



--everything-is-local



- [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. Введение**

1. 1.1 [О системе контроля версий](#)

2. 1.2 [Краткая история Git](#)

- 3. 1.3 [Что такое Git?](#)
- 4. 1.4 [Командная строка](#)
- 5. 1.5 [Установка Git](#)
- 6. 1.6 [Первоначальная настройка Git](#)
- 7. 1.7 [Как получить помощь?](#)
- 8. 1.8 [Заключение](#)

2. [**Основы Git**](#)

- 1. 2.1 [Создание Git-репозитория](#)
- 2. 2.2 [Запись изменений в репозиторий](#)
- 3. 2.3 [Просмотр истории коммитов](#)
- 4. 2.4 [Операции отмены](#)
- 5. 2.5 [Работа с удалёнными репозиториями](#)
- 6. 2.6 [Работа с тегами](#)
- 7. 2.7 [Псевдонимы в Git](#)
- 8. 2.8 [Заключение](#)

3. [**Ветвление в Git**](#)

- 1. 3.1 [О ветвлении в двух словах](#)
- 2. 3.2 [Основы ветвления и слияния](#)
- 3. 3.3 [Управление ветками](#)
- 4. 3.4 [Работа с ветками](#)
- 5. 3.5 [Удалённые ветки](#)
- 6. 3.6 [Перебазирование](#)

7. 3.7 [Заключение](#)

4. **4. [Git на сервере](#)**

1. 4.1 [Протоколы](#)

2. 4.2 [Установка Git на сервер](#)

3. 4.3 [Генерация открытого SSH ключа](#)

4. 4.4 [Настраиваем сервер](#)

5. 4.5 [Git-демон](#)

6. 4.6 [Умный HTTP](#)

7. 4.7 [GitWeb](#)

8. 4.8 [GitLab](#)

9. 4.9 [Git-хостинг](#)

10. 4.10 [Заключение](#)

5. **5. [Распределенный Git](#)**

1. 5.1 [Распределенный рабочий процесс](#)

2. 5.2 [Участие в проекте](#)

3. 5.3 [Сопровождение проекта](#)

4. 5.4 [Заключение](#)

1. **6. [GitHub](#)**

1. 6.1 [Настройка и конфигурация учетной записи](#)

2. 6.2 [Внесение собственного вклада в проекты](#)

- 3. 6.3 [Сопровождение проекта](#)
- 4. 6.4 [Управление организацией](#)
- 5. 6.5 [Scripting GitHub](#)
- 6. 6.6 [Заключение](#)

2. **7. Инструменты Git**

- 1. 7.1 [Выбор ревизии](#)
- 2. 7.2 [Интерактивное индексирование](#)
- 3. 7.3 [Припрятывание и очистка](#)
- 4. 7.4 [Подпись](#)
- 5. 7.5 [Поиск](#)
- 6. 7.6 [Перезапись истории](#)
- 7. 7.7 [Раскрытие тайн reset](#)
- 8. 7.8 [Продвинутое слияние](#)
- 9. 7.9 [Rereget](#)
- 10. 7.10 [Обнаружение ошибок с помощью Git](#)
- 11. 7.11 [Подмодули](#)
- 12. 7.12 [Создание пакетов](#)
- 13. 7.13 [Замена](#)
- 14. 7.14 [Хранилище учётных данных](#)
- 15. 7.15 [Заключение](#)

3. **8. Настройка Git**

- 1. 8.1 [Конфигурация Git](#)

- 2. 8.2 [Атрибуты Git](#)
- 3. 8.3 [Хуки в Git](#)
- 4. 8.4 [Пример принудительной политики Git](#)
- 5. 8.5 [Заключение](#)

4. **9. Git и другие системы контроля версий**

- 1. 9.1 [Git как клиент](#)
- 2. 9.2 [Переход на Git](#)
- 3. 9.3 [Заключение](#)

5. **10. Git изнутри**

- 1. 10.1 [Сантехника и Фарфор](#)
- 2. 10.2 [Объекты Git](#)
- 3. 10.3 [Ссылки в Git](#)
- 4. 10.4 [Раск-файлы](#)
- 5. 10.5 [Спецификации ссылок](#)
- 6. 10.6 [Протоколы передачи данных](#)
- 7. 10.7 [Обслуживание репозитория и восстановление данных](#)
- 8. 10.8 [Переменные окружения](#)
- 9. 10.9 [Заключение](#)

1. **A1. Приложение A: Git в других окружениях**

- 1. A1.1 [Графические интерфейсы](#)

2. A1.2 [Git в Visual Studio](#)
3. A1.3 [Git в Visual Studio Code](#)
4. A1.4 [Git в Eclipse](#)
5. A1.5 [Git в IntelliJ / PyCharm / WebStorm / PhpStorm / RubyMine](#)
6. A1.6 [Git в Sublime Text](#)
7. A1.7 [Git в Bash](#)
8. A1.8 [Git в Zsh](#)
9. A1.9 [Git в PowerShell](#)
10. A1.10 [Заключение](#)

2. **A2. Приложение В: Встраивание Git в ваши приложения**

1. A2.1 [Git из командной строки](#)
2. A2.2 [Libgit2](#)
3. A2.3 [JGit](#)
4. A2.4 [go-git](#)
5. A2.5 [Dulwich](#)

3. **A3. Приложение С: Команды Git**

1. A3.1 [Настройка и конфигурация](#)
2. A3.2 [Клонирование и создание репозитория](#)
3. A3.3 [Основные команды](#)
4. A3.4 [Ветвление и слияния](#)
5. A3.5 [Совместная работа и обновление проектов](#)
6. A3.6 [Осмотр и сравнение](#)

- 7. A3.7 [Отладка](#)
- 8. A3.8 [Внесение исправлений](#)
- 9. A3.9 [Работа с помощью электронной почты](#)
- 10. A3.10 [Внешние системы](#)
- 11. A3.11 [Администрирование](#)
- 12. A3.12 [Низкоуровневые команды](#)

2nd Edition

1.5 Введение - Установка Git

Установка Git

Прежде чем использовать Git, вы должны установить его на своём компьютере. Даже если он уже установлен, наверное, это хороший повод, чтобы обновиться до последней версии. Вы можете установить Git из собранного пакета или другого установщика, либо скачать исходный код и скомпилировать его самостоятельно.

Примечание В этой книге используется Git версии **2.8.0**. Хотя большинство используемых нами команд должны работать даже в старых версиях Git, некоторые из них могут не работать или действовать немного иначе, если вы используете старую версию. Поскольку Git отлично справляется с сохранением обратной совместимости, любая версия после 2.8 должна работать нормально.

Установка в Linux

Если вы хотите установить Git под Linux как бинарный пакет, это можно сделать, используя обычный менеджер пакетов вашего дистрибутива. Если у вас Fedora (или другой похожий дистрибутив, такой как RHEL или CentOS), можно воспользоваться dnf:

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

Если же у вас дистрибутив, основанный на Debian, например, Ubuntu, попробуйте apt:

```
$ sudo apt install git
```

Чтобы воспользоваться дополнительными возможностями, посмотрите инструкцию по установке для нескольких различных разновидностей Unix на сайте Git <https://git-scm.com/download/linux>.

Установка на Mac

Существует несколько способов установки Git на Mac. Самый простой — установить Xcode Command Line Tools. В версии Mavericks (10.9) и выше вы можете добиться этого просто первый раз выполнив 'git' в терминале.

```
$ git --version
```

Если Git не установлен, вам будет предложено его установить.

Если Вы хотите получить более актуальную версию, то можете воспользоваться бинарным установщиком. Установщик Git для OS X доступен для скачивания с сайта Git <https://git-scm.com/download/mac>.



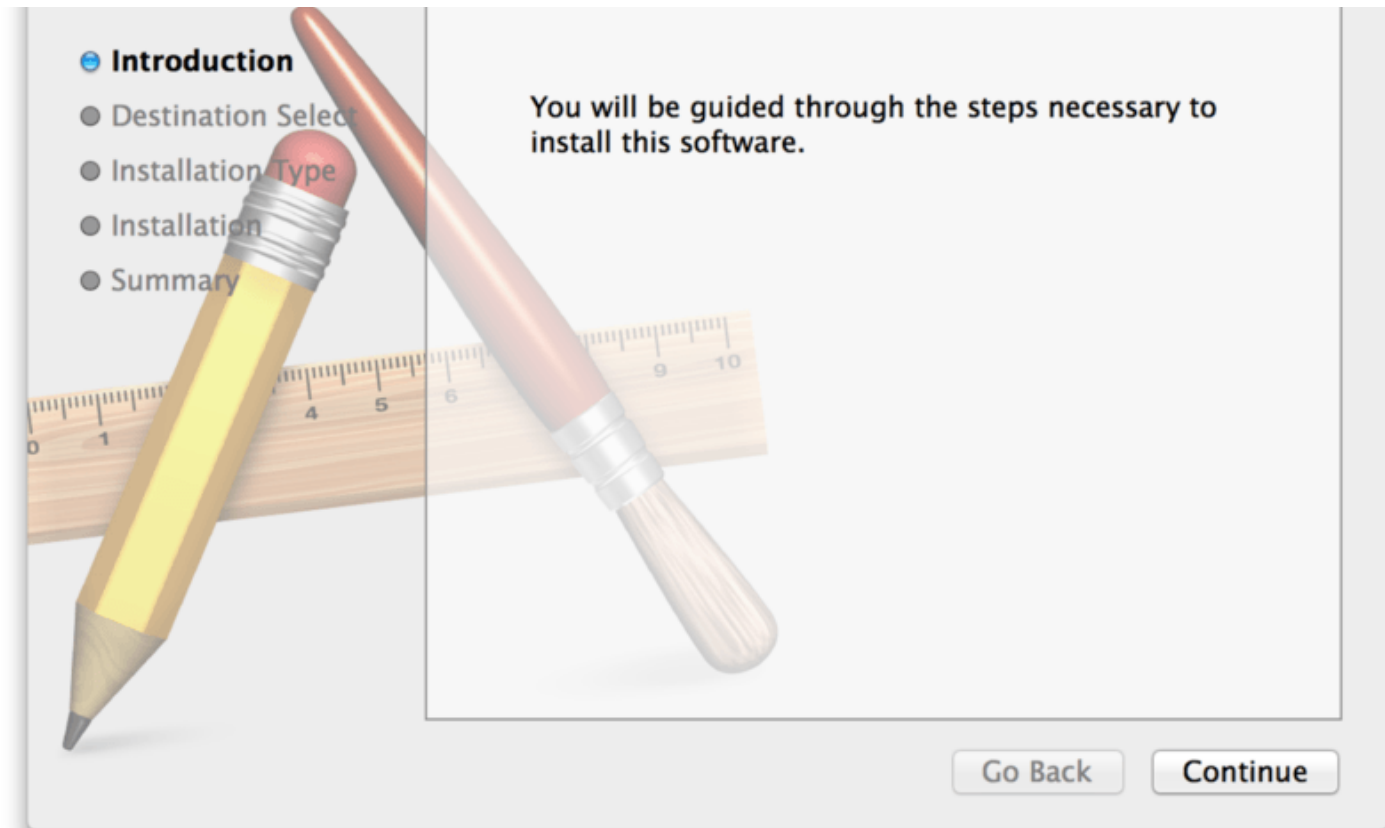


Рисунок 7. OS X инсталлятор Git

Установка в Windows

Для установки Git в Windows также имеется несколько способов. Официальная сборка доступна для скачивания на официальном сайте Git. Просто перейдите на страницу <https://git-scm.com/download/win>, и загрузка запустится автоматически. Обратите внимание, что это отдельный проект, называемый Git для Windows; для получения дополнительной информации о нём перейдите на <https://gitforwindows.org>.

Для автоматической установки вы можете использовать [пакет Git Chocolatey](#). Обратите внимание, что пакет Chocolatey поддерживается сообществом.

Установка из исходников

Многие предпочитают устанавливать Git из исходников, поскольку такой способ позволяет получить самую свежую версию. Обновление бинарных инсталляторов, как правило, немного отстаёт, хотя в последнее время разница не столь существенна.

Если вы действительно хотите установить Git из исходников, у вас должны быть установлены следующие библиотеки, от которых он зависит: autotools, curl, zlib, openssl, expat, and libiconv. Например, если в вашей системе используется dnf (Fedora) или apt-get (системы на базе Debian), вы можете использовать одну из следующих команд для установки всех зависимостей, используемых для сборки и установки бинарных файлов Git:

```
$ 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
```

Для того, чтобы собрать документацию в различных форматах (doc, html, info), понадобится установить дополнительные зависимости:

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

Примечание Пользователи RHEL и производных от неё (таких как CentOS или Scientific Linux) должны [подключить репозиторий EPEL](#) для корректной установки пакета docbook2X

Если вы используете систему на базе Debian (Debian/Ubuntu/Ubuntu-производные), вам так же понадобится установить пакет install-info:

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

Если вы используете систему на базе RPM (Fedora/RHEL/RHEL-производные), вам так же понадобится установить пакет getopt, который уже установлен в системах на базе Debian:

```
$ sudo dnf install getopt
```

К тому же из-за различий имён бинарных файлов вам понадобится сделать следующее:

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

Когда все необходимые зависимости установлены, вы можете пойти дальше и скачать самый свежий архив с исходниками из следующих мест: с сайта Kernel.org <https://www.kernel.org/pub/software/scm/git>, или зеркала на сайте GitHub <https://github.com/git/git/releases>. Конечно, немного проще скачать последнюю версию с сайта GitHub, но на странице kernel.org релизы имеют подписи, если вы хотите проверить, что скачиваете.

Затем скомпилируйте и установите:

```
$ 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
```

После этого вы можете получать обновления Git посредством самого Git:

```
$ 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](#)