# 精通Git（影印版）

Mastering Git

Jakub Narębski 著

[PACKT] PUBLISHING

东南大学出版社
SOUTHEAST UNIVERSITY PRESS

# 精通 Git(影印版)
# Mastering Git

Jakub Narębski 著

南京　东南大学出版社

图书在版编目(CIP)数据

精通 Git:英文/(波)贾科布·拉瑞斯基(Jakub Narębski)著. —影印本. —南京:东南大学出版社,2017.10

书名原文:Mastering Git
ISBN 978-7-5641-7363-0

Ⅰ.①精…　Ⅱ.①贾…　Ⅲ.①软件工具-程序设计　Ⅳ.①TP311.561

中国版本图书馆 CIP 数据核字(2017)第 196706 号
图字:10-2017-113 号

精通 Git(影印版)

出版发行:东南大学出版社
地　　址:南京四牌楼 2 号　　邮编:210096
出 版 人:江建中
网　　址:http://www.seupress.com
电子邮件:press@seupress.com
印　　刷:常州市武进第三印刷有限公司
开　　本:787 毫米×980 毫米　16 开本
印　　张:26
字　　数:509 千字
版　　次:2017 年 10 月第 1 版
印　　次:2017 年 10 月第 1 次印刷
书　　号:ISBN 978-7-5641-7363-0
定　　价:84.00 元

本社图书若有印装质量问题,请直接与营销部联系。电话(传真):025-83791830

# Credits

**Author**
Jakub Narębski

**Reviewer**
Markus Maiwald

**Commissioning Editor**
Dipika Gaonkar

**Acquisition Editor**
Vinay Argekar

**Content Development Editor**
Athira Laji

**Technical Editor**
Shivani Kiran Mistry

**Copy Editor**
Akshata Lobo

**Project Coordinator**
Bijal Patel

**Proofreader**
Safis Editing

**Indexer**
Hemangini Bari

**Graphics**
Disha Haria

**Production Coordinator**
Conidon Miranda

**Cover Work**
Conidon Miranda

# About the Author

**Jakub Narębski** followed Git development from the very beginning of its creation. He is one of the main contributors to the gitweb subsystem (the original web interface for Git), and is an unofficial gitweb maintainer. He created, announced, and analyzed annual Git User's Surveys from 2007 till 2012—all except the first one (you can find his analysis of those surveys on the Git Wiki). He shares his expertise with the technology on the StackOverflow question-and-answer website.

He was one of the proofreaders of the *Version Control by Example* by Eric Sink, and was the reason why it has chapter on Git.

He is an assistant professor in the faculty of mathematics and computer science at the Nicolaus Copernicus University in Toruń, Poland. He uses Git as a version control system of choice both for personal and professional work, teaching it to computer science students as a part of their coursework.

# About the Reviewer

**Markus Maiwald** is an internet service provider, business webhoster, and domain provider. As an example, he offers agencies complete white labeling solutions for their customers (from registering a domain to deploying a webserver).

Therefore, his slogan is: *I build the systems your business runs on.*

Professionally, he is a consultant and systems administrator with over 15 years of Linux experience. He likes building high performance server systems and he develops usable and standard-compliant systems with focus on security.

As a true webworker 2.0, he runs his own international business with customers all over the world, from an insurance company in Europe to a web developer studio in Thailand.

This is the main reason why he was so passionate to work on this book. As a great team player and with a lot of experience in international teamwork, he brings in a great knowledge of tools such as Git.

I have to thank Bijal Patel, my project co-ordinator from Packt Publishing. I received outstanding support and had a great time.

I would also like to thank Sarah for her patience and encouragement while I finished this project.

# Preface

*Mastering Git* is meticulously designed to help you gain deeper insights into Git's architecture and its underlying concepts, behavior, and best practices.

*Mastering Git* starts with a quick implementation example of using Git for the collaborative development of a sample project to establish the foundation knowledge of Git's operational tasks and concepts. Furthermore, as you progress through the book, subsequent chapters provide detailed descriptions of the various areas of usage: from the source code archaeology, through managing your own work, to working with other developers. Version control topics are accompanied by in-detail description of relevant parts of Git architecture and behavior.

This book also helps augment your understanding to examine and explore your project's history, create and manage your contributions, set up repositories and branches for collaboration in centralized and distributed workflows, integrate work coming from other developers, customize and extend Git, and recover from repository errors. By exploring advanced Git practices, and getting to know details of Git workings, you will attain a deeper understanding of Git's behavior, allowing you to customize and extend existing recipes, and write your own.

## What this book covers

*Chapter 1, Git Basics in Practice,* serves as a reminder of version control basics with Git. The focus will be on providing the practical aspects of the technology, showing and explaining basic version control operations for the development of an example project, and the collaboration between two developers.

*Chapter 2, Exploring Project History,* introduces the concept of the Directed Acyclic Graph (DAG) of revisions and explains how this concept relates to the ideas of branches, tags, and the current branch in Git. You will learn how to select, filter, and view the range of revisions in the history of a project, and how to find revisions using different criteria.

*Chapter 3, Developing with Git*, describes how to create such history and how to add to it. You will learn how to create new revisions and new lines of development. This chapter introduces the concept of the staging area for commits (the index), and explains how to view and read differences between the working directory, the index, and the current revision.

*Chapter 4, Managing Your Worktree*, focuses on explaining how to manage the working directory (the worktree) to prepare contents for a new commit. This chapter will teach the reader how to manage their files in detail. It will also show how to manage files that require special handling, introducing the concepts of ignored files and file attributes.

*Chapter 5, Collaborative Development with Git*, presents a bird's eye view of the various ways to collaborate, showing different centralized and distributed workflows. It will focus on the repository-level interactions in collaborative development. You will also learn here the concept of the chain of trust, and how to use signed tags, signed merges, and signed commits.

*Chapter 6, Advanced Branching Techniques*, goes deeper into the details of collaboration in a distributed development. It explores the relations between local branches and branches in remote repositories, and describes how to synchronize branches and tags. You will learn here branching techniques, getting to know various ways of utilizing different types of branches for distinct purposes (including topic branch workflow).

*Chapter 7, Merging Changes Together*, teaches you how to merge together changes from different parallel lines of development (that is, branches) using merge and rebase. This chapter will also explain the different types of merge conflicts, how to examine them, and how to resolve them. You will learn how to copy changes with cherry-pick, and how to apply a single patch and a patch series.

*Chapter 8, Keeping History Clean*, explains why one might want to keep clean history, when it can and should be done, and how it can be done. Here you will find step-by-step instructions on how to reorder, squash, and split commits. This chapter also demonstrates how can one recover from a history rewrite, and explains what to do if one cannot rewrite history: how to revert the effect of commit, how to add a note to it, and how to change the view of project's history.

*Chapter 9, Managing Subprojects – Building a Living Framework*, explains and shows different ways to connect different projects in the one single repository of the framework project, from the strong inclusion by embedding the code of one project in the other (subtrees), to the light connection between projects by nesting repositories (submodules). This chapter also presents various solutions to the problem of large repositories and of large files.

*Chapter 10, Customizing and Extending Git*, covers configuring and extending Git to fit one's needs. You will find here details on how to set up command line for easier use, and a short introduction to graphical interfaces. This chapter explains how to automate Git with hooks (focusing on client-side hooks), for example, how to make Git check whether the commit being created passes specified coding guidelines.

*Chapter 11, Git Administration*, is intended to help readers who are in a situation of having to take up the administrative side of Git. It briefly touches the topic of serving Git repositories. Here you will learn how to use server-side hooks for logging, access control, enforcing development policy, and other purposes.

*Chapter 12, Git Best Practices*, presents a collection of version control generic and Git-specific recommendations and best practice. Those cover issues of managing the working directory, creating commits and a series of commits (pull requests), submitting changes for inclusion, and the peer review of changes.

## What you need for this book

To follow the examples used in this book and run the provided commands, you will need the Git software, preferably version 2.5.0 or later. Git is available for free on every platform (such as Linux, Windows, and Mac OS X). All examples use the textual Git interface, using the bash shell.

To compile and run sample program, which development is tracked in *Chapter 1, Git Basics in Practice*, as a demonstration of using version control, you would need working C compiler and the `make` program.

## Who this book is for

If you are a Git user with reasonable knowledge of Git and you are familiar with basic concepts such as branching, merging, staging, and workflows, this is the book for you. If you have been using Git for a long time, this book will help you understand how Git works, make full use of its power, and learn about advanced tools, techniques, and workflows. The basic knowledge of installing Git and its software configuration management concepts is necessary.

## Conventions

In this book, you will find a number of text styles that distinguish between different kinds of information. Here are some examples of these styles and an explanation of their meaning.

Code words in text, commands and their options, folder names, filenames, file extensions, pathnames, branch and tag names, dummy URLs, user input, environment variables, configuration options and their values are shown as follows: "For example, writing `git log -- foo` explicitly asks for the history of a path `foo`."

Additionally, the following convention is used: `<file>` denotes user input (here, the name of a file), `$HOME` denotes the value of environment variable, and tilde in a pathname is used to denote user's home directory (for example `~/.gitignore`).

A block of code, or a fragment of a configuration file, is set as follows:

```
void init_rand(void)
{
    srand(time(NULL));
}
```

When we wish to draw your attention to a particular part of a code block (which is quite rare), the relevant lines or items are set in bold:

```
void init_rand(void)
{
    srand(time(NULL));
}
```

Any command-line input or output is written as follows:

```
carol@server ~$ mkdir -p /srv/git
carol@server ~$ cd /srv/git
carol@server /srv/git$ git init --bare random.git
```

**New terms** and **important words** are shown in bold. Words that you see on the screen, for example, in menus or dialog boxes, appear in the text like this: "The default description that Git gives to a stash (**WIP on branch**)."

> Warnings or important notes appear in a box like this.

> Tips and tricks appear like this.

## Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or disliked. Reader feedback is important for us as it helps us develop titles that you will really get the most out of.

To send us general feedback, simply e-mail `feedback@packtpub.com`, and mention the book's title in the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide at `www.packtpub.com/authors`.

## Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

### Downloading the example code

You can download the example code files from your account at `http://www.packtpub.com` for all the Packt Publishing books you have purchased. If you purchased this book elsewhere, you can visit `http://www.packtpub.com/support` and register to have the files e-mailed directly to you.

### Downloading the color images of this book

We also provide you with a PDF file that has color images of the screenshots/ diagrams used in this book. The color images will help you better understand the changes in the output. You can download this file from `https://www.packtpub.com/sites/default/files/downloads/MasteringGit_ColorImages.pdf`.

### Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you could report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting `http://www.packtpub.com/submit-errata`, selecting your book, clicking on the **Errata Submission Form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website or added to any list of existing errata under the Errata section of that title.

To view the previously submitted errata, go to https://www.packtpub.com/books/content/support and enter the name of the book in the search field. The required information will appear under the **Errata** section.

## Piracy

Piracy of copyrighted material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works in any form on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at copyright@packtpub.com with a link to the suspected pirated material.

We appreciate your help in protecting our authors and our ability to bring you valuable content.

## Questions

If you have a problem with any aspect of this book, you can contact us at questions@packtpub.com, and we will do our best to address the problem.

# Table of Contents