Extracted from:

tmux

Productive Mouse-Free Development

This PDF file contains pages extracted from *tmux*, 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 printer versions; the content is otherwise identical.

tmux

Productive Mouse-Free Development
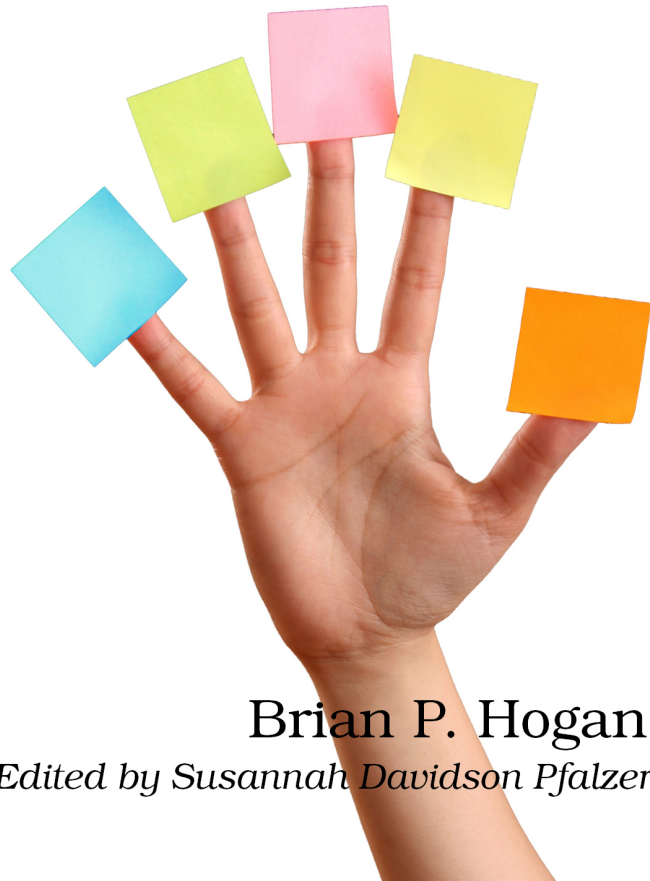
Brian P. Hogan

Edited by Susannah Davidson Pfalzer

tmux

Productive Mouse-Free Development

Brian P. Hogan

Pragmatic Bookshelf

Our Pragmatic courses, workshops, and other products can help you and your team create better software and have more fun. For more information, as well as the latest Pragmatic titles, please visit us at *http://pragprog.com*.

The team that produced this book includes:

Preface

Your mouse is slowing you down.

The mouse, when introduced, created a new way for us to interact with our computers. We could click, double-click, triple-click, and now even swipe to interact with our applications. The mouse, along with graphical interfaces, made computers just a little easier to use for average users. 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, web server, and a text editor running at the same time. Switching between these with the mouse eats up valuable time and can break your focus. 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 be pretty distracting.

Using tmux, you can create an environment like the one shown in . Using tmux's windows, you can easily manage the text editor, the database console, and the 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, 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, 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 find out how to use tmux to work remotely with others. With tmux, you can create a work environment that keeps almost everything you need at your fingertips.
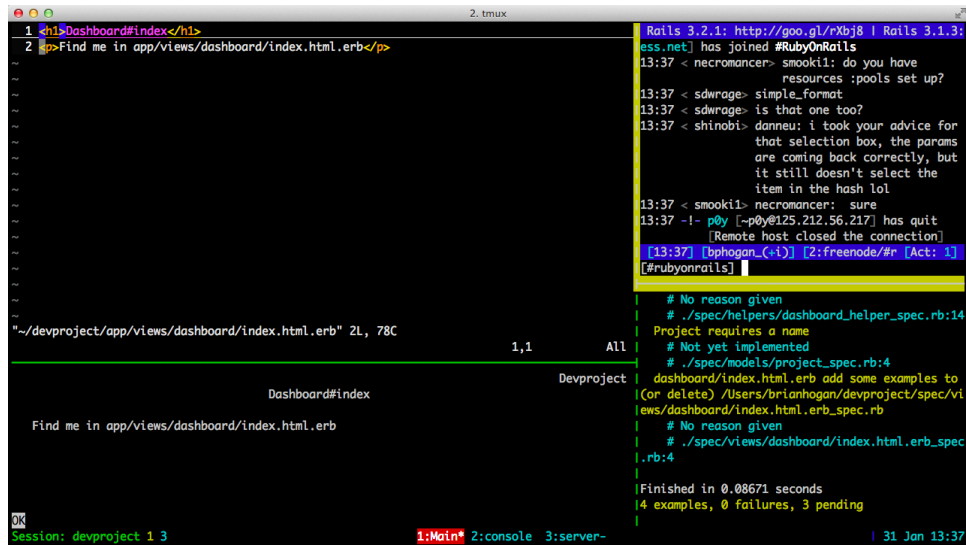
Figure 1—tmux as a development environment

What Is tmux?

tmux is a *terminal multiplexer*. It lets us use a single environment to launch multiple terminals, or windows, each running its own process or program. For example, we can launch tmux and load up the Vim text editor. We 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. We can divide windows into horizontal or vertical panes, which means we can run two or more programs on the same screen side by side. And we can do it all without using the mouse.

We can also *detach* from a session, meaning we can leave our 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, we 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 us 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

This book aims to help application developers who work on OS X or Linux operating systems and spend a good part of their time using the terminal.

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 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 learn how to redefine many of the default keybindings and the appearance of tmux.

In Chapter 3, *Scripting Customized tmux Environments*, on page ?, you'll see how to 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 see how to use keyboard keys to move backwards through the buffer, how to select and copy text, and how to work with multiple paste buffers.

Next, in Chapter 5, *Pair Programming with tmux*, on page ?, you'll learn how to 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 This Release

tmux is evolving and this book has a few changes to keep it up to date with the most recent version.

- All examples require at least tmux 1.7, and will work with version 1.8.

- The section on opening panes in the current directory under OS X was removed.

- *Maximizing and Restoring Panes*, on page ? now states that this behavior is already supported in tmux 1.8.

- The Workflows chapter contains a new section, *Issuing Commands In Many Panes Simultaneously*, on page ?, on using the "synchronize panes" feature.

- All errata have been dealt with, including a couple of nasty typos.

What You Need

In order to use tmux, you'll need a computer that runs Mac OS X or a flavor of UNIX or Linux. Unfortunately, tmux does not run under Windows, but it will run great on a virtual machine, VPS, or shared hosting environment running Linux.

While not required, experience with text editors such as Vim or Emacs will be helpful. tmux works very much the same way, and it has some predefined keyboard shortcuts that users of these editors will find familiar.

Conventions

tmux is a tool that's driven by the keyboard. You'll encounter many keyboard shortcuts throughout the book. Since tmux supports both lower- and upper-case 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 CONTROL 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.

- Finally, I'll show some terminal commands throughout the book, like

```
$ tmux new-session
```

The dollar sign is simply the prompt from the shell session and you won't need to type it when you type the command.

Online Resources

The book's web site[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 build 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/bhtmux