Vim-Plugin

c-support.vim

Version 5.12

Hot keys

Key mappings for Vim with and without GUI.
Plugin: http://vim.sourceforge.net

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

	Comments	
\cl	end-of-line comment	(n,v,i)
\cj	adjust end-of-line comment	(n,v,i)
\cs	set end-of-line comment column	(n)
\c∗	code ⇒ comment /∗ ∗/	(n,v)
\cc	code ⇒ comment //	(n,v)
\co	comment ⇒ code	(n,v)
\cfr	frame comment	(n,i)
\cfu	function comment	(n,i)
\cme	method description	(n,i)
\ccl	class description	(n,i)
\cfdi	file description (implementation)	(n,i)
\cfdh	file description (header)	(n,i)
\ccs	C/C++–file sections (tab compl.)	(n,i)
\chs	H–file sections (tab compl.)	(n,i)
\ckc	keyword comment (tab compl.)	(n,i)
\csc	special comment (tab compl.)	(n,i)
\cd	date	(n,v,i)
\ct	date & time	(n,v,i)

	Statements	
\sd	do { } while	(n,v,i)
\sf	for	(n,i)
\sfo	for { }	(n,v,i)
\si	if	(n,i)
\sif	if { }	(n,v,i)
\sie	if else	(n,v,i)
\sife	if { } else { }	(n,v,i)
\se	else { }	(n,v,i)
\sw	while	(n,i)
\swh	while { }	(n,v,i)
\ss	switch	(n,v,i)
\sc	case	(n,i)
\s{ \sb	{ }	(n,v,i)

	Preprocessor	
\ps	choose a Std. Lib. include	(n,i)
\pc	choose a C99 include	(n,i)
\p<	#include<...>	(n,i)
\p"	#include"..."	(n,i)
\pd	#define	(n,i)
\pu	#undef	(n,i)
\pie	#if #else #endif	(n,v,i)
\pid	#ifdef #else #endif	(n,v,i)
\pin	#ifndef #else #endif	(n,v,i)
\pind	#ifndef #def #endif	(n,v,i)
\pi0	#if 0 #endif	(n,v,i)
\pr0	remove #if 0 #endif	(n,i)
\pe	#error	(n,i)
\pl	#line	(n,i)
\pp	#pragma	(n,i)

	Snippet	
\nr	read code snippet	(n,i)
\nw	write code snippet	(n,v,i)
\ne	edit code snippet	(n,i)
\np	pick up prototype	(n,v,i)
\ni	insert prototype(s)	(n,i)
\nc	clear prototype(s)	(n,i)
\ns	show prototype(s)	(n,i)
\ntl	edit local templates	(n,i)
\ntg	edit global templates[1]	(n,i)
\ntr	reread the templates	(n,i)
\nts	change templates style	(n,i)

	Idioms	
\if	function	(n,v,i)
\isf	static function	(n,v,i)
\im	main()	(n,v,i)
\i0	for(x=0; x<n; x+=1)	(n,v,i)
\in	for(x=n-1; x>=0; x-=1)	(n,v,i)
\ie	enum + typedef	(n,v,i)
\is	struct + typedef	(n,v,i)
\iu	union + typedef	(n,v,i)
\ip	printf()	(n,i)
\isc	scanf()	(n,i)
\ica	p=calloc()	(n,i)
\ima	p=malloc()	(n,i)
\isi	sizeof()	(n,v,i)
\ias	assert()	(n,v,i)
\ii	open input file	(n,v,i)
\io	open output file	(n,v,i)

[1] systemwide installation only

		C++			***R**un*	
`\+co`	`cout << << endl;`	(n,i)	`\rc`	save and compile	(n,i)
`\+"`	`<< ""`	(n,i)	`\rl`	link	(n,i)
`\+c`	class	(n,i)	`\rr`	run	(n,i)
`\+ps`	`#include<...>` STL	(n,i)	`\ra`	set comand line arguments	(n,i)
`\+pc`	`#include<c..>` C	(n,i)	`\rm`	run make	(n,i)
`\+cn`	class (using `new`)	(n,i)	`\rma`	cmd. line arg. for `make`	(n,i)
`\+ci`	class implementation	(n,i)	`\rp`	run `splint`[1]	(n,i)
`\+cni`	class (using `new`) implementation	(n,i)	`\rpa`	cmd. line arg. for `splint`	(n,i)
`\+mi`	method implementation	(n,i)	`\rk`	run CodeCheck[2]	(n,i)
`\+ai`	accessor implementation	(n,i)	`\rka`	cmd. line arg. for CodeCheck	(n,i)
`\+tc`	template class	(n,i)	`\rd`	run `indent`	(n,i)
`\+tcn`	template class (using `new`)	(n,i)	`\rh`	hardcopy buffer	(n,i,v)
`\+tci`	template class implementation	(n,i)	`\rs`	show plugin settings	(n,i)
`\+tcni`	template class (using `new`) impl.	(n,i)	`\rx`	set xterm size (n,i, only Unix & GUI)	
`\+tmi`	template method implementation	(n,i)	`\ro`	change output destination	(n,i)
`\+tai`	template accessor implementation	(n,i)		**Additional Mappings**	

typing	expansion	
`/*`	`/*  */`	(i)
`/*`	`/*` (multiline) marked text `*/`	(v)

`/*<CR>`	`/*`
	` * |`
	` */`
		(i)

`{<CR>`	`{`
	`   |`
	`}`
		(i)

`{<CR>`	`{`
	`   `(multiline) marked text
	`}`
		(v)

`\+tf`	template function	(n,i)
`\+ec`	error class	(n,i)
`\+tr`	`try...catch`	(n,v,i)
`\+ca`	`catch`	(n,v,i)
`\+c.`	`catch(...)`	(n,v,i)
	***H**elp and Menus*	
`\hm`	show manual	(n,i)
`\hp`	help (c-support)	(n,i)
`\lcs`	load Menus	(n & GUI only)
`\ucs`	unload Menus	(n & GUI only)

Ex commands:

CFileSection C/C++–file sections (same as `\ccs`)

HFileSection H–file sections (same as `\chs`)

KeywordComment keyword comment (same as `\ckc`)

SpecialComment special comment (same as `\csc`)

IncludeStdLibrary standard library includes (same as `\ps`)

IncludeC99Library C99 includes (same as `\pc`)

IncludeCppLibrary STL includes (same as `\+ps`)

IncludeCppCLibrary C includes (same as `\+pc`)

CStyle C99 include (same as `\nts`)

Use tab expansion to show the items to choose from.

---

[1] www.splint.org
[2] **CodeCheck**^{TM} is a product of Abraxas Software, Inc.