

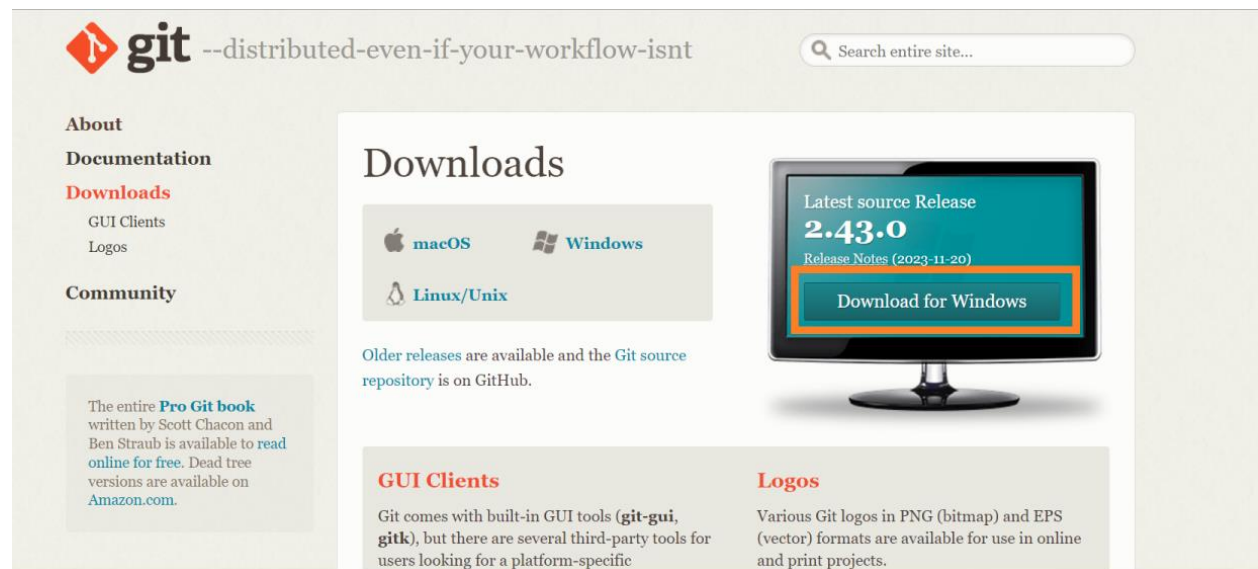
# Set Up GIT in your windows machine.

**Step 1:** Check if git is previously installed or not.

```
C:\Users\Z004CS0M>git --version
'git' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Z004CS0M>
```

**Step 2:** Go to → <https://www.git-scm.com/downloads> , Download the latest release for windows.



# Download for Windows

[Click here to download](#) the latest (**2.43.0**) **64-bit** version of **Git for Windows**. This is the most recent [maintained build](#). It was released **about 2 months ago**, on 2023-11-20.

## Other Git for Windows downloads

### Standalone Installer

[32-bit Git for Windows Setup](#).

[64-bit Git for Windows Setup](#).

### Portable ("thumbdrive edition")

[32-bit Git for Windows Portable](#).

[64-bit Git for Windows Portable](#).

### Using winget tool

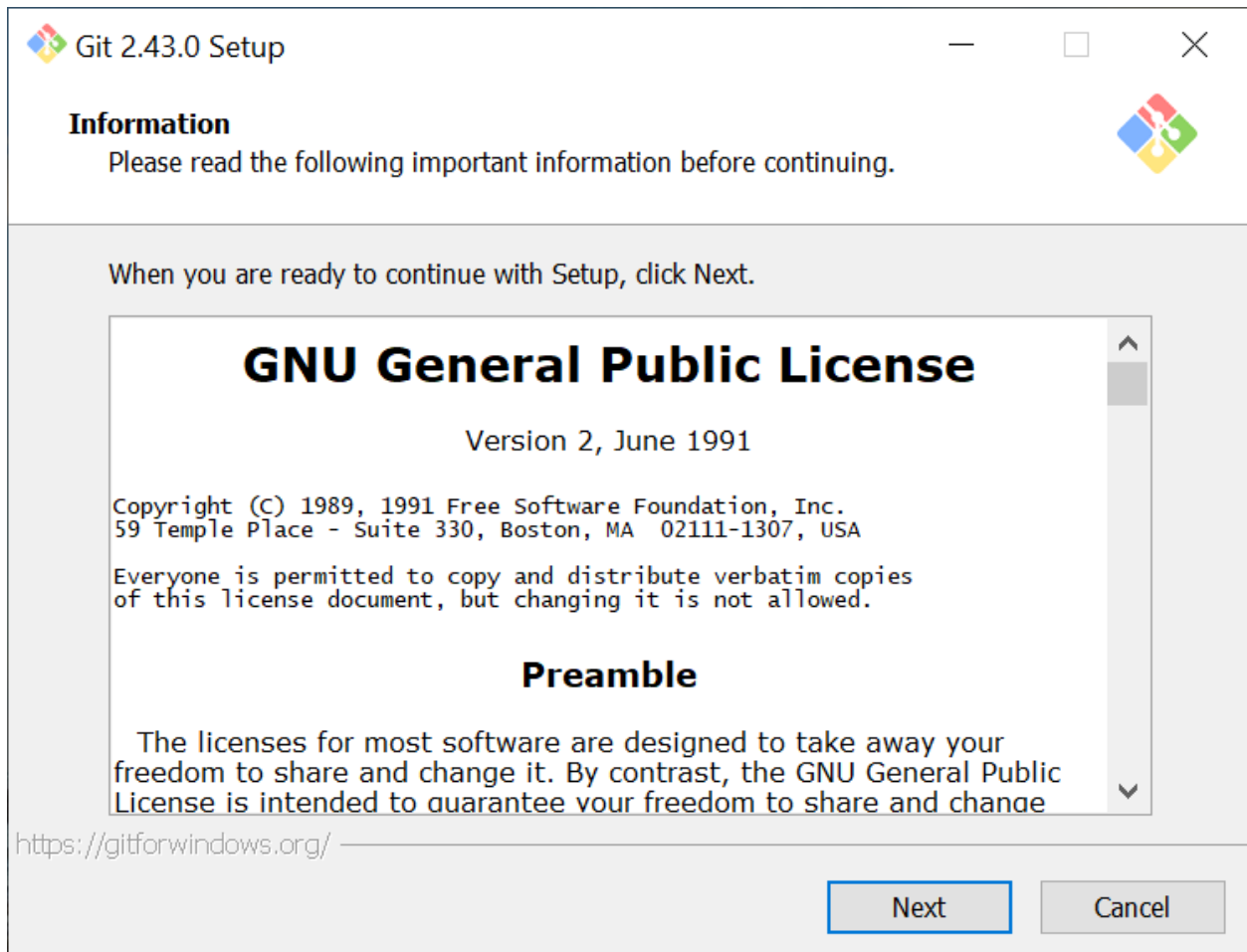
Install [winget tool](#) if you don't already have it, then type this command in command prompt or Powershell.

```
winget install --id Git.Git -e --source winget
```

The current source code release is version **2.43.0**. If you want the newer version, you can build it from [the source code](#).

**Git-2.43.0-64-bit.exe** will be downloaded on your machine.

**Step 3:** Double click on downloaded file and follow below steps.



**Click Next.**

## 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 320.9 MB of free disk space is required.

<https://gitforwindows.org/>

Back

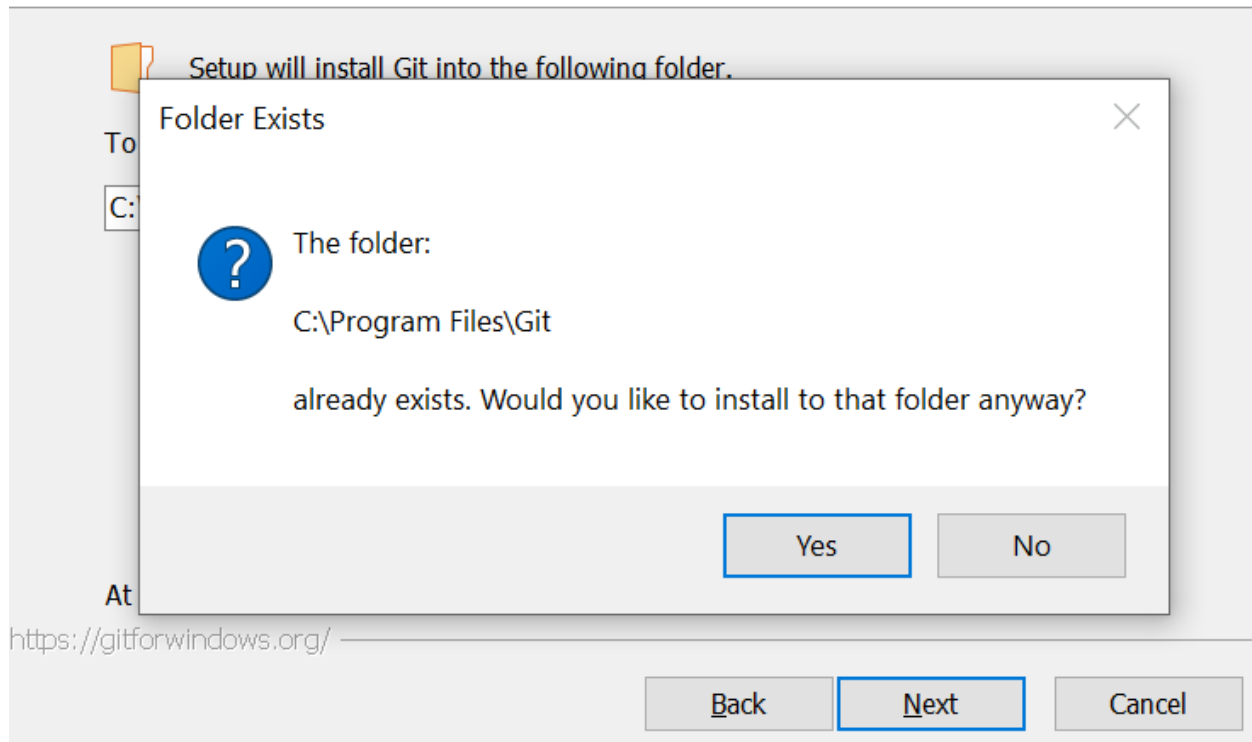
Next

Cancel

## Click Next

## Select Destination Location

Where should Git be installed?



Click Yes.



Git 2.43.0 Setup



## 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.

<input type="checkbox"/>	Additional icons	^
<input type="checkbox"/>	On the Desktop	
<input checked="" type="checkbox"/>	Windows Explorer integration	
<input checked="" type="checkbox"/>	Open Git Bash here	
<input checked="" type="checkbox"/>	Open Git GUI here	
<input checked="" type="checkbox"/>	Git LFS (Large File Support)	
<input checked="" type="checkbox"/>	Associate .git* configuration files with the default text editor	
<input checked="" type="checkbox"/>	Associate .sh files to be run with Bash	
<input type="checkbox"/>	Check daily for Git for Windows updates	
<input type="checkbox"/>	(NEW!) Add a Git Bash Profile to Windows Terminal	v

Current selection requires at least 320.8 MB of disk space.

<https://gitforwindows.org/>

Back

Next

Cancel

## Click Next

### 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

## Click Next

### Choosing the default editor used by Git

Which editor would you like Git to use?



Use Notepad as Git's default editor



(NEW!) Notepad is a simple GUI editor that comes with Windows.

<https://gitforwindows.org/>

Back

Next

Cancel

Click Next



## 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.

<https://gitforwindows.org/>

Back

Next

Cancel

Click Next

## 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

Click Next



Git 2.43.0 Setup



### Choosing the SSH executable

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



☒ **Use bundled OpenSSH**

This uses ssh.exe that comes with Git.

☐ **Use (Tortoise)Plink**

To use PuTTY, specify the path to an existing copy of (Tortoise)Plink.exe:

C:\Program Files\PuTTY\plink.exe



☐ Set ssh.variant for Tortoise Plink

☐ **Use external OpenSSH**

**NEW!** This uses an external ssh.exe. Git will not install its own OpenSSH (and related) binaries but use them as found on the PATH.

<https://gitforwindows.org/>

Back

Next

Cancel

Click Next

### 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

Click Next

### 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

Click Next

**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 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/>

Back

Next

Cancel

**Click Next**

### Choose the default behavior of `git pull`

What should `git pull` do by default?



☒ **Fast-forward or merge**

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. This is the standard behavior of `git pull`.

<https://gitforwindows.org/>

Back

Next

Cancel

Click Next

### 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/>

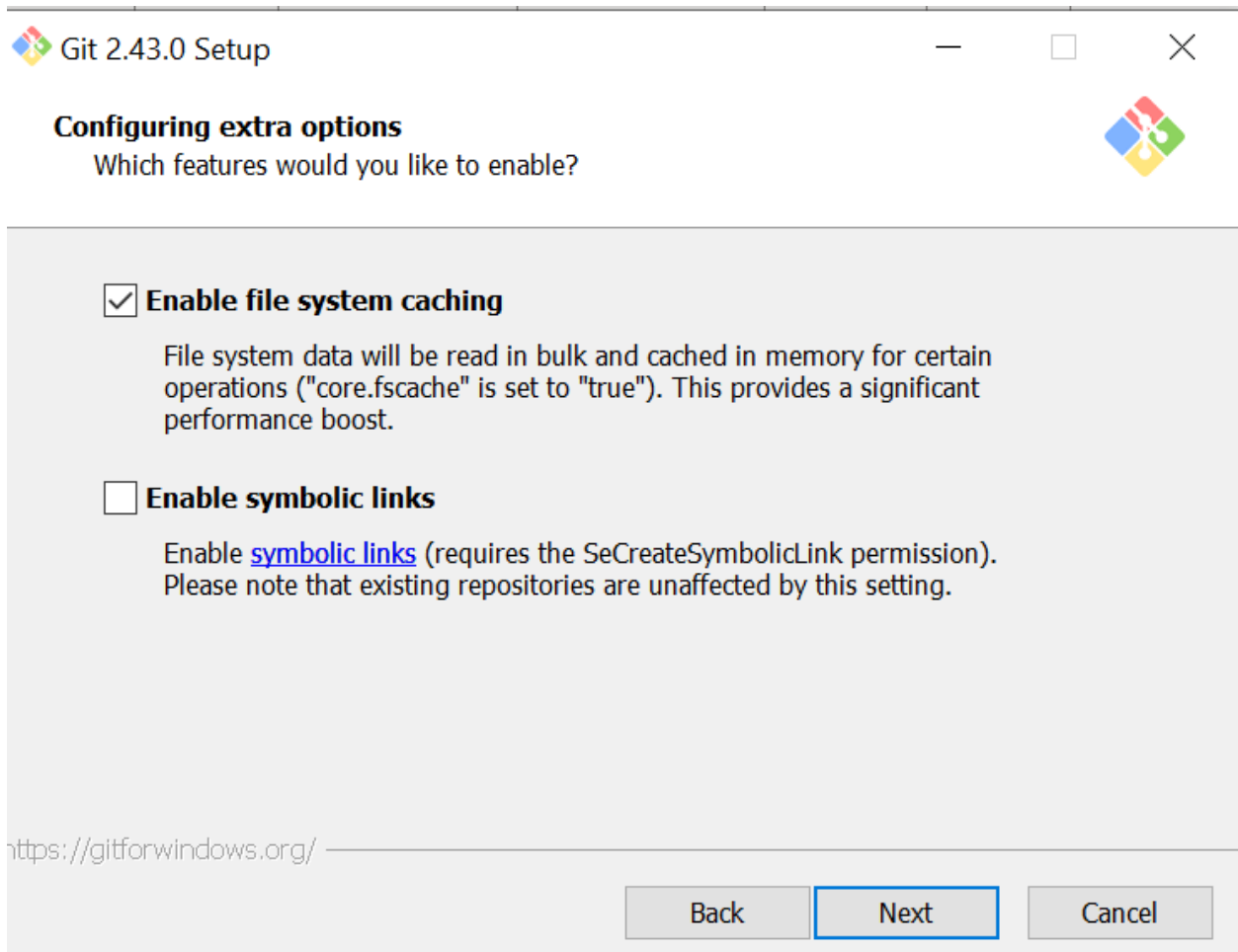
Back

Next

Cancel

Click Next





Click Next

### Configuring experimental options

These features are developed actively. Would you like to try them?



☐ **Enable experimental support for pseudo consoles.**

This allows running native console programs like Node or Python in a Git Bash window without using winpty, but is unfortunately not quite stable yet.

☐ **Enable experimental built-in file system monitor**

**(NEW!)** Automatically run a [built-in file system watcher](#), to speed up common operations such as ``git status``, ``git add``, ``git commit``, etc in worktrees containing many files.

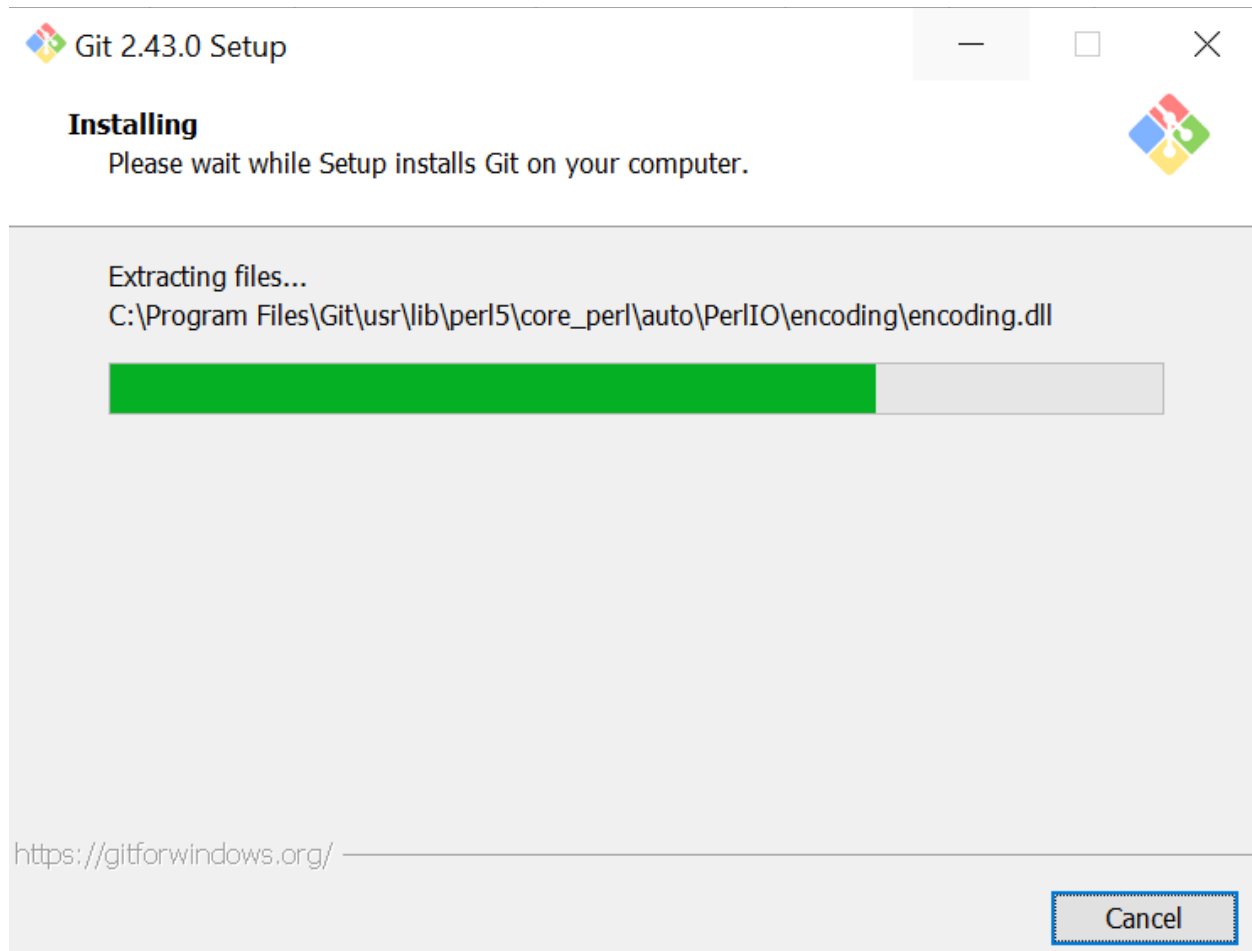
<https://gitforwindows.org/>

Back

Install

Cancel

Click Install



**Git is installed, Click on Finish.**

## 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

**Step 4:** Verify if git is installed.

```
C:\Users\Z004CS0M>git --version
git version 2.43.0.windows.1

C:\Users\Z004CS0M>
```

Git is installed successfully! Now create an account in github.

**<END OF DOCUMENT>**