

The Ultimate Dotfile Setup

An opinionated presentation by [@colindrake](#)

First: Why Spend the Time?

- Work more effectively
 - Reduce common commands to short aliases
 - Set up shell keybindings you're quick with (vi, emacs)
 - Leverage community maintained plugins with this system
- Able able to switch to dotfiles on new systems in seconds
 - Have configurations at your fingertips on remote systems
 - Alternate between coworker configs while pairing!

Basic Recommendations

- *A Github repository*, for easy access
- *GNU Stow* for portable and flexible file management
- *ZSH*, for the massive plugin community

Other Recommendations

Plugins, utilities, etc:

- `z` , `ag` , `rlwrap` , `trash`
- `extract` , `common-aliases` , `git` , `osx`
- `zsh-autosuggestions` , `zsh-syntax-highlighting`

Managing Your Dotfiles: Git and GNU Stow

Create the Repository

Personally, I like to store mine at `~/dotfiles` .

```
$ cd ~  
$ mkdir .dotfiles  
$ cd .dotfiles  
$ git init .
```

Installing Stow

```
$ brew install stow
```

Directory Structure

Human-readable, organizable "buckets" with the real, underlying structure beneath:

```
~/ .dotfiles/  
  git/  
    .gitconfig  
    .config/  
      hub  
  nvim/  
    .config/  
      nvim/  
        init.vim  
  zsh/  
    .zshrc
```


Running Stow

Given the previous directory structure, this:

```
$ stow --target=$HOME {git,nvim,zsh}
```

will create symlinks at:

```
$HOME/  
  .gitconfig  
  config/  
    hub  
    nvim/  
      init.vim  
  .zshrc
```

Advantages

- Separates out configuration files nicely for editing
- Installable *and reversible* in one command
 - Can work in piecemeal fashion or in full
- Manageable in a git repository separate from `$HOME`
- Easily switch between work, server-specific, personal, ... configs

ZSH: Upgrading Your Shell

zsh

Many advanced differences, but for day to day work:

- Almost fully `bash` compatible, unlike `fish`
- Smarter tab completions
 - Case insensitivity for files
 - Try `cd /u/l/b<TAB>`
 - Custom completion hooks: subcommands for `git`, etc.
- Shared history buffer with *all* open sessions
- But really... *access to the largest shell plugin ecosystem*

Use `zsh` for interactive shell, `bash` for portable scripting.

zplug

- ZSH plugin manager
- Smaller/quicker than `oh-my-zsh` , but uses same plugin repo
 - And more! ex: I have a [forked Pure theme](#) stored in a Gist.
- Minimal boilerplate to auto update and plugins, etc.

Installing zsh and zplug

```
$ brew install zsh zplug
```

Setup Walkthrough