:

```
[general]
glatex_set_koma_active=true
glatex_set_toolbar_active=true
glatex_set_autocompletion=true

[autocompletion]
glatex_set_autobraces=false
```

**Note:** The feature in general is only working, if `glatex_set_autocompletion=true` is also set to `true`.

### 7.3.6 Autoadding of {} after a command

The plugin can autoadd a pair of braces {} on hitting return after typing a command. The function will search for a \ and will stop once it finds a space, some {} or a second \ as on \\. This can be configured also by using the hidden preference `glatex.set_autobraces` described in chapter 7.3.5, page 15.

## 8 Contribution to the plugin

If you like the plugin, there are a number of ways, how to contribute to the development of the plugin.

### 8.1 Extending plugin

#### 8.1.1 Adding a new translation

Currently the plugin is available in English and German language but we are always looking for other translations to. There are two major topics in translation:

1. **Translation of plugin:** Adding a new translation and improving an existing one is easy to do. After catching the source tarball and extracting you can find all needed files inside the `po/` folder. Please contact the authors if you plan to update/add a translation to ensure nobody else is currently working on and avoid double work and to get some further information about translation (see Chapter 13).
2. **Translation of documentation:** Since this document is currently only available in English it would be helpful for not English speaking people to have a translated version. If you like to do an translation, please also contact one of the authors for details (see Chapter 13).

#### 8.1.2 Adding a new feature

New features are always highly welcome. The TODO file inside source code archive gives a good idea of current wished features and which are being worked on. Also you can have a look onto the feature request tracker of geany-plugins project at [http://sourceforge.net/tracker/?group\\_id=222729](http://sourceforge.net/tracker/?group_id=222729) whether you find something interesting. Of course we are also open for not in the sources mentioned before listed items. Just contact one of the authors (see Chapter 13).

When sending a patch which is adding a new feature, please check whether you did also care about some documentation for it. As the user will need some, it might can increase the speed a patch is applied. Of course you should also check chapter 9.1.1, page 17 for maybe some more detailed information before.

### 8.2 Testing & bug reporting

Geany<sup>L</sup>A<sub>T</sub>E<sub>X</sub> is tested mainly on x86 and x86\_64 architecture running GNU/Linux. Also it was tested on some Windows 32 versions like XP SP3 very briefly. Since there are also other systems available, testing on other platforms and maybe reporting of issues is highly appreciate.

## 8.3 Packaging

Geany $\text{\LaTeX}$  is part of the geany-plugins project even though there are releases independent of a major release of the project. Therefore there are two things you can do here:

1. Package the plugin for your operating system or distribution. As you might can imagine, the authors unfortunately cannot support all possible platforms.
2. Help to keep releases and packages of geany-plugins project up to date for current version of Geany.

## 8.4 Improving and extending of documentation

Documentation is never complete. There are spelling mistakes, paragraphs that needs to be extended or rewritten because they are not clear or topics that were missed out at all.

The documentation is written in  $\text{\LaTeX}$  so all you need is to get the tex file from doc folder and add or update the content. After this, just send a diff or complete file to one of the authors.

## 8.5 Providing additional data for plugin

You can also contribute to the plugin's development by providing additional data as for example customized templates for the  $\text{\LaTeX}$ -Wizard. If you build up one, you might like to send it to one of the authors.

## 8.6 Propaganda

And of course, tell others of Geany and this plugin. If you like to do a talk about Geany $\text{\LaTeX}$  and/or Geany in general, there is some code available on <http://git.geany.org/talks/> you might can use as a start point for preparing your own presentation. If your favourite language is not yet available there, please feel free to do your own translation and in best case send your translation to one of Geany's<sup>3</sup> development team so it can be added to archive.

# 9 Development

## 9.1 Development version

You can checkout the current source code from the Subversion repository at Sourceforge.net. Get the code from:

svn checkout <http://geany-plugins.svn.sourceforge.net/svnroot/geany-plugins/trunk/geanylatex>

### 9.1.1 Sending a patch

If you want to create a patch, please respect the license of Geany $\text{\LaTeX}$  as well as intellectual property of third. Patches that should be included to the default distribution must be licensed under the same conditions as Geany $\text{\LaTeX}$  by the copyright owner.

---

<sup>3</sup>Check for addresses <http://www.geany.org>

## 10 Known issues

At time of the the documentation was created no issue were known. Since this is only a snapshot, you will find more recent information for all reported issues bug tracking system of SF at

[http://sourceforge.net/tracker/?group\\_id=222729](http://sourceforge.net/tracker/?group_id=222729)

## 11 Recommendations to improve work with L<sup>A</sup>T<sub>E</sub>X and Geany

Geany is offering a number of nice features that can be used to make daily work more easy without need to write a new plugin or extend GeanyL<sup>A</sup>T<sub>E</sub>X.

### 11.1 Geany's build system

On Geany you can define a couple of commands for the build system to improve work with your source file.

#### 11.1.1 Document backward search

When working on a document it happens taht you find a typing error or some more generic issue on your document. Once this happend, its hard to find the correct position in your tex file. If you are using xdvi you can use the backward search function to jump to the right place of your document. An example configuration line for Geany's build system could look similar to this snippet:

```
xdvi -editor "geany --line %l '%%f'" "%f"
```

### 11.2 Geany's template system

If you don't need a dynamic template as described in Chapter 6.1.3, page 6 you can also use Geany's buildin template function which allows to also add customised templates, including placeholders for e.g. author's name, but in a more general and nonL<sup>A</sup>T<sub>E</sub>X-specific way. Nevertheless you should give it a try as it is useful in many cases. For information on how to create your own template using Geany's built-in feature, please check the manual.

### 11.3 Geany's code snippet function

Geany allows you to define code snippets and re-insert them easily at different places throughout your document.

A possible snippet for snippets.conf could be:

A snapshot of the authors' last version for LaTeX can be found on <http://www.geany.org/Download/Extras>

```
[LaTeX]
frame=\\begin{frame}\\n\\ws%\\frametitle{%cursor%}\\n\\ws%cursor%\\n\\end{frame}
block=\\begin{block}{%cursor%}\\n\\ws%cursor%\\n\\end{block}
itemize=\\begin{itemize}\\n\\ws%\\item %cursor%\\n\\end{itemize}
enumerate=\\begin{enumerate}\\n\\ws%\\item %cursor%\\n\\end{enumerate}
description=\\begin{description}\\n\\ws%\\item %cursor%\\n\\end{description}
```

## 11.4 Other useful plugins

As mentioned before, a number of useful functions are already implemented in other plugins. Below you will find a list with the authors's recommendations. More nice plugins can be found on Geany's plugins page at <http://www.geany.org>.

### 11.4.1 GeanyLipsum

This plugin implements an easy way for inserting Lorem Ipsum text into a document. The length of the inserted text is configurable so the plugin can be very helpful on testing layout.

**Homepage:** <http://frank.uvena.de/en/Geany/geanylipsum/>

### 11.4.2 geanyVC

When working on bigger documents a version control system like Subversion could be useful to keep versions. GeanyVC is adding a easy to use frontend for a number of popular version control systems such as git, Subversion, CVS, Bazaar or Mercurial.

**Homepage:** <http://plugins.geany.org/geanyvc/>

### 11.4.3 Spellcheck

Nobody is perfect - in special with typing mistakes on writing a text. Spellcheck is offering a way on Geany to make usage of a common spellchecking system as aspell, myspell or hunspell. Wrong spelled words can be marked with an red line and the plugin is offering suggestions for correct the word. Unfortunately right now it's not supporting some special things common in  $\text{T}_{\text{E}}\text{X}$  and  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ .

**Homepage:** <http://plugins.geany.org/spellcheck/>

### 11.4.4 tasks out of the addons plugins

A plugin that is recognising TODO or FIXME tags inside a document and allows to easy jump to these entries. This function is similar to the `todo` package but doesn't require recompiling of the document. Recognised tags will be inserted to another tab in Geany's message widget.

**Homepage:** <http://plugins.geany.org/addons/>

## 12 License

Geany<sup>L</sup>A<sup>T</sup>E<sub>X</sub> and all its parts is distributed under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. A copy of this license can be found in the file COPYING included with the source code of this program. If not, you will be able to get a copy by contacting the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

## 13 Bugs, questions, homepage

If you found any bugs or want to provide a patch, please contact Frank Lanitz ([frank\(at\)geany\(dot\)org](mailto:frank(at)geany(dot)org)). Please also do so, if you got any questions and visiting <http://frank.uvena.de/en/Geany/geanylatex/> didn't help you to figure out the answer. Visiting the website is also a good start if you want to check for any update on this plugin.