



pathogen.vim: manage your runtimepath http://www.vim.org/scripts/script.php?script_id=2332

🕒 109 commits

🔗 1 branch

📦 6 releases

👤 11 contributors



🔗 branch: master ▾

vim-pathogen / +

Update comment to match readme



graywh authored a month ago

latest commit 1270dceb1f 📄

➔ tpope committed 4 days ago

📁 autoload	Update comment to match readme	4 days ago
📄 CONTRIBUTING.markdown	Add a CONTRIBUTING file	4 months ago
📄 README.markdown	Fix typo	5 months ago

📖 README.markdown

pathogen.vim

Manage your `'runtimepath'` with ease. In practical terms, pathogen.vim makes it super easy to install plugins and runtime files in their own private directories.

Installation

Install to `~/.vim/autoload/pathogen.vim` . Or copy and paste:

```
mkdir -p ~/.vim/autoload ~/.vim/bundle; \  
curl -Sso ~/.vim/autoload/pathogen.vim \  
https://raw.githubusercontent.com/tpope/vim-pathogen/master/autoload/pathogen.vim
```

If you're using Windows, change all occurrences of `~/.vim` to `~\vimfiles` .

Runtime Path Manipulation

Add this to your vimrc:

```
execute pathogen#infect()
```

If you're brand new to Vim and lacking a vimrc, `vim ~/.vimrc` and paste in the following super-minimal example:

```
execute pathogen#infect()  
syntax on  
filetype plugin indent on
```

<> Code

🔔 Issues 8

🔗 Pull Requests 1

🔔 Pulse

📊 Graphs

🔗 Network

HTTPS clone URL

`https://github.com/`

You can clone with HTTPS, Subversion, and other methods.

📄 Download ZIP

Now any plugins you wish to install can be extracted to a subdirectory under `~/.vim/bundle`, and they will be added to the `'runtimepath'`. Observe:

```
cd ~/.vim/bundle
git clone git://github.com/tpope/vim-sensible.git
```

Now [sensible.vim](#) is installed. If you really want to get crazy, you could set it up as a submodule in whatever repository you keep your dot files in. I don't like to get crazy.

If you don't like the directory name `bundle`, you can pass a runtime relative glob as an argument:

```
execute pathogen#infect('stuff/{'))
```

The `{}` indicates where the expansion should occur. Currently only a trailing `{}` is supported.

You can also pass an absolute path instead. I keep the plugins I maintain under `~/src`, and this is how I add them:

```
execute pathogen#infect('bundle/{', '~/src/vim/bundle/{'))
```

Normally to generate documentation, Vim expects you to run `:helptags` on each directory with documentation (e.g., `:helptags ~/.vim/doc`). Provided with `pathogen.vim` is a `:Helptags` command that does this on every directory in your `'runtimepath'`. If you really want to get crazy, you could even invoke `Helptags` in your `vimrc`. I don't like to get crazy.

Finally, `pathogen.vim` has a rich API that can manipulate `'runtimepath'` and other comma-delimited path options in ways most people will never need to do. If you're one of those edge cases, look at the source. It's well documented.

Runtime File Editing

`:Vopen`, `:Vedit`, `:Vsplit`, `:Vvsplit`, `:Vtabedit`, `:Vpedit`, and `:Vread` have all moved to [scriptease.vim](#).

FAQ

Can I put `pathogen.vim` in a submodule like all my other plugins?

Sure, stick it under `~/.vim/bundle`, and prepend the following to your `vimrc`:

```
runtime bundle/vim-pathogen/autoload/pathogen.vim
```

Or if your bundles are somewhere other than `~/.vim` (say, `~/src/vim`):

```
source ~/src/vim/bundle/vim-pathogen/autoload/pathogen.vim
```

Will you accept these 14 pull requests adding a `.gitignore` for `tags` so I don't see untracked changes in my dot files repository?

No, but I'll teach you how to ignore `tags` globally:

```
git config --global core.excludesfile ~/.cvsignore
echo tags >> ~/.cvsignore
```

While any filename will work, I've chosen to follow the ancient tradition of `.cvsignore` because utilities like

rsync use it, too. Clever, huh?

What about Vimballs?

If you really must use one:

```
:e name.vba  
:mkdir ~/.vim/bundle/name  
:UseVimball ~/.vim/bundle/name
```

Contributing

If your [commit message](#) sucks, I'm not going to accept your pull request. I've explained very politely dozens of times that [my general guidelines](#) are absolute rules on my own repositories, so I may lack the energy to explain it to you yet another time. And please, if I ask you to change something, `git commit --amend`.

Beyond that, don't be shy about asking before patching. What takes you hours might take me minutes simply because I have both domain knowledge and a perverse knowledge of Vim script so vast that many would consider it a symptom of mental illness. On the flip side, some ideas I'll reject no matter how good the implementation is. "Send a patch" is an edge case answer in my book.

Self-Promotion

Like [pathogen.vim](#)? Follow the repository on [GitHub](#) and vote for it on [vim.org](#). And if you're feeling especially charitable, follow [tpope](#) on [Twitter](#) and [GitHub](#).

License

Copyright (c) Tim Pope. Distributed under the same terms as Vim itself. See `:help license`.

