

Automatic LaTeX Plugin for Vim



MAIN PAGE
DOWNLOADS

NEWS

RSS feeds

 *News*
 *Activity*
 *Releases*

COMMENTS

FEATURE LIST
VIDEOS
REQUIREMENTS
VIM PLUGINS

SCRIPTS

HELP
LINKS
MAILING LIST

STATISTICS

SOURCEFORGE

FEATURE LIST

This list is not exhaustive!

Compilation Features:

- * **background T_EX/L_AT_EX processing** with status info (current page number processed by L_AT_EX in status line - what I call *progress bar*) (see a [video](#))
- * interface for [latexmk](#)
- * **forward/reverse searching** for: **xpdf** (ver. ≥ 3.03), **okular**, **xdvi** (*xpdf, skim, okular, xdvi \rightarrow vim*)
- * **forward searching** (*vim \rightarrow xpdf, skim, okular, xdvi*) : [SyncTex](#)
- * **go to *named destination***: using *hyperref* package you can put named destinations into pdf file, you can go to them with one command from vim (Xpdf only)
- * **Making L_AT_EX projects** with just one command : [MakeLatex](#)
/ bibliography, index, table of contents, lists of: figures, tables, theorems [with *ntheorem* package], algorithms /
- * **Making parts of projects** using [subfiles](#) package (*compilation in background and forward/reverse searching are supported*)
- * **various *debug modes***: silent/debug/Debug/verbose.
/ automatically close *QuickFix window* (: *copen*) if there are no errors in it /
- * [show errors](#), warnings, citation/reference warnings, font warnings, files used ...
/ *error file is automatically read after compilation* /
- * [open log](#), in log file you can *sync the tex source with the log file* (then errors in the source file will be highlighted!) or the *xpdf viewer with the log file*. There are also maps to go to next/previous error/warning/info/... message. And there is syntax file for tex log files.

Editing Features:

- * indentation
- * selection tools for [environments](#) (*vie, vae*) , in line math (*vim, vam*), paragraphs (*vip, vap*)
- * *gw* which runs vim *gw* command over inner paragraph,
- * changing environment names
/ and changes the label if you use the labeling system; this needs activation /
- * changing the star (add remove a star from an environment)
- * : *Dictinoary* an interface to [English Usage Dictionary](#) by J.Trzeciak.

Completions:

- * **[completion](#) for commands** (also user defined in associate files), environments, citations, references, *tikz* commands, keywords, provided abbreviations, and much more
/ there are two completion modes: when you know what

you want with and when you search for a command with Shift+Tab /

Some completion features: /see [here](#)/

- + programable completion for package names, package options (and their values), class names, and class options

- + programable completion for \LaTeX commands and their values

- + math commands are completed only in math mode

- + *tikz* commands/keywords are completed only in *tikz* environments

- + **completion of `\ref`, `\eqref` by numbers from the output** (`\ref{1.3 ...}`) (see a [video](#))

- + completion is connected to **closing last environment/bracket functions** so that you have all under one key by default

- + completion for `\usefont{ }{ }{ }{ }`, `\fonttype{ }`, commands

 - / finds completions in font definition files /

- + **completion for `\cite{ }` supports regular expression** which are passed to the searching tool in bibliography files

- * **closing brackets and environments** / preserves nested environments in 100% /

 - / which is included into completion with key /

- * many useful [maps](#)

 - / some of them are configurable (on the fly): for example you can change if * is added to environments which are put by imaps. /

- * [abbreviations](#) for environments (with *tab completion*)

Moving Tools:

- * [table of contents](#) (see a [video](#))

 - / which allows moving through files, chapters, sections, opening a preview window, but also yanking (copying), pasting section-wise /

- * **tag file:** with [:LatexTags](#) (or just [:Tags](#)) ATP will make a very nice tag file to be used with tag feature of vim

- * list of [labels](#)

 - / as with toc, you can see the context of the label in a preview window /

- * [:GotoLabel](#) command, besides the label name you can pass the label number which appears in the output.

- * [:GotoNamedDest](#) - jump to named destination in Xpdf (named destinations are inserted with `\hypertarget` commands).

- * moving commands to [next environment](#) and [next section](#) (or previous)

 - / configured with vim normal commands: *n* and *N* for repeating /

 - / also to next [in line/displayed] math environment /

 - / support of project files /

- * % **operates on** \$: \$, \(: \backslash) and \[: \backslash] (in plain tex also on \$\$: \$\$), brackets and \begin: \end pairs, which also gets highlighted
- * *L*^A*T*_E*X*/*T*_E*X* oriented normal [*gf*](#) command
 - / *gf* = go to file under cursor, if there is no file under the cursor gives a list of input files /
 - + with nice indentation for the list of files to chose
 - + with *g:atp_developer=1* you can easily open package (.sty) and class (.cls) files

Searching Tools:

- * **reverse/inverse searching**
- * [*tags*](#) - python script to make a tag file.
- * search for definitions (\def) declared in input files ([*:Dsearch*](#))
 - / like [D command but works with multiple line definitions /
- * [**extended search**](#) in bib files
- * **query [AMS Ref](#) web page for the bibliography data** (see a [video](#))
- * white space mapped to _s\+ in searching command lines / and ?:
 - / this is a handy way to search in text which is broken into lines, can be turned off/on with F2 key/

Support for Projects:

- * [**project script**](#) which restores project specific variables (for example: [*b:atp_MainFile*](#), [*b:ListOfFiles*](#) - the list of files included in the project, ...)
 - / The project script name is .project.vim /
 - / It is possible to store many projects in one directory! /
 - / There is no need to make a project file to mark the project directory/
- * Completion/Viewing of parts of the project using the [*subfiles*](#) package. Also inverse/reverse searching is supported.
- * [*:GotoFile*](#) function (replace vim *gf* map), which list all the input files for the main project file and let you go to one of them (it also sets the main file variable for the new buffer to correct value automatically).
- * [*:S*](#) - a command to **search in project files** (like it would be just one long file)
 - + supports flags: bcsW of vim search() function and the vim 'wrapscan' option.
- * [*:Labels*](#) command (which list all defined labels) support project files.
- * [*:TOC*](#) command (Table of Contents) almost supports project files (work in progress) (see a [video](#)).
- * [*:NInput*](#)/[*:PInput*](#) commands : to go to next/previous input file.