spf13 presents

The Ultimate Vim Distribution

spf13-vim is a distribution of vim plugins and resources for Vim, GVim and MacVim.

It is a completely cross platform distribution that stays true to the feel of vim while providing modern features like a plugin management system, autocomplete, tags and tons more.

Install Now or

Learn More

Installation easily installed on *nix, os x and windows

Easy Installation *nix and os x

The easiest way to install spf13-vim is to use our automatic installer by simply copying and pasting the following line into a terminal. This will install spf13-vim and backup your existing vim configuration.

If you are upgrading from a prior version (before 3.0) this is also the recommended installation.

```
curl http://j.mp/spf13-vim3 -L -o - | sh
```

Updating to the latest version

```
cd $HOME/to/spf13-vim/
git pull
vim +BundleInstall! +BundleClean +q
```

Installing on Windows

On Windows and *nix Git and Curl are required.

Installing dependencies

Install msysgit

After installation try running `git --version` within *command prompt* (press Win-R, type `cmd`, press Enter) to make sure all good:

```
C:\> git --version
git version 1.7.4.msysgit.0
```

Setup Curl

Instructions blatently copied from vundle readme Installing Curl on Windows is easy as Curl is bundled with msysgit! But before it can be used with Vundle it's required make `curl` run in *command prompt*. The easiest way is to create `curl.cmd` with this content

```
@rem Do not use "echo off" to not affect any child calls.
@setlocal

@rem Get the absolute path to the parent directory, which is assumed to be the
@rem Git installation root.
@for /F "delims=" %%I in ("%~dp0..") do @set git_install_root=%%~fI
@set PATH=%git_install_root%\bin;%git_install_root%\mingw\bin;%PATH%

@if not exist "%HOME%" @set HOME=%HOMEDRIVE%%HOMEPATH%
```

```
@if not exist "%HOME%" @set HOME=%USERPROFILE%

@curl.exe %*
```

And copy it to `C:\Program Files\Git\cmd\curl.cmd`, assuming msysgit was installed to `c:\Program Files\Git`

to verify all good, run:

```
C:\> curl --version
curl 7.21.1 (i686-pc-mingw32) libcurl/7.21.1 OpenSSL/0.9.8k zli
b/1.2.3
Protocols: dict file ftp ftps http https imap imaps ldap ldaps
 pop3 pop3s rtsp smtp smtps telnet tftp
Features: Largefile NTLM SSL SSPI libz
```

Installing spf13-vim on Windows

The easiest way is to download and run the spf13-vim-windows-install.cmd file.

Excellent Configuration A highly optimized .vimrc config file

The .vimrc file is suited to programming. It is extremely well organized and folds in sections. Each section is labeled and each option is commented.

It fixes many of the inconveniences of vanilla vim including

- A single config can be used across Windows, Mac and linux

- Eliminates swap and backup files from littering directories, preferring to store in a central location.
- Fixes common typos like :W, :Q, etc
- Setup a solid set of settings for Formatting (change to meet your needs)
- Setup the interface to take advantage of vim's features including
  - omnicomplete
  - line numbers
  - syntax highlighting
  - A better ruler & status line
  - & more
- Configuring included plugins

Customization

Create `~/.vimrc.local` and `~/.gvimrc.local` for any local customizations.

For example, to override the default color schemes:

```
echo colorscheme ir_black  >> ~/.vimrc.local
```

Featured Plugins All the modern features you expect from your IDE, now in Vim

Vundle The best plugin manager

Vundle is an excellent system built on the same principles as Pathogen, but with an integrated plugin management system that is Git and Github aware.

spf13-vim uses the Vundle plugin management system to have a well organized vim directory (Similar to mac's app folders). Vundle also ensures that the latest versions of your plugins are installed and makes it easy to keep them up to date.

NERDTree file navigation

NERDTree is a file explorer plugin that provides "project drawer" functionality to your vim editing. You can learn more about it with :help NERDTree or checkout my post on NERDTree.

QuickStart Launch using `<Leader>e`.

Customizations:

- Use `<C-E>` to toggle NERDTree
- Use `<leader>e` or `<leader>nt` to load NERDTreeFind which opens NERDTree where the current file is located.
- Hide clutter ('.pyc', '.git', '.hg', '.svn', '.bzr')
- Treat NERDTree more like a panel than a split.

ctrlp fast file finder

Ctrlp replaces the Command-T plugin with a 100% viml plugin. It provides an intuitive and fast mechanism to load files from the file system (with regex and fuzzy find), from open buffers, and from recently used files.

QuickStart Launch using `<c-p>`.

Surround managing all the '"[{}]"' etc

This plugin is a tool for dealing with pairs of "surroundings." Examples of surroundings include parentheses, quotes, and HTML tags. They are closely related to what Vim refers to as text-objects. Provided are mappings to allow for removing, changing, and adding surroundings.

Details follow on the exact semantics, but first, consider the following examples. An asterisk (*) is used to denote the cursor position.

NERDCommenter comment++

NERDCommenter allows you to wrangle your code comments, regardless of filetype. View `help :NERDCommenter` for all the details.

QuickStart Toggle comments using `<Leader>c<space>` in Visual or Normal mode.

Syntastic integrated syntax checking

```
  Old text                  Command     New te
xt ~
  "Hello *world!"           ds"         Hello
 world!
  [123+4*56]/2              cs])        (123+4
56)/2
  "Look ma, I'm *HTML!"     cs"<q>      <q>Loo
k ma, I'm HTML!</q>
  if *x>3 {                 ysW(        if ( x
>3 ) {
  my $str = *whee!;         vllllS'     my $st
r = 'whee!';
```

For instance, if the cursor was inside `"foo bar"`, you could type `cs"'` to convert the text to `'foo bar'`.

There's a lot more, check it out at `:help surround`

Syntastic is a syntax checking plugin that runs buffers through external syntax checkers as they are saved and opened. If syntax errors are detected, the user is notified and is happy because they didn't have to compile their code or execute their script to find them.

numbers.vim better line numbers

This plugin will alternate between relative numbering (normal mode) and absolute numbering (insert mode) depending on the mode you are in. This allows you to easily move code around with the relative line numbers when in normal mode. As well as providing accurate line numbers when writing code in insert mode

neocomplcache

autocomplete++

NeoComplCache is an amazing autocomplete plugin with additional support for snippets. It can complete simulatiously from the dictionary, buffer, omnicomplete and snippets. This is the one true plugin that brings Vim autocomplete on par with the best editors.

QuickStart Just start typing, it will autocomplete where possible

Customizations:

Fugitive deep git

integration

Fugitive adds pervasive git support to git directories in vim. For more information, use `:help fugitive`

Use `:Gstatus` to view `git status` and type `-` on any file to stage or unstage it. Type `p` on a file to enter `git add -p` and stage specific hunks in the file.

Use `:Gdiff` on an open file to see what changes have been made to that file

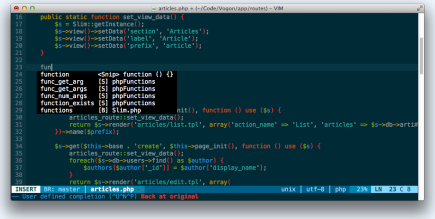
PIV PHP editing

The most feature complete and up to date PHP Integration for Vim with proper support for PHP 5.3+ including latest syntax, functions, better fold support, etc.

PIV provides:

- PHP 5.3 support
- Auto generation of PHP Doc (,pd on (function, variable, class) definition line)
- Autocomplete of classes, functions, variables, constants and language keywords
- Better indenting

- Automatically present the autocomplete menu
- Support tab and enter for autocomplete
- `<C-k>` for completing snippets.

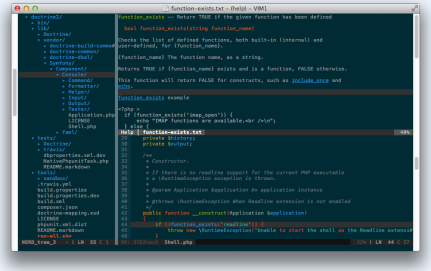


QuickStart `<leader>gs` to bring up git status

Customizations:

- `<leader>gs` :Gstatus
- `<leader>gd` :Gdiff
- `<leader>gc` :Gcommit
- `<leader>gb` :Gblame
- `<leader>gl` :Glog
- `<leader>gp` :Git push
- :Git ___ will pass anything along to git.

- Full PHP documentation manual (hit K on any function for full docs)



Tabularize align everything

Tabularize lets you align statements on their equal signs and other characters

Customizations:

- `<Leader>a=` :Tabularize /=
- `<Leader>a:` :Tabularize /:
- `<Leader>a::` :Tabularize /:\zs
- `<Leader>a,` :Tabularize /,

Tagbar tag generation and navigation

spf13-vim includes the Tagbar plugin. This plugin requires exuberant-ctags and will automatically generate tags for your open files. It also provides a panel to navigate easily via tags

QuickStart `CTRL-]` while the cursor is on a keyword (such as

EasyMotion jump anywhere

EasyMotion provides an interactive way to use motions in Vim.

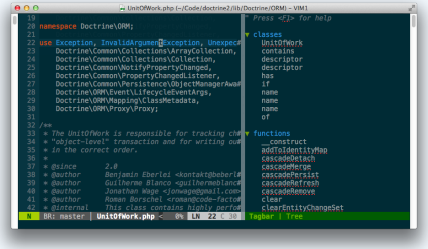
It quickly maps each possible jump destination to a key allowing very fast and straightforward movement.

QuickStart EasyMotion is triggered using the normal

- `<Leader>a<Bar>` :Tabularize /

a function name) to jump to it's definition.

Customizations: spf13-vim binds `<Leader>tt` to toggle the tagbar panel

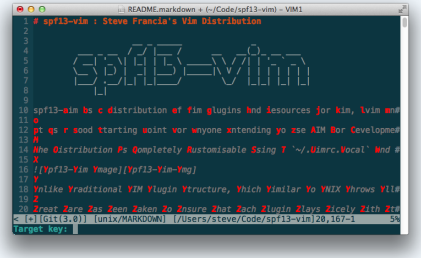


Note: For full language support, run `brew install ctags` to install exuberant-ctags.

Tip: Check out `:help ctags` for information about VIM's built-in ctag support. Tag navigation creates a stack which can traversed via `Ctrl-]` (to find the source of a token) and `Ctrl-T` (to jump back up one level).

movements, but prefixing them with `<leader><leader>`

For example this screen shot demonstrates pressing `,,w`



Customize spf13-vim is easily customizable to completely to fit your needs

Adding Your Own Bundles

Custom Settings

Create `~/.vimrc.bundles.local` for any additional bundles.

To add a new bundle

```
echo Bundle \'spf13/vim-colors\' >> ~/.vimrc.bundles.local
```

Create `~/.vimrc.local` and `~/.gvimrc.local` for any local customizations.

For example, to override the default color schemes:

```
echo colorscheme ir_black  >> ~/.vimrc.local
```

Other Awesome Stuff

Additional Syntaxes

spf13-vim ships with a few additional syntaxes:

- Markdown (bound to *.markdown, *.md, and *.mk)
- Twig
- Git commits (set your `EDITOR` to `mvim -f`)

Amazing Colors

spf13-vim includes solarized and spf13 vim color pack:

- ir_black
- molokai
- peaksea

Use `:color molokai` to switch to a color scheme.

Snippets

It also contains a very complete set of snippets for use with snipmate or NeoComplCache.

VIM Introduction

New to Vim? Here's how to get started

Tutorials

- Type `vimtutor` into a shell to go through a brief interactive tutorial

Modes

- VIM has two (common) modes:

Useful commands

- Use `:q` to exit vim

inside VIM.
- Read the slides at VIM: Walking Without Crutches.

- insert mode- stuff you type is added to the buffer
- normal mode- keys you hit are interpreted as commands

- To enter insert mode, hit `i`
- To exit insert mode, hit `<ESC>`

- Certain commands are prefixed with a `<Leader>` key, which by default maps to `\`.
- Spf13-vim uses `let mapleader = ","` to change this to `,` which is in a consistent and convenient location.
- Keyboard cheat sheet.

Community For the greater good

Mailing List

Be notified of major updates

spf13-vim-announcement - Google Groups

Discussion Group

Discuss issues, ideas,

plugins

spf13-vim-discuss - Google Groups

Contributing

GitHub makes for easy

contribution

All development is done via GitHub. Fork and issue a pull request or file an issue.

Site and distribution by @spf13

Back to top