- Login to Edit

Ubuntu Documentation > Community Documentation > VimHowto

VimHowto

Parent page: Programming Applications

Vim is an advanced text editor that provides the power of the de-facto Unix editor 'Vi' with a more complete feature set. Vim is often called a "programmer's editor," and is so useful for programming that many consider it an entire IDE. It's not just for programmers, though. Vim is perfect for all kinds of text editing, from composing email to editing configuration files.

This article aims to get help you install vim, and give you a basic introduction to vim.

Conteúdo

Installing vim

The console version of vim comes preinstalled with ubuntu, hence there is no need to install it (see note below). However, if you prefer to work with a GUI based vim, install the package **vim-gnome** from synaptic or type in a terminal

```
sudo apt-get install vim-gnome
```

NOTE: As of Ubuntu 6.10 (Edgy Eft), the default Vim install is a cut-down version called "vim-tiny". In order to get the full version complete with the online help documentation, and many of the advanced functions such as code folding; you will need to install "vim":

```
sudo apt-get install vim
```

Building Vim

If you want the very latest version of Vim, you can build it yourself. Tony Mechelynck maintains a page on building Vim under Linux. The Vim packages page lists the packages you will need to install on Ubuntu.

A Quick Introduction

vim has a formidable learning curve, but by getting comfortable with vim and its great features, you will became very efficient at manipulating text. A pre-condition to that, is the programmer (*novice*) should learn Touch Typing to experience the power.

You can start vim in console mode by typing *vi* or **vim** at the terminal or vim in graphical mode by typing **gvim**. Doing so should bring up a blank screen, with details about vim. However, any attempts to type text will fail! Which brings us to the most confusing feature for beginners - modes.

Modes

One of the most confusing things about vim is that it has four modes.

- **Insert:** To type text
- **Command:** To issue commands. Also called as Normal mode.
- **Ex:** To issue *colon* commands
- **Visual** To select text visually

The *Insert* mode is not default, you must press **i** to move into insert mode. Type some text in the screen. Press the **<Esc>** button to get out of insert mode into **Command** mode. The command mode is used to move about, and to manipulate text, sometimes in interesting ways. The **Visual** mode is used to select text, press **v** to enter it and select some text, then you can issue commands that will apply only to the selected area, type **<Esc>** again to return to **Command** mode. The **Ex** mode is used to issue colon commands, which is used for operations like saving, search & replace and configuring vim. Save the text you just typed in by going to the *Ex* mode by pressing **:** from the normal mode and typing **:w filename<Enter>**. Quit vim by executing the colon command **:q**. To summarize,

```
vim (to start vim)
i (to insert text)
<type text>
<Esc> (to come to command mode)
v (to select some text)
<Esc> (back to command mode)
:w filename (to save the text to the file 'filename')
:q (to quit the file)
:q! (to quit without saving)
vim filename (to open the file you just saved directly in vim)
```

However, it is best to learn vim by using it. You can quickly learn the basics of vim by using the inbuilt vim tutorial, by typing **vim-tutor** (**vimtutor** on dapper) in the terminal. Using the Ex: command **:help** from inside Vim is often very usefull.

Configuration

vim is a highly configurable editor, and it is best to configure vim to your liking as vim by default has all the nice features turned off. A list of files and their locations are given below.

- ~/.vimrc is the vim configuration file which vim reads on startup
- ~/.gvimrc is the gvim configuration file which gvim reads on startup. It's best to keep only gui specific settings here, as it will take preference over the settings in your .vimrc file.
- ~/.vim/ is the directory in which the user can add utility plugins, syntax highlighting plugins, and indent plugins.

Enable Syntax Highlighting

Turning syntax highlighting on is quite simple.

If you want to just enable syntax highlighting for a session, you can simply issue a colon command

```
:syn on
```

Syntax highlighting can be turned off by issuing another 'colon' command

```
:syn off
```

To make this permanent everytime you open a file, just add the following line to your vimrc.

```
syntax on
```

Enable Autoindenting

To enable Auto-Indenting of code, just type the following colon command.

```
:set ai
```

The code you type will indent automatically. If it does not indent correctly, you might need to obtain a indenting plugin for the language you are writing in from vim site.

TO make this permanent, add the following lines to your vimrc.

```
filetype indent on
set autoindent
```

Sample .vimrc file

Below is a basic .vimrc file with basic configuration. Please note that lines beginning with the character **"** are comments.

```
" Turn on line numbering. Turn it off with "set nonu"
set nu

" Set syntax on
syntax on

" Indent automatically depending on filetype
filetype indent on
set autoindent

" Case insensitive search
set ic

" Higlhight search
set hls

" Wrap text instead of being on one line
set lbr

" Change colorscheme from default to delek
colorscheme delek
```

You can also learn more by looking at a more elaborate vimrc file at **/usr/share/vim/vim70/vimrc_example.vim**. You can find this example explained in detail at the Vim documentation.

You can also find several .vimrc files online on the dotfiles website.

Editing docbook documents with vim

To contribute to the Ubuntu Documentation, you will need to use the docbook format. If so, you might be interested in the VIM *filetype plugin* xmledit.

Add the following to your ~/.vimrc

```
map! ,e <emphasis>
map! ,p <para>
map <F3> v/>^Mx
```

If you are at the beginning of an opening XML tag you can just press F3 and the tag gets cut to the buffer. Go the end of the section and press 'p' (=paste) and it will be appended after the current char.

This is useful to add tags after the text is already written. A typical usecase is when it is necessary to add formatting to current documents which have been copy/pasted from a web site.

Editing the Ubuntu Wiki

You can use VIM to edit articles in the Ubuntu wiki. Since we use the **MoinMoin** engine, we can use the Vim syntax plugin moin to get syntax highlighting for the wiki text in vim. Enable the plugin by using the instructions in the previous link.

Just click the **More Actions:** drop down list on the page you want to edit. Then select *Show Raw Text*. Copy the source of the wiki page you are editing, and paste it into vim. If you are using the console version of vim, it might be a good idea to turn off autoindenting, as vim autoformats the text as you paste it. This is not a issue in the gui version of vim.

Note: This wiki article was edited in vim ☺

Online Sources

You can find valuable information about vim at the following pages

- The vim homepage is the place to go for scripts and plugins. The tips section has a RSS feed which gives many useful tips about vim.
- The vim documentation is the central location for documentation regarding vim. You can download the Vim User Manual, and the Vim Book there.
- Vim Power tips give tips about effectively using vim.
- The author of Vim, Bram Moolenaar has written an article called Seven habits of effective text editing.

VimHowto (última edição 2010-07-27 15:02:44 efectuada por https://login.launchpad.net/+id/CrKWrb3 @ 160.129.18.220[160.129.18.220]:graywh)

- Page History