VIM-PLUGIN

c-support.vim

VERSION 5.19

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
[n]\cl	end-of-line comment		(n,v,i)
[n]\cj	adjust end-of-line comment		(n,v,i)
\cs	set end-of-line comment column		(n)
[n]\c*	code ⇒ comment /* */		(n,v)
[n]\cc	code ⇒ comment //		(n,v,o)
[n]\co	comment ⇒ code		(n,v,o)
\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)
[n]\cx	toggle comments: C ↔ C++		(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)
\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)
\pif	#if #endif	(n,v,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)
[n]\nf [n]\np	pick up function prototype	(n,v,i)
[n]\nm	pick up method 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)
[n]\i0	for(x=0; x<n; x+=1)	(n,v,i)
[n]\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)
\ire	p=realloc()	(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)
\ifs	fscanf	(n,i)
\ifp	fprintf	(n,i)

[1] system-wide installation only

		C++
\+ps	#include<...> STL	(n,i)
\+pc	#include<c..> C	(n,i)
\+c	class	(n,i)
\+cn	class (using new)	(n,i)
\+tc	template class	(n,i)
\+tcn	template class (using new)	(n,i)
\+tf	template function	(n,i)
\+ec	error class	(n,i)
\+ci	class implementation	(n,i)
\+cni	class (using new) implementation	(n,i)
\+tci	template class implementation	(n,i)
\+tcni	template class (using new) impl.	(n,i)
\+mi	method implementation	(n,i)
\+ai	accessor implementation	(n,i)
\+tmi	template method implementation	(n,i)
\+tai	template accessor implementation	(n,i)
\+tr	try...catch	(n,v,i)
\+ca	catch	(n,v,i)
\+c.	catch(...)	(n,v,i)

		*H*elp
\hm	manual for word under cursor	(n,i)
\hp	help (c-support)	(n,i)

		***R**un*
\rc	save and compile	(n,i)
\rl	link	(n,i)
\rr	run	(n,i)
\ra	set comand line arguments	(n,i)
\rm	run make[1]	(n,i)
\rmc	run make clean[1]	(n,i)
\rcm	choose a makefile [1]	(n,i)
\rme	executable to run[1]	(n,i)
\rma	cmd. line arg. for make[1]	(n,i)
\rp	run splint[2]	(n,i)
\rpa	cmd. line arg. for splint	(n,i)
\rk	run CodeCheck[3]	(n,i)
\rka	cmd. line arg. for CodeCheck	(n,i)
\rd	run indent	(n,i)
[n]\rh	hardcopy buffer	(n,i,v)
\rs	show plugin settings	(n,i)
\rx	set xterm size (n,i, only Unix & GUI)	
\ro	change output destination	(n,i)

Additional Mappings

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

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

{<CR>	{
	  |
	}
		(i)

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

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] also working for filetype **make**
[2] *www.splint.org*
[3] **CodeCheck**^{TM} is a product of Abraxas Software, Inc.