Dhaval Joshi
Roll No:-14 T.E(IT)

Experiment No:2

Aim: To understand Version Control System / Source Code Management, install git and create a GitHub account.

Theory:

Version Control System:

A Version Control System (VCS) is a way to make changes to files without worrying about something that will get lost or things will fall out of the flow. Version Control also offers backup and history of any changes for any files line-by-line.

The success of the DevOps depends on the Source Control. Version Control or Source Control in DevOps helps to manage the changes done during development process in a project. It can be versions of code, documents, or even environment configuration.

Source Code Management:

Source Code Management or SCM is a DevOps automation tool that maintains a track of versions (revisions) made to the program. Each version has a timestamp and the person who made the changes. These versions can be compared and merged. SCM is also known as Version Control.

Benefits of using Version Control (VCS):

- Provides a complete history of every file, made by the individuals over the years. These changes include the creation, deletion, and edits to the files. History also displays the author, date, and notes of why the change was made. This will help to find the root cause of the problem in the future.

- VC protects the source code from any accidental changes, human mistakes, and unintended circumstances. For any software project, the source code is the most important asset. Hence it is crucial to protect it.

Because source code contains all the data and revision history related to a project.

- The facility of Branching and Merging. A Branch in VCS will keep various streams of work independent from each other. At the same time it will also provide the facility to merge that work back together. There are many workflows to choose from. The team can choose the workflow depending on how to make use of branching and merging facility in the VCS.
- Source Control or Version Control helps to understand the difference between various versions and also the purpose of making the changes.
- Version Control is especially beneficial in case of large distributed teams, where developers work on multiple projects and its difficult to keep a track of changes and updates across teams.
- Version Control ensures that Dev and Ops are in there in the source control. This is the beauty of DevOps. This makes it clear and transparent to everyone to see what is there in the source and each version of it.

There are many Version Control tools for DevOps available in the market.

Example: -

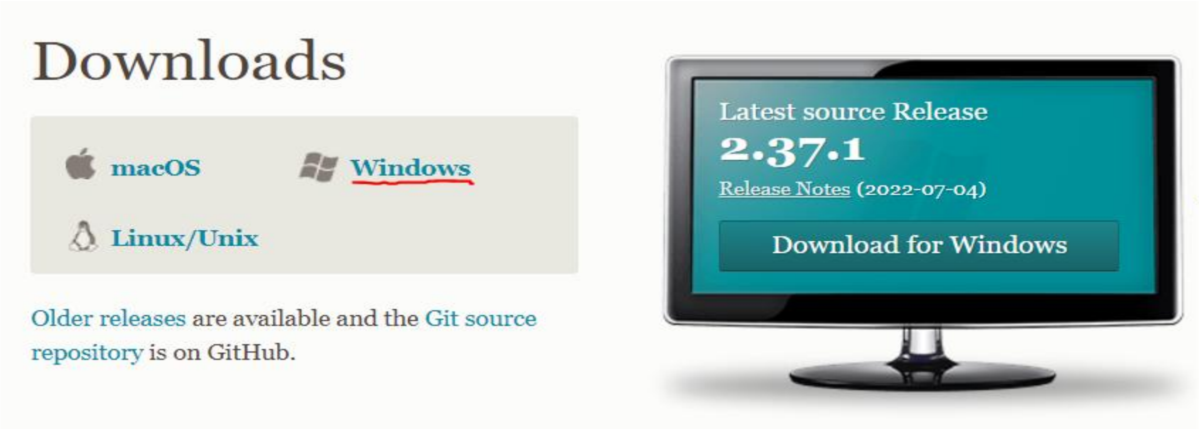
GitHub: Git is an open-source Version Control System (VCS), it is completely free. Git is designed to work in small to large level projects. Git will help to merge and maintain the history of code changes. GitHub is the repository where all the source code is kept by Git users. GitHub offers local branching and multiple workflows. It is easy to learn and offers faster operation speed.

Installation of Git:-

STEPS:

1. Browse to the official Git website: https://git-scm.com/downloads

2. Click the download link for Windows and allow the download to complete.



Downloads

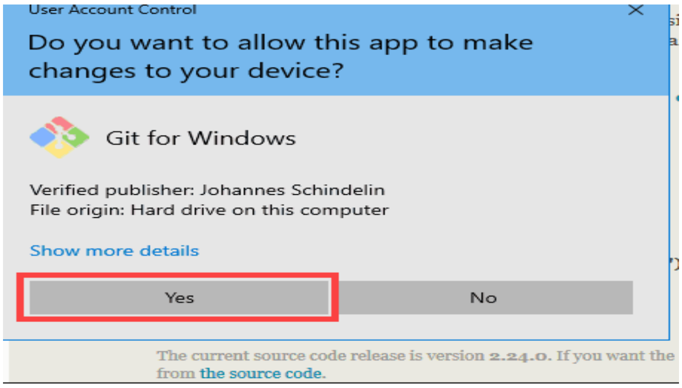
Older releases are available and the Git source repository is on GitHub.

Latest source Release
2.37.1
Release Notes (2022-07-04)

Fig 2.1

3. Browse to the download location (or use the download shortcut in your browser). Double-click the file to extract and launch the installer.

4. Allow the app to make changes to your device by clicking **Yes** on the User Account Control dialog that opens.



User Account Control

Do you want to allow this app to make changes to your device?

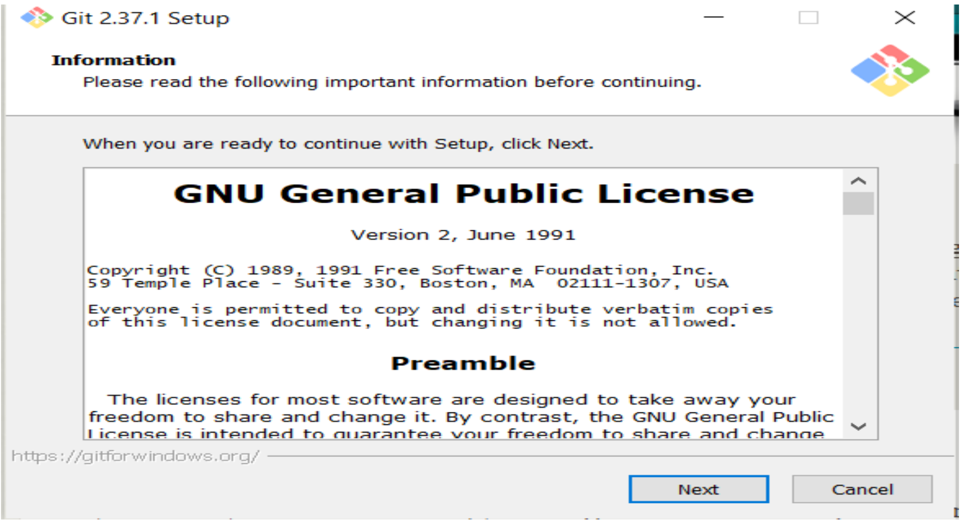
Git for Windows

Verified publisher: Johannes Schindelin
File origin: Hard drive on this computer

The current source code release is version 2.24.0. If you want the
from the source code.

Fig 2.2

5. Review the GNU General Public License, and when you're ready to install, click **Next**.



Git 2.37.1 Setup

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

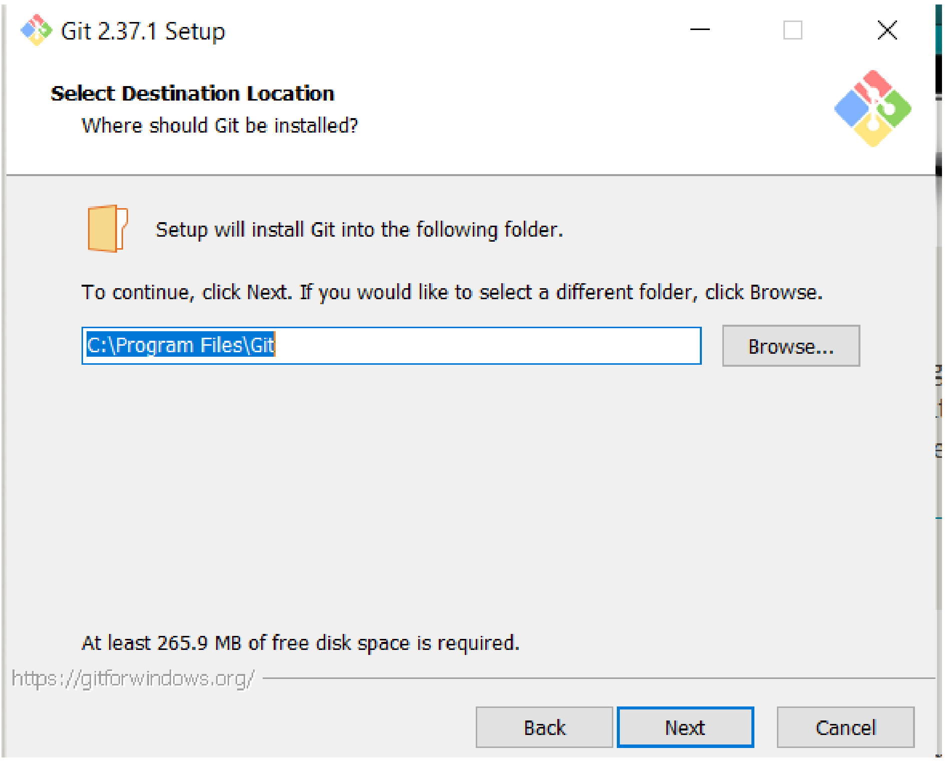
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

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

Fig 2.3

6. The installer will ask you for an installation location. Leave the default, unless you have reason to change it, and click **Next**.



Git 2.37.1 Setup

Select Destination Location

Where should Git be installed?

Setup will install Git into the following folder.

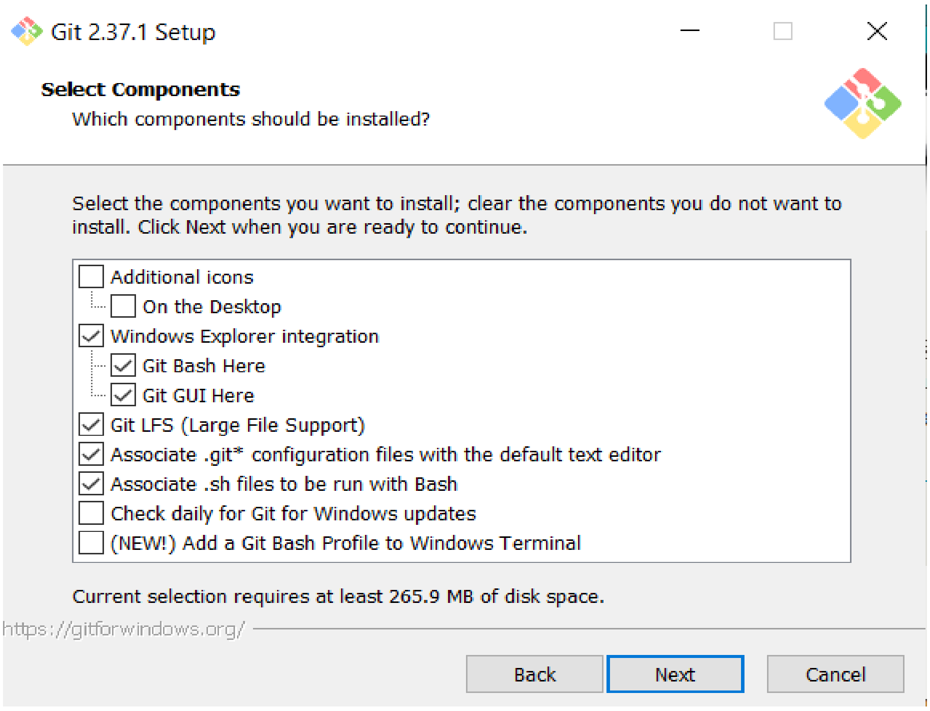
To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\Git

At least 265.9 MB of free disk space is required.

Fig 2.4

7. A component selection screen will appear. Leave the defaults unless you have a specific need to change them and click **Next**.



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.

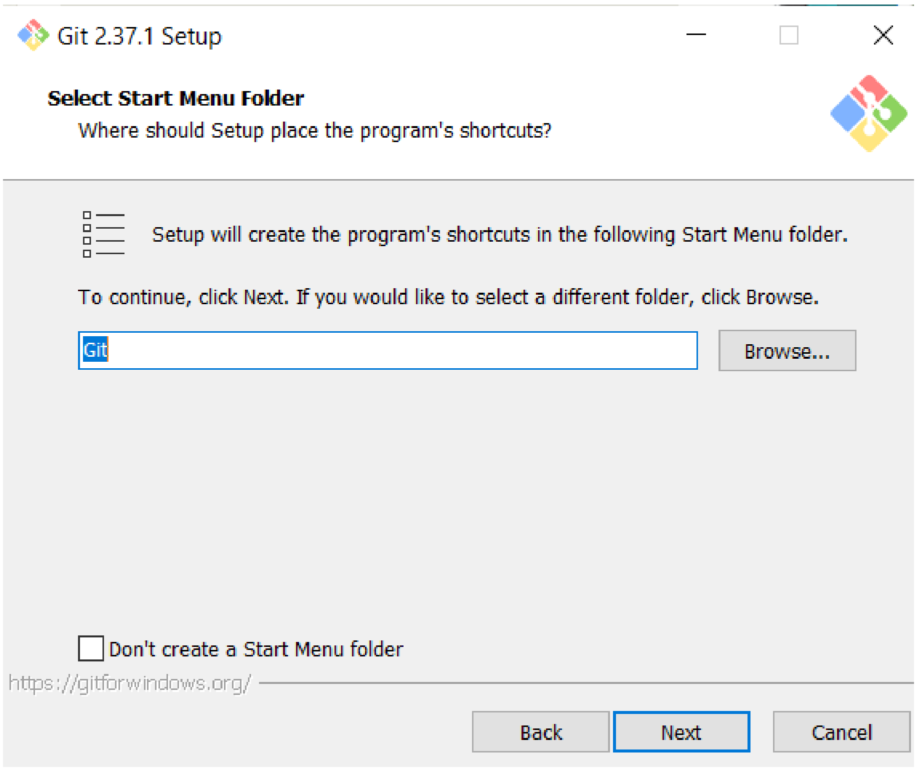
- □ Additional icons
  - □ On the Desktop
- ☑ Windows Explorer integration
  - ☑ Git Bash Here
  - ☑ Git GUI Here
- ☑ Git LFS (Large File Support)
- ☑ Associate .git* configuration files with the default text editor
- ☑ Associate .sh files to be run with Bash
- □ Check daily for Git for Windows updates
- □ (NEW!) Add a Git Bash Profile to Windows Terminal

Current selection requires at least 265.9 MB of disk space.

https://gitforwindows.org/

Fig 2.5

8. The installer will offer to create a start menu folder. Simply 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.

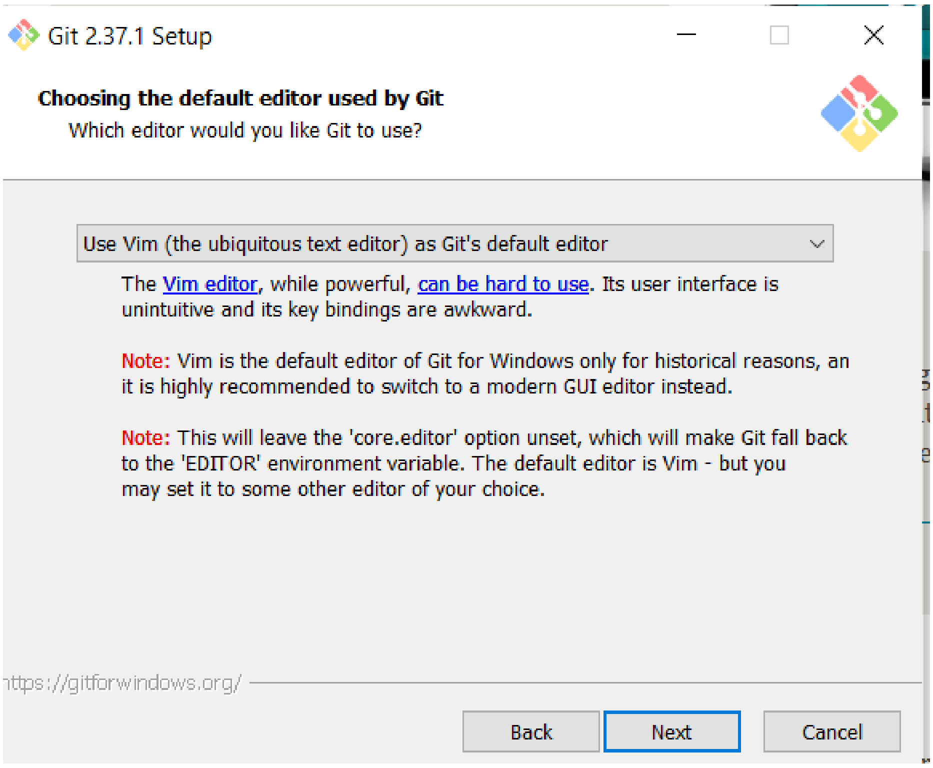
Git

□ Don't create a Start Menu folder

https://gitforwindows.org/

Fig 2.6

9. Select a text editor you'd like to use with Git. Use the drop-down menu to select whichever text editor you prefer and click **Next**.



Choosing the default editor used by Git

Which editor would you like Git to use?

Use Vim (the ubiquitous text editor) as Git's default editor

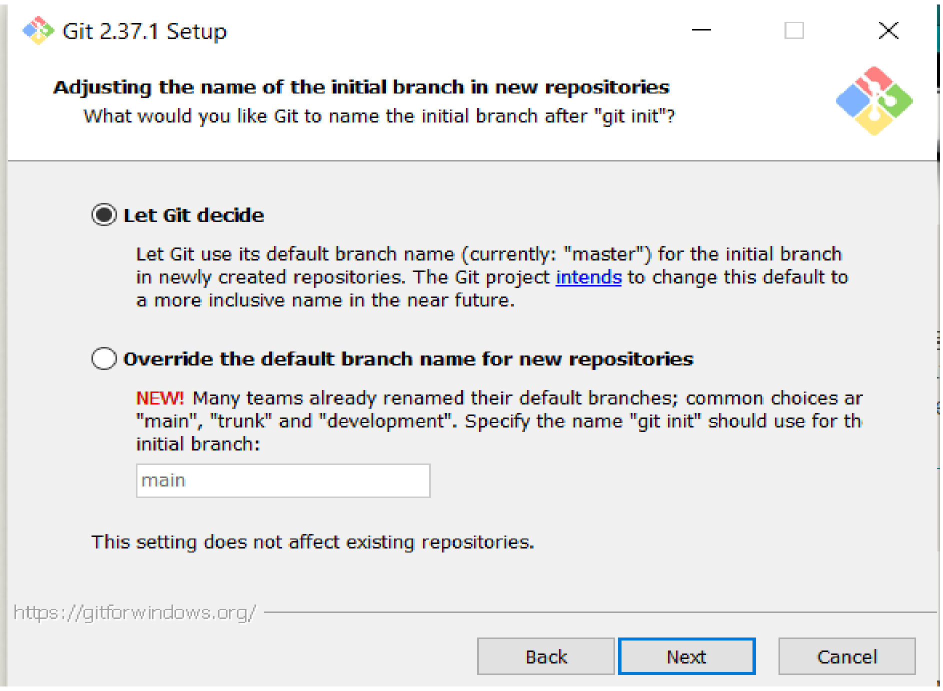
The Vim editor, while powerful, can be hard to use. Its user interface is unintuitive and its key bindings are awkward.

Note: Vim is the default editor of Git for Windows only for historical reasons, an it is highly recommended to switch to a modern GUI editor instead.

Note: This will leave the 'core.editor' option unset, which will make Git fall back to the 'EDITOR' environment variable. The default editor is Vim - but you may set it to some other editor of your choice.

Fig 2.7

10. The next step allows you to choose a different name for your initial branch. The default is 'master.' Unless you're working in a team that requires a different name, leave the default option and 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**

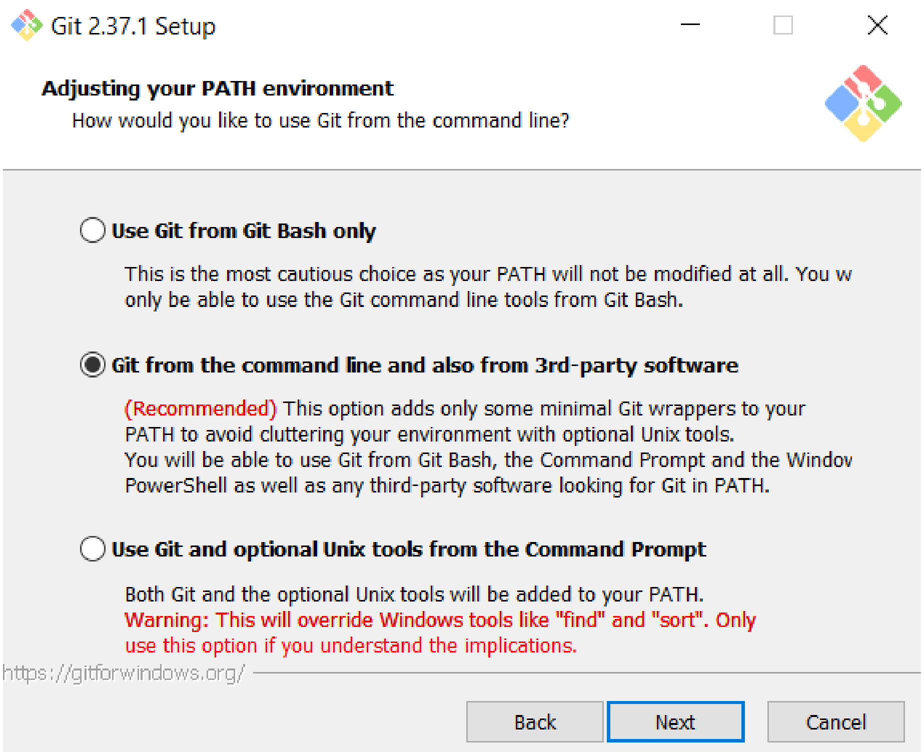
NEW! Many teams already renamed their default branches; common choices ar "main", "trunk" and "development". Specify the name "git init" should use for th initial branch:

main

This setting does not affect existing repositories.

Fig 2.8

11. This installation step allows you to change the **PATH environment**. The **PATH** is the default set of directories included when you run a command from the command line. Leave this on the middle (recommended) selection and click **Next**.



Git 2.37.1 Setup

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 w only be able to use the Git command line tools from Git Bash.

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

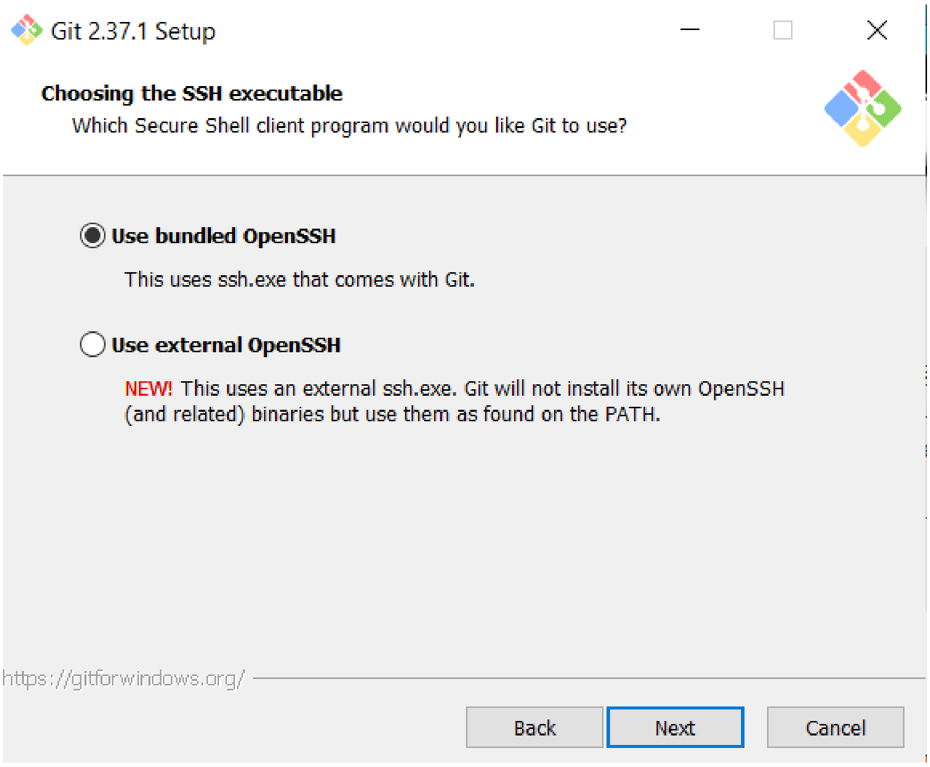
(Recommended) This option adds only some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from Git Bash, the Command Prompt and the Windov PowerShell as well as any third-party software looking for Git in PATH.

○ **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.

Fig 2.9

12. The installer now asks which SSH client you want Git to use. Git already comes with its own SSH client, so if you don't need a specific one, leave the default option and click **Next.**



Git 2.37.1 Setup

Choosing the SSH executable

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

● **Use bundled OpenSSH**

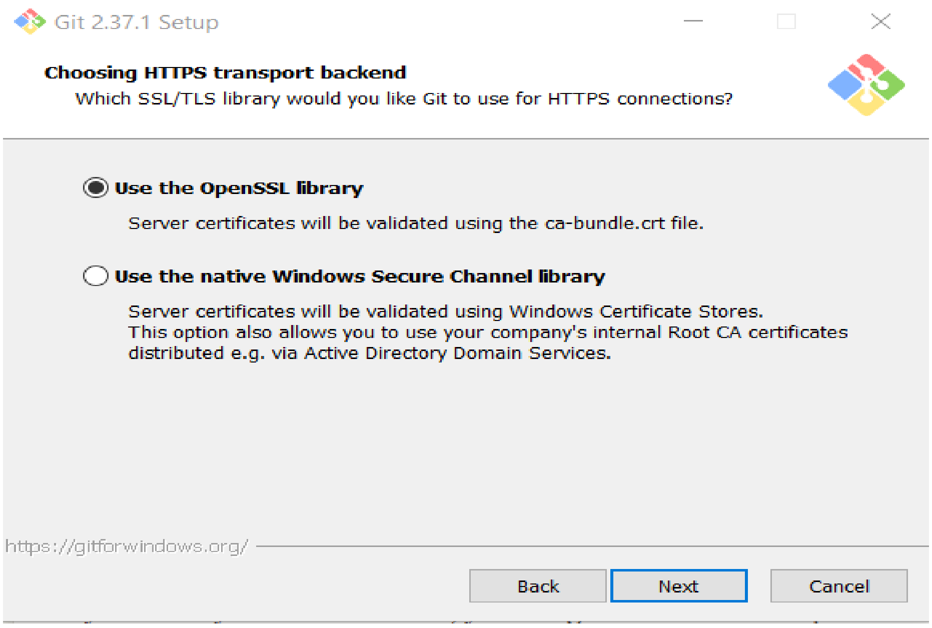
This uses ssh.exe that comes with Git.

○ **Use 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.

Fig 2.10

13. The next option relates to server certificates. Most users should use the default. If you're working in an Active Directory environment, you may need to switch to Windows Store certificates. Click **Next**.



Git 2.37.1 Setup

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.

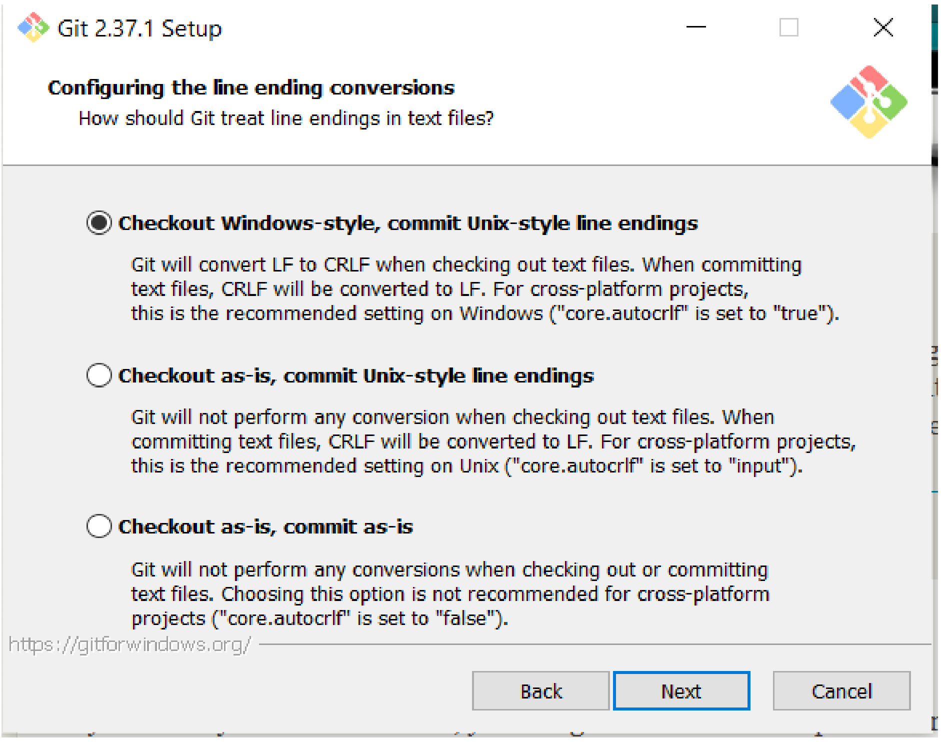
○ **Use the native Windows Secure Channel library**

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/

Fig 2.11

14. The next selection converts line endings. It is recommended that you leave the default selection. This relates to the way data is formatted and changing this option may cause problems. Click **Next**.



Git 2.37.1 Setup

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").

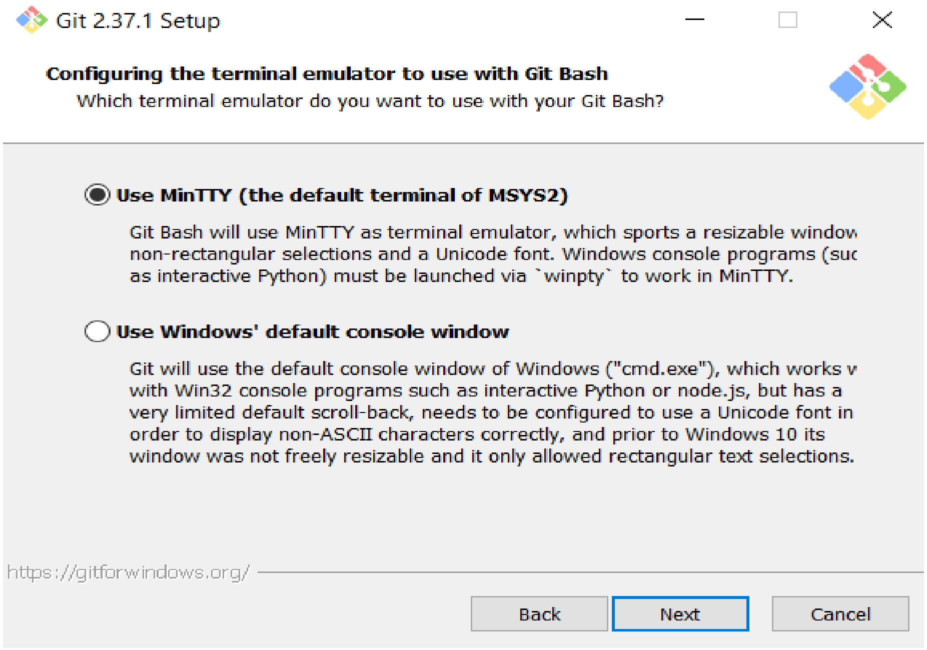
○ **Checkout as-is, commit as-is**

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/

Fig 2.12

15. Choose the terminal emulator you want to use. The default MinTTY is recommended, for its features. Click **Next**.



Git 2.37.1 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)**

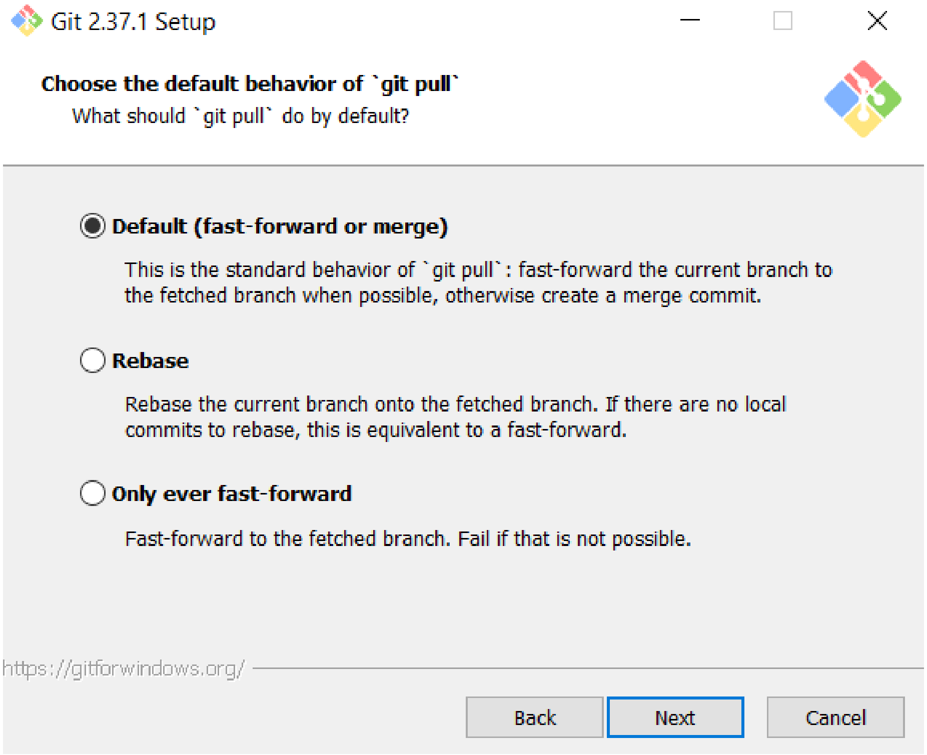
Git Bash will use MinTTY as terminal emulator, which sports a resizable window non-rectangular selections and a Unicode font. Windows console programs (suc as interactive Python) must be launched via `winpty` to work in MinTTY.

○ **Use Windows' default console window**

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

Fig 2.13

16. The installer now asks what the `git pull` command should do. The default option is recommended unless you specifically need to change its behavior. Click **Next** to continue with the installation.



Git 2.37.1 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**

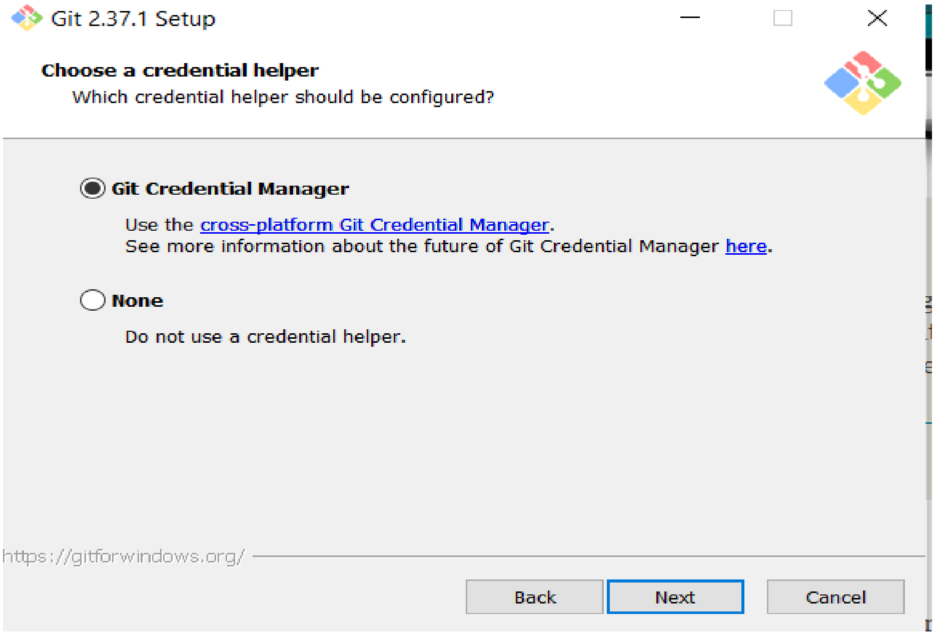
Rebase the current branch onto the fetched branch. If there are no local commits to rebase, this is equivalent to a fast-forward.

○ **Only ever fast-forward**

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

Fig 2.14

17. Next you should choose which credential helper to use. Git uses credential helpers to fetch or save credentials. Leave the default option as it is the most stable one, and click **Next**.



Git 2.37.1 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.

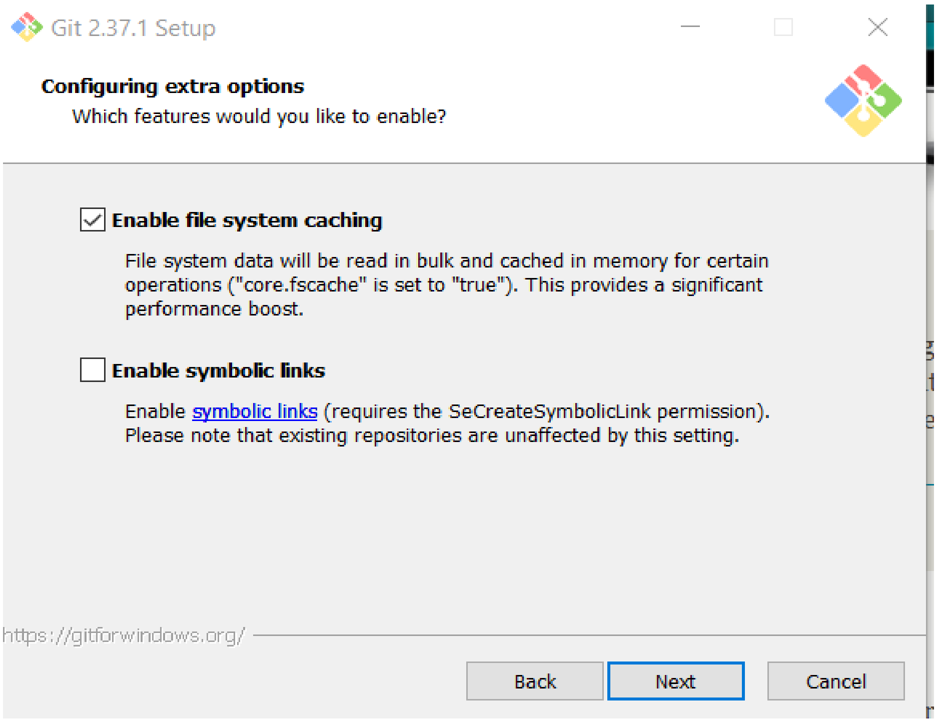
○ **None**

Do not use a credential helper.

https://gitforwindows.org/

Fig 2.15

18. The default options are recommended, however this step allows you to decide which extra option you would like to enable. If you use symbolic links, which are like shortcuts for the command line, tick the box. Click **Next**.



Git 2.37.1 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.

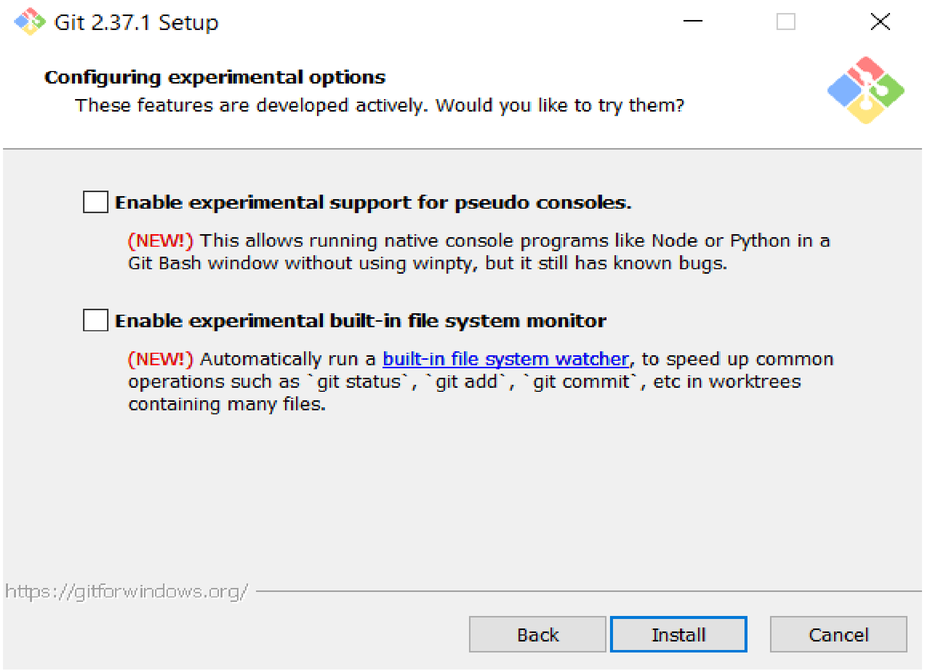
☐ **Enable symbolic links**

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

https://gitforwindows.org/

Fig 2.16

19. Depending on the version of Git you're installing, it may offer to install experimental features. At the time this article was written, the options to include support for pseudo controls and a built-in file system monitor were offered. Unless you are feeling adventurous, leave them unchecked and click **Install**.



Configuring experimental options

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

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

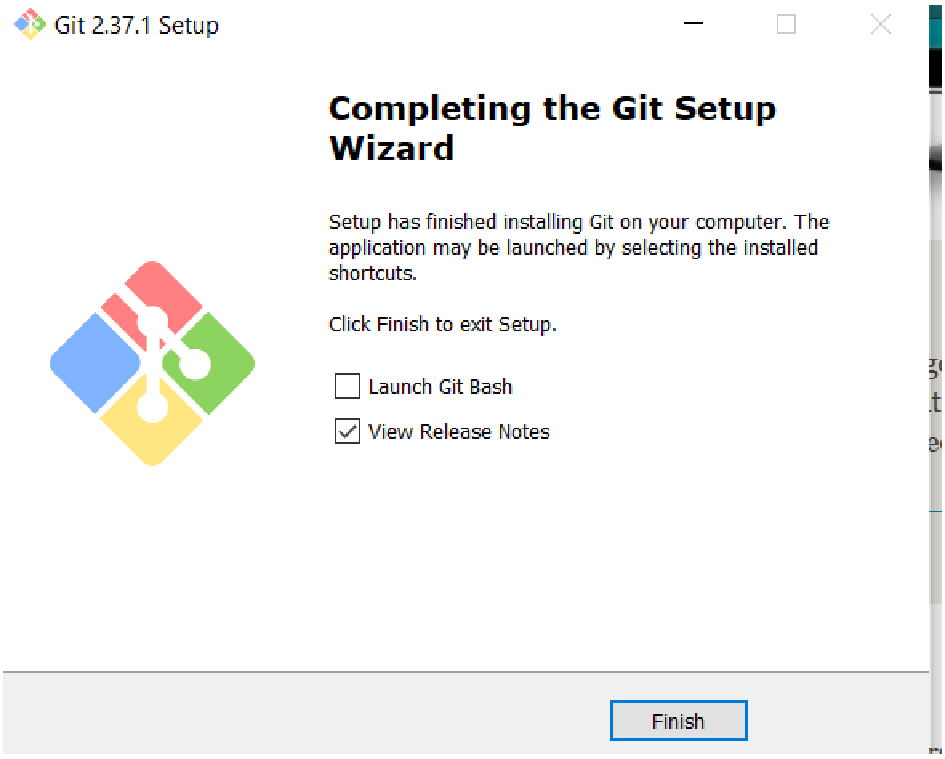
(NEW!) This allows running native console programs like Node or Python in a Git Bash window without using winpty, but it still has known bugs.

□ **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.

Fig 2.17

20. Once the installation is complete, tick the boxes to view the Release Notes or Launch Git Bash, then click **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.

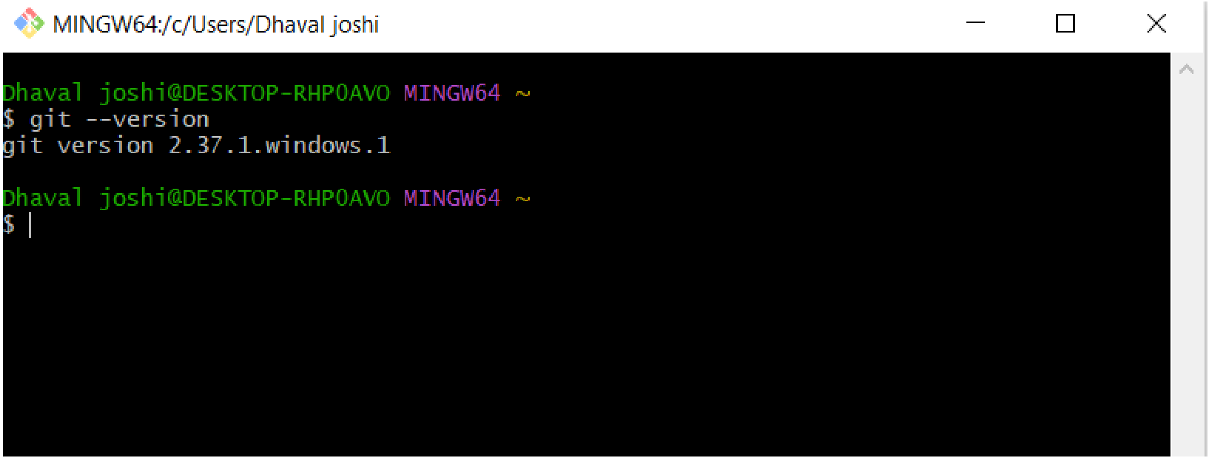
□ Launch Git Bash

☑ View Release Notes

Fig 2.18

Dhaval Joshi
Roll No:-14 T.E(IT)

Git Bash:-

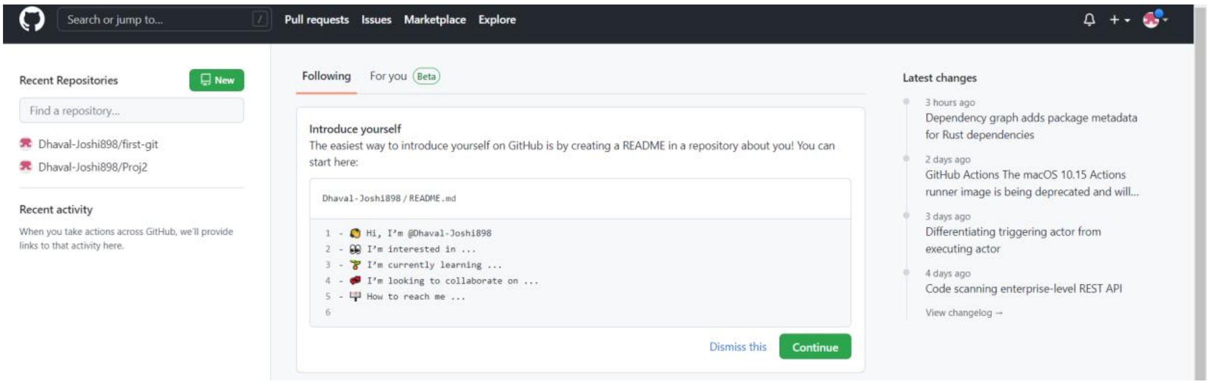


```
Dhaval joshi@DESKTOP-RHP0AVO MINGW64 ~
$ git --version
git version 2.37.1.windows.1

Dhaval joshi@DESKTOP-RHP0AVO MINGW64 ~
$ |
```

Fig 2.19

GitHub account:-



Dhaval-Joshi898/first-git
Dhaval-Joshi898/Proj2

Recent activity

When you take actions across GitHub, we'll provide links to that activity here.

Introduce yourself

The easiest way to introduce yourself on GitHub is by creating a README in a repository about you! You can start here:

```
Dhaval-Joshi898 / README.md

- Hi, I'm @Dhaval-Joshi898
- I'm interested in ...
- I'm currently learning ...
- I'm looking to collaborate on ...
- How to reach me ...
```

3 hours ago
Dependency graph adds package metadata for Rust dependencies

2 days ago
GitHub Actions The macOS 10.15 Actions runner image is being deprecated and will...

3 days ago
Differentiating triggering actor from executing actor

4 days ago
Code scanning enterprise-level REST API

View changelog →

Fig 2.20

Conclusions: In this experiment I understood the version control/source code management. I installed git and configured it into my system and created an account on a GitHub which acts as a holder for our code and repositories.