
vimtool 参考手册

前言

vimtool 实质上就是一个 vim+插件+配置的工具包，它尽可能实现 Windows 下的 sourceinsight 的功能需求，有人说它配置好了比 sourceinsight 更高效，我个人两个我都喜欢，sourceinsight 是 Windows 下的得力助手，那么 vim 就是 Linux 下的得力助手就是 vim。vimtool 帮你完成大量繁琐的安装配置工作，以优化你的 vim 工作环境为理念。我们不重复造轮子，但我们可以灵活将这些轮子和部件组装成自己心仪的跑车！

vimtool 的原作者 seafly 起初因为在使用虚拟机学习编程过程中，因为频繁用 sourceinsight 通过 samba 访问 Linux 内的项目代码，其中又要有 samba 的支持和网络的各种配置，所以作者在考虑 Linux 下是否也有类似 sourceinsight 这种工具呢？于是作者开始网上寻找并最终锁定自己熟悉的 vim 根据以前的 vim 使用基础经验加上网络上新的操作的不断累积和完善，最终为自己编写自动化脚本，并打包为以后快速搭建 vim 环境而使用。是为不想屡次安装配置 vim 环境的繁琐过程为自己编写的。所以“懒惰使人进步”在这里又似乎变得合理了，哈哈`(*∩_∩*)'。

该脚本的功能非常强大并不断被改进和完善，这里原作者本人也热烈欢迎希望广大 vimtool 受益者能帮忙完善和改进 vimtool 让更多使用人群受益。

目前 vimtool 有开发版和使用版，开发版本唯一明显区别是仅仅就是里面的 vimtool 参考手册为可编辑和多了一份可编辑的专门用于开发的 vimtool 开发手册，根据该手册来新增功能的同时也要修改 vimtool 里面的脚本和配置文件等来实现这个新增功能。该工具包完全免费开源，但是版权必须遵循原作者的版权，这也是尊重原作者劳动成果的一种方式吧。

前面的什么各种插件配置设置什么的大部分工作都已经由 vimtool 帮你完成了！

但你得有一定 Linux 使用基础，否则我这个教程对你也就没什么意义了~

vim 各种操作中如果不熟悉的跟着下面这些操作自己动手跟着实验会更加容易掌握，印象会更加深刻，所以不熟悉的操作多动手跟着下面练练就 OK 了。因为我也是这样过来的。

原作者 QQ：1052061602 或 1989291498，邮箱：seafly0616@qq.com

下面这些问答操作你值得好好看看，因为在实际项目管理中都是用得到的。

vimtool 安装部分

初次使用者建议在解压后的 vimtool 目录下首先执行 `./build_all help` 获取简单的安装帮助信息。并先提前浏览本篇文档大致内容，它能帮助你。

如何完整安装 vimtool？

首先你得有这个 vimtool.tgz 压缩包，

然后打开命令行终端，进入 vimtool.tgz 所在目录，

然后解压这个 vimtool.tgz，

不会解压的看括号内的参考命令 (tar -xzf ./vimtool.tgz)

解压完成之后当前目录会有一个 vimtool 目录，进入这个目录，

然后根据需要安装，

我们这里是选择完整安装，所以直接执行当前目录的 build_all 即可。

完整安装直接执行括号内的命令 (./build_all)

为什么我执行 build_all 时候提示权限不够并无法继续执行？

如图所示：

```
16K -rw-rw-rw-. 1 root root 16K 1月 6 11:32 build_all
4.0K drwxrwxrwx. 2 root root 4.0K 1月 4 12:07 config
4.0K drwxrwxrwx. 2 root root 4.0K 1月 2 20:03 doc
4.0K drwxrwxrwx. 4 root root 4.0K 1月 6 03:10 plugin
4.0K drwxrwxrwx. 2 root root 4.0K 1月 6 11:29 vim
[root@redhat6 vimtool-devel]# ./build_all
-bash: ./build_all: 权限不够
[root@redhat6 vimtool-devel]#
```

答：那是因为这个文件没有可执行权限，你用下面命令加上可执行权限即可：

加可执行权限命令：chmod +x build_all

如何只安装或更新 vim 编辑器？

执行 ./build_all only_vim

如果是更新 vim 版本的话还需要修改 build_all 里配置清单相应位置，

然后再执行 ./build_all only_vim

如何不用安装 vim 主编辑器，只安装插件？

首先你得有这个 vimtool.tgz 压缩包，

然后打开命令行终端，进入 vimtool.tgz 所在目录，

然后解压这个 vimtool.tgz，

不会解压的看括号内的参考命令 (`tar -xzvf ./vimtool.tgz`)

解压完成之后当前目录会有一个 vimtool 目录，进入这个目录，

然后根据需要安装，

我们这里是选择只安装 vim 插件，所以直接执行 `./build_all no_vim` 即可。

只安装插件直接执行括号内的命令 (`./build_all no_vim`)

如何只安装或更新非源码插件？（这里指不需要经过编译的插件）

首先你得有这个 vimtool.tgz 压缩包，

然后打开命令行终端，进入 vimtool.tgz 所在目录，

然后解压这个 vimtool.tgz，

不会解压的看括号内的参考命令 (`tar -xzvf ./vimtool.tgz`)

解压完成之后当前目录会有一个 vimtool 目录，进入这个目录，

然后根据需要安装，

我们这里是选择只安装非源码 vim 插件，所以直接执行当前目录的 `build_all` 即可。

只安装非源码插件直接执行括号内的命令 (`./build_all script_plugin`)

如何只安装源码插件？（这里指需要经过编译的插件）

首先你得有这个 vimtool.tgz 压缩包，

然后打开命令行终端，进入 vimtool.tgz 所在目录，

然后解压这个 vimtool.tgz，

不会解压的看括号内的参考命令 (`tar -xzf ./vimtool.tgz`)

解压完成之后当前目录会有一个 vimtool 目录，进入这个目录，

然后根据需要安装，

我们这里是选择只安装非源码 vim 插件，所以直接执行当前目录的 build_all 即可。

只安装非源码插件直接执行括号内的命令 (`./build_all source_plugin`)

如何为 vim 添加新的非源码插件？或者如何更新原有插件版本？（这里主要讲非源码插件）

方法 01：如果你知道怎么添加就自己添加并配置和调试

方法 02：借助 vimtool 的自动安装脚本来帮你安装

走方法 01 的都知道怎么做了，我们这里主要讲解走方法 02 的操作：

step01 把你要添加的插件丢到 vimtool/plugin/script 目录里，

step02 然后打开命令行终端，进入 vimtool 目录，

step03 根据实际情况给 vimtool/config/vimrc 里的文件添加这个新插件的配置。

step04 然后执行命令 (`./build_all script_plugin`) 即可。

(例如： `./build_all script_plugin`)

如何只更新配置文件？

比如你并没有更新插件或 vim，只是想向配置文件里添加新的配置代码。

方法 01：如果你知道怎么添加就自己添加并配置和调试

方法 02：借助 vimtool 的自动安装脚本来帮你安装

走方法 01 的都知道怎么做了，我们这里主要讲解走方法 02 的操作：

step02 然后打开命令行终端，进入 vimtool 目录，

step03 修改 vimtool/config/里的相应配置文件

step04 然后执行命令 (`./build_all update_config`) 即可。

(例如：`./build_all update_config`)

如何为 vim 添加新插件？或者如何更新原有插件版本？（这里主要讲源码插件）

方法 01：如果你知道怎么添加就自己添加并配置和调试

方法 02：借助 vimtool 的自动安装脚本来帮你安装

走方法 01 的都知道怎么做了，我们这里主要讲解走方法 02 的操作：

step01 把你要添加的插件丢到 vimtool/plugin/source 目录里，

step02 然后打开命令行终端，进入 vimtool 目录，

step03 根据实际情况给 vimtool/config/里的文件添加这个新插件的配置。

step04 然后执行命令 (`./build_all source_plugin`) 即可。

(例如：`./build_all source_plugin`)

(注意包名是不带路径的全名)

我从刚开始使用 vimtool 到现在，已经向里面添加或更新了不少东西，怎么在下次重新搭建的时候可以直接搭建成我目前使用的配置？

答：这个我本人建议你吧 vimtool 目录单独放在一个文件夹里，然后每次需要修改配置或

更新时候就直接在这里面操作，然后及时记得打包备份，然后上传到云盘，下次就直接从你的云盘下载下来使用就可以还原成你目前的配置了。

vimtool 操作部分

如何用 vimtool 开始管理项目代码？

比如我们拿 Linux 内核代码为例，每个项目代码都有一个总目录，比如 `linux-2.6.39` 目录，该目录里面就是关于各种内核源代码等文件。我们称这个总目录名为该内核项目代码的 **顶层目录**。通过 vim 管理的工程都需要先进入这个 **顶层目录**。然后执行后续的工程操作。

如何为目标项目或工程做准备工作？

进入目标项目 **顶层目录** 之后，直接执行 `object.sh` 命令，然后会在当前目录下生成项目依赖文件，生成依赖文件所需时间根据项目大小来定。我这里以 Linux 内核代码为例生成需要 1 分钟左右时间。生成好了就可以打开 vim 了。打开 vim 之后，在 **vim 命令模式** 下分别按下 F3 和 F4 来导入我们前面生成的依赖文件。之后你就可以操作你的工程文件了，这里你还可以按 F12 打开窗口管理器（关闭窗口管理器同样再次按一下 F12）。

如何退出工程管理？

这里我们退出工程管理可不是乱退的，是要有一定顺序退出的。如果我们打开了窗口管理器，那么我们最先关闭窗口管理器（打开和关闭都按 F12），然后如果我们打开了 quickfix 窗

口，那么我们其次就得关闭 quickfix 窗口（vim 命令模式下输入 `:cclose`），其次就是关闭缓冲区，把缓冲区所有标签一个不剩都清除掉，记得所有文件都要保存好了才能顺利完成这些清除工作，然后根据 vim 退出习惯就能退出 vim 了（`:q` 或 `:q!`）

+++++

如何进入 vim 编辑模式？

常用的就如下几种进入编辑模式的方式：

一般直接输入 `i` 就进入，输入之后就从此光标前插入。

也可以输入 `a`，输入之后就从此光标后插入。

也可以输入大写 `A`，输入之后就从此行末尾插入。

也可以输入 `o`，输入之后就在此行下新增一行并把光标移到新增行开头插入。

也可以输入大写的 `O`，输入之后就在此行上方新增一行并把光标移到新增行开头插入。

具体什么效果还要自己去跟着上面做试验去体验一把印象更深刻。

如何进入 vim 命令模式或 normal 模式？

答：这里本人理解的 vim 模式无外乎就两种模式，一种是编辑模式，一种就是命令模式，进入编辑模式的方法有很多种，可以参考网上 vim 基础教程，这里就不讲怎么进入编辑模式了，这里主要讲进入命令模式，直接按 `ESC` 键就进入命令模式了，多按几下 `ESC` 也一样，你开心就好。

vim 方向键有哪些？

vim 方向键也分上下左右键，分别是 `kjhl`（上下左右），如何操作 vim 方向键呢？

就是进入 vim 命令模式下直接按这几个字母就 OK。以后的 vim 方向键就特指这几个字母键，

我们一般不会去操作那个带箭头的方向键，所以建议大家尽早熟悉这几个字母方向键，掌握方法就是多在实际操作中练习就 OK 了。

如何在一行内快速移动光标？

vim 命令模式下输入数字 0 则定位到该行最开始位置（也就是第一列）。

vim 命令模式下输入符号 ^ 则定位到该行开始处非空字符处。

vim 命令模式下输入符号 \$ 则定位到该行末尾。

vim 命令模式下使用 vim 方向键来单个字符单个字符地移动光标。

vim 命令模式下按一下 w 就往后跳一个单词的距离。

具体什么效果还要自己亲自动手跟着上面实验了才能体会到！

输入一个类或结构体变量之后要自动补全成员怎么操作？

输入一个长串函数名如何智能补全？

答：输入一部分字符之后再按 Ctrl-n 就能够补全。

为什么我输入部分字符 Ctrl-n 之后不能补全？

答：说明你使用的这个函数或类或结构体是你新增的，你需要在工程中再次运行 object.sh 来更新工程依赖文件，然后按 F3, F4 导入依赖文件。

如何在项目管理同时更新项目依赖文件呢？

vim 命令模式下输入 :AsyncRun object.sh 就可以更新且不影响工程操作，更新过程可以通过打开 quickfix 窗口查看，更新完成后 quickfix 窗口会计时你的更新所花费的时间（也就是你这个 object.sh 命令执行完花费时间），更新完成之后在 vim 命令模式下分别按

F3 和 F4 来重新导入更新好的依赖文件即可。

如何打开 winmanager 窗口？

vim 命令模式下按 F12 即可打开，关闭同样再次按 F12 就关闭了。

那么如何直接在工程中执行 object.sh 呢？

答：在 vim 命令模式下输入 `:AsyncRun object.sh` 即可。然后你稍等片刻，可以打开 quickfix 窗口查看进度。

如何打开 quickfix 窗口呢？

答：在 vim 命令模式下输入 `:copen` 即可打开 quickfix 窗口查看生成进度。

如何调整 quickfix 窗口大小呢？

答：在 vim 命令模式下输入 `:copen 15` 即可调整 quickfix 显示 15 行大小。

问：如何高亮标签某个变量或函数名？

答：光标移到该变量名或函数名上，然后在 vim 命令模式下直接输入 `,m` 之后 1 秒钟左右时间，就会高亮标签你要标注的函数名或变量名。除了标注一个，vim 能同时标注多个名字，比如我标注了这个变量名是紫色的，那么我标注另一个变量名就是其它颜色了。

问：如何取消高亮标签某个变量或函数名？

答：光标移到该变量名或函数名上，然后在 vim 命令模式下直接输入 `,m` 之后 1 秒钟左右时间，就会取消高亮标签你要标注的函数名或变量名。

问：通过/或?查找关键字之后这个关键字高亮，怎么取消它高亮？

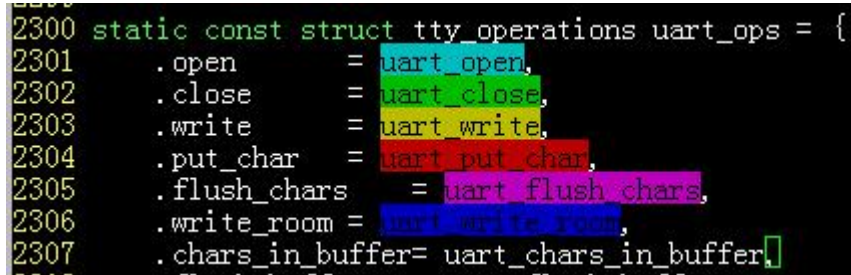
答：这个只需在 vim 命令模式下输入 `:set nohlsearch` 即可。

如果想再次打开高亮，则输入 `:set hlsearch` 即可。

请问标签高亮最多同时支持多少个标签高亮呢？

答：目前（2016 版本 vimtools）最多支持同时 6 种高亮，分别会以不同背景色区分。

如下图：



```
2300 static const struct tty_operations uart_ops = {
2301     .open      = uart_open,
2302     .close     = uart_close,
2303     .write     = uart_write,
2304     .put_char  = uart_put_char,
2305     .flush_chars = uart_flush_chars,
2306     .write_room = uart_write_room,
2307     .chars_in_buffer = uart_chars_in_buffer,
2308 }
```

能支持代码块标签高亮吗？

答：能，选中代码块(Vj)然后进行标签高亮操作(,m)即可。取消高亮必须选中和刚刚加高亮的代码块时一样范围，否则会认为你是选择新的代码块并会加上新的高亮标签。

如何给源文件添加文件头？

答：源文件一般都要添加文件头的，所谓文件头就是表明作者和许可证等信息。文件头类型有 Apache 许可证类型的，GNU 许可证类型的，还有精简版的等等。vimtool 配置默认使用的文件头是 GNULicense 类型的文件头，比如我们添加默认的 GNULicense 文件头，我们直接在 vim 命令模式下输入 `:AddGNULicense` 或者 vim 命令模式下直接按 `F2` 即可。

如何打开/关闭 quickfix 窗口？

打开 quickfix 窗口直接在 vim 命令模式下输入 `:copen` 即可打开。

关闭 quickfix 窗口直接在 vim 命令模式下输入 `:cclose` 即可打开。

调整 quickfix 窗口大小直接在 vim 命令模式下输入 `:copen 15` 即可调整为 15 行大小。

如何在工程中打开某个文件？

方法 01：

vim 命令模式下输入 `:edit filename` 直接在该编辑区打开某个文件(别用 tabedit 等 tab 系列命令！)

方法 02：

通过 winmanager 窗口的左边上面的文件窗口选择你要打开的文件回车打开它。

方法 03：

vim 命令模式下按 `F5`，然后输入你要打开的文件名。

如何正确退出 vimtool 管理？

step01 如果打开 winmanager 那就最先退出 winmanager 窗口。

step02 如果打开了 quickfix 窗口那就先退出 quickfix 窗口。

step03 如果 minibuf 区有标签就全部必须清空,一个都不剩。(清空前所有文件都保存)

step04 接下来你就可以像退出普通 vim 那样退出 vimtool 管理了。

下面是一些插件常用 vim 命令：

`:cscope find g usage_point` #找到 usage_point 函数实现处(定义处)

:cscope help 打开 cscope 帮助信息

:cscope reset 重置所有连接

:cscope show 显示连接

:cscope ./cscope.out ./ 添加一个新的数据库

Ctrl - w - [h|i|j|k] 切换窗口 (焦点)

Ctrl +] #跟踪到宏或变量定义处或函数实现处

Ctrl + t 或 Ctrl+o #返回跟踪

,m 高亮当前 word , 取消高亮就在当前 word 处再次执行(可以高亮多词)

F5 或 :LookupFile 从项目中查找并打开某个文件

(lookupfile) Ctrl - n|p 从列表中选择, 回车表示确认打开

:LUBufs 从缓冲中查找并打开某个文件

:LUWalk 打开并显示当前目录

<ESC> <Ctrl-w> q 退出 lookupfile 窗口, 或直接 ESC 之后:q 退出 lookupfile

minibuffer 窗口管理

minibuffer 文件标签切换:<Tab>

minibuffer 文件标签的删除: d

minibuffer 在某个编辑区打开特定标签: :bnumber

FileExplorer 窗口管理

vim 命令模式下按<F5> 更新列表

Taglist 窗口管理

vim 命令模式下输入 u 更新列表

项目管理经验以及注意事项

- ** 养成闲置时把光标（焦点）切换到编辑区的好习惯: <Ctrl+w> {vim 方向键}。
- ** 打开一个新源文件: 先 touch, 然后更新 FileExplorer<F5>, 然后从 FileExplorer 内打开。或者直接:edit newfile
- ** 其他文件打开: 用 edit 命令,千万别用 tabedit, :edit filename 就可以了。
- ** 在该编辑区内直接跳转到某个标签: bn, 比如跳转到 10 号标签, :b10
- ** 全选内容: VG
- ** 选中光标以下所有内容: VG
- ** 批量注释源码: <Ctrl+v>选中目标行,然后块右移(>),然后把 Tab 替换为/(r/)。
- ** 最后退出工程时: 先退出 winmanager(:WMToggle), 然后删除 minibuf 所有标签, 最后退出 vim.