

使用假病毒模式

在 FakeVim 模式下，您可以以类似于 Vim 编辑器的方式运行主编辑器。要在 FakeVim 模式下运行编辑器，请选择 **编辑>高级>使用 Vim 样式的编辑**，或按 **Alt+V**，**Alt+V**。

支持的模式和命令

在 FakeVim 模式下，主编辑器中的大多数击键将被拦截并以类似于 Vim 的方式进行解释。大多数受支持的命令可以跟在运动命令之后或在可视模式下执行，或者它们使用寄存器，或者可以以多次重复为前缀。

以下各节描述了在受支持模式下模拟的命令，以及它们在功能上如何与 Vim 不同：

- › 正常
- › 视觉
- › 命令行 (:)
- › 插入和替换

有关使用 Vim 的更多信息，请参阅 Vim 网站上的[文档](#)。

正常和可视模式

- › 基本运动，如 `、 、 、 、 、 、 、 、 h/j/k/l<C-U><C-D><C-F><C-B>ggG0^$`
- › 字移动，如 `、 、 web`
- › 内部/a 运动，例如 `、 、 ya{ciw3daw`
- › f和运动t
- › [and \c]运动
- › { 和 } 段落移动
- › 使用寄存器删除/更改/拉扯/粘贴
- › 撤消和重做
- › `<C-A>` 并增大或减小十进制、八进制或十六进制格式的数字（例如，在将数字更改为 `<C-X>128<C-A>"0x0ff""0x17f"`）
- › `.重复上一次更改`
- › `/search, , , , , -` 大多数正则表达式语法在 Vim 中使用，除了 `和` 与 QRegExp 中的语法相同？
`search*#nN\<\>\b`
- › `@和`（宏记录和执行）特殊键保存为 `q<S-Left>`
- › 注册表

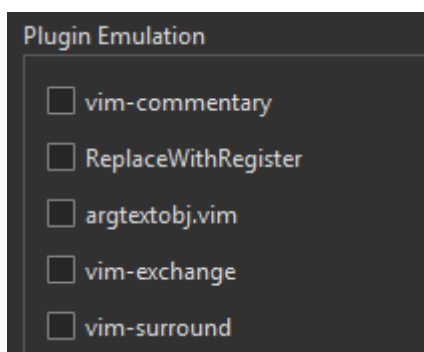
- › 使用、`<`、`>`、缩进，移动、订数和仕可伪模式 `卜=<<>>`
- › 到上/下，例如，`,`，`~gUgu`
- › `i`、`,`、`,`、和 进入插入模式 `aoIAO`
- › 滚动窗口，如、`ztzbzz`
- › 换行移动，如、`,`、`,`、`,`、`gJgkg0g^g$`

命令行模式

- › `:map`、`,`、等 `:unmap` `:inoremap`
- › `:source` 提供非常基本的 `vimrc` 文件逐行源
- › `:substitute` 替换区域中的表达式
- › `: '<,'>!cmd` 通过外部命令进行筛选（例如，使用 `:%!sort`）
- › `:<range> sor[t][!]`
- › `:.!cmd` 插入外部命令的标准输出
- › `:read`
- › `:yank`、`,`、`:delete` `:change`
- › `:move`、`:` `:join`
- › `:20` 转到地址
- › `:history`
- › `:registers`、`:` `:display`
- › `:nohlsearch`
- › `:undo`、`:` `:redo`
- › `:normal`
- › `:<,:>`
- › `set formatoptions=`，请参阅 Vim 文档中的。目前支持的信函：`:h fo-tablefo-j`

Plugin Emulation

FakeVim also emulates some popular vim plugins. To enable plugin emulation for particular plugins, select **Edit > Preferences > FakeVim > General > Plugin Emulation**.



Currently emulated plugins:

- › **vim-commentary**: action to comment code regions. For example, `,, gcgccgc2jgcip`
- › **ReplaceWithRegister**

› `["x]grr` to replace the current line.

- › `argtextobj.vim`: Defines the and text objects for function parameters. `i a a a`
- › `vim-exchange`: A text exchange operator for vim.
- › `vim-surround`: Adds mappings for deleting, adding and changing surroundings.

Insert Mode

- › `<C-O>` executes a single command and returns to insert mode
- › `<C-V>` inserts a raw character
- › `<insert>` toggles replace mode

Options

Use to set the options listed in the following table: `:set ...`

Long Name	Short Name	Arguments
autoindent	ai	
backspace	bs	indent,,eolstart
blinkincursor	bc	
clipboard	cb	
expandtab	et	
hlsearch	hls	
ignorecase	ic	
incsearch	is	
iskeyword	isk	A combination of the following characters: , , , , , @48-57_192-255a-zA-Z
relativenumber	rnu	
scrolloff	so	
shiftwidth	sw	
showcmd	sc	
smartcase	scs	
smartindent	si	
smarttab	sta	
startofline	sol	
tabstop	ts	
tildeop	top	
usecoresearch	ucs	
wrapscan	ws	

```

" highlight matched
set hlsearch
" case insensitive search
set ignorecase
set smartcase
" search while typing
set incsearch
" wrap-around when searching
set wrapscan
" show pressed keys in lower right corner
set showcmd
" tab -> spaces
set expandtab
set tabstop=4
set shiftwidth=4
" keep a 5 line buffer for the cursor from top/bottom of window
set scrolloff=5
" X11 clipboard
set clipboard=unnamed
" use ~ with movement
set tildeop

" mappings
nnoremap ; :
inoremap jj <Esc>

" clear highlighted search term on space
noremap <silent> <Space> :nohl<CR>

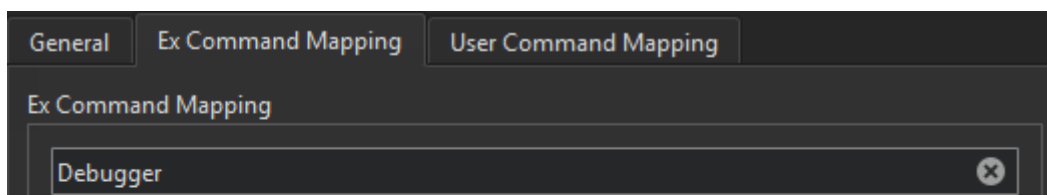
" reselect visual block after indent
vnoremap < <gv
vnoremap > >gv

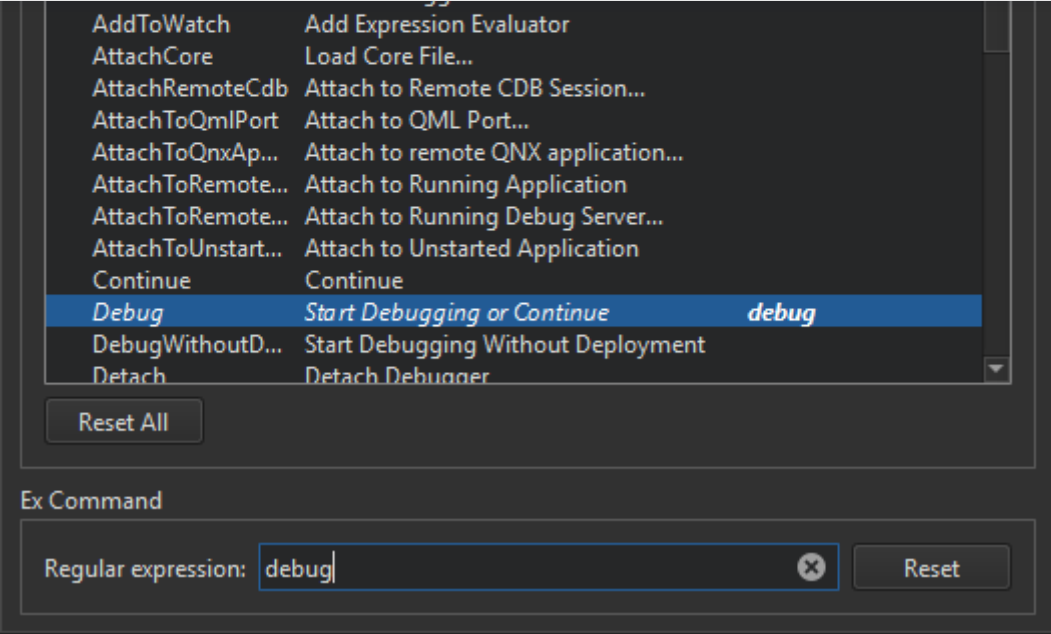
" MOVE LINE/BLOCK
nnoremap <C-S-J> :m+<CR>==
nnoremap <C-S-K> :m-2<CR>==
inoremap <C-S-J> <Esc>:m+<CR>==gi
inoremap <C-S-K> <Esc>:m-2<CR>==gi
vnoremap <C-S-J> :m'+<CR>gv=gv
vnoremap <C-S-K> :m-2<CR>gv=gv

```

Mapping FakeVim Commands

To map commands entered on the **FakeVim** command line to Qt Creator functions, select **Edit > Preferences > FakeVim > Ex Command Mapping**. Enter a string in the **Filter** field to search for a specific Qt Creator function.

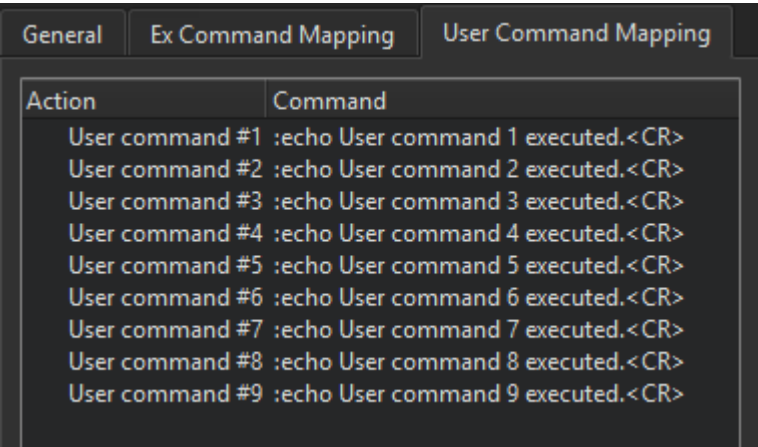




Select a function in the list, and enter a string that will trigger the function in the **Regular expression** field. You can view the trigger expression in the **Ex Trigger Expression** field. To remove the trigger expression, select **Reset**.

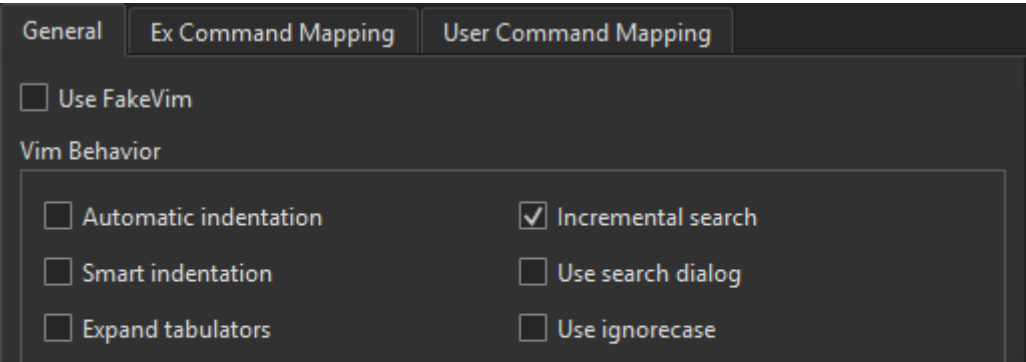
To reset the trigger expressions for all functions, select **Reset All**.

To map *user commands* to keyboard shortcuts, select **Edit > Preferences > FakeVim > User Command Mapping**. The user command mapped to the shortcut is executed by FakeVim as if you were typing it (as when replaying a macro).



Setting FakeVim Preferences

To make changes to the Vim-style settings, select **Edit > Preferences > FakeVim > General**.



☒ Show partial command
 ☐ Show position of text marks

☒ Start of line
 ☐ Pass control keys

☒ Pass keys in insert mode
 ☐ Show line numbers relative to cursor

☐ Blinking cursor
 ☐ Use tildeop

Shift width:
 Tabulator size:
 Scroll offset:

Backspace:

Keyword characters:

☐ Read .vimrc from location:

Plugin Emulation

☐ vim-commentary
 ☐ ReplaceWithRegister
 ☐ argtextobj.vim
 ☐ vim-exchange
 ☐ vim-surround

To preselect the indentation settings specified for the text editor, select **Copy Text Editor Settings**. To preselect the Qt coding style, select **Set Qt Style**. To preselect a simple indentation style, select **Set Plain Style**. You can then change any of the preselected settings.

To use a Vim-style color scheme, select **Edit > Preferences > Text Editor > Fonts & Color**. In the **Color Scheme** list, select **Vim (dark)**.

Quitting FakeVim Mode

To quit the FakeVim mode, unselect **Edit > Preferences > FakeVim > Use FakeVim** or press **Alt+V,Alt+V**.

You can temporarily escape FakeVim mode to access the normal Qt Creator keyboard shortcuts like **Ctrl-R** for **Run** by first pressing the comma key (,).

[< Specifying Text Editor Settings](#)

[Using Language Servers >](#)

© 2022 The Qt Company Ltd. Documentation contributions included herein are the copyrights of their respective owners. The documentation provided herein is licensed under the terms of the [GNU Free Documentation License version 1.3](#) as published by the Free Software Foundation. Qt and respective logos are trademarks of The Qt Company Ltd in Finland and/or other countries worldwide. All other trademarks are property of their respective owners.



Contact Us

Company

- About Us
- Investors
- Newsroom
- Careers
- Office Locations

Support

- Support Services
- Professional Services
- Partners
- Training

Community

- Contribute to Qt
- Forum
- Wiki
- Downloads
- Marketplace

Licensing

- Terms & Conditions
- Open Source
- FAQ

For Customers

- Support Center
- Downloads
- Qt Login
- Contact Us
- Customer Success