
Git Revision Control

Raymond E. Marcil
<marcilr@gmail.com>

Revision 26 (August 10, 2015)

Abstract

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It out-classes SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.¹

¹Git - <https://git-scm.com/>

Contents

Contents	2
List of Figures	3
List of Tables	3
List of Definitions and Abbreviations	4
Introduction	5
Appendix	6

List of Figures

List of Tables

List of Definitions and Abbreviations

- **Branch** - [FIXME: Need data]
- **Git** - Quoting Linus: “I’m an egotistical bastard, and I name all my projects after myself. First ‘Linux’, now ‘Git’”.
(‘git’ is British slang for “pig headed, think they are always correct, argumentative”).²
- **Tag** - [FIXME: Need data]

²Git FAQ

https://git.wiki.kernel.org/index.php/GitFaq#Why_the_.27Git.27_name.3F

Introduction

Git is a distributed revision control system with an emphasis on speed,³ data integrity,⁴ and support for distributed, non-linear workflows.⁵ Git was initially designed and developed by Linus Torvalds for Linux kernel development in 2005, and has since become one of the most widely adopted version control systems for software development.⁶

As with most other distributed revision control systems, and unlike most clientserver systems, every Git working directory is a full-fledged repository with complete history and full version-tracking capabilities, independent of network access or a central server.⁷ Like the Linux kernel, Git is free software distributed under the terms of the GNU General Public License version 2.⁸

³ Torvalds, Linus (2005-04-07). “Re: Kernel SCM saga...” linux-kernel (Mailing list). “So I’m writing some scripts to try to track things a whole lot faster.”

⁴ Torvalds, Linus (2007-06-10). “Re: fatal: serious inflate inconsistency”. git (Mailing list). A brief description of Git’s data integrity design goals.

⁵Linus Torvalds (2007-05-03). [Google tech talk: Linus Torvalds on git](#). Event occurs at 02:30. Retrieved 2007-05-16.

⁶ “[Eclipse Community Survey 2014 results — Ian Skerrett](#)”. ianskerrett.wordpress.com. 2014-06-23. Retrieved 2014-06-23.

⁷Chacon, Scott (24 December 2014). [Pro Git](#) (2nd ed.). New York, NY: Apress. pp. 2930. ISBN 978-1484200773.

⁸Git (software), From Wikipedia, the free encyclopedia, [https://en.wikipedia.org/wiki/Git_\(software\)](https://en.wikipedia.org/wiki/Git_(software))

Appendix

Git

<https://git-scm.com/>

Git (software)

From Wikipedia, the free encyclopedia

[https://en.wikipedia.org/wiki/Git_\(software\)](https://en.wikipedia.org/wiki/Git_(software))

Git About

<https://git-scm.com/about>

Git branching and tagging best practices

Excellent details and semantics.

<http://programmers.stackexchange.com/questions/165725/git-branching-and-tagging-best-practices>

Git FAQ

<https://git.wiki.kernel.org/index.php/GitFAQ>