## VIM-PLUGIN bash-support.vim VERSION 3.12

## 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

[n]\c1         end-of-line comments         (n, i, v)           [n]\c2         adjust end-of-line comments         (n, i, v)           \cs         set end-of-line comment column         (n)           \cfr         frame comment         (n, i)           \cfu         function description         (n, i)           \check         file header         (n, i)           \ckb         keyword BUG         (n, i)           \ckt         keyword TODO         (n, i)           \ckt         keyword TRICKY         (n, i)           \ckw         keyword WARNING         (n, i)           \ckn         keyword TRICKY         (n, i)           \ckn         keyword warning         (n, i, v)           \cdot         date         (n, i, v)           \cdot         date         (n, i, v)           \cdot         date         (n, i, v)           \cdot         date & time         (n, i, v)           \cdot         date & time         (n, i, v)           \cdot         date & time         (n, i, v)           \cdot         (r)         (n, i, v)           \cdot         (r)         (r)           \cdot         (r)         (r)		<b>C</b> o	mments
\cs         set end-of-line comment column         (n, i)           \cfr         frame comment         (n, i)           \cfu         function description         (n, i)           \ckb         keyword BUG         (n, i)           \ckb         keyword TODO         (n, i)           \ckr         keyword TRICKY         (n, i)           \ckw         keyword WARNING         (n, i)           \ckw         keyword: new keyword         (n, i, v)           \cd         date         (n, i, v)           \cd         echo "<1ine>"         (n, i, v)           \cd         date         (n, i, v)           \cd         date         (m, i, v)           \cd         echo "<1ine>"         (n, i, v)           \cd         keyword comment         (tab compl.)         (n, i)           \cd	[n]\cl	end-of-line comment	(n, i, v)
\cfr         frame comment         (n, i)           \cfu         function description         (n, i)           \ch         file header         (n, i)           \ckb         keyword BUG         (n, i)           \ckt         keyword TODO         (n, i)           \ckt         keyword TRICKY         (n, i)           \ckw         keyword: new keyword         (n, i)           \ck         date         (n, i, v)           \ck         date         (n, i, v)           \ck         date         (m, i, v)           \ck         date         (m, i, v)           \sc         case in	[n]\cj	adjust end-of-line comments	(n, i, v)
\cfu   function description	\cs	set end-of-line comment column	(n)
\ch         file header         (n, i)           \ckb         keyword BUG         (n, i)           \ckt         keyword TODO         (n, i)           \ckr         keyword TRICKY         (n, i)           \ckw         keyword WARNING         (n, i)           \ckn         keyword: new keyword         (n, i)           \ckn         keyword: new keyword         (n, i)           \ckn         keyword: new keyword         (n, i)           \cdot         date         (n, i, v)           \cdot         echo "<1ine>"         (n, i)           \cdot         echo "<1ine>"         (n, i)           \cdot         echo "<1ine>"         (n, i)           \cdot         keyword comment         (tab compl.)         (n, i)           \cdot         keyword comment         (tab compl.)         (n, i)           \sec         case in esac         (n, i)           \sec         case in esac	\cfr	frame comment	(n, i)
\ckb         keyword BUG         (n, i)           \ckt         keyword TODO         (n, i)           \ckr         keyword TRICKY         (n, i)           \ckw         keyword WARNING         (n, i)           \ckw         keyword inew keyword         (n, i)           \ckn         keyword: new keyword         (n, i)           \ckn         (n, i, v)           \cd         date         (n, i, v)           \cd         echo " <li>'<li>'<li>(n, i, v)           \cd         echo "<li>'         (n, i, v)           \se         case in esac         (n, i, v)           \se         for in do done         (n, i, v)           \sie         if then fi         (n, i,</li></li></li></li>	\cfu	function description	(n, i)
\ckt         keyword TODO         (n, i)           \ckr         keyword TRICKY         (n, i)           \ckw         keyword WARNING         (n, i)           \ckn         keyword: new keyword         (n, i)           \ckn         keyword: new keyword         (n, i)           \ckn         toggle comment         (n, i, v)           \cd         date         (n, i, v)           \cd         date & time         (n, i, v)           \center center         (n, i)         (n, i)           \center center         (n, i, v)           \set         case in esac         (n, i)           \set         case in esac         (n, i)           \set         case in esac         (n, i)           \set         clif then         (n, i, v)           \sfo         for in do done         (n, i, v)	\ch	file header	(n, i)
\ckr         keyword TRICKY         (n, i)           \ckw         keyword WARNING         (n, i)           \ckn         keyword: new keyword         (n, i)           \notice         toggle comment         (n, i, v)           \cd         date         (n, i, v)           \cd         date         (n, i, v)           \cd         date & time         (n, i, v)           \cd         date & time         (n, i, v)           \cd         echo " <li>line&gt;"         (n, i)           \cdots         ceho "<li>line&gt;"         (n, i)           \cdots         keyword comment         (tab compl.)         (n, i)           \sc         case in esac         (n, i, v)           \sf</li></li>	\ckb	keyword BUG	(n, i)
\ckw         keyword WARNING         (n, i)           \ckn         keyword: new keyword         (n, i)           [n]\cc         toggle comment         (n, i, v)           \cd         date         (n, i, v)           \ct         date & time         (n, i, v)           \ct         date & time         (n, i)           \ct         echo " <li>line&gt;"         (n, i)           \ct         remove echo         (n, i)           \ct         keyword comment         (tab compl.)         (n, i)           \sec         case in esac         (n, i)         (n, i)           \sec         case in esac         (n, i, v)           \sf</li>	\ckt	keyword TODO	(n, i)
\ckn         keyword: new keyword         (n, i)           [n]\cc         toggle comment         (n, i, v)           \cd         date         (n, i, v)           \ct         date & time         (n, i, v)           \ce         echo " <li>line&gt;"         (n, i)           \cr         remove echo         (n, i)           \cs         script sections         (tab compl.)         (n, i)           \ckc         keyword comment         (tab compl.)         (n, i)           \sec         case in esac         (n, i)         (n, i)           \sec         case in esac         (n, i)         (n, i, v)           \sef         for in do done         (n, i, v)           \sie         if then else fi         (n, i, v)           \sum         while do done         (n, i, v)           <t< td=""><td>\ckr</td><td>keyword TRICKY</td><td>(n, i)</td></t<></li>	\ckr	keyword TRICKY	(n, i)
[n]\cc         toggle comment         (n, i, v)           \cd         date         (n, i, v)           \ce         echo " <li>ine&gt;"         (n, i)           \cr         remove echo         (n, i)           \cr         remove echo         (n, i)           \cs         script sections         (tab compl.)         (n, i)           \ckc         keyword comment         (tab compl.)         (n, i)           \sec         case in esac         (n, i)           \sec         for (n, i)         (n, i, v)           \sec         for (()) do done         (n,</li>	\ckw	keyword WARNING	(n, i)
\cd date \( (n, i, v) \) \ct date & time \( (n, i, v) \) \ce echo " <li>\cr remove echo \( (n, i) \) \cs script sections \( (tab compl.) \) \( (n, i) \) \ck keyword comment \( (tab compl.) \) \( (n, i) \) \cv vim modeline \( (n, i) \) \set elif then \( (n, i) \) \set elif then \( (n, i, v) \) \sf for in do done \( (n, i, v) \) \si if then fi \( (n, i, v) \) \si if then else fi \( (n, i, v) \) \ss select in do done \( (n, i, v) \) \su until do done \( (n, i, v) \) \sw while do done \( (n, i, v) \) \sw while do done \( (n, i, v) \) \ss echo -e "" \( (n, i, v) \) \sa array element \( \{ .[ .] \} \) \( (n, i, v) \) \sa array element \( \{ .[ .] \} \) \( (n, i, v) \) \sa subarray \( \{ .[ .] \} \) \( (n, i, v) \) \sa subarray \( \{ .[ .] \} \) \( (n, i, v) \) \sa i list of indices \( \{ .[ .] \} \) \( (n, i, v) \) \sa i list of indices \( \{ .[ .] \} \) \( (n, i, v) \)</li>	\ckn	keyword: new keyword	(n, i)
\cd date \( (n, i, v) \) \ct date & time \( (n, i, v) \) \ce echo " <li>\cr remove echo \( (n, i) \) \cs script sections \( (tab compl.) \) \( (n, i) \) \ck keyword comment \( (tab compl.) \) \( (n, i) \) \cv vim modeline \( (n, i) \) \set elif then \( (n, i) \) \set elif then \( (n, i, v) \) \sf for in do done \( (n, i, v) \) \si if then fi \( (n, i, v) \) \si if then else fi \( (n, i, v) \) \ss select in do done \( (n, i, v) \) \su until do done \( (n, i, v) \) \sw while do done \( (n, i, v) \) \sw while do done \( (n, i, v) \) \ss echo -e "" \( (n, i, v) \) \sa array element \( \{ .[ .] \} \) \( (n, i, v) \) \sa array element \( \{ .[ .] \} \) \( (n, i, v) \) \sa subarray \( \{ .[ .] \} \) \( (n, i, v) \) \sa subarray \( \{ .[ .] \} \) \( (n, i, v) \) \sa i list of indices \( \{ .[ .] \} \) \( (n, i, v) \) \sa i list of indices \( \{ .[ .] \} \) \( (n, i, v) \)</li>	[n]\CC	toggle comment	(n, i, v)
\ce echo " <li>\cr remove echo (n, i) \cr remove echo (n, i) \css script sections (tab compl.) (n,i) \ckc keyword comment (tab compl.) (n,i) \cv vim modeline (n, i)  \sc case in esac (n, i) \sei elif then (n, i, v) \sf for in do done (n, i, v) \sf for in do done (n, i, v) \si if then fi (n, i, v) \si if then else fi (n, i, v) \ss select in do done (n, i, v) \su until do done (n, i, v) \su until do done (n, i, v) \sw while do done (n, i, v) \sw while do done (n, i, v) \sy printf "%s" (n, i, v) \sa array element \${.[.]} (n, i, v) \sa array elem.s (all) \${.[@]} (n, i, v) \sa array elem.s (1 word) \${.[*]} (n, i, v) \sa array elem.s \${.[@]::} (n, i, v) \sa array elem.s \${.[@]::} (n, i, v) \sa array elem.s \${.[@]} (n, i, v)</li>	\cd		(n, i, v)
\ce echo " <li>\cr remove echo (n, i) \cr remove echo (n, i) \css script sections (tab compl.) (n,i) \ckc keyword comment (tab compl.) (n,i) \cv vim modeline (n, i)  \sc case in esac (n, i) \sei elif then (n, i, v) \sf for in do done (n, i, v) \sf for in do done (n, i, v) \si if then fi (n, i, v) \si if then else fi (n, i, v) \ss select in do done (n, i, v) \su until do done (n, i, v) \su until do done (n, i, v) \sw while do done (n, i, v) \sw while do done (n, i, v) \sy printf "%s" (n, i, v) \sa array element \${.[.]} (n, i, v) \sa array elem.s (all) \${.[@]} (n, i, v) \sa array elem.s (1 word) \${.[*]} (n, i, v) \sa array elem.s \${.[@]::} (n, i, v) \sa array elem.s \${.[@]::} (n, i, v) \sa array elem.s \${.[@]} (n, i, v)</li>		date & time	
\css   script sections   (tab compl.)   (n,i) \ \ckc   keyword comment   (tab compl.)   (n,i) \ \cv   vim modeline   (n, i) \	\ce	echo " <line>"</line>	(n, i)
\ckc keyword comment (tab compl.) (n,i) \cv vim modeline (n, i)    Statements	\cr	remove echo	(n, i)
Cev         vim modeline         (n, i)           Statements           \sc         case in esac         (n, i)           \sei         elif then         (n, i, v)           \sf         for in do done         (n, i, v)           \sf         for (()) do done         (n, i, v)           \si         if then fi         (n, i, v)           \si         if then else fi         (n, i, v)           \su         until do done         (n, i, v)           \sw         while do done         (n, i, v)           \sw         while do done         (n, i, v)           \sw         echo -e ""         (n, i, v)           \se         echo -e ""         (n, i, v)           \sa         array element \${.[.]}         (n, i, v)           \sa         arr. elem.s (all) \${.[@]}         (n, i, v)           \sa         subarray \${.[@]::}         (n, i, v)           \sa         subarray \${.[@]::}         (n, i, v)           \sa         list of indices \${.[*]}         (n, i, v)	\css	script sections (tab compl.)	(n,i)
Statements           \sc         case in esac         (n, i)           \sei         elif then         (n, i, v)           \sf         for in do done         (n, i, v)           \sf         for (()) do done         (n, i, v)           \si         if then fi         (n, i, v)           \si         if then else fi         (n, i, v)           \su         until do done         (n, i, v)           \sw         while do done         (n, i, v)           \sw         while do done         (n, i, v)           \sw         echo -e ""         (n, i, v)           \sp         printf "%s"         (n, i, v)           \sa         array element \${.[.]}         (n, i, v)           \sa         arr. elem.s (all) \${.[@]}         (n, i, v)           \sa         subarray \${.[@]::}         (n, i, v)           \sa         list of indices \${.[*]}         (n, i, v)	\ckc		(n,i)
\sc	\cv	vim modeline	(n, i)
\sei elif then		Sta	tements
\sf for in do done	\sc	case in esac	(n, i)
\sfo for (()) do done	∖sei	elif then	(n, i)
\si	\sf	for in do done	(n, i, v)
\sie if then else fi	\sfo	for (()) do done	(n, i, v)
\ss	\si	if then fi	(n, i, v)
\su until do done (n, i, v) \sw while do done (n, i, v) \sfu function (n, i, v) \se echo -e "" (n, i, v) \sp printf "%s" (n, i, v) \sa array element \${.[.]} (n, i, v) \sa arr. elem.s (all) \${.[@]} (n, i, v) \sa1 arr. elem.s (1 word) \${.[*]} (n, i, v) \ssa subarray \${.[@]::} (n, i, v) \ssa no. of arr. elem.s \${.[@]} (n, i, v) \sai list of indices \${.[*]} (n, i, v)  \Test \t1 unary operator, 1 arg.: [-] (n, i)	\sie	if then else fi	(n, i, v)
\sw while do done \( (n, i, v) \) \sfu function \( (n, i, v) \) \se echo -e "" \( (n, i, v) \) \sp printf "%s" \( (n, i, v) \) \sa array element \\${.[.]} \( (n, i, v) \) \sa arr. elem.s (all) \\${.[@]} \( (n, i, v) \) \sa aur. elem.s (1 word) \\${.[*]} \( (n, i, v) \) \sa subarray \\${.[@]::} \( (n, i, v) \) \sa no. of arr. elem.s \\${.[@]} \( (n, i, v) \) \sai list of indices \\${.[*]} \( (n, i, v) \) \Test \( (n, i, v) \)	\ss	select in do done	(n, i, v)
\sfu function	\su	until do done	(n, i, v)
\se echo -e "" (n, i, v) \sp printf "%s" (n, i, v) \sa array element \${.[.]} (n, i, v) \sa arr. elem.s (all) \${.[@]} (n, i, v) \sa1 arr. elem.s (1 word) \${.[*]} (n, i, v) \sa subarray \${.[@]::} (n, i, v) \san no. of arr. elem.s \${.[@]} (n, i, v) \sai list of indices \${.[*]} (n, i, v)  Test \t1 unary operator, 1 arg.: [-] (n, i)	\sw	while do done	(n, i, v)
\sp \ \printf \ \\ss \ \ \sa \ \ \array \element \ \\${.[.]} \ \ (n, i, v) \ \sa \ \ \array \element \ \\${.[.]} \ \ (n, i, v) \ \sa \ \ \arr. \ \elem.s \ (all) \ \\${.[@]} \ \ (n, i, v) \ \sa \ \ \arr. \ \elem.s \ (1 \ \word) \ \\${.[*]} \ \ (n, i, v) \ \sa \ \ \subarray \ \\${.[@]::} \ \ (n, i, v) \ \sa \ \ \sa \ \ \no. \ \of \arr. \ \elem.s \ \\${.[@]} \ \ (n, i, v) \ \sa \ \ \sa \ \ \list \ \of \ind \text{indices} \ \\${.[*]} \ \ (n, i, v) \ \ \ \sa \ \ \ \tan \ \ \tan \ \of \text{indices} \ \\${.[*]} \ \ \ \ (n, i, v) \ \ \tan \ \ \tan \ \tan \ \of \text{indices} \ \\${.[*]} \ \ \ \ \ \ \tan \ \of \text{indices} \ \\${.[*]} \ \ \ \ \ \ \ \ \ \ \ \tan \ \of \text{indices} \ \\${.[*]} \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	\sfu		(n, i, v)
\sa array element \\$\{.[.]\} \ (n, i, v) \ \saa arr. elem.s (all) \\$\{.[@]\} \ (n, i, v) \ \sa1 arr. elem.s (1 word) \\$\{.[*]\} \ (n, i, v) \ \saa subarray \\$\{.[@]::\} \ (n, i, v) \ \saa no. of arr. elem.s \\$\{.[@]\} \ (n, i, v) \ \sai list of indices \\$\{.[*]\} \ (n, i, v) \ \Test \ \t1 unary operator, 1 arg.: [-] \ (n, i)	\se	echo -e	(n, i, v)
\saa arr. elem.s (all) \\$\[.[@]\} (n, i, v) \sa1 arr. elem.s (1 word) \\$\[.[*]\} (n, i, v) \saa subarray \\$\[.[@]::\} (n, i, v) \san no. of arr. elem.s \\$\[.[@]\] (n, i, v) \sai list of indices \\$\[.[*]\] (n, i, v) \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	\sp	printf "%s"	(n, i, v)
\sa1 arr. elem.s (1 word) \${.[*]} (n, i, v) \ssa subarray \${.[@]::} (n, i, v) \san no. of arr. elem.s \${.[@]} (n, i, v) \sai list of indices \${.[*]} (n, i, v) \\  \tag{Test} \t1 unary operator, 1 arg.: [-] (n, i)	\sa		(n, i, v)
\ssa	\saa		
\san       no. of arr. elem.s \${.[@]}       (n, i, v)         \sai       list of indices \${.[*]}       (n, i, v)         Test         \t1       unary operator, 1 arg.: [ - ]       (n, i)	\sa1	arr. elem.s (1 word) \${.[*]}	(n, i, v)
\sai   list of indices \${.[*]} (n, i, v)	∖ssa		(n, i, v)
\sai   list of indices \${.[*]} (n, i, v)   Test   \t1   unary operator, 1 arg.: [ - ] (n, i)	∖san	no. of arr. elem.s \${.[@]}	(n, i, v)
\t1 unary operator, 1 arg.: [ - ] (n, i)	∖sai	list of indices \${.[*]}	
			Test
	\t1	unary operator, 1 arg.: [ - ]	(n, i)
$\mid \  \mid \  \mid $ tz $\mid \  \mid $ binary operator, 2 arg.: $\mid \  \mid \  \mid \  \mid $ $\mid \  \mid $	\t2	binary operator, 2 arg.: [ - ]	(n, i)

	Regular Expressions
\xm	match: [[ =~ ]] (n, i)
	POSIX Character Classes
\pan	[:alnum:] (n, i)
\pal	[:alpha:] (n, i)
\pas	[:ascii:] (n, i)
\pb	[:blank:] (n, i)
\pc	[:cntrl:] (n, i)
\pd	[:digit:] (n, i)
\pg	[:graph:] (n, i)
\pl	[:lower:] (n, i)
\ppr	[:print:] (n, i)
\ppu	[:punct:] (n, i)
\ps	[:space:] (n, i)
\pu	[:upper:] (n, i)
\pw	[:word:] (n, i)
\px	[:xdigit:] (n, i)
	<b>Sn</b> ippets
\nr	read code snippet (n, i)
\nv	view code snippet (n, i)
\nw	write code snippet (n, v, i)
\ne	edit code snippet (n, i)
\ntl	edit local templates (n, i)
\ntg	edit global templates <sup>1</sup> (n, i)
\ntr	reread the templates (n, i)
\nts	switch template style (n, i)
	Run
\rr	update file, run script $(n, i, v^2)$
\ra	set script command line arguments (n, i)
\rc	update file, check syntax (n, i)
\rco	syntax check options (n, i)
\rd	start debugger <sup>2</sup> (n, i)
\re	make script executable <sup>2</sup> (n, i)
\rba	set Bash command line arguments (n, i)
\rh	hardcopy buffer (n, i, v)
\rs	settings and hotkeys (n, i)
\rt	set xterm size <sup>2</sup> (n, i, GUI only)
\ro	change output destination (n, i)
	Help
\hb	Bash manual (n, i)
\hh	help (Bash builtins) (n, i)
\hm	manual (command line utilities) (n, i)
\hbs	Bash-support plugin (n, i)
-	Menu(s)
\lbs	load menus (n & GUI only)
\ubs	unload menus (n & GUI only)
(	Additional Mappings
4	
typin	
,,	single quotes around a WORD (n)
""	double quotes around a WORD (n)

Page 1 / 1

December 26, 2012

bash-support.vim

<sup>1</sup> system-wide installation only 2 Linux/U\*\*x only