

VIM-PLUG-IN  
**bash-support.vim**  
 VERSION 4.3  
**HOT KEYS**

Key mappings for Vim and gVim.

<http://www.vim.org> — Wolfgang Mehner, wolfgang-mehner@web.de

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

<i>Bash</i>		
<b>\bps</b>	parameter substitution (list)	(n, i)
<b>\bsp</b>	special parameters (list)	(n, i)
<b>\ben</b>	environment (list)	(n, i)
<b>\bbu</b>	builtins (list)	(n, i)
<b>\bse</b>	set options (list)	(n, i)
<b>\bso</b>	shopts (list)	(n, i)
<i>Comments</i>		
<b>[n]\cl</b>	end-of-line comment	(n, i, v)
<b>[n]\cj</b>	adjust end-of-line comments	(n, i, v)
<b>\cs</b>	set end-of-line comment col.	(n)
<b>[n]\cc</b>	code → comment	(n, i, v)
<b>[n]\co</b>	uncomment code	(n, i, v)
<b>\cfr</b>	frame comment	(n, i)
<b>\cfu</b>	function description	(n, i)
<b>\ch</b>	file header	(n, i)
<b>\cd</b>	date	(n, i)
<b>\ct</b>	date & time	(n, i)
<b>\css</b>	script sections	(n, i)
<b>\ckc</b>	keyword comments	(n, i)
<b>\cma</b>	plug-in macros	(n, i)
<b>\ce</b>	echo "actual line"	(n, i)
<b>\cr</b>	remove echo from actual line	(n, i)

<i>Statements</i>		
<b>\sc</b>	case in ... esac	(n, i)
<b>\sei</b>	elif then	(n, i)
<b>\sf</b>	for in do done	(n, i, v)
<b>\sfo</b>	for ((...)) do done	(n, i, v)
<b>\si</b>	if then fi	(n, i, v)
<b>\sie</b>	if then else fi	(n, i, v)
<b>\ss</b>	select in do done	(n, i, v)
<b>\su</b>	until do done	(n, i, v)
<b>\sw</b>	while do done	(n, i, v)
<b>\sfu</b>	function	(n, i, v)
<b>\se</b>	echo -e ""	(n, i, v)
<b>\sp</b>	printf "%s"	(n, i, v)
<b>\sae</b>	array element \${.[.]}	(n, i, v)
<b>\saa</b>	arr. elem.s (all) \${.[@]}	(n, i, v)
<b>\sas</b>	arr. elem.s (1 word) \${.[*]}	(n, i, v)
<b>\ssa</b>	subarray \${.[@]::}	(n, i, v)
<b>\san</b>	no. of arr. elem.s \${.[@]}	(n, i, v)
<b>\sai</b>	list of indices \${.[*]}	(n, i, v)
<i>Tests</i>		
<b>\ta</b>	arithmetic tests	(n, i)
<b>\tfp</b>	file permissions	(n, i)
<b>\tft</b>	file types	(n, i)
<b>\tfc</b>	file characteristics	(n, i)
<b>\ts</b>	string comparisons	(n, i)
<b>\toe</b>	option is enabled	(n, i)
<b>\tvs</b>	variables has been set	(n, i)
<b>\tfd</b>	file descr. refers to a terminal	(n, i)
<b>\tm</b>	string matches regexp	(n, i)
<i>IO-Redirection</i>		
<b>\ior</b>	IO-redirections (list)	(n, i)
<b>\ioh</b>	here-document	(n, i)

<i>Pattern Matching</i>		
<b>pzo</b>	zero or one, ?(   )	(n, i)
<b>pzm</b>	zero or more, *(   )	(n, i)
<b>pom</b>	one or more, +(   )	(n, i)
<b>peo</b>	exactly one, @(   )	(n, i)
<b>pae</b>	anything except, !(   )	(n, i)
<b>ppc</b>	POSIX classes	(n, i)
<b>pbr</b>	\${BASH_REMATCH[0..3]}	(n, i)
<i>Snippets</i>		
<b>\nr</b>	read code snippet	(n, i)
<b>\nv</b>	view code snippet	(n, i)
<b>\nw</b>	write code snippet	(n, i, v)
<b>\ne</b>	edit code snippet	(n, i)
<b>\ntl</b>	edit local templates	(n, i)
<b>\ntc</b>	edit custom templates	(n, i)
<b>\ntp</b>	edit personal templates	(n, i)
<b>\ntr</b>	reread the templates	(n, i)
<b>\ntw</b>	template setup wizard	(n, i)
<b>\nts</b>	choose style	(n, i)
<i>Run</i>		
<b>\rr</b>	update file, run script	(n, i, v <sup>1</sup> )
<b>\ra</b>	set script cmd. line arguments	(n, i)
<b>\rba</b>	set Bash cmd. line arguments	(n, i)
<b>\rc</b>	update file, check syntax	(n, i)
<b>\rco</b>	syntax check options	(n, i)
<b>\rd</b>	start debugger <sup>1</sup>	(n, i)
<b>\re</b>	make script executable/not exec. <sup>1</sup>	(n, i)
<b>\rh</b>	hardcopy buffer	(n, i, v)
<b>\rs</b>	plug-in settings	(n, i)
<b>\rx</b>	set xterm size <sup>1,2</sup>	(n, i)
<b>\ro</b>	change output destination	(n, i)
<i>Help</i>		
<b>\hb</b>	display the Bash manual	(n, i)
<b>\hh</b>	help (Bash builtins)	(n, i)
<b>\hm</b>	show manual (cmd. line utilities)	(n, i)
<b>\hp</b>	help (plug-in)	(n, i)

<sup>1</sup> Linux/U\*\*x only  
<sup>2</sup> GUI only