

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 2.0 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>.

Git macOS installer

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

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

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
sudo apt-get 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-2.0.0.tar.gz
cd git-2.0.0
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
```

Revision #1

Created 16 April 2019 19:47:35 by ClassCloud

Updated 16 April 2019 19:48:01 by ClassCloud