



--fast-version-control

- [About](#)
 - [Branching and Merging](#)
 - [Small and Fast](#)
 - [Distributed](#)
 - [Data Assurance](#)
 - [Staging Area](#)
 - [Free and Open Source](#)
 - [Trademark](#)
 - [Documentation](#)
 - [Reference](#)
 - [Book](#)
 - [Videos](#)
 - [External Links](#)
 - [Downloads](#)
 - [GUI Clients](#)
 - [Logos](#)
 - [Community](#)
-

This book is available in [English](#).

Full translation available in

[azərbaycan dili](#),
[български език](#),
[Deutsch](#),
[Español](#),
[Français](#),
[Ελληνικά](#),
[日本語](#),
[한국어](#),
[Nederlands](#),
[Русский](#),
[Slovenščina](#),
[Tagalog](#),
[Українська](#),
[简体中文](#),

Partial translations available in

[Čeština](#),
[Македонски](#),
[Polski](#),
[Српски](#),
[Ўзбекча](#),
[繁體中文](#),

Translations started for

[Беларуская](#),
[فارسی](#),
[Indonesian](#),
[Italiano](#),
[Bahasa Melayu](#),
[Português \(Brasil\)](#),
[Português \(Portugal\)](#),
[Svenska](#),
[Türkçe](#).

The source of this book is [hosted on GitHub](#).
Patches, suggestions and comments are welcome.

[Chapters ▼](#)

1. **1. Getting Started**

- 1. 1.1 [About Version Control](#)
- 2. 1.2 [A Short History of Git](#)
- 3. 1.3 [What is Git?](#)
- 4. 1.4 [The Command Line](#)
- 5. 1.5 [Installing Git](#)
- 6. 1.6 [First-Time Git Setup](#)
- 7. 1.7 [Getting Help](#)
- 8. 1.8 [Summary](#)

2. **2. Git Basics**

- 1. 2.1 [Getting a Git Repository](#)
- 2. 2.2 [Recording Changes to the Repository](#)
- 3. 2.3 [Viewing the Commit History](#)
- 4. 2.4 [Undoing Things](#)
- 5. 2.5 [Working with Remotes](#)
- 6. 2.6 [Tagging](#)
- 7. 2.7 [Git Aliases](#)
- 8. 2.8 [Summary](#)

3. **3. Git Branching**

- 1. 3.1 [Branches in a Nutshell](#)
- 2. 3.2 [Basic Branching and Merging](#)
- 3. 3.3 [Branch Management](#)
- 4. 3.4 [Branching Workflows](#)
- 5. 3.5 [Remote Branches](#)
- 6. 3.6 [Rebasing](#)
- 7. 3.7 [Summary](#)

4. **4. Git on the Server**

- 1. 4.1 [The Protocols](#)
- 2. 4.2 [Getting Git on a Server](#)
- 3. 4.3 [Generating Your SSH Public Key](#)
- 4. 4.4 [Setting Up the Server](#)

- 5.4.5 [Git Daemon](#)
- 6.4.6 [Smart HTTP](#)
- 7.4.7 [GitWeb](#)
- 8.4.8 [GitLab](#)
- 9.4.9 [Third Party Hosted Options](#)
- 10.4.10 [Summary](#)

5. [5. Distributed Git](#)

- 1.5.1 [Distributed Workflows](#)
- 2.5.2 [Contributing to a Project](#)
- 3.5.3 [Maintaining a Project](#)
- 4.5.4 [Summary](#)

1. [6. GitHub](#)

- 1.6.1 [Account Setup and Configuration](#)
- 2.6.2 [Contributing to a Project](#)
- 3.6.3 [Maintaining a Project](#)
- 4.6.4 [Managing an organization](#)
- 5.6.5 [Scripting GitHub](#)
- 6.6.6 [Summary](#)

2. [7. Git Tools](#)

- 1.7.1 [Revision Selection](#)
- 2.7.2 [Interactive Staging](#)
- 3.7.3 [Stashing and Cleaning](#)
- 4.7.4 [Signing Your Work](#)
- 5.7.5 [Searching](#)
- 6.7.6 [Rewriting History](#)
- 7.7.7 [Reset Demystified](#)
- 8.7.8 [Advanced Merging](#)
- 9.7.9 [Rerere](#)
- 10.7.10 [Debugging with Git](#)
- 11.7.11 [Submodules](#)
- 12.7.12 [Bundling](#)
- 13.7.13 [Replace](#)
- 14.7.14 [Credential Storage](#)
- 15.7.15 [Summary](#)

3. [8. Customizing Git](#)

- 1.8.1 [Git Configuration](#)
- 2.8.2 [Git Attributes](#)
- 3.8.3 [Git Hooks](#)
- 4.8.4 [An Example Git-Enforced Policy](#)
- 5.8.5 [Summary](#)

4. [9. Git and Other Systems](#)

- 1.9.1 [Git as a Client](#)
- 2.9.2 [Migrating to Git](#)
- 3.9.3 [Summary](#)

5. **10. Git Internals**

1. 10.1 [Plumbing and Porcelain](#)
2. 10.2 [Git Objects](#)
3. 10.3 [Git References](#)
4. 10.4 [Packfiles](#)
5. 10.5 [The Refspec](#)
6. 10.6 [Transfer Protocols](#)
7. 10.7 [Maintenance and Data Recovery](#)
8. 10.8 [Environment Variables](#)
9. 10.9 [Summary](#)

1. **A1. Appendix A: Git in Other Environments**

1. A1.1 [Graphical Interfaces](#)
2. A1.2 [Git in Visual Studio](#)
3. A1.3 [Git in Visual Studio Code](#)
4. A1.4 [Git in IntelliJ / PyCharm / WebStorm / PhpStorm / RubyMine](#)
5. A1.5 [Git in Sublime Text](#)
6. A1.6 [Git in Bash](#)
7. A1.7 [Git in Zsh](#)
8. A1.8 [Git in PowerShell](#)
9. A1.9 [Summary](#)

2. **A2. Appendix B: Embedding Git in your Applications**

1. A2.1 [Command-line Git](#)
2. A2.2 [Libgit2](#)
3. A2.3 [JGit](#)
4. A2.4 [go-git](#)
5. A2.5 [Dulwich](#)

3. **A3. Appendix C: Git Commands**

1. A3.1 [Setup and Config](#)
2. A3.2 [Getting and Creating Projects](#)
3. A3.3 [Basic Snapshotting](#)
4. A3.4 [Branching and Merging](#)
5. A3.5 [Sharing and Updating Projects](#)
6. A3.6 [Inspection and Comparison](#)
7. A3.7 [Debugging](#)
8. A3.8 [Patching](#)
9. A3.9 [Email](#)
10. A3.10 [External Systems](#)
11. A3.11 [Administration](#)
12. A3.12 [Plumbing Commands](#)

2nd Edition

1.5 Getting Started - Installing Git

Installing Git

Before you start using Git, you have to make it available on your computer. Even if it's already installed, it's probably a good idea to update to the latest version. You can either install it as a package or via another installer, or download the source code and compile it yourself.

This book was written using Git version **2.8.0**. Though most of the commands we use should work even in ancient versions of Git, some of them might not or might act slightly differently if you're using an older version. Since Git is quite excellent at preserving backwards compatibility, any version after 2.8 should work just fine.

Installing on Linux

If you want to install the basic Git tools on Linux via a binary installer, you can generally do so through the package management tool that comes with your distribution. If you're on Fedora (or any closely-related RPM-based distribution, such as RHEL or CentOS), you can use `dnf`:

```
$ sudo dnf install git-all
```

If you're on a Debian-based distribution, such as Ubuntu, try `apt`:

```
$ sudo apt install git-all
```

For more options, there are instructions for installing on several different Unix distributions on the Git website, at <https://git-scm.com/download/linux>.

Installing on macOS

There are several ways to install Git on a Mac. The easiest is probably to install the Xcode Command Line Tools. On Mavericks (10.9) or above you can do this simply by trying to run `git` from the Terminal the very first time.

```
$ git --version
```

If you don't have it installed already, it will prompt you to install it.

If you want a more up to date version, you can also install it via a binary installer. A macOS Git installer is maintained and available for download at the Git website, at <https://git-scm.com/download/mac>.



Figure 7. Git macOS Installer

You can also install it as part of the GitHub for macOS install. Their GUI Git tool has an option to install command line tools as well. You can download that tool from the GitHub for macOS website, at <https://desktop.github.com>.

Installing on Windows

There are also a few ways to install Git on Windows. The most official build is available for download on the Git website. Just go to <https://git-scm.com/download/win> and the download will start automatically. Note that this is a project called Git for Windows, which is separate from Git itself; for more information on it, go to <https://gitforwindows.org>.

To get an automated installation you can use the [Git Chocolatey package](#). Note that the Chocolatey package is community maintained.

Another easy way to get Git installed is by installing GitHub Desktop. The installer includes a command line version of Git as well as the GUI. It also works well with PowerShell, and sets up solid credential caching and sane CRLF settings. We'll learn more about those things a little later, but suffice it to say they're things you want. You can download this from the [GitHub Desktop website](#).

Installing from Source

Some people may instead find it useful to install Git from source, because you'll get the most recent version. The binary installers tend to be a bit behind, though as Git has matured in recent years, this has made less of a difference.

If you do want to install Git from source, you need to have the following libraries that Git depends on: autotools, curl, zlib, openssl, expat, and libiconv. For example, if you're on a system that has `dnf` (such as Fedora) or `apt-get` (such as a

Debian-based system), you can use one of these commands to install the minimal dependencies for compiling and installing the Git binaries:

```
$ sudo dnf install dh-autoreconf curl-devel expat-devel gettext-devel \
openssl-devel perl-devel zlib-devel
$ sudo apt-get install dh-autoreconf libcurl4-gnutls-dev libexpat1-dev \
gettext libz-dev libssl-dev
```

In order to be able to add the documentation in various formats (doc, html, info), these additional dependencies are required:

```
$ sudo dnf install asciidoc xmlto docbook2X
$ sudo apt-get install asciidoc xmlto docbook2x
```

Note Users of RHEL and RHEL-derivatives like CentOS and Scientific Linux will have to [enable the EPEL repository](#) to download the docbook2X package.

If you're using a Debian-based distribution (Debian/Ubuntu/Ubuntu-derivatives), you also need the `install-info` package:

```
$ sudo apt-get install install-info
```

If you're using a RPM-based distribution (Fedora/RHEL/RHEL-derivatives), you also need the `getopt` package (which is already installed on a Debian-based distro):

```
$ sudo dnf install getopt
```

Additionally, if you're using Fedora/RHEL/RHEL-derivatives, you need to do this:

```
$ sudo ln -s /usr/bin/db2x_docbook2texi /usr/bin/docbook2x-texi
```

due to binary name differences.

When you have all the necessary dependencies, you can go ahead and grab the latest tagged release tarball from several places. You can get it via the kernel.org site, at <https://www.kernel.org/pub/software/scm/git>, or the mirror on the GitHub website, at <https://github.com/git/git/releases>. It's generally a little clearer what the latest version is on the GitHub page, but the kernel.org page also has release signatures if you want to verify your download.

Then, compile and install:

```
$ tar -zxf git-2.8.0.tar.gz
$ cd git-2.8.0
$ make configure
$ ./configure --prefix=/usr
$ make all doc info
$ sudo make install install-doc install-html install-info
```

After this is done, you can also get Git via Git itself for updates:

```
$ git clone git://git.kernel.org/pub/scm/git/git.git
```

[prev](#) | [next](#)
[About this site](#)

Patches, suggestions, and comments are welcome.

Git is a member of [Software Freedom Conservancy](#).