
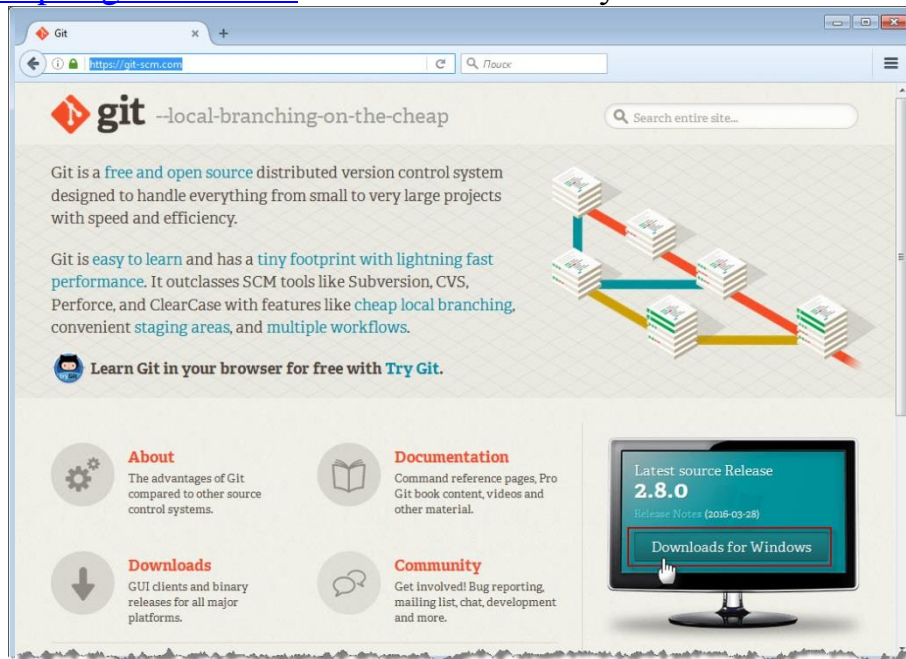


УСТАНОВКА СИСТЕМЫ КОНТРОЛЯ ВЕРСИЙ GIT

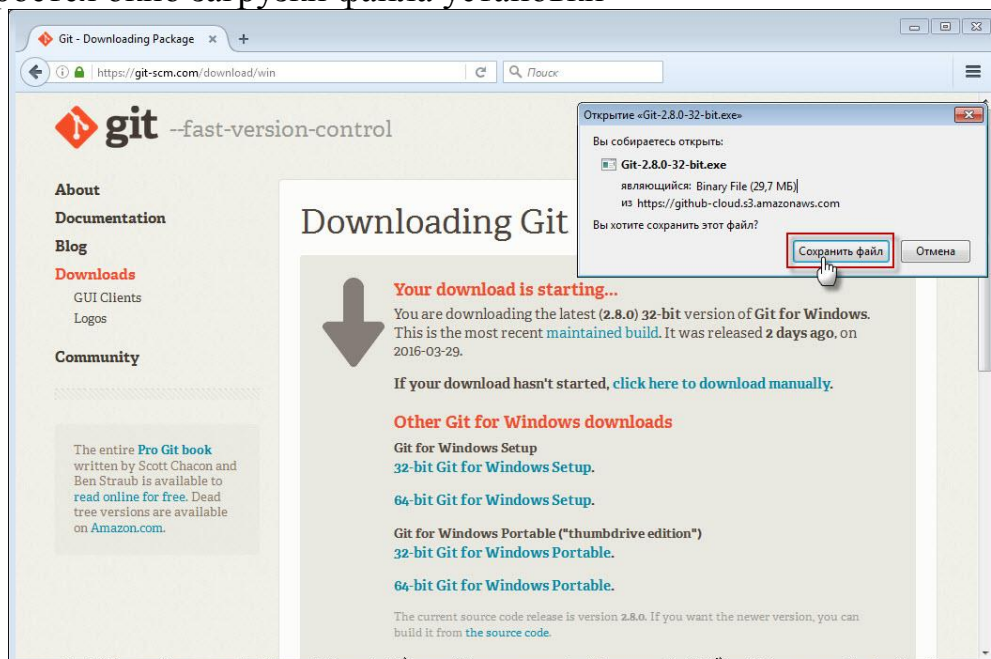
Git – свободно распространяемая мощная и быстрая распределённая система контроля версий с открытым кодом, предназначенная для работы с любыми, как очень маленькими, так и очень большими проектами. В данном документе описан процесс установки **Git** на компьютер с операционной системой *Windows*. Описание установки приведено на примере релиза 2.8.0.


Загрузка файла установки

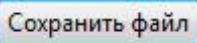
Для загрузки файла установки вам необходимо открыть в веб-браузере страницу <https://git-scm.com/> и нажать на кнопку .



Откроется окно загрузки файла установки

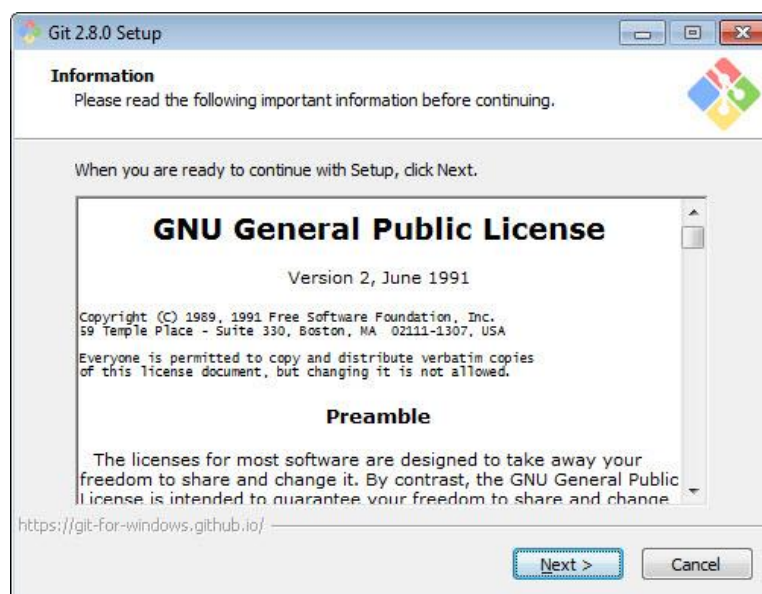


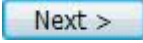
Программа установки автоматически определяет тип операционной системы и предлагает сохранить соответствующий файл установки. Однако вы можете отказаться от скачивания предлагаемого файла, нажав на кнопку , и выбрать другой файл для установки.

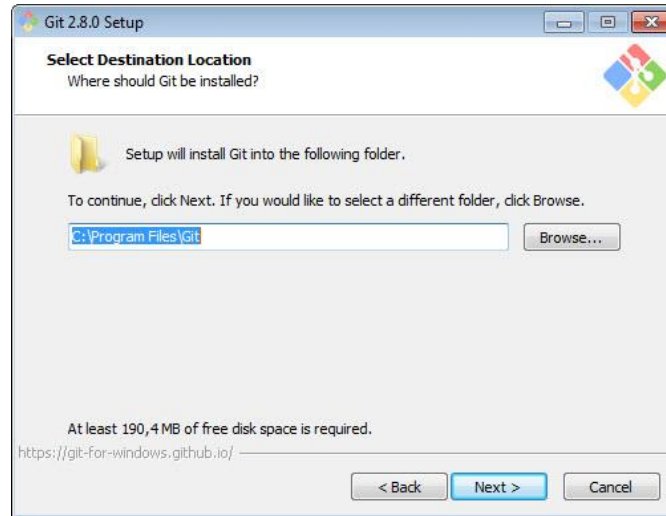
Если предлагаемый для загрузки файл вас устраивает, нажмите на кнопку . Файл будет помещён в папку, назначенную для загрузки файлов.

Установка *Git*. После того, как файл загрузится, можно начать установку. Для этого:

1. Запустите на выполнение загруженный файл (в нашем примере файл Git-2.8.0-32-bit.exe). Откроется окно с текстом лицензионного соглашения. *Git* распространяется в рамках *GNU General Public License* - универсальной общественной лицензии GNU, по которой программное обеспечение передаётся автором в общественную собственность.



2. Прочтите текст лицензионного соглашения, после чего щёлкните по кнопке . Откроется окно выбора папки установки.



3. Вы можете выбрать любую другую папку, имеющуюся на диске, с помощью стандартной процедуры *Windows*, щёлкнув по кнопке **Browse...** или ввести путь в строке «C:\Program Files\Git». Рекомендуется оставить папку, предложенную по умолчанию. В нижней части окна приведены требования к объёму свободного места на диске. Для продолжения установки щёлкните по кнопке **Next >**. Практически на любом этапе установки вы можете вернуться к предыдущему шагу, щёлкнув по кнопке **< Back** или прекратить процесс установки, щёлкнув по кнопке **Cancel**.

4. Откроется окно для выбора устанавливаемых компонентов.



Установкой/снятием флагов сформируйте для себя набор компонентов для установки:

Additional icons - вывод на рабочий стол значков для запуска установленных компонентов;

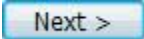
Windows Explore Integration - установка способов вызова **Git** командой контекстного меню на папке (правой кнопкой мыши):

- Git Bash Here** - запуск **Git** в виде консоли из командной строки;
- Git GUI Here** - запуск **Git** в виде графического приложения.

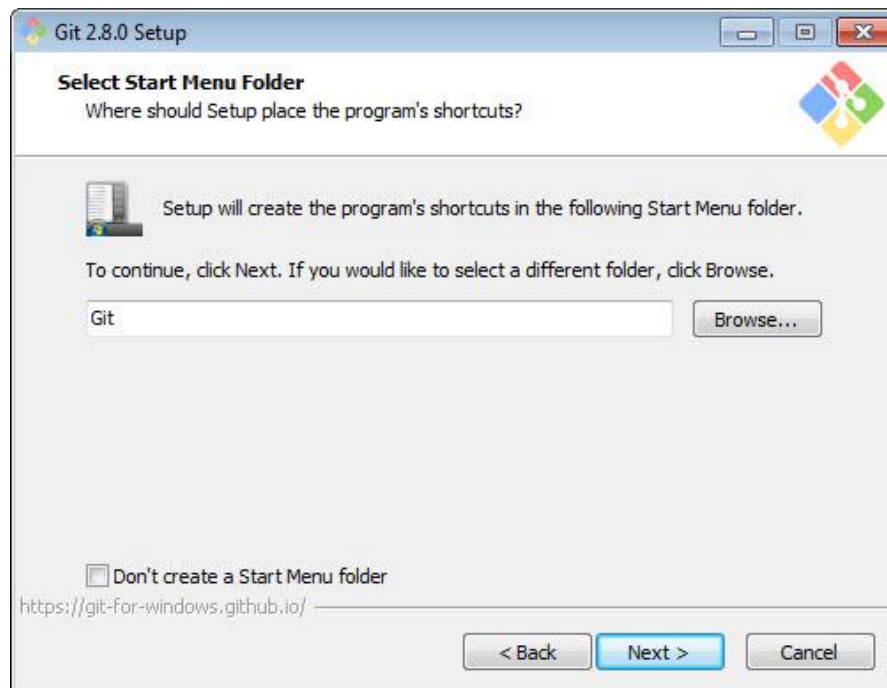
Accociate .git* configuration files with the default text editor - при установленном флаге файлы .git* будут автоматически открываться для редактирования текстовым редактором, установленным по умолчанию;

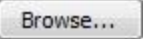
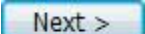
Accociate .sh files to be run with Bash - при установленном флаге файлы .sh* будут автоматически открываться в командной оболочке *Bash*;

Use a TrueType font in all console windows - при установленном флаге во всех консольных окнах будет использоваться шрифт TrueType.

По окончании конфигурирования набора компонентов для продолжения установки щёлкните по кнопке .

5. Откроется окно выбора папки для размещения ярлыков запуска программ.



По умолчанию предлагается папка «Git» («C:\Program Files\Git»). Вы можете выбрать любую другую папку, щёлкнув по кнопке . Можно вовсе отказаться от создания ярлыков запуска, установив флаг «**Don't create a Start Menu Folder**». Для продолжения установки щёлкните по кнопке .

6. Откроется окно настройки переменных окружения.

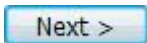


Предлагаются три варианта запуска **Git**. В зависимости от выбранного способа программа установки внесёт соответствующие изменения в переменную окружения *PATH*:

Use Git from Git Bash only - способ установки, при котором переменная *PATH* не затрагивается совсем. Запуск **Git** в этом случае будет возможен только командами контекстного меню «*Git Bash Here*» и «*Git GUI Here*». Данный способ не рекомендуется, так как может потребовать дополнительной работы по конфигурации после установки **Git**;

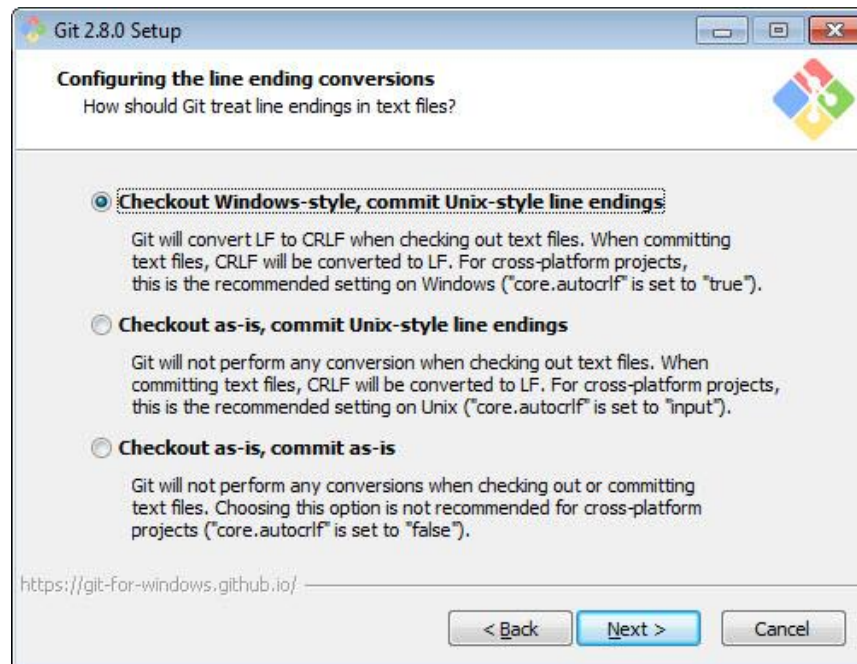
Use Git from the Windows Command Prompt - значение, предлагаемое по умолчанию. При выборе данной опции программа установки вносит минимальные изменения в *PATH*, прописывая только путь к консольной версии, чтобы не загромождать окружение инструментарием Unix. Команду контекстного меню «*Git Bash Here*» также можно использовать. Рекомендуется использовать данный вариант запуска **Git**;

Use Git and optional Unix tools from the Windows Command Prompt - программа установки вносит бóльшие изменения в *PATH*, чем в предыдущем варианте, прописывая пути к различным утилитам Unix. **Предупреждение:** при выборе данного способа вместо Windows-утилит (таких, например, как *find* и *sort*) будут вызываться Unix-утилиты с аналогичными именами. Выбирайте данный способ только если вы полностью осознаёте, к какому результату он приведёт.

Установите переключатель в нужное положение и щёлкните по кнопке  для продолжения установки.

7. Откроется окно, в котором вам нужно установить способ обработки конца строк в текстовых файлах. Команды перевода строк в

Unix-системах и в не Unix-системах различаются. Если в Unix-системах для «перевода строки» и «возврата каретки» используется команда LF (*line feed* - «подача бумаги на строку»), то в не Unix-системах используются две отдельные команды: LF+CR (*line feed* + *carriage return* - «возврат каретки»). Поэтому, на данном шаге требуется указать программе-установщику, файлы какого типа у вас используются и как **Git** должен обрабатывать окончание строки в текстовых файлах.



Здесь также возможны три варианта:

Checkout Windows-style, commit Unix-style line ending - значение по умолчанию. При выборе данного способа **Git** будет конвертировать LF в CRLF при проверке текстовых файлов, а во время фиксирования изменений (коммитов) CRLF будет конвертироваться в LF. Данный способ рекомендуется для Windos-систем при работе с кросс-платформенными проектами (настройка проверки core.autocrlf установлена в true);

Checkout as-is, commit Unix-style line ending - при установке переключателя в это положение **Git** не будет проводить конвертацию при проверке текстовых файлов. Во время коммитов CRLF будет конвертироваться в LF. Данный способ рекомендуется для Unix-систем при работе с кросс-платформенными проектами (настройка проверки core.autocrlf установлена в input);

Checkout as-is, commit as-is - при установке переключателя в данное положение конвертирование производиться не будет. Выбор данного варианта не рекомендуется для кросс-платформенных проектов (настройка проверки core.autocrlf установлена в false).

Установите переключатель в нужное положение и щёлкните по кнопке

Next >

для продолжения установки.

8. Откроется окно для выбора эмулятора терминала, который будет использоваться с **Git Bash**.



В качестве эмулятора терминала можно использовать:

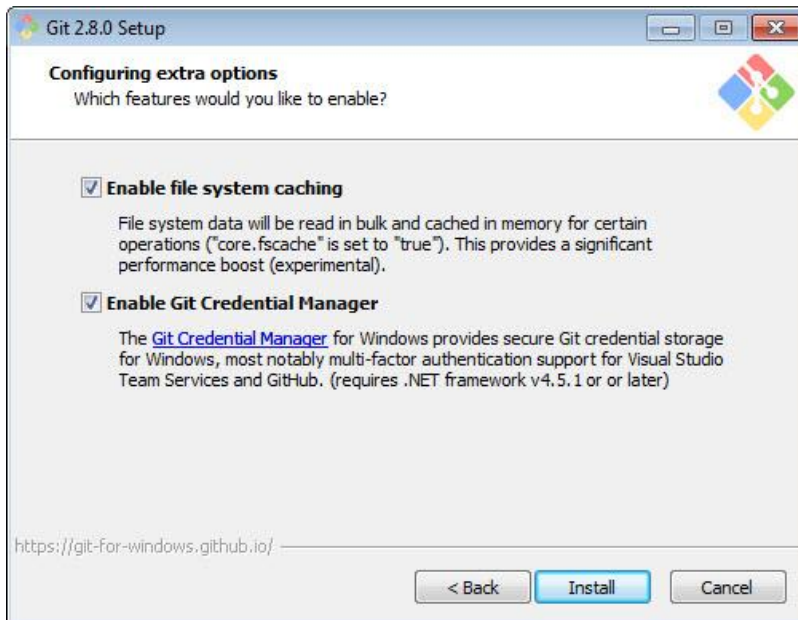
Use MinTTY (the default terminal of MSYS2) - при установке переключателя в это положение **Git Bash** будет использовать MinTTY в качестве эмулятора терминала. MinTTY поддерживает окно с изменяемыми размерами, выделение непрямоугольной области и шрифт Unicode. Консольные Windows-программы (такие как интерактивный *Python*), должны запускаться через winpty для работы в MinTTY;

Use Windows' default console window - при выборе этого варианта **Git** будет использовать консоль Windows (cmd.exe) в качестве консоли по умолчанию. Она хорошо работает с Win32-консольными программами, такими как интерактивная *Python* или *node.js*, но имеет очень ограниченную прокрутку назад для того, чтобы быть сконфигурированной для использования шрифта Unicode с целью корректного отображения не ASCII-символов. При работе с Windows 10, консольное окно не имеет свободно изменяемых размеров и позволяет только выделение прямоугольной области текста. Рекомендуется установить данный эмулятор терминала. Установите переключатель в нужное положение и щёлкните по кнопке

Next >


 для продолжения установки.

9. Откроется окно для установки дополнительных опций.

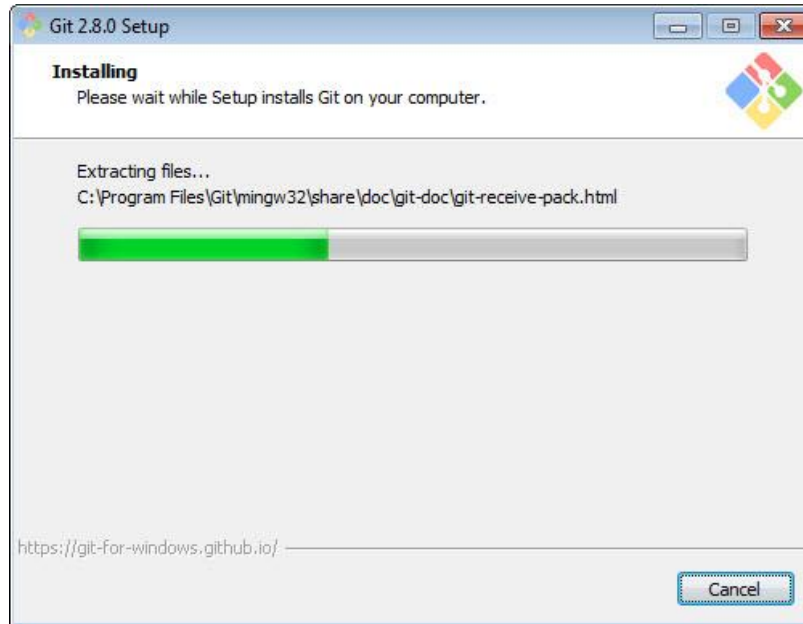


Дополнительными опциями являются:

Enable file system caching – если опция установлена, данные файловой системы будут прочитаны все вместе и заэкшированы в память для выполнения определённых операций. Это обеспечивает значительный прирост производительности (экспериментальные данные);

Enable Git Credential Manager - [Git Credential Manager](https://git-for-windows.github.io/) для Windows обеспечивает безопасное хранение учётных данных **Git** для Windows, в особенности поддержку многофакторной аутентификации для служб Visual Studio и GitHub (должна быть установлена .NET framework версии 4.5.1 или более поздняя). Установите флаги у тех дополнительных опций, которые вы хотите подключить к **Git**, и щёлкните по кнопке  для запуска непосредственно процедуры установки.

10. Откроется окно копирования файлов на ваш компьютер.



Прервать процесс копирования и отменить установку вы можете, щёлкнув по кнопке **Cancel**. Дождитесь окончания копирования. Откроется окно завершения установки:



В окне вы можете установить флаги:

Launch Git Bash - запустить **Git Bash** сразу по завершению установки;

View Release Notes - просмотр файла с описанием релизов **Git** для Windows.

Для завершения установки щёлкните по кнопке **Finish**. Программа установки завершит свою работу. На этом установка **Git** закончена.