Vim Tips and Tricks

Install new syntax file for Vim

1. Download the .vim file.
2. Copy it to <vim installation directory>
3. Edit your .vimrc file.
4. Add a line like this: “**au BufNewFile,BufRead *.ft set filetype=ft**”, where “ft” is the file extension and file type you are installing support for.

Custom Vim settings in Cygwin

If you want to have custom startup setting for Vim in a Cygwin session, do the following:

1. Start a Cygwin Bash session.
2. Go to “/usr/share/vim/vim72″. (You may need to modify the “72″ to match your installed version of Vim.)
3. Copy vimrc_example.vim to ~/.vimrc
4. Go to your home directory, edit the .vimrc file, and modify the settings to your liking.

Customize syntax highlighting in VIM by extension

Sometimes, VIM does not correctly apply syntax highlighting rules. In my case, I found that this happens a lot with Visual Basic files. You can override the syntax highlighting rules, forcing a set of syntax rules to be applied based on a file’s extension, as follows. For example, if you find that .cls (VB class files) are not being handled as Visual Basic files, add the following to your vimrc file:

“**au BufNewFile,BufRead *.cls set filetype=vb**”

For other file types, just add a line similar to the one above, changing the “*.cls” and “vb” values as needed.

Maximize Vim at startup

If you want Vim to be automatically maximized when you run it, enter the following as the last statement in your vimrc file: “**au GUIEnter * simalt ~x**” (without the quotes)

Search and Replace

(from here)

The **:substitute** command searches for a text pattern, and replaces it with a text string. There are many options, but these are what you probably want:

```
:%s/foo/bar/g
```

…finds each occurrence of ‘foo’ (in all lines), and replaces it with ‘bar’.

```
:s/foo/bar/g
```

…finds each occurrence of ‘foo’ (in the current line only), and replaces it with ‘bar’.

```
:%s/foo/bar/gc
```

…changes each ‘foo’ to ‘bar’, but asks for confirmation first.

```
:%s/\<foo\>/bar/gc
```

…changes only whole words exactly matching ‘foo’ to ‘bar’; asks for confirmation.

```
:%s/foo/bar/gci
```

…changes each ‘foo’ (case insensitive) to ‘bar’; asks for confirmation. This may be wanted after using **:set noignorecase** to make searches case sensitive (the default).

```
:%s/foo/bar/gcI
```

…changes each ‘foo’ (case sensitive) to ‘bar’; asks for confirmation. This may be wanted after using **:set ignorecase** to make searches case insensitive.

From:
https://kbase.devtoprd.com/ - **Knowledge Base**

Permanent link:
https://kbase.devtoprd.com/doku.php?id=vim_tips_and_tricks

Last update: **2025/06/08 07:25**