

Extracted from:

Modern Vim

Craft Your Development Environment
with Vim 8 and Neovim

This PDF file contains pages extracted from *Modern Vim*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

Copyright © 2018 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

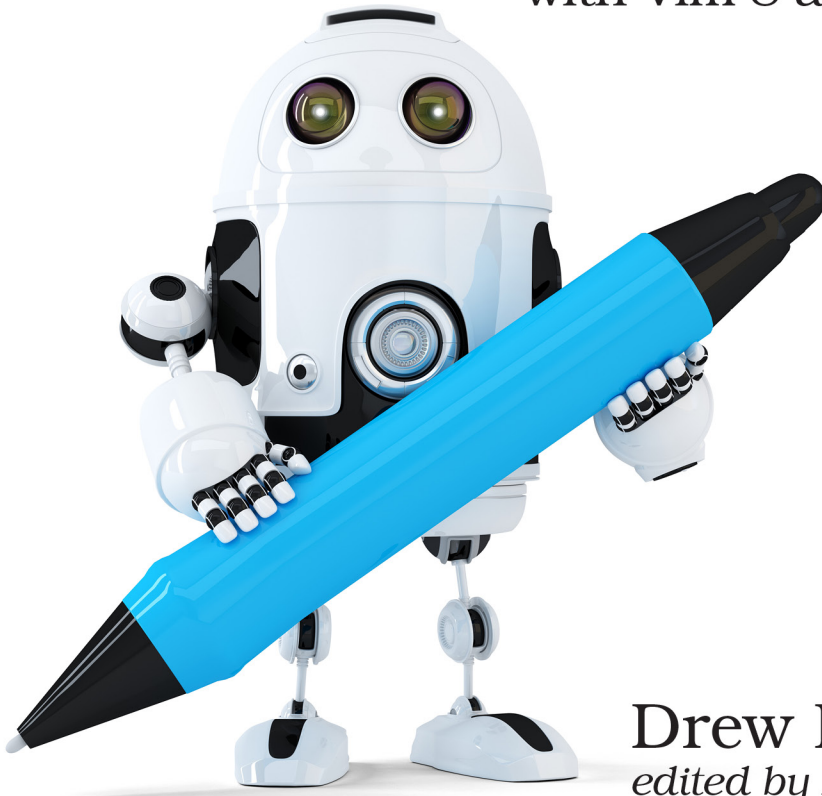
The Pragmatic Bookshelf

Raleigh, North Carolina

The
Pragmatic
Programmers

Modern Vim

Craft Your Development Environment
with Vim 8 and Neovim



Drew Neil
edited by Katharine Dvorak

Modern Vim

Craft Your Development Environment
with Vim 8 and Neovim

Drew Neil

The Pragmatic Bookshelf

Raleigh, North Carolina



Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic books, screencasts, and audio books can help you and your team create better software and have more fun. Visit us at <https://pragprog.com>.

The team that produced this book includes:

Publisher: Andy Hunt
VP of Operations: Janet Furlow
Managing Editor: Brian MacDonald
Supervising Editor: Jacquelyn Carter
Development Editor: Katharine Dvorak
Copy Editor: Jasmine Kwityn
Indexing: Potomac Indexing, LLC
Layout: Gilson Graphics

For sales, volume licensing, and support, please contact support@pragprog.com.

For international rights, please contact rights@pragprog.com.

Copyright © 2018 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

Printed in the United States of America.

ISBN-13: 978-1-68050-262-6

Encoded using the finest acid-free high-entropy binary digits.

Book version: P1.0—May 2018

Switching to Neovim



Neovim only

Neovim can be used as a drop-in replacement for Vim. In this tip, you'll find out how to install Neovim on Debian and macOS, and how to make Neovim use your existing vimrc and Vim plugins. If you want to install Neovim on another system, Neovim's wiki contains comprehensive installation instructions for many systems.²

Installing Neovim on Linux

Linux users should be able to get Neovim with their package manager. For example, on Debian, you'd install Neovim by running:

```
⇒ $ sudo apt-get install neovim
```

If that doesn't work, you may need to install the Personal Package Archive (PPA):

```
⇒ $ sudo add-apt-repository ppa:neovim-ppa/stable
```

```
⇒ $ sudo apt-get install neovim
```

Alternatively, you can install Neovim as an AppImage,³ which is a universal package that should work on most modern Linux distributions. You can get the latest nightly build from the Neovim releases page.⁴ After downloading, you need to make the package executable:

```
⇒ $ curl -LO https://github.com/neovim/neovim/releases/download/nightly/nvim.appimage
```

```
⇒ $ chmod u+x nvim.appimage
```

You could then launch Neovim by running:

```
⇒ $ ./nvim.appimage
```

If you choose this option, you may want to set up an alias so that you can launch Neovim without typing so many characters.

2. <https://github.com/neovim/neovim/wiki/Installing-Neovim>

3. <http://appimage.org>

4. <https://github.com/neovim/neovim/releases>

The Origins of Neovim

In 2014, when version 7.4 of Vim was current, Thiago de Arruda submitted a patch that introduced multi-threading capabilities to Vim. The patch wasn't accepted, and he didn't receive any feedback to suggest why the contribution was rejected. In response, Thiago created a fork of Vim and named it *Neovim*. To support his work on this project, Thiago ran a fundraising campaign,^a which raised over \$33,000.

Thiago led the development of Neovim for 18 months and built a strong community around the project. When Thiago stepped down in late 2015, Justin Keyes stepped in as lead developer.

Today, reading the original project goals on BountySource, I'm struck by how ambitious the Neovim project was. And I'm impressed to realize that all of the project's original goals have been met.

a. <https://www.bountysource.com/teams/neovim/fundraiser>

Installing Neovim on macOS

On macOS, you can install Neovim using Homebrew:

```
⇒ $ brew install neovim
```

Launching Neovim

“Neovim” is the name of the software; “neovim” is the name of the package; and the executable command is abbreviated to `nvim`. When you've installed it, launch Neovim by running:

```
⇒ $ nvim
```

The act of typing “vim” is probably burned into your fingers' muscle memory. You might want to configure your shell with an alias so that typing “vim” starts Neovim instead of Vim. You could also set the `$VISUAL` variable to `nvim`, so that programs that launch a text editor (such as `git commit`) will use Neovim. In bash, you could set that up as follows:

```
nvim-aliases.sh
# Use Neovim as "preferred editor"
export VISUAL=nvim

# Use Neovim instead of Vim or Vi
alias vim=nvim
alias vi=nvim
```

For a couple of examples showing how `$VISUAL` can be useful, skip ahead to [Tip 22, Using an Existing nvim Instance as the Preferred Editor, on page ?](#).

Reusing Your Vim Configuration

For a smooth transition from Vim to Neovim, it helps to reuse your existing configuration. Neovim can load your Vim runtime configuration files, but first you have to tell it where to find them.

When Vim starts up, it looks in your `~/.vim` directory for a `vimrc` configuration file. The equivalent configuration file for Neovim is located in a `~/.config/nvim` directory and is called `init.vim` (:help base-directories).

You'll have to create the configuration directory for Neovim:

```
⇒ $ mkdir -p ~/.config/nvim
```

Next, create and save a `~/.config/nvim/init.vim` file with the following contents:

```
init.vim
set runtimepath^=~/.vim runtimepath+=~/.vim/after
let &packpath = &runtimepath
source ~/.vim/vimrc
```

Next time you launch `nvim`, it should load the same runtime files as `vim`. That means that your Vim customizations now apply to Neovim.

Vim Script Compatibility

Most Vim plugins written in Vim script should *just work*TM in Neovim. The one area where you have to be cautious is with any plugin that uses job control to perform work asynchronously. Both Vim and Neovim support this functionality, but their APIs are different. Since the job control feature is relatively new, this issue doesn't affect many plugins.

Contextual Instructions for Neovim

Throughout this book, you'll come across generalized instructions that look like this:

```
⇒ $ mkdir -p $VIMCONFIG/pack/bundle/start
⇒ $ mkdir -p $VIMDATA/undo
```

When running Neovim on Unix, you could execute those commands by running:

```
⇒ $ mkdir -p ~/.config/nvim/pack/bundle/start
⇒ $ mkdir -p ~/.local/share/nvim/undo
```

Alternatively, you could set the `$VIMCONFIG` and `$VIMDATA` variables for your shell. For example, in `bash` you would run:

```
⇒ $ export VIMCONFIG=~/.config/nvim
```

⇒ `$ export VIMDATA=~/.local/share/nvim`

Having set these variables, you could then run the `mkdir -p $VIMCONFIG/pack/bundle/start` and `mkdir -p $VIMDATA/undo` commands verbatim.
