

VIM-PLUGIN
perl-support.vim
VERSION 4.12

HOT KEYS

Key mappings for Vim and gVim.
Plugin: <http://vim.sourceforge.net>
Fritz Mehner (mehner@fh-swf.de)

(i) insert mode, (n) normal mode, (v) visual mode

| <i>Load / Unload Perl Support</i> | | |
|-----------------------------------|------------------------------------|-----------|
| \lps | load menues | (n) |
| \ups | unload menues | (n) |
| <i>Help</i> | | |
| \rp \h | read perldoc for word under cursor | (n) |
| \hp | help (plugin) | (n,i) |
| <i>Comments</i> | | |
| \cl | end-of-line comment | (n, v, i) |
| \cj | adjust end-of-line comments | (n, v, i) |
| \cs | set end-of-line comment col. | (n) |
| \cfr | frame comment | (n, i) |
| \cfu | function description | (n, i) |
| \cm | method description | (n, i) |
| \chpl | file header (.pl) | (n) |
| \chpm | file header (.pm) | (n) |
| \cht | file header (.t) | (n) |
| \chpo | file header (.pod) | (n) |
| \ckb | keyword comm. BUG | (n, i) |
| \ckt | keyword comm. TODO | (n, i) |
| \ckw | keyword comm. WARNING | (n, i) |
| \cko | keyword comm. WORKAROUND | (n, i) |
| \ckn | keyword comm. new keyword | (n, i) |
| \cc | code ↔ comment | (n, v) |
| \cb | code block → comment | (n, v) |
| \cn | uncomment code block | (n) |
| \cd | date | (n, i) |
| \ct | date & time | (n, i) |
| \cv | vim modeline | (n, i) |

| <i>Statements</i> | | |
|-------------------|------------------------------------|-----------|
| \sd | do { } while | (n, v, i) |
| \sf | for { } | (n, v, i) |
| \sfe | foreach { } | (n, v, i) |
| \si | if { } | (n, v, i) |
| \sie | if { } else { } | (n, v, i) |
| \se | else { } | (n, v, i) |
| \sei | elsif { } | (n, v, i) |
| \su | unless { } | (n, v, i) |
| \sue | unless { } else { } | (n, v, i) |
| \st | until { } | (n, v, i) |
| \sw | while { } | (n, v, i) |
| \s{ \sb | { } | (n, v, i) |
| <i>Idioms</i> | | |
| \id \ \$ | my \$; | (n, i) |
| \id= \ \$= | my \$ = ; | (n, i) |
| \idd \ \$ \$ | my (\$, \$); | (n, i) |
| \ia \ @ | my @; | (n, i) |
| \ia= \ @= | my @ = (, ,); | (n, i) |
| \ih \ % | my %; | (n, i) |
| \ih= \ %= | my % = (=> , => ,); | (n, i) |
| \ir | my \$rgx_ = q//; | (n, i) |
| \im | \$ =~ m//xm | (n, i) |
| \is | \$ =~ s//xm | (n, i) |
| \it | \$ =~ tr//xm | (n, i) |
| \isu | subroutine | (n, v, i) |
| \ifu | | (n, v, i) |
| \ip | print "... \n"; | (n, i) |
| \ii | open input file | (n, v, i) |
| \io | open output file | (n, v, i) |
| \ipi | open pipe | (n, v, i) |
| <i>Snippet</i> | | |
| \nr | read code snippet | (n) |
| \nw | write code snippet | (n, v) |
| \ne | edit code snippet | (n) |
| \ntl | edit local templates | (n) |
| \ntg | edit global templates ¹ | (n) |
| \ntr | reread the templates | (n) |

| <i>Regular Expressions</i> | | |
|--------------------------------|---------------------------------|--------|
| \xr | pick up Regex | (n, v) |
| \xs | pick up string | (n, v) |
| \xf | pick up flag(s) | (n, v) |
| \xm | match | (n) |
| \xmm | match multiple (Regex/target) | (n) |
| \xe | explain Regex | (n, v) |
| <i>POSIX Character Classes</i> | | |
| \pa | [:alnum:] | (n, i) |
| \ph | [:alpha:] | (n, i) |
| \pi | [:ascii:] | (n, i) |
| \pb | [:blank:] | (n, i) |
| \pc | [:cntrl:] | (n, i) |
| \pd | [:digit:] | (n, i) |
| \pg | [:graph:] | (n, i) |
| \pl | [:lower:] | (n, i) |
| \pp | [:print:] | (n, i) |
| \pn | [:punct:] | (n, i) |
| \ps | [:space:] | (n, i) |
| \pu | [:upper:] | (n, i) |
| \pw | [:word:] | (n, i) |
| \px | [:xdigit:] | (n, i) |
| <i>POD</i> | | |
| \pod | run podchecker | (n) |
| \podh | convert POD data to .html file | (n) |
| \podm | Convert POD data to *roff input | (n) |
| \podt | Convert POD data to ASCII text | (n) |
| <i>Profiling</i> | | |
| \rps | run SmallProf | (n) |
| \rpf | run FastProf | (n) |
| \rpn | run NYTProf | (n) |
| \rpnc | open CSV file (NYTProf) | (n) |
| \rpnh | browse HTML files (NYTProf) | (n) |

¹ systemwide installation only

| <i>Run</i> | | |
|-------------|--|---------------|
| \rr | update file, run script | (n) |
| \rs | update file, check syntax | (n) |
| \ra | set command line arguments | (n) |
| \rw | set Perl cmd. line switches | (n) |
| \rm | run make | (n) |
| \rma | command line arguments for make | (n) |
| \rd | start debugger | (n) |
| \re | make script executable | (n) |
| \ri | show installed Perl modules | (n) |
| \rg | generate Perl module list | (n) |
| \ry | run perltidy | (n, v) |
| \rc | run perlritic | (n) |
| \rt | save buffer with timestamp | (n) |
| \rh | hardcopy buffer | (n, v) |
| \rk | settings and hotkeys | (n) |
| \rx | set xterm size | (n, GUI only) |
| \ro | change output destination | (n) |

perlritic

Ex commands for **perlritic** (version 1.01+)

Use tab expansion to choose the severity or the verbosity.

```
:CriticSeverity 1      2      3      4      5
                  brutal cruel harsh stern gentle
:CriticVerbosity 1...11
:CriticOptions  option(s), see perlritic(1)
```

Profiling

The following ex commands can be used to sort a profiler report in the quickfix window.

Use tab expansion to choose the sort criterion or the file name.

For **Devel::SmallProf**

```
:SmallProfSort  file-name|line-number|line-count|time|ctime
```

For **Devel::FastProf**

```
:FastProfSort  file-name|line-number|time|line-count
```

For **Devel::NYTProf**

```
:NYTProfCSV      Read a CSV-file.
:NYTProfHTML     Read the HTML-reports with an external viewer (GUI only).
:NYTProfSort     file-name|line-number|time|calls|time-call
```