

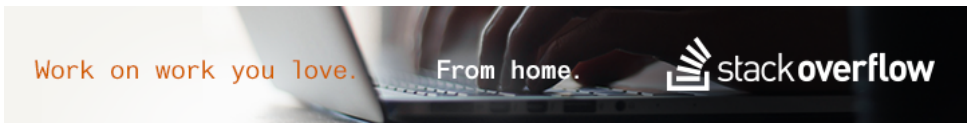
## Announcing Stack Overflow Documentation

We started with Q&A. Technical documentation is next, and we need your help.

Whether you're a beginner or an experienced developer, you *can* contribute.

[Sign up and start helping →](#)[Learn more about Documentation →](#)

## Vim and Ctags tips and tricks [closed]



I have just installed [Ctags](#) (to help with C++ development) with my Vim (or rather gVim), and would like to find out your favorite commands, macros, shortcuts, tips that go along with it...

Share your best arsenal. What other Vim add-ons you would recommend for C++ on Vim development?

**EDIT** What other add-on you would use in conjunction with Ctags?

**EDIT2** What version of gVim you use with tags? Does it make a difference?

**EDIT3** How do you enhance your programming experience for both big and small projects?

[vim](#) [ctags](#) [vim-macros](#)

edited Feb 2 at 12:53



[kenorb](#)

18.2k 7 129 113

asked Feb 19 '09 at 1:37

[Sasha](#)

**locked** by [Jon Clements ♦](#) Jun 20 at 9:44

This question exists because it has historical significance, but **it is not considered a good, on-topic question for this site**, so please do not use it as evidence that you can ask similar questions here. This question and its answers are frozen and cannot be changed. More info: [help center](#).

**closed** as not constructive by [Kev](#) Dec 8 '11 at 1:12

As it currently stands, this question is not a good fit for our Q&A format. We expect answers to be supported by facts, references, or expertise, but this question will likely solicit debate, arguments, polling, or extended discussion. If you feel that this question can be improved and possibly reopened, [visit the help center](#) for guidance.

If this question can be reworded to fit the rules in the [help center](#), please [edit the question](#).

15 Make sure you're using exuberant ctags (a modern implementation) for best results: [ctags.sourceforge.net](#) – [idbrii](#) Feb 23 '11 at 18:59

17 Perhaps the kev is not constructive? – [Floyd](#) Jun 6 '13 at 11:29

27 though closed as not constructive almost two years ago, it is so constructive for me – [sazary](#) Aug 10 '13 at 22:01

15 I think the real reason for closing this question should have been: "This question, despite being enormously helpful, doesn't fit in StackOverflow policy of Q&A. Let this question and answers move to Quora and grow that site while we stubbornly keep closing helpful questions and answers". – [Diego Pino](#) Feb 17 '14 at 11:25





2 I understand why this question was closed, given the rules of SO. But flagging a >400 upvoted question with a >300-upvoted answer for *deletion* is a bit on the grumpy side, isn't it? – [DevSolar](#) Aug 17 '15 at 10:54

## 15 Answers



+ - go to definition  
 + - Jump back from the definition.  
 + + - Open the definition in a horizontal split

Add these lines in vimrc

```
map <C-\> :tab split<CR>:exec("tag ".expand("<cword>"))<CR>
map <A-J> :vsp <CR>:exec("tag ".expand("<cword>"))<CR>
```

 +  - Open the definition in a new tab  
 +  - Open the definition in a vertical split

After the tags are generated. You can use the following keys to tag into and tag out of functions:

 + Left MouseClick - Go to definition  
 + Right MouseClick - Jump back from definition

edited Jun 8 '14 at 4:53



Olivier Pons  
6,828 11 53 104

answered Feb 19 '09 at 4:54



Amjith  
11.3k 5 29 35

- 8 C - ] => (control key) and ] key C - T => (control key) and T key Just for newbies to ctags. Thanks for new tab trick - [kumar](#) Dec 8 '09 at 10:52
- 6 Similarly, after splitting the window with `Ctrl-w C-]` you can do `C-w T` (aka Shift-t) to change that split into a new tab. - [dash-tom-bang](#) Sep 29 '10 at 17:25
- 1 @Will If you are using vim from the terminal then you have to change the terminal preferences to make option key behave like an alt key. If you are using MacVim then add this line to vimrc 'set macmeta'. There is doc for macmeta, :h macmeta. - [Amjith](#) Jan 31 '12 at 13:31
- 17 `g C-]` is very useful. It opens a quick dialog to select one between multiple definitions. - [Vincenzo Pii](#) Mar 22 '12 at 10:23
- 3 `g]` is also useful. You can position the cursor over the tag name and press `g]` to get a list of matching tags. - [zeekvfu](#) Aug 23 '13 at 9:23

Add  projects to your  **stackoverflow** profile.

One line that always goes in my .vimrc:

```
set tags=./tags; /
```

This will look in the current directory for "tags", and work up the tree towards root until one is found. IOW, you can be anywhere in your source tree instead of just the root of it.

edited Dec 16 '11 at 21:52



Conspicuous Compiler  
5,105 24 44

answered Apr 12 '09 at 8:58



uzi  
3,371 2 9 10

- 37 Going up to root is excessive. I suggest only going up to home instead: `tags+=tags;$HOME` - [gotgenes](#) May 12 '09 at 6:03
- 37 The source trees I work on aren't always in my home directory... and being that \$HOME tends to be two directories away from root... well, I can afford two more directory lookups. :) - [uzi](#) May 13 '09 at 19:50
- 6 I agree w/ uzi. If my cwd is /home/me/programming/foo/bar/baz/bang/bam that's only 8 directory lookups. I suppose it depends on your tolerance for speed, but the time to lookup 1 directory isn't noticeably different from 8 on my machine. - [Nate Murray](#) Oct 23 '09 at 15:04
- 1 It does depend on your scenario, I need to use gotgenes suggestion on one of my machines because there are roughly 8,000 directories in /home, and that tends to be a bit sluggish. - [jkerian](#) Feb 22 '11 at 15:50
- 1 Is there something similar for `cscope` ? - [szx](#) Dec 7 '12 at 11:19

Another useful plugin for C development is [cscope](#) Just as Ctags lets you jump to definitions, Cscope jumps to the *calling* functions.

If you have cscope in your ~/bin/ directory, add the following to your .vimrc and use `g^]` to go to the calling function (see :help cscope).

```
if has("cscope")
    set csprg=~/.bin/cscope
    set cst=0
    set cst
    set nocsverb
    " add any database in current directory
    if filereadable("cscope.out")
        cs add cscope.out
    " else add database pointed to by environment
    elseif $CSCOPE_DB != ""
        cs add $CSCOPE_DB
    endif
endif
```

Almost forgot... Just as ctags - you have to generate (and periodically update) the database. I use the following script

```
select_files > cscope.files
ctags -L cscope.files
```

```
ctags -e -L cscope.files
cscope -ub -i cscope.files
```

Where 'select\_files' is another script that extracts the list of C and header files from the Makefile. This way I index only the files actually used by the project.

answered Feb 19 '09 at 15:40



[nimrodm](#)

13.4k 2 39 44

What is the advantage of cscope over ctags? Do they compliment eachother? – [Whaledawg](#) Feb 20 '09 at 4:53

4 They complement each other. Ctags finds the definition of a symbol or a function. Cscope can find all the places calling a function. using both of them makes navigation in large source trees very easy. – [nimrodm](#) Feb 20 '09 at 7:51

1 However, AFAIK cscope has a limited comprehension of C++. – [Luc Hermitte](#) Feb 21 '09 at 11:35

It is limited, and not very good. But for what it does (jumping to USAGES), there isn't any better alternative. Its main problem is that when you want to jump to function calls (f), you often need to jump to anywhere that symbol is used (s) instead and filter the results mentally... – [Greg Rogers](#) Apr 13 '09 at 13:17

2 @Whaledawg Yes, when I press CTRL+] my sound card sends "clap.wav" in output :) – [Patrizio Rullo](#) Feb 2 '11 at 14:14

You can add directories to your ctags lookup. For example, I have a ctags index built for Qt4, and have this in my .vimrc:

```
set tags+=/usr/local/share/ctags/qt4
```

answered Feb 19 '09 at 1:41



[strager](#)

57.4k 13 97 148

All of the above and...

code\_complete : function parameter complete, code snippets, and much more.

- [http://www.vim.org/scripts/script.php?script\\_id=1764](http://www.vim.org/scripts/script.php?script_id=1764)

taglist.vim : Source code browser (supports C/C++, java, perl, python, tcl, sql, php, etc)

- [http://www.vim.org/scripts/script.php?script\\_id=273](http://www.vim.org/scripts/script.php?script_id=273)

edited Jun 6 '11 at 10:52

answered Mar 5 '09 at 19:45



[Johan](#)

7,954 20 69 101

5 I think it would be good to add that taglist.vim is the most downloaded vim plugin. – [phaedrus](#) Nov 28 '09 at 16:09

I use ALT-left and ALT-right to pop/push from/to the tag stack.

```
" Alt-right/left to navigate forward/backward in the tags stack
map <M-Left> <C-T>
map <M-Right> <C-]>
```

If you use hjkl for movement you can map <M-h> and <M-l> instead.

edited Feb 19 '09 at 1:47

answered Feb 19 '09 at 1:40



[strager](#)

57.4k 13 97 148

isn't the same as ctrl -] and ctrl-t -- I guess, your suggestion is a bit cleaner. – [Sasha](#) Feb 19 '09 at 1:47

@Sasha, Pardon? Don't know what you're saying. =X To note, this also works in :help. – [strager](#) Feb 19 '09 at 1:48

2 I think this is more intuitive and cleaner than the default. – [Ayman](#) Sep 10 '09 at 7:56

1 @Ayman But also violates the principle of not having to move the hands off the homerow, like using hjkl vs the arrow keys. – [TankorSmash](#) Aug 29 '13 at 22:38

@TankorSmash, what principle? – [strager](#) Aug 30 '13 at 5:07

### Several definitions of the same name

<C-w>g<C-]> open the definition in a split, but also do :tjump which either goes to the definition or, if there are several definitions, presents you with a list of definitions to choose

from.

answered Apr 22 '10 at 19:06



Heikki Naski

1,184 1 9 11

2 Similar to the first command, vim has a `:ptag` command that opens the definition in a preview window. – [idbrii](#) Nov 24 '11 at 17:24

1 The preview window is nice because you can close it with `:pc` from any other window, without having to first move the cursor to the preview window. – [Heikki Naski](#) Mar 22 '12 at 13:59

There's also a neat autocmd trick with `ptag` at `:h CursorHold-example` which displays the tag for the item under the cursor in the preview window after a configurable timeout. – [dash-tom-bang](#) May 28 '13 at 0:11

+1 for `:tjump` i've now bound `:tjump <C-r><C-w><CR>` works better than `<C-j>` imho Would love to combine this behaviour with `:ptag` – [Andreas Fliesberg](#) Jun 17 '13 at 10:38

`:help` is awesome sometimes... see `:help CTRL-W_g` and `:ptjump` – [Andreas Fliesberg](#) Jun 17 '13 at 11:19

The command I am using most is `C-j` which jumps to the definition of the function under the cursor. You can use it more often to follow more calls. After that, `C-o` will bring you back one level, `C-i` goes deeper again.

answered Feb 19 '09 at 1:45



Raim

3,003 10 24

16 `C-t` goes back one level in the `C-j` stack, while `C-o` goes back to a previous cursor position after a movement. So if you `C-j` to a function then page down, `C-t` will go back immediately while `C-o` will go back to the start of the function, *then* back to the call. – [Greg Hewgill](#) Feb 19 '09 at 2:21

I've found the [taglist](#) plug-in a must-have. It lists all tags that it knows about (files that you have opened) in a separate window and makes it very easy to navigate larger files.

I use it mostly for Python development, but it can only be better for C/C++.

answered Feb 21 '09 at 9:52



Walter

4,757 17 28

I've encapsulated tags manipulation in an [experimental plugin](#) of mine.

Regarding C++ development in vim, I've already answered [there](#): I use my own suite, and a few other plugins.

answered Feb 19 '09 at 15:03



Luc Hermitte

19.1k 3 31 50

I've been adapting my vim plugins for two years to support big enough c++ project. You can take a look at them.

They use ctags and cscope.

[http://www.vim.org/scripts/script.php?script\\_id=1638](http://www.vim.org/scripts/script.php?script_id=1638)  
[http://www.vim.org/scripts/script.php?script\\_id=2507](http://www.vim.org/scripts/script.php?script_id=2507)

edited Nov 14 '13 at 14:47



Jens

35k 8 61 102

answered Mar 5 '09 at 10:50



Mykola Golubyev

32k 8 61 85

I put the following in my `.gvimrc` file, which searches up the tree from any point for a tags file when gvim starts:

```
function SetTags()
  let curdir = getcwd()

  while !filereadable("tags") && getcwd() != "/"
    cd ..
  endwhile

  if filereadable("tags")
    execute "set tags=" . getcwd() . "/tags"
  endif
endfunction
```

```

    execute "cd " . curdir
endfunction

call SetTags()

```

I then periodically regenerate a tags file at the top of my source tree with a script that looks like:

```

#!/bin/bash

find . -regex ".*\.(c|h|hpp|cc|cpp)" -print | ctags --totals --recurse --
extra="+qf" --fields="+i" -L -

```

answered Apr 17 '09 at 23:22



alanwj

209 2 2

- 
- 7 You just re-invented upward search. See :h file-searching. See also [stackoverflow.com/questions/563616/vimctags-tips-and-tricks/...](http://stackoverflow.com/questions/563616/vimctags-tips-and-tricks/...) – gotgenes May 12 '09 at 6:10
- 

I use vim in macos, and the original ctags doesn't work well, so I download newest and configure make make install it. I install ctgas in /usr/local/bin/ctags(to keep original one)

```

"taglist
let Tlist_Ctags_Cmd = "/usr/local/bin/ctags"
let Tlist_WinWidth = 50
map <leader>ta :TlistToggle<cr>
map <leader>bta :!/usr/local/bin/ctags -R .<CR>
set tags=tags;/
map <M-j> <C-]>
map <M-k> <C-T>

```

edited Dec 27 '10 at 6:16

answered Dec 27 '10 at 6:11



Zhenyu Li

113 9

I adapted the SetTags() search function above (which should be replaced by the equivalent `set tags+=./tags;/`) to work for cscope. Seems to work!

```

"cscope file-searching alternative
function SetCscope()
    let curdir = getcwd()

    while !filereadable("cscope.out") && getcwd() != "/"
        cd ..
    endwhile

    if filereadable("cscope.out")
        execute "cs add " . getcwd() . "/cscope.out"
    endif

    execute "cd " . curdir
endfunction

call SetCscope()

```

answered May 4 '11 at 18:50



dergachev

19 1

- 
- 3 As gotgenes said, You just re-invented upward search. See :h file-searching. See [this plugin](#) for a relevant example. – idbrii Nov 24 '11 at 17:25
- 

Another iteration on the SetCscope() function above. That sets cscope pre-path to get matches without being on the dir where "cscope.out" is:

```

function s:FindFile(file)
    let curdir = getcwd()
    let found = curdir
    while !filereadable(a:file) && found != "/"
        cd ..
        let found = getcwd()
    endwhile
    execute "cd " . curdir
    return found
endfunction

if has('cscope')
    let $CSCOPE_DIR=s:FindFile("cscope.out")
    let $CSCOPE_DB=$CSCOPE_DIR."/cscope.out"
    if filereadable($CSCOPE_DB)
        cscope add $CSCOPE_DB $CSCOPE_DIR
    endif

    command -nargs=0 Cscope !cscope -ub -R &
endif

```

answered Jun 13 '11 at 16:53

[saoret.one](#)

21 1

---

3 As gotgenes said, You just re-invented upward search. See :h file-searching. See [this plugin](#) for a relevant example. – [idbrii](#) Nov 24 '11 at 17:30

---