Extracted from:

tmux 2

Productive Mouse-Free Development

The Pragmatic Bookshelf

Raleigh, North Carolina

The Pragmatic Programmers

Pragmatic exPress

tmux 2

Productive Mouse-Free Development

Brian P. Hogan

Edited by Susannah Davidson Pfalzer

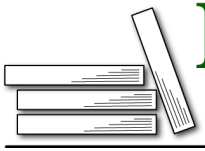
tmux 2

Productive Mouse-Free Development

Brian P. Hogan

The Pragmatic Bookshelf

Raleigh, North Carolina



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:

Susannah Davidson Pfalzer (editor)
Nicole Abramowitz (copyedit)
Gilson Graphics (layout)
Janet Furlow (producer)

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

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

Printed in the United States of America.
ISBN-13: 978-1-68050-221-3
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—November 2016

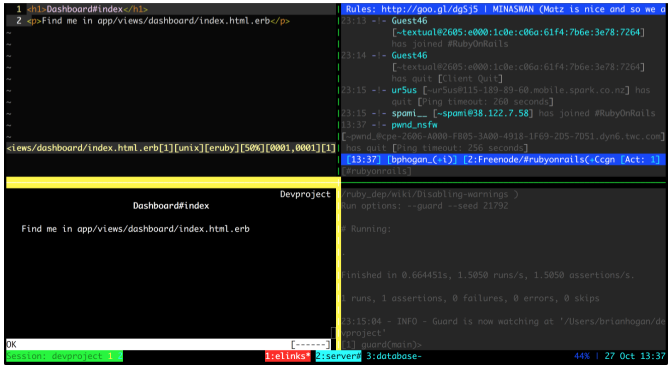
Preface

Your mouse is slowing you down.

When it was first introduced, the mouse created a new way for people to interact with computers. We can click, double-click, and even triple-click to open documents, switch windows, and select text. And thanks to trackpads, we can even swipe and use gestures to interact with our applications. The mouse, along with graphical interfaces, made computers just a little easier to use for the average person. But there's a downside to the mouse, especially for programmers.

As we build software, we work with multiple programs throughout the course of our day. A web developer, for example, might have a database console, a web server, and a text editor running at the same time. Switching between these with the mouse can slow you down. It may not seem like much, but moving your hand off of the keyboard's home row, placing it on the mouse, locating the pointer, and performing the task can eat up time and break your focus. And it can also induce strain on your wrist, arm, or shoulder. That repetitive movement of reaching for your mouse can cause some serious discomfort if you're not careful about how you hold that mouse.

Using tmux, you can create an environment like this, right in your terminal, managed entirely without a mouse:



Using tmux's windows, you can easily manage a text editor, a database console, and a local web server within a single environment. And you can split tmux windows into sections, so multiple apps can run side by side. This means you can run a text-based browser, an IRC client, or your automated tests in the same window as your main editor.

Best of all, you can quickly move between these windows and panes using only the keyboard. Over time, the keystrokes you use to manage your environment will become second nature to you, which will greatly increase both your concentration and your productivity.

In this book, you'll learn how to configure, use, and customize tmux. You'll learn how to manage multiple programs simultaneously, write scripts to create custom environments, and use tmux to work remotely with others. With tmux, you can create a work environment that keeps almost everything you need at your fingertips.

What Is tmux?

tmux is a *terminal multiplexer*. It lets you use a single environment to launch multiple terminals, or windows, each running its own process or program. For example, you can launch tmux and load up the Vim text editor. You can then create a new window, load up a database console, and switch back and forth between these programs all within a single session.

If you use a modern operating system and a terminal that has tabs, this doesn't sound like anything new. But running multiple programs simultaneously is only one of tmux's features. You can divide your terminal windows into horizontal or vertical panes, which means you can run two or more programs on the same screen side by side. And you can do it all without using the mouse.

You can also *detach* from a session, meaning you can leave your environment running in the background. If you've used GNU-Screen before, you're familiar with this feature. In many ways, tmux is like GNU-Screen with a lot of extra features and a much simpler configuration system. And since tmux uses a client-server model, you can control windows and panes from a central location, or even jump between multiple sessions from a single terminal window. This client-server model also lets you create scripts and interact with tmux from other windows or applications.

Over the course of this book, we'll explore all of these features and more.

Who Should Read This Book

Whether you're a system administrator or a software developer who spends a good part of your time using the terminal and command-line tools, this book aims to help you work faster.

If you're a software developer, you'll see how to use tmux to build a development environment that can make working with multiple terminal sessions a breeze. And if you're already comfortable using Vim or Emacs, you'll see how tmux can accelerate your workflow even more.

If you're a system administrator or a developer who spends some time working with remote servers, you'll be interested in how you can leverage tmux to create a persistent dashboard for managing or monitoring servers.

What's in This Book

This book will show you how to incorporate tmux into your work by taking you through its basic features and showing you how you might apply them to everyday situations.

In Chapter 1, *Learning the Basics*, on page ?, you'll learn about the basic features of tmux as you create sessions, panes, and windows and learn how to perform basic navigation.

In Chapter 2, *Configuring tmux*, on page ?, you'll redefine many of the default keybindings and customize how tmux looks.

In Chapter 3, *Scripting Customized tmux Environments*, on page ?, you'll script your own development environment using the command-line interface, configuration files, and the tmuxinator program.

After that, you'll work with text in Chapter 4, *Working With Text and Buffers*, on page ?. You'll use the keyboard to move backwards through the buffer, select and copy text, and work with multiple paste buffers.

Next, in Chapter 5, *Pair Programming with tmux*, on page ?, you'll set up tmux so that you and a coworker can work together on the same codebase from different computers using tmux.

Finally, Chapter 6, *Workflows*, on page ? covers more advanced ways to manage windows, panes, and sessions, and shows you how to be even more productive with tmux.

Changes in the Second Edition

This new edition has some notable changes from the first edition. tmux 2.1 and 2.2 introduced several backwards-incompatible changes that this edition addresses; this edition also introduces some new options. And tmux is now more popular than it was, so there are more tools and tricks you can use to improve your workflow. Here's what's new:

- All examples require at least tmux 2.3.
- This book now covers installation on Windows 10, where tmux is supported under Microsoft's Windows Subsystem for Linux.
- Chapter 2, *Configuring tmux*, on page ? includes more options for identifying the active pane, uses more updated methods for controlling tmux's visual styles, and removes some outdated configuration options that no longer work.
- Chapter 3, *Scripting Customized tmux Environments*, on page ? contains updated instructions for Tmuxinator and its new configuration format, as well as information on how to export tmux scripts from Tmuxinator.
- Chapter 4, *Working With Text and Buffers*, on page ? has an updated method for getting text to and from system clipboards on Linux and Mac.
- Chapter 5, *Pair Programming with tmux*, on page ? now includes instructions on generating an SSH key, and discusses how to use tmate as a quick alternative.
- Chapter 6, *Workflows*, on page ? contains several new sections:
  - *Opening a Pane in the Current Directory*, on page ?
  - *Keeping Specific Configuration Separate*, on page ?
  - *Integrating Seamlessly with Vim*, on page ?
  - *Extending tmux with Plugins*, on page ?

What You Need

In order to use tmux, you'll need a computer that runs Mac OS X, Windows 10 with Bash support, or a flavor of Unix or Linux. Unfortunately, tmux doesn't run under the regular Windows Command Prompt or Powershell, but it will run great on a virtual machine, VPS, or shared hosting environment running Linux or FreeBSD.

You should also have a good grasp of using command-line tools on a Linux or Unix system. We'll use the Bash shell in this book, and being comfortable with creating directories and text files, as well as some basic scripting, will help you move more quickly through the examples.

While not required, experience with text editors such as Vim or Emacs might be helpful. tmux works much the same way, and it has some predefined keyboard shortcuts that you may find familiar if you use one of these text editors.

Conventions

tmux is a tool that's driven by the keyboard. You'll encounter many keyboard shortcuts throughout the book. Since tmux supports both lowercase and uppercase keyboard shortcuts, it may sometimes be unclear which key the book is referencing.

To keep it simple, these are the conventions I've used.

- Ctrl-b means "press the Ctrl and b keys simultaneously."
- Ctrl-R means you'll press the Ctrl and r keys simultaneously, but you'll need to use the Shift key to produce the capital "R." I won't explicitly show the Shift key in any of these keystrokes.
- Ctrl-b d means "press the Ctrl and b keys simultaneously, then release, and then press d." In Chapter 1, *Learning the Basics*, on page ?, you'll learn about the *command prefix*, which will use this notation, but shortened to Prefix d.
- I'll show some terminal commands throughout the book, like

  ```
  $ tmux new-session
  ```

  The dollar sign represents the prompt from the Bash shell session. You won't need to type it when you type the command. It just denotes that this is a command you should type.

- Finally, as you'll see in Chapter 2, *Configuring tmux*, on page ?, you can configure tmux with a configuration file called tmux.conf. Filenames starting with a period don't show up in directory listings on most systems or text editors by default. Code listings in this book have a header that points to the file in the book's source code download, like this:

  config/tmux.conf
  ```
  # Setting the prefix from C-b to C-a
  set -g prefix C-a
  ```

To make it easy for you to find the file in the source code download, I've named the example file tmux.conf, without the leading period. The headers above the code listing reference that file.

Online Resources

The book's website[1] has links to an interactive discussion forum as well as a place to submit errata for the book. You'll also find the source code for the configuration files and scripts we use in this book. You can click the box above the code excerpts to download that source code directly.

Working with tmux has made me much more productive, and I'm excited to share my experiences with you. Let's get started by installing tmux and working with its basic features.

1. http://pragprog.com/titles/bhtmux2