



Information

Please read the following important information before continuing.



When you are ready to continue with Setup, click Next.

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your
freedom to share and change it. By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change

<https://gitforwindows.org/>

☒ Only show new options

Install

Cancel



Select Destination Location

Where should Git be installed?



Setup will install Git into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\Git

Browse...

At least 260.0 MB of free disk space is required.

<https://gitforwindows.org/>

< Back

Next >

Cancel



Select Components

Which components should be installed?



Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- ☐ Additional icons
 - ☐ On the Desktop
- ☒ Windows Explorer integration
 - ☒ Git Bash Here
 - ☒ Git GUI Here
- ☒ Git LFS (Large File Support)
- ☒ Associate .git* configuration files with the default text editor
- ☒ Associate .sh files to be run with Bash
- ☐ Use a TrueType font in all console windows
- ☐ Check daily for Git for Windows updates

Current selection requires at least 260.0 MB of disk space.

<https://gitforwindows.org/>

< Back

Next >

Cancel



Select Start Menu Folder

Where should Setup place the program's shortcuts?



Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

Git

Browse...

☐ Don't create a Start Menu folder

<https://gitforwindows.org/>

< Back

Next >

Cancel

Choosing the default editor used by Git

Which editor would you like Git to use?



Use Visual Studio Code as Git's default editor

1

[Visual Studio Code](#) is an Open Source, lightweight and powerful editor running as a desktop application. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

(WARNING!) This will be installed only for this user.

Use this option to let Git use Visual Studio Code as its default editor.

<https://gitforwindows.org/>

< Back

Next >

2

Cancel



Adjusting the name of the initial branch in new repositories



What would you like Git to name the initial branch after "git init"?

☒ **Let Git decide**

Let Git use its default branch name (currently: "master") for the initial branch in newly created repositories. The Git project [intends](#) to change this default to a more inclusive name in the near future.

☐ **Override the default branch name for new repositories**

NEW! Many teams already renamed their default branches; common choices are "main", "trunk" and "development". Specify the name "git init" should use for the initial branch:

This setting does not affect existing repositories.



Adjusting your PATH environment



How would you like to use Git from the command line?

☐ **Use Git from Git Bash only**

This is the most cautious choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.

☒ **Git from the command line and also from 3rd-party software**

(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from Git Bash, the Command Prompt and the Windows PowerShell as well as any third-party software looking for Git in PATH.

☐ **Use Git and optional Unix tools from the Command Prompt**

Both Git and the optional Unix tools will be added to your PATH.
Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

<https://gitforwindows.org/>

< Back

Next >



Cancel



Choosing the SSH executable

Which Secure Shell client program would you like Git to use?



☒ **Use OpenSSH**

This uses `ssh.exe` that comes with Git. The `GIT_SSH` and `SVN_SSH` environment variables will not be modified.

☐ **Use (Tortoise)Plink**

PuTTY sessions were found in your Registry. You may specify the path to an existing copy of (Tortoise)Plink.exe from the TortoiseGit/SVN/CVS or PuTTY applications. "ssh.variant" will be set in the GIT configuration. The `GIT_SSH` and `SVN_SSH` environment variables will be adjusted to point to the following executable:

C:\Program Files\PuTTY\plink.exe

...

☐ Set ssh.variant for Tortoise Plink



Choosing HTTPS transport backend



Which SSL/TLS library would you like Git to use for HTTPS connections?

☒ **Use the OpenSSL library**

Server certificates will be validated using the ca-bundle.crt file.

☐ **Use the native Windows Secure Channel library**

Server certificates will be validated using Windows Certificate Stores.
This option also allows you to use your company's internal Root CA certificates distributed e.g. via Active Directory Domain Services.

<https://gitforwindows.org/>

< Back

Next >

Cancel



Configuring the line ending conversions

How should Git treat line endings in text files?



☒ **Checkout Windows-style, commit Unix-style line endings**

Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").

☐ **Checkout as-is, commit Unix-style line endings**

Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").

☐ **Checkout as-is, commit as-is**

Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").

<https://gitforwindows.org/>

< Back

Next >

Cancel



Configuring the terminal emulator to use with Git Bash



Which terminal emulator do you want to use with your Git Bash?

☒ **Use MinTTY (the default terminal of MSYS2)**

Git Bash will use MinTTY as terminal emulator, which sports a resizable window non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via ``winpty`` to work in MinTTY.

☐ **Use Windows' default console window**

Git will use the default console window of Windows ("cmd.exe"), which works with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.



Choose the default behavior of `git pull`

What should `git pull` do by default?



☒ **Default (fast-forward or merge)**

This is the standard behavior of `git pull` : fast-forward the current branch to the fetched branch when possible, otherwise create a merge commit.

☐ **Rebase**

Rebase the current branch onto the fetched branch. If there are no local commits to rebase, this is equivalent to a fast-forward.

☐ **Only ever fast-forward**

Fast-forward to the fetched branch. Fail if that is not possible.

<https://gitforwindows.org/>

< Back

Next >

Cancel



Choose a credential helper

Which credential helper should be configured?



☒ **Git Credential Manager Core**

(NEW!) Use the new, [cross-platform version of the Git Credential Manager](#).
See more information about the future of Git Credential Manager [here](#).

☐ **Git Credential Manager**

(DEPRECATED) The [Git Credential Manager for Windows](#) handles credentials e.g. for Azure DevOps and GitHub (requires .NET framework v4.5.1 or later).

☐ **None**

Do not use a credential helper.



Configuring extra options



Which features would you like to enable?



Enable file system caching

File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.



Enable symbolic links

Enable [symbolic links](#) (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.



Configuring experimental options



Which bleeding-edge features would you like to enable?

☐

Enable experimental support for pseudo consoles.

(NEW!) This allows running native console programs like Node or Python in a Git Bash window without using winpty, but it still has known bugs.

<https://gitforwindows.org/>

< Back

Install

Cancel



Completing the Git Setup Wizard

Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.



Click Finish to exit Setup.

☐ Launch Git Bash

☒ View Release Notes

Finish