SpaceVim

A modular Vim/Neovim configuration

Home » FAQ

This is a list of the frequently asked questions about SpaceVim. Including questions about installation, configuration and usage.

Installation

Where is my old vim configuration?

In Linux/MacOS, the old vim configuration file `~/.vimrc` will be renamed to `~/.vimrc_back`, and the directory `~/.vim` also will be renamed to `~/.vim_back`.

How to uninstall SpaceVim?

The installation script does not remove your vimrc, it just changes the name from `~/.vim` to `~/.vim_back`. and if you uninstalll SpaceVim, your vimrc will come back. you can run:

```
curl -sLf https://spacevim.org/install.sh | bash -s -- --uninstall
```

How to install SpaceVim manually?

The following section will document how to install SpaceVim manually on Linux. First, you need to clone the repository to `~/.SpaceVim`.

```
git clone https://spacevim.org/git/repos/SpaceVim/ ~/.SpaceVim
```

Then, backup your old Neovim/Vim configuration file:

```
mv ~/.vimrc ~/.vimrc_back
mv ~/.vim ~/.vim_back
mv ~/.config/nvim ~/.config/nvim_back
```

Link `~/.SpaceVim` to Vim and Neovim user folder:

```
ln -s ~/.SpaceVim ~/.vim
ln -s ~/.SpaceVim ~/.config/nvim
```

Configuration

Can I try SpaceVim without overwriting my vimrc?

The SpaceVim install script will move your `~/.vimrc` to `~/.vimrc_back`. If you want to have a try SpaceVim without overwriting your own Vim configuration you can:

Clone SpaceVim manually.

```
git clone https://spacevim.org/git/repos/SpaceVim/ ~/.SpaceVim
```

Then, start Vim via `vim -u ~/.SpaceVim/vimrc`. You can also put this alias into your bashrc.

```
alias svim='vim -u ~/.SpaceVim/vimrc'
```

Why use toml as the default configuration file format?

In the old version of SpaceVim, we used a Vim file (`init.vim`) for configuration. This introduced a lot of problems. When loading a Vim file the file content is executed line by line. This means that when there was an error the content before the error was still executed. This led to unforeseen problems.

We decided going forward to use a more robust configuration mechanism in SpaceVim. SpaceVim must be able to load the whole configuration file and if there are syntax errors in the configuration file, the entire configuration needs to be discarded.

We compared TOML, YAML, XML, and JSON. We chose TOML as the default configuration language. Here are some of the drawbacks we found with the other choices considered:

1. YAML: It is error-prone due to indentation being significant and when configuring transitions.
2. XML: Vim lacks a parsing library for XML and XML is hard for humans to write.
3. JSON: Is a good configuration format and Vim has a parsing function. However, JSON does not support comments.

Where should I put my configuration?

SpaceVim loads custom global configuration from `~/.SpaceVim.d/init.toml`. It also supports project specific configuration. That means it will load `.SpaceVim.d/init.toml` from the root of your project.

Why are the options in toml file not applied?

Many people have encountered the same problem. The options have been added to `init.toml` but SpaceVim do not use it. One possibility is that there is a syntax error in toml. For example:

```
[options]
    enable_statusline_mode = true
    enable_tabline_filetype_icon = true
    enable_os_fileformat_icon = true
    statusline_unicode_symbols = true
    line_on_the_fly = false
[[layers]]
    name = 'core'
    enable_filetree_gitstatus = true
    enable_filetree_filetypeicon = true

[options]
    bootstrap_before = 'myspacevim#before'
```

In this example, only `bootstrap_before` option will be used.

In SpaceVim should have only one `[options]` section in toml file. In the example above, the `bootstrap_before` line should be moved before `[[layers]]`.

E492: Not an editor command: ^M

The problem was git auto added ^M when cloning, solved by:

```
git config --global core.autocrlf input
```

Why SpaceVim can not display default colorscheme?

By default, SpaceVim uses true colors, so you should make sure your terminal supports true colors. This is an article about what true colors are and which terminals support true colors.

Why can’t I update plugins?

Sometimes you will see `Updating failed, The plugin dir is dirty`. Since the plugin dir is a git repo, if the directory is dirty (has changes that haven’t been committed to git) you can not use `git pull` to update plugin. To fix this issue, just move your cursor to the error line, and press `gf`, then run `git reset --hard HEAD` or `git checkout .`. For more info please read git documentation.

How to reload `init.toml`?

You can not reload `init.toml` after startup. After editing the `init.toml` file, you need to restart your vim or neovim.

How to enable +py and +py3 in Neovim?

In Neovim we can use `g:python_host_prog` and `g:python3_host_prog` to config python prog. In SpaceVim the custom configuration file is loaded after SpaceVim core code. So in SpaceVim itself, if we using `:py` command, it may cause errors. So we introduce two new environment variables: `PYTHON_HOST_PROG` and `PYTHON3_HOST_PROG`.

For example:

```
export PYTHON_HOST_PROG='/home/q/envs/neovim2/bin/python'
export PYTHON3_HOST_PROG='/home/q/envs/neovim3/bin/python'
```

Why does Vim freeze after pressing Ctrl-s?

This is a feature of terminal emulators. You can use `Ctrl-q` to unfreeze Vim. To disable this feature you need the following in either `~/.bash_profile` or `~/.bashrc`:

```
stty -ixon
```

How to use telescope layer only for nvim?

If you use both Nvim and Vim, you can use following configuration to select corresponding layer.

```
[[layers]]
    name = 'telescope'
    enable = 'has("nvim")'
[[layers]]
    name = 'leaderf'
    enable = '!has("nvim")'
```