

VIM-PLUGIN  
**perl-support.vim**  
VERSION 4.13

# HOT KEYS

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

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

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

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

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

<sup>1</sup> systemwide installation only

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

## perlcritic

Ex commands for **perlcritic** (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 perlcritic(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
```