

Vim 中自动加载 cscope.out

下面来分享下自己的 vimrc 脚本关于 cscope 的一部分，该脚本可以实现在项目的任一子目录下，自动的向上查找 cscope.out，并把他 add 进来。

首先来分享一个小脚本，可以帮助我们在项目中生成 cscope.out 和 ctags。

我把这个文件命名为 cs.sh

[Java] [view plaincopy](#)

```
1. #!/bin/sh
2. find . -name "*.h" -o -name "*.c" -o -name "*.cpp" -o -name "*.java" \
3.     >cscope.files
4. cscope -bkq -i cscope.files
5. ctags -R
```

需要注意的是，上面的脚本中，生成的 cscope.files 中保存的是相对路径，而不是绝对路径。这样的话，在 cscope 中查找 tags 的时候，是没法打开相应的包含查找结果的文件的。怎么避免这个问题呢？有如下两个方法：

1， 在 cs.sh 的脚本中，find 命令后接一个变量 PRO_PATH，这个 PRO_PATH 是当前项目的跟目录，这个时候 cscope.files 中保存的就是所有项目文件的绝对路径了。

2，在:cs add 的时候，接一个 prepend path

[Java] [view plaincopy](#)

```
1. :cs add [cscope.out] [pre-path]
```

我用了第二个方法，所以就有了下面的 vimrc 片段，

[Java] [view plaincopy](#)

```
1. if has("cscope")
2.     set csprg=/usr/bin/cscope
3.     set cst=0
4.     set cst
5.     set csverb
6.     set csprg=3
7.     "add any database in current dir
8.     if filereadable("cscope.out")
9.         cs add cscope.out
10.    "else search cscope.out elsewhere
11.    else
```

```
12. let cscope_file=findfile("cscope.out", ".;")
13. let cscope_pre=matchstr(cscope_file, ".*/")
14. if !empty(cscope_file) && filereadable(cscope_file)
15.     exe "cs add" cscope_file cscope_pre
16. endif
17. endif
18. endif
19.
```