Join the discussion @ p2p.wrox.com

Wrox Programmer to Programmer™

Professional Git®

Brent Laster

PROFESSIONAL GIT®

INTRODUCTION xxiii

▶ PART I UNDERSTANDING GIT CONCEPTS

CHAPTER 1 What Is Git? 3

CHAPTER 2 Key Concepts 19

CHAPTER 3 The Git Promotion Model 31

CONNECTED LAB 1 Installing Git 43

▶ PART II USING GIT

CHAPTER 4 Configuration and Setup 49

CHAPTER 5 Getting Productive 73

CONNECTED LAB 2 Creating and Exploring a Git Repository and Managing Content 99

CHAPTER 6 Tracking Changes. 105

CONNECTED LAB 3 Tracking Content through the File Status Life Cycle 127

CHAPTER 7 Working with Changes over Time and Using Tags 131

CONNECTED LAB 4 Using Git History, Aliases, and Tags 155

CHAPTER 8 Working with Local Branches 159

CONNECTED LAB 5 Working with Branches 189

CHAPTER 9 Merging Content 193

CONNECTED LAB 6 Practicing with Merging. 243

CHAPTER 10 Supporting Files in Git. 245

CHAPTER 11 Doing More with Git 263

CONNECTED LAB 7 Deleting, Renaming, and Stashing 317

CHAPTER 12 Understanding Remotes—Branches and Operations 321

CONNECTED LAB 8 Setting Up a GitHub Account
and Cloning a Repository. 353

CHAPTER 13 Understanding Remotes—Workflows for Changes 355

CONNECTED LAB 9 Using the Overall Workflow
with a Remote Repository 377

CHAPTER 14 Working with Trees and Modules in Git 381

CONNECTED LAB 10 Working with Worktrees 413

CONNECTED LAB 11 Working with Submodules. 415

CONNECTED LAB 12 Working with Subtrees. 419

CHAPTER 15 Extending Git Functionality with Git Hooks. 423

INDEX 443

PROFESSIONAL

Git®

Brent Laster

wrox™
A Wiley Brand

Professional Git®

Published by
John Wiley & Sons, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Copyright © 2017 by John Wiley & Sons, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-1-119-28497-0
ISBN: 978-1-119-28498-7 (ebk)
ISBN: 978-1-119-28500-7 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

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, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at http://www.wiley.com/go/permissions.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at http://booksupport.wiley.com. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2016956722

Trademarks: Wiley, the Wiley logo, Wrox, the Wrox logo, Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. Git is a registered trademark of Software Freedom Conservancy, Inc. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc., is not associated with any product or vendor mentioned in this book.

To my soul mate and wife, Anne-Marie,

who has taught me what it means to have your dreams come true.

And to my boys Walker, Chase, and Tanner,

who inspire me every day to do the best I can.

ABOUT THE AUTHOR

BRENT LASTER is a senior manager, software development, in the Research and Development Division at SAS in Cary, North Carolina. He manages several groups involved with release engineering processes and internal tooling. He also serves as a resource for the use of open-source technologies and conducts internal training classes in technologies such as Git, Gerrit, Gradle, and Jenkins, both in the U.S. and abroad.

In addition to corporate training, Brent creates and presents workshops for a wide variety of technical conferences. He has presented workshops and informational sessions on open-source technologies (and how to apply them) at such conferences as the Rich Web Experience/Continuous Delivery Experience, ÜberConf, OSCON, and others. He is also a contributor to publications such as *No Fluff Just Stuff* magazine. Brent also conducts live web training from time to time.

Brent's passion is teaching, and he does so in a way that makes difficult concepts relatable to all. He has been involved in technical training for over 25 years and continues to seek out ways to show others how technology can be used to simplify and automate their work.

You can learn more about Brent and his work on his LinkedIn page at `http://linkedin.com/in/BrentLaster` or on Twitter at `@BrentCLaster`.

ABOUT THE TECHNICAL EDITOR

CHAIM KRAUSE is a Simulation Specialist for the U.S. Army in Leavenworth, Kansas. Although he holds a BA in Political Science from the University of Chicago, Chaim is an autodidact when it comes to computers, programming, and electronics. He wrote his first computer game in BASIC on a TRS-80 Model I Level I and stored the program on a cassette tape. Amateur radio introduced him to electronics, while the Arduino and the Raspberry Pi provided a medium to combine computing, programming, and electronics into one hobby.

In his spare time, he likes to play PC games and occasionally develops his own. He has recently taken up the sport of golf to spend more time with his significant other, Ivana.

ABOUT THE TECHNICAL PROOFREADER

PHILIPPE CHARRIÈRE is a solution engineer for GitHub. He is also a speaker at many technical events about programming, including JDD (Poland), Devoxx France, SoftShake (Geneva), and Voxxed Days Luxembourg. He started programming with a TI-99/4A (Texas Instruments) a long time ago. In his spare time, he develops IOT solutions.

CREDITS

PROJECT EDITOR
Adaobi Obi Tulton

TECHNICAL EDITOR
Chaim Krause

TECHNICAL PROOFREADER
Philippe Charrière

PRODUCTION EDITOR
Barath Kumar Rajasekaran

COPY EDITOR
Marylouise Wiack

MANAGER OF CONTENT DEVELOPMENT & ASSEMBLY
Mary Beth Wakefield

PRODUCTION MANAGER
Kathleen Wisor

MARKETING MANAGER
Carrie Sherrill

PROFESSIONAL TECHNOLOGY & STRATEGY DIRECTOR
Barry Pruett

BUSINESS MANAGER
Amy Knies

EXECUTIVE EDITOR
Jim Minatel

PROJECT COORDINATOR, COVER
Brent Savage

PROOFREADER
Nancy Bell

INDEXER
Johnna VanHoose

COVER DESIGNER
Wiley

COVER IMAGE
Doug Lemke/Shutterstock

ACKNOWLEDGMENTS

Over the course of this project, I have been amazed to discover how many dedicated people it takes to produce a book of this sort. The team at John Wiley & Sons has been outstanding and deserves more thanks than I can offer here.

First, thanks to Jim Minatel for taking on this project and seeing the potential for a new book on Git. I've appreciated his vision for the project as well as his practical guidance. Thanks also to Adaobi Obi Tulton, an amazing project editor who provided excellent direction and advice while juggling what seemed to be a million details, and never dropped the ball. This book is only possible because of her dedicated management and assistance. Thanks to Marylouise Wiack the copy editor, for making my writing readable and clear; Barath Kumar Rajasekaran, the production editor; and Nancy Bell, the proofreader, for bringing everything together to create a final, polished product.

Sincere thanks to Chaim Krause, the technical editor, for all of the effort and time he put in to making sure my explanations, examples, and labs were correct and made sense. I appreciated his careful attention to detail, and his suggestions based on his own experience toward making this book even more useful to readers. Thanks also to Philippe Charrière for volunteering on short notice to be a second technical reviewer. I've appreciated the commitment, comments, and suggestions that he has provided as an experienced professional.

Thanks to the management at SAS for supporting my initiatives to create and present corporate training courses over the years to employees across the company. I especially thank Glenn Musial, Cyndi Schnupper, and Andy Diggelmann for their encouragement and positive feedback; Barbara Miller for the years she spent managing all the logistics of scheduling and preparing materials for the training sessions as they expanded into so many areas and internationally; and my colleague, Lee Greene, for helping to teach the classes and review the materials.

On the conference side, a big thanks to Jay Zimmerman, the founder and organizer of the No Fluff Just Stuff conference series, for giving me the opportunity to speak at their events all across the country. It's great to be able to participate in quality technical events like these, which provide meaningful information and training for all.

Thanks also to everyone who's attended one of my training sessions or workshops and asked a question or provided feedback. This input can be invaluable in making the content better and more applicable.

An obscure thank you here to my fourth-grade assistant teacher, Ms. King, who told me all those years ago that I should be a writer. I'm not sure this is the kind of book she or I had in mind, but I'm counting it.

Finally, the biggest thanks of all must go to my wife, Anne-Marie, and to my children. This book was written over a long period of time, mostly nights and weekends, which took time away from them. Nonetheless, they never failed in their words of encouragement. Anne-Marie, you have been my inspiration, my friend, and my greatest supporter during this project, even though I know how foreign much of it seemed to you. Thank you for making each day a joy and our life together amazing, and for sharing my dreams and, most of all, my life.

CONTENTS

INTRODUCTION *xxiii*

PART I: UNDERSTANDING GIT CONCEPTS

CHAPTER 1: WHAT IS GIT? 3

History of Git 4
Industry-Standard Tooling 4
The Git Ecosystem 5
Core Git 5
Git-Hosting Sites 6
Self-Hosting Packages 7
Ease-of-Use Packages 8
Plug-ins 10
Tools That Incorporate Git 10
Git Libraries 10
Git's Advantages and Challenges 11
The Advantages 11
The Challenges 14
Summary 18

CHAPTER 2: KEY CONCEPTS 19

Design Concepts: User-Facing 19
Centralized Model 20
Distributed Model 20
Design Concepts: Internal 22
Delta Storage 22
Snapshot Storage 23
Git's Storage Requirements 23
Repository Design Considerations 25
Repository Scope 25
File Scope 27
Summary 29

CHAPTER 3: THE GIT PROMOTION MODEL 31

The Levels of Git 31
Dev-Test-Prod and Git 32

The Core Git Commands for Moving Content 39
Summary 41
About Connected Lab 1: Installing Git 42

CONNECTED LAB 1 INSTALLING GIT 43

Installing Git for Windows 43
Steps 43
Installing Git on Mac OS X 46
Installing Git on Linux 46

PART II: USING GIT

CHAPTER 4: CONFIGURATION AND SETUP 49

Executing Commands in Git 49
Operand Types 50
Porcelain versus Plumbing Commands 51
Specifying Arguments 53
Auto-complete 54
Configuring Git 55
Telling Git Who You Are 56
Configuration Scope 57
Default Editor 60
End of Line Settings 61
Aliases 63
Windows Filesystem Cache 64
Initializing a Repository 64
Git Init 64
Git Clone 65
Advanced Topics 66
Git Init Demystified 67
Looking Further into a Git Repository 67
Mapping Config Commands to Configuration Files 69
Creating Parameterized Aliases 70
Summary 71

CHAPTER 5: GETTING PRODUCTIVE 73

Getting Help 73
The Multiple Repositories Model 76
Adding Content to Track—Add 77
Staging Scope 78

Finalizing Changes—Commit 86
Prerequisites 87
Commit Scope 88
Putting It All Together 88
Amending Commits 88
Results of a Commit 91
Advanced Topics 94
Using the Verbose Option 95
The Full Commit Message Experience 95
Autocorrect and Auto Execute 96
Summary 97
About Connected Lab 2: Creating and Exploring a Git Repository and Managing Content 97

CONNECTED LAB 2 CREATING AND EXPLORING A GIT REPOSITORY AND MANAGING CONTENT 99

Prerequisites 99
Optional Advanced Deep-Dive into the Repository Structure 99
Steps 99

CHAPTER 6: TRACKING CHANGES 105

Git Status 105
Workflow Example with Status 106
Status Command Short Form 110
Git Diff 111
Important Symbolic Names in Git 112
How to Think about Git's Approach to Diffing 112
Diff Names Only 115
Word-diff 116
Ignoring Non-critical Changes 116
Diffing Two Commits 118
Visual Diffing 121
Other Diff Tricks 124
Summary 125
About Connected Lab 3: Tracking Content through the File Status Life Cycle 125

CONNECTED LAB 3 TRACKING CONTENT THROUGH THE FILE STATUS LIFE CYCLE 127

Prerequisites 127
Steps 127

CHAPTER 7: WORKING WITH CHANGES OVER TIME AND USING TAGS 131

The Log Command 132
Common Display and Filtering Options 132
Time-Limiting Options 133
History by Files and Paths 134
Log Output Format 135
Searching History 136
Git Blame 138
Seeing History Visually 140
Tags 141
Seeing Tag Details 142
Modifying Tags 143
Quick Tagging Example 144
Undoing Changes in History 145
Reset—Rolling Back Changes 145
Completely Resetting the Local Environment 146
Revert—Canceling Out Changes 147
Advanced Topics 151
Signing Commits and Tags 151
Reflogs 153
Summary 154
About Connected Lab 4: Using Git History, Tags, and Aliases 154

CONNECTED LAB 4 USING GIT HISTORY, ALIASES, AND TAGS 155

Prerequisites 155
Steps 155

CHAPTER 8: WORKING WITH LOCAL BRANCHES 159

What Is a Branch? 159
Example from Another Source Management System 160
The Git Model of Branches 160
Creating a Branch 162
Checking Out a Branch 163
Adding Content to a Branch 164
One Working Directory—Many Branches 164
Getting Information about Branches 168
Deleting or Renaming a Branch 169
Developing with Branches 172

Advanced Topics 180
Checking Out Non-branch Commits 181
Checking Out Individual Files 186
Summary 187
About Connected Lab 5: Working with Branches 188

CONNECTED LAB 5 WORKING WITH BRANCHES 189

Prerequisites 189
Steps 189

CHAPTER 9: MERGING CONTENT 193

The Basics of Merging 194
The Merge Command 194
Preparing for a Merge 194
Types of Merges 195
Rebasing—Merging with History 199
Cherry-Picking 203
Differences between Cherry-Pick and Rebase 205
Merge Operations 206
Dealing with Conflicts 210
Merging Is a State 210
Error Messages for Conflicts 211
Aborting the Operation 212
Dealing with Conflicts—the Workflow 213
Resolution Options and Strategies 217
Visual Merging 223
Selecting a Merging Tool 226
Making Merge Tools Available to Git 226
Advanced Topics 227
Alternative Style for Conflict Markers 227
Advanced Rebasing Scenario 229
Interactive Rebasing 234
Summary 241
About Connected Lab 6: Practicing with Merging 242

CONNECTED LAB 6 PRACTICING WITH MERGING 243

Prerequisites 243
Steps 243

CHAPTER 10: SUPPORTING FILES IN GIT 245

The Git Attributes File 246
What Can You Do with This File? 246
The Scope of Git Attributes 246
The File Format 248
Common Use Cases 249
Getting Attribute Information for Files 256
The Git Ignore File 256
The Scope of Git Ignore 257
The File Format 258
Getting Ignore Information for Files 260
Summary 261

CHAPTER 11: DOING MORE WITH GIT 263

Modifying the Layout of Files and Directories in Your Local Environment 263
stash 264
Staging Commands 270
mv 271
rm 272
Commands for Searching 272
grep 273
Git Log Searches 275
Working with Patches and Archives for Changes 278
archive 278
bundle 279
Sharing Patches through E-mail 280
apply 281
am 282
Mailing Patch Files 283
Commands for Cleaning Up 285
clean 286
gc 288
notes 288
Advanced Topics 290
filter-branch 291
rev-list 291
bisect 298
rerere 306
Summary 314
About Connected Lab 7: Deleting, Renaming, and Stashing 315

CONNECTED LAB 7 DELETING, RENAMING, AND STASHING 317

Prerequisites 317
Steps 317

CHAPTER 12: UNDERSTANDING REMOTES—BRANCHES AND OPERATIONS 321

Remotes 322
Remote Access Protocols 324
The Remote Command 326
How Git Interacts with the Remote Environment 328
Remote Tracking Branches 328
Git Clone 330
Viewing Information about Remote Branches 333
Push 338
Fetch 346
Pull 349
Summary 352
About Connected Lab 8: Setting Up a GitHub Account and Cloning a Repository 352

CONNECTED LAB 8 SETTING UP A GITHUB ACCOUNT AND CLONING A REPOSITORY 353

Prerequisites 353
Steps 353

CHAPTER 13: UNDERSTANDING REMOTES—WORKFLOWS FOR CHANGES 355

The Basic Conflict and Merge Resolution Workflow in Git 356
How the Remote Side Handles Conflicts 356
Hosted Repositories 368
Models for Collaboration with Git 369
Putting It All Together 374
Summary 375
About Connected Lab 9: Using the Overall Workflow with a Remote Repository 375

CONNECTED LAB 9 USING THE OVERALL WORKFLOW WITH A REMOTE REPOSITORY 377

Prerequisites 377
Steps 377

CHAPTER 14: WORKING WITH TREES AND MODULES IN GIT 381

Worktrees 381
Adding a Worktree 382
Listing Out the Working Trees 385
Pruning a Worktree 385
Submodules 386
Understanding How Submodules Work 388
Adding a Submodule 388
Determining Submodule Status 389
Processing Multiple Submodules 393
Updating Submodules from Their Remotes 394
Viewing Submodule Differences 396
Superproject versus Submodules 397
The Problem with Submodules 399
Updating the Submodule References 399
Updating Submodules When the Superproject Is Updated 401
Pushing Changes from Submodules 402
Submodules and Merging 404
Unregistering a Submodule 404
Subtrees 405
Adding a Project as a Subtree 407
Updating a Subtree 409
Using the Subtree Split Functionality 410
Creating a New Project from the Split Content 410
Subtree Push 411
Summary 411
About Connected Labs 10–12 411
About Connected Lab 10: Working with Worktrees 412
About Connected Lab 11: Working with Submodules 412
About Connected Lab 12: Working with Subtrees 412

CONNECTED LAB 10 WORKING WITH WORKTREES 413

Prerequisites 413
Steps 413

CONNECTED LAB 11 WORKING WITH SUBMODULES 415

Prerequisites 415
Steps 415

CONNECTED LAB 12 WORKING WITH SUBTREES 419

Prerequisites 419
Steps 419

CHAPTER 15: EXTENDING GIT FUNCTIONALITY WITH GIT HOOKS 423

Installing Hooks 424
Updating Hooks 425
Common Hook Attributes 425
Hook Domain 425
Return Code to Control Workflow 426
Working Directory Access 426
Environment Variables 427
Hook Descriptions 427
Applypatch-msg 427
Pre-applypatch 427
Post-applypatch 428
Pre-commit 428
Prepare-commit-msg 429
Commit-message 431
Post-commit 432
Pre-rebase 433
Post-checkout 434
Post-merge 434
Pre-push 434
Pre-receive 435
Update 436
Post-receive 436
Post-update 437
Other Hooks 438
Push-to-checkout 438
Pre-auto-gc 438
Post-rewrite 438
Hooks Quick Reference 439
Summary 441

INDEX *443*

INTRODUCTION

Welcome. If your job or interests involve designing, creating, or testing software, or managing any part of a software development lifecycle, chances are that you've heard of Git and, at some level, have tried to use and understand it. This book will help you reach that goal. To put it simply, *Professional Git* is intended to help you understand and use Git to get your job done, whether that job is a personal project or a professional requirement. In the process, it will also make Git part of your professional comfort zone. Throughout the book, I've provided the background and concepts that you need to know (and understand) to make sense of Git, while you learn how to interact with it.

This section will provide you with a quick introduction to the book. It will explain how this book is unique from other books about Git, the intended target audience, the book's overall structure and content, and some of the value it offers you.

I encourage you to take a few minutes and read through this section. Then, you can dive into the material at your own pace, and build your skills and understanding of Git through the text and the included hands-on labs. Or, if you'd like to quickly see additional information about the range of content, you can browse the table of contents.

Thanks for taking a look at *Professional Git*.

HOW THIS BOOK IS UNIQUE

While many books about Git are already on the market, most are aimed at providing the technical usage of the application as their major and singular goal. *Professional Git* will provide you with that, but it will also provide you with an understanding of Git in terms of concepts that you probably already know. As well, most books do not provide practical ways to integrate the concepts they describe. Learning is most effective when you have actual examples to work through so you can internalize the concepts and gain proficiency at your pace. *Professional Git* includes Connected Labs that you can work through to absorb what you've just read.

I've included simple, clear illustrations to help you visualize key ideas and workflows. I've also included Advanced Topics sections at the end of many chapters. These sections provide additional explanations of how to use some lesser-known features of Git as well as how to go beyond the standard Git features to gain extra value.

It is easy to experience a bad transition from another source management system to Git, if you don't understand Git. To be most effective, you need to comprehend the Git model and workflow. You should also know what to watch out for as you make the transition and why it's important to consider not only the commands and workflow, but also the structure and scope of its underlying repositories. I cover all of this in *Professional Git*.

TARGET AUDIENCE

This book is based on my years of training people on Git; these people worked at all levels and came from many different backgrounds—developers, testers, project managers, people managers, documentation specialists, and so on. I have presented the basic materials outlined in this book through many workshops at industry conferences and corporate training sessions. I've presented them at locations across the United States, as well as internationally. I've been successful in helping people to walk away with a newfound confidence in using Git.

I only make one assumption in this book: that you have experience with at least one source management system. It doesn't matter which one: CVS, Subversion, Mercury—any will do. I just assume that you have a basic awareness of what a source management system does as well as fundamental concepts such as checking in and checking out code and branching. Beyond that, you do not require any prior knowledge or experience. And even if you have significant experience with Git or another system, you'll find something of benefit here. In fact, if you're reading this, then you probably fall into one of the following categories:

- You are new to Git and know that you need to learn it.
- You have used Git but have been trying to use it the same way you used your previous source control system.
- You have used Git and feel that you know "just enough to be dangerous."
- You are getting by with Git, but really want to understand why it works the way it does and how to really use it as intended.
- You work with, or manage, people who either use Git or need to learn it. Given that association, you need to know about Git and to understand the fundamental concepts.
- You've heard about the potential benefits of Git, and so you are curious about it and about what it can do for you and the organization you work with.

You may actually see yourself in more than one of these categories. However, you probably just want to be able to get your job done (whether that job is a personal or professional goal). This book was built on that premise.

Git requires a mind shift. In fact, it requires a series of mind shifts. However, each shift is easy to understand once you can relate it to something you already know. Understanding each of these shifts will, in turn, allow you to be more productive and to harness the features of this powerful tool—and that's what this book is about.

STRUCTURE AND CONTENT

This book is organized as a series of chapters that present Git from the ground up, teaching you what you need to know and build on to become proficient before adding new concepts.

In the first three chapters, I cover the foundational concepts of Git: how it's different from other systems, the ecosystem that's been built around it, its advantages and challenges, and the model that allows you to understand its workflow and manage content effectively with it. This section will provide you with a basic understanding of the ideas, goals, and essential terminology of Git.

In the remaining chapters of the book, I cover the usage and features of Git, from performing basic operations to create repositories and commit changes into them, to creating branches, doing merges, and working with content in public repositories.

Notice that I don't have you using Git right away. (If you want to do that, feel free to jump ahead to Chapter 4, which quickly enables you to start getting hands-on with Git.) However, I highly recommend reading the first three chapters. If you're new to Git (or it's been a while), the background reading, especially in Chapters 2 and 3, will provide the foundation you need to understand the remaining chapters. And even if you've used Git before, reading these chapters may clear up questions that you've had about Git, give you a better mental model to work from, and form a basis to understand some of the more advanced concepts.

READER VALUE

Throughout the book, you'll find examples and guidance on the commands and workflows you need to be productive with Git. Each chapter includes ways to relate concepts to what you already know and understand. In addition to the text, you'll find many illustrations to help you understand concepts visually. As I've already mentioned, this book also adds a feature that allows you to get hands-on experience with Git, via Connected Labs interspersed throughout the chapters. These labs are designed to reinforce the concepts presented in the text of the preceding chapter(s) and to get you actively involved in the learning process, allowing you to better grasp the concepts. To get the most out of the book, you should take the time to complete each lab—usually only a few minutes. You'll find that these simple steps will greatly increase your overall understanding and confidence when using Git.

As well, take a look at the Advanced Topics sections, located at the end of some chapters. You'll likely find explanations and ideas to leverage Git functionality in ways you may not have considered before, or you may find out how to use that feature you've always wondered about.

For the later labs, custom Git repositories with example content are provided for the user at http://github.com/professional-git. In addition, downloadable copies of the code for the hooks from the last chapter are available in http://github.com/professional-git/hooks. In the event that GitHub is not available, you can find the needed files at www.wrox.com/go/professionalgit

NEXT STEPS

If this sounds like the book for you, then I encourage you to keep reading and to start making the connections and mind shifts that will help you succeed with Git. As you progress through the book, you'll find many ideas, insights, and "a-ha" moments that will serve you well. And with that knowledge, you'll soon be working at the level of "Professional Git."

PART I
Understanding Git Concepts

- **CHAPTER 1:** What Is Git?
- **CHAPTER 2:** Key Concepts
- **CHAPTER 3:** The Git Promotion Model