



Git 2.35.1 Setup



### Choosing the default editor used by Git

Which editor would you like Git to use?



Use Visual Studio Code as Git's default editor

[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).

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

<https://gitforwindows.org/>

☐ Only show new options

Back

Next

Cancel



Git 2.35.1 Setup



### 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:

main

This setting does not affect existing repositories.

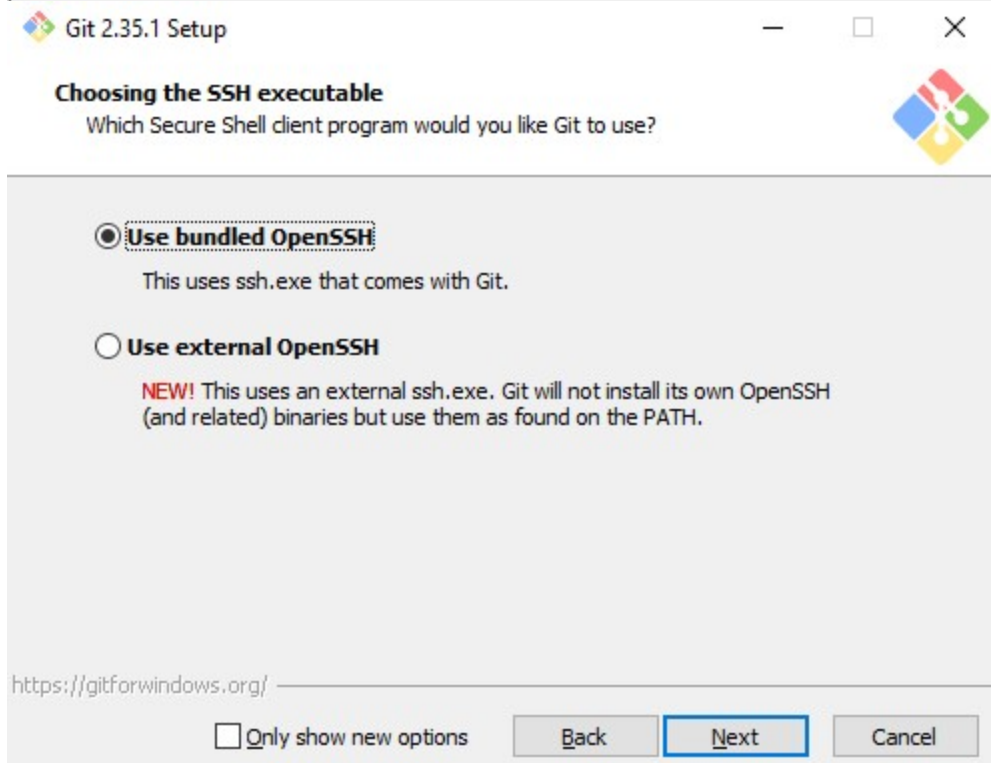
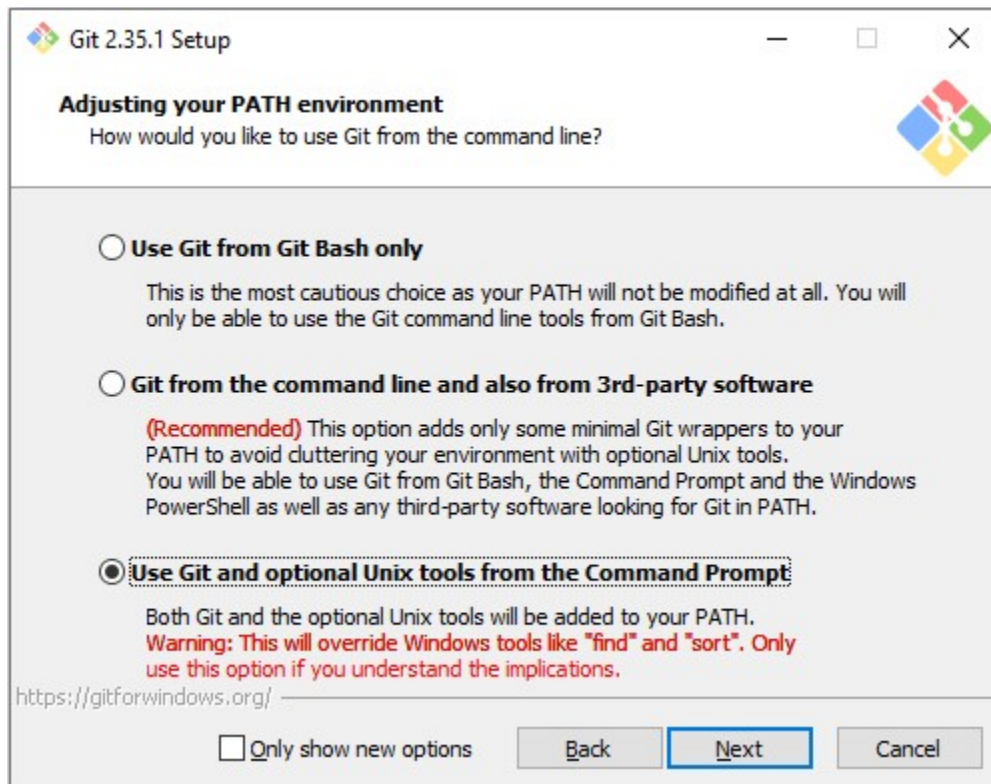
<https://gitforwindows.org/>

☐ Only show new options

Back

Next

Cancel



**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/>☐ Only show new options

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/>☐ Only show new options

Back

Next

Cancel



 Git 2.35.1 Setup


—

□

×

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 `wintpy` to work in MinTTY.

☒ Use Windows' default console window

Git will use the default console window of Windows (`cmd.exe`), which works well 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.

<https://gitforwindows.org/>

☐ Only show new options

Back

Next

Cancel

 Git 2.35.1 Setup

—

□

×

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/>

☐ Only show new options

Back

Next

Cancel

Git 2.35.1 Setup

**Choose a credential helper**  
Which credential helper should be configured?

☒ **Git Credential Manager**  
Use the [cross-platform Git Credential Manager](#).  
See more information about the future of Git Credential Manager [here](#).

☐ **None**  
Do not use a credential helper.

<https://gitforwindows.org/>

☐ Only show new options

Back Next Cancel

Git 2.35.1 Setup

**Configuring extra options**  
Which features would you like to enable?

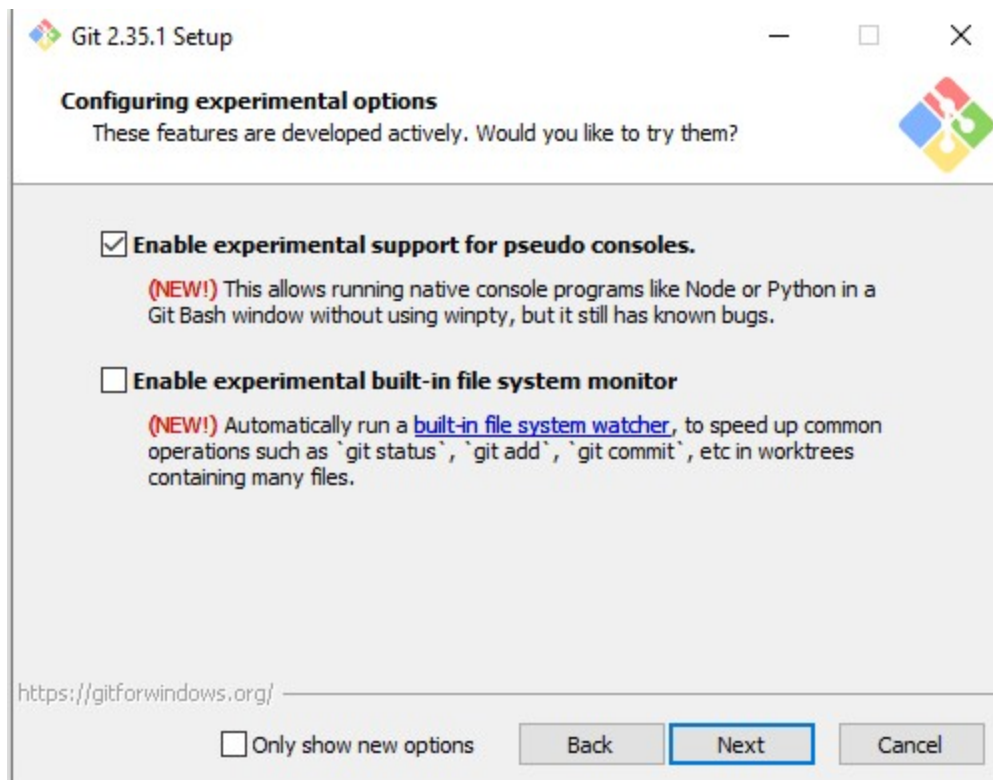
☒ **Enable file system caching**  
File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.

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

<https://gitforwindows.org/>

☐ Only show new options

Back Next Cancel



Laukti :)

