VIM PLUGIN
c-support.vim
VERSION 4.3

Plugin: http://vim.sourceforge.net
Fritz Mehner (mehner@fh-swf.de)
November 2006

HOT KEYS

Key mappings for Vim without GUI.

All mappings also work for gVim.

*L*oad / *U*nload C/C++ Support		
\lcs	Load Menus	(normal, GUI only)
\ucs	Unload Menus	(normal, GUI only)

*H*elp		
\h	show plugin help	

*C*omments		
\ce	line end comment /**/	(normal, vis.)
\cn	line end comment //	(normal, visual)
\cl	set end comment column	(normal)
\ci	multiline comment /**/	(normal, vis.)
\c*	code ⇒ comment /**/	(normal, visual)
\c\	code ⇒ comment //	(normal, visual)
\co	comment ⇒ code /**/	(normal, visual)
\cf	frame comment	(normal)
\cu	function comment	(normal)
\cm	method description	(normal)
\ca	class description	(normal)
\cd	date	(normal)
\ct	date & time	(normal)
\cy	change comment style	(normal)

*S*tatements		
\sd	`do { } while`	(normal, visual)
\so	`for`	(normal, visual)
\sr	`for { }`	(normal, visual)
\si	`if`	(normal, visual)
\se	`if else`	(normal, visual)
\sf	`if { }`	(normal, visual)
\sl	`if { } else { }`	(normal, visual)
\sw	`while`	(normal, visual)
\sh	`while { }`	(normal, visual)
\ss	`switch`	(normal, visual)
\sc	`case`	(normal, visual)
\s{	`{ }`	(normal, visual)

*P*reprocessor		
\p<	`#include<...>`	(normal)
\p"	`#include"..."`	(normal)
\pd	`#define`	(normal)
\pu	`#undef`	(normal)
\pi	`#if #else #endif`	(normal, visual)
\pf	`#ifdef #else #endif`	(normal, visual)
\pn	`#ifndef #else #endif`	(normal, visual)
\pe	`#ifndef #def #endif`	(normal, visual)
\p0	`#if 0 #endif`	(normal, visual)
\pr	remove `#if 0 #endif`	(normal)

*I*dioms		
\if	`function`	(normal, visual)
\it	`static function`	(normal, visual)
\im	`main()`	(normal)
\i0	`for( x=0; x<n; x+=1 )`	(normal)
\in	`for( x=n-1; x>=0; x-=1 )`	(normal)
\ie	`enum + typedef`	(normal)
\is	`struct + typedef`	(normal)
\iu	`union + typedef`	(normal)
\ip	`printf()`	(normal)
\ic	`scanf()`	(normal)
\il	`p=calloc()`	(normal)
\ia	`p=malloc()`	(normal)
\iz	`sizeof()`	(normal, visual)
\ir	`assert()`	(normal, visual)
\ii	open input file	(normal)
\io	open output file	(normal)

*S*nippet		
\nr	read code snippet	(normal & GUI only)
\nw	write code snippet	(norm. vis. & GUI only)
\ne	edit code snippet	(normal & GUI only)
\np	pick up prototype	(normal, visual)
\ni	insert prototype(s)	(normal)
\nc	clear prototype(s)	(normal)
\ns	show prototype(s)	(normal)

C++		
\+d	method implementation	(normal)
\+l	class	(normal)
\+n	class (using **new**)	(normal)
\+o	error class	(normal)
\+t	template class	(normal)
\+w	template class (using **new**)	(normal)
\+f	template function	(normal)
\+y	`try ... catch`	(normal, visual)
\+h	`catch`	(normal, visual)
\+.	`catch(...)`	(normal, visual)

		R*un*
\rc	save and compile	(normal)
\rl	link	(normal)
\rr	run	(normal)
\ra	set comand line arguments	(normal)
\rm	run `make`	(normal)
\rg	cmd. line arg. for `make`	(normal)
\rp	run `splint`[1]	(normal)
\ri	cmd. line arg. for `splint`	(normal)
\rk	run `CodeCheck`[2]	(normal)
\re	cmd. line arg. for `CodeCheck`	(normal)
\rd	run `indent`	(normal, visual)
\rh	hardcopy buffer	(normal, visual)
\rs	show plugin settings	(normal)
\rx	set xterm size	(normal, only Unix & GUI)
\ro	change output destination	(normal)

---

[1]`splint` must be installed (`www.splint.org`).
[2]`CodeCheck`^{TM} must be installed.
CodeCheck^{TM} is a product of Abraxas Software, Inc.
(`www.abraxas-software.com`).