SE489 DevOps Engineering

Lab 1

Lab 1: Version Control Systems

Objective: After successfully completing this lab session, student will be able to understand, distinguish between Git and Github. This lab will train student how to download, customize, install and configure Git, lab further introduces students with basic git commands and operations.

a. Git

Definition1:

Git is software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows. – Wikipedia

Definition2:

Git is a DevOps tool used for source code management.

It's equally interesting to note that *Linus Torvald* himself created git in 2005 for maintaining the code of Linux Kernel.

As opposed to the common perception, git and github are two different entities, following tabulated figure shows major difference between the two

git	GitHub
1. It is a software	1. It is a service
2. It is installed locally on the system	2. It is hosted on Web
3. It is a command line tool	3. It provides a graphical interface
4. It is a tool to manage different versions of edits, made to files in a git repository	4. It is a space to upload a copy of the **Git** repository
5. It provides functionalities like Version Control System Source Code Management	5. It provides functionalities of Git like VCS, Source Code Management as well as adding few of its own features

But both of them work in tandem to achive code maintenance, as shown in the following figure-

Before you install git on your system, make sure

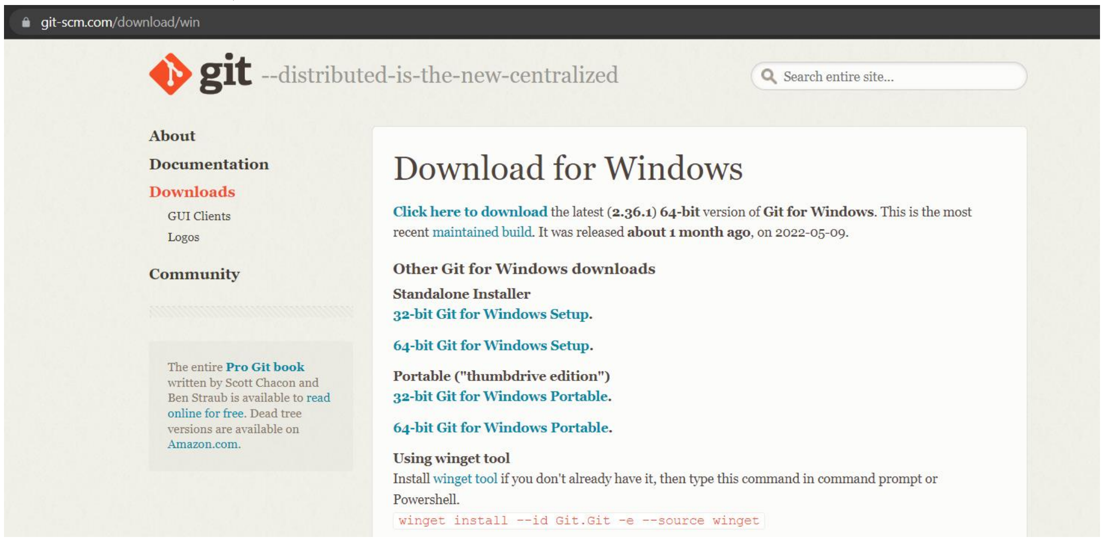
1. You have administrative privileges on the system, you want to install git.
2. There is enough space in the hard disk of the system

To install Git at your system, there are few steps which are to be followed

Steps shown below are for windows 11, but they will work on almost all versions of windows with minimal changes

Step1: Download git

Download, latest *standalone* version of the git from its website. (https://git-scm.com/download/win), at the time of this manual, it was 2.36.1



The entire Pro Git book written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

Download for Windows

Click here to download the latest (2.36.1) 64-bit version of Git for Windows. This is the most recent maintained build. It was released about 1 month ago, on 2022-05-09.

Other Git for Windows downloads

Standalone Installer
32-bit Git for Windows Setup.

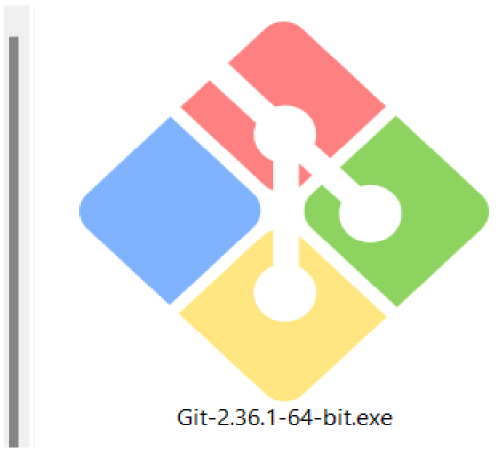
64-bit Git for Windows Setup.

Portable ("thumbdrive edition")
32-bit Git for Windows Portable.

64-bit Git for Windows Portable.

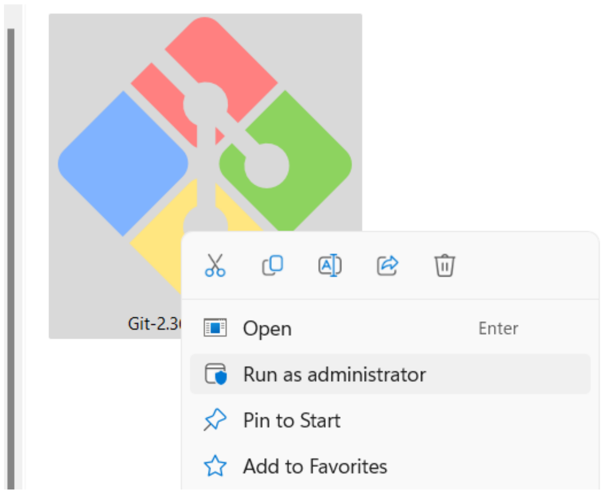
Using winget tool
Install winget tool if you don't already have it, then type this command in command prompt or Powershell.

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

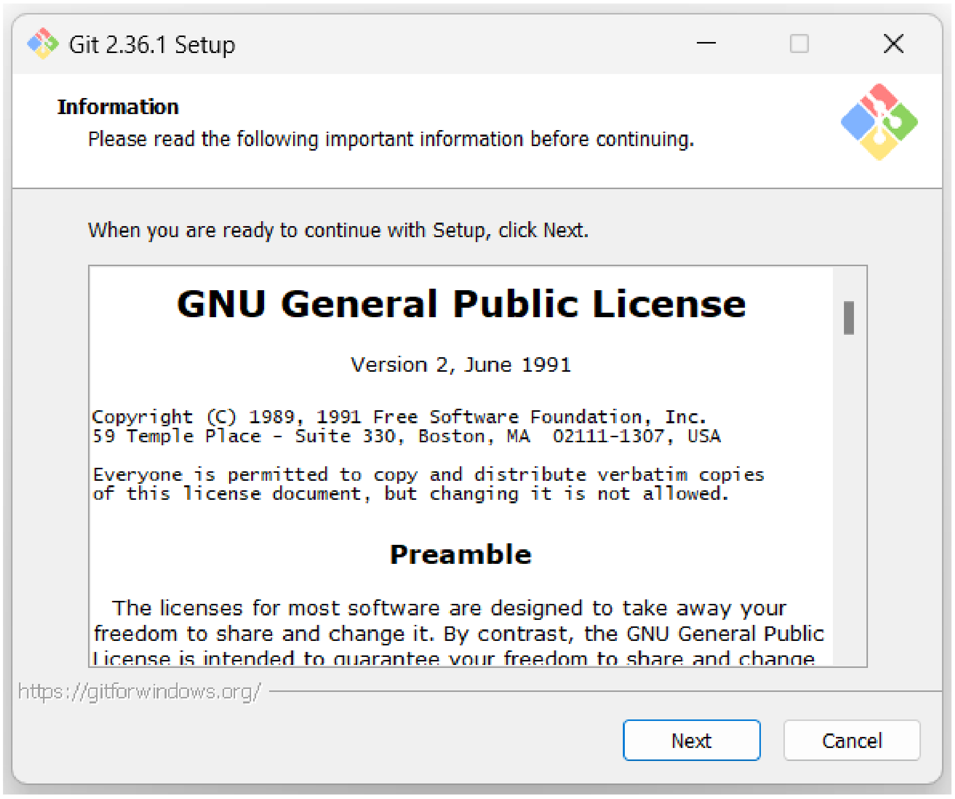


Step 2: Run Installer

Right click on the installer just downloaded and select, Run as Administrator



Click on next


Git 2.36.1 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

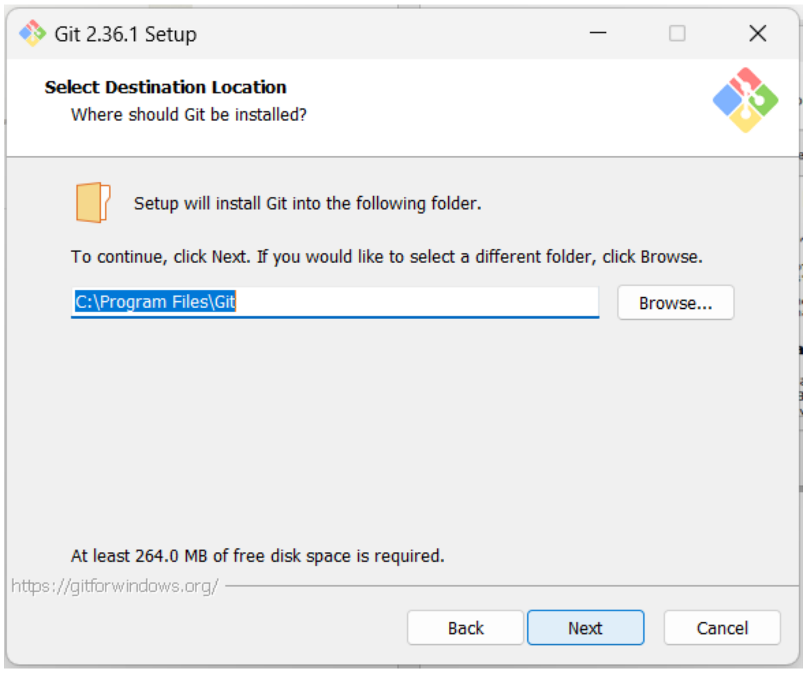
```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA  02111-1307, USA

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

Approve or change the default location


Git 2.36.1 Setup

Select Destination Location

Where should Git be installed?

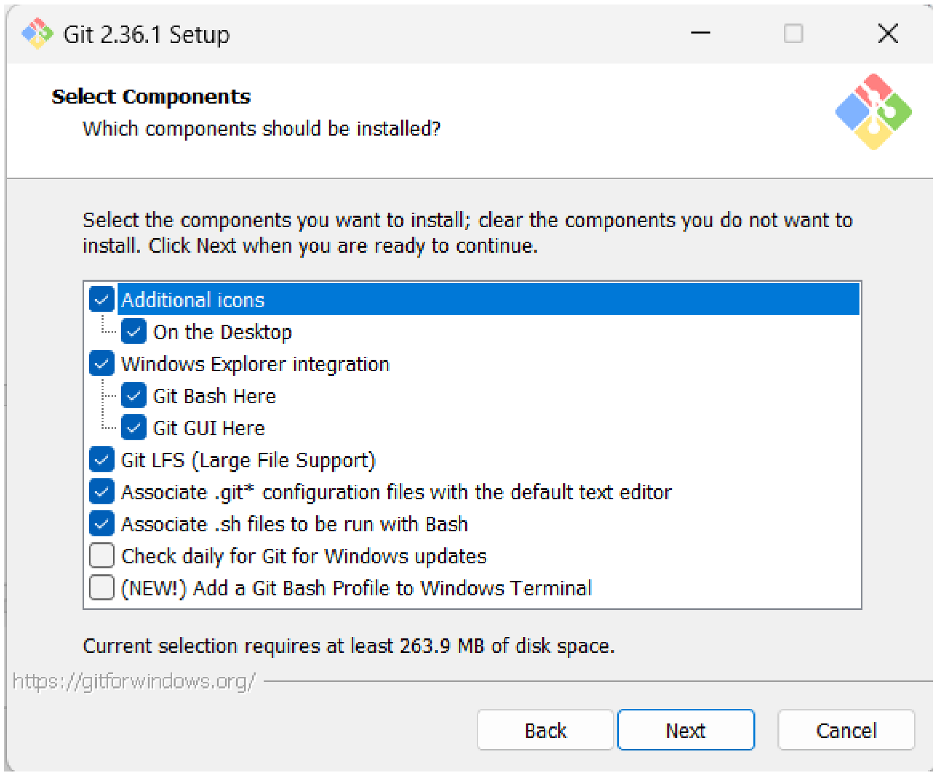
Setup will install Git into the following folder.

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

C:\Program Files\Git

At least 264.0 MB of free disk space is required.

Click on next to accept the default selections or change some additional one, as I have selected additional icons


Git 2.36.1 Setup

Select Components

Which components should be installed?

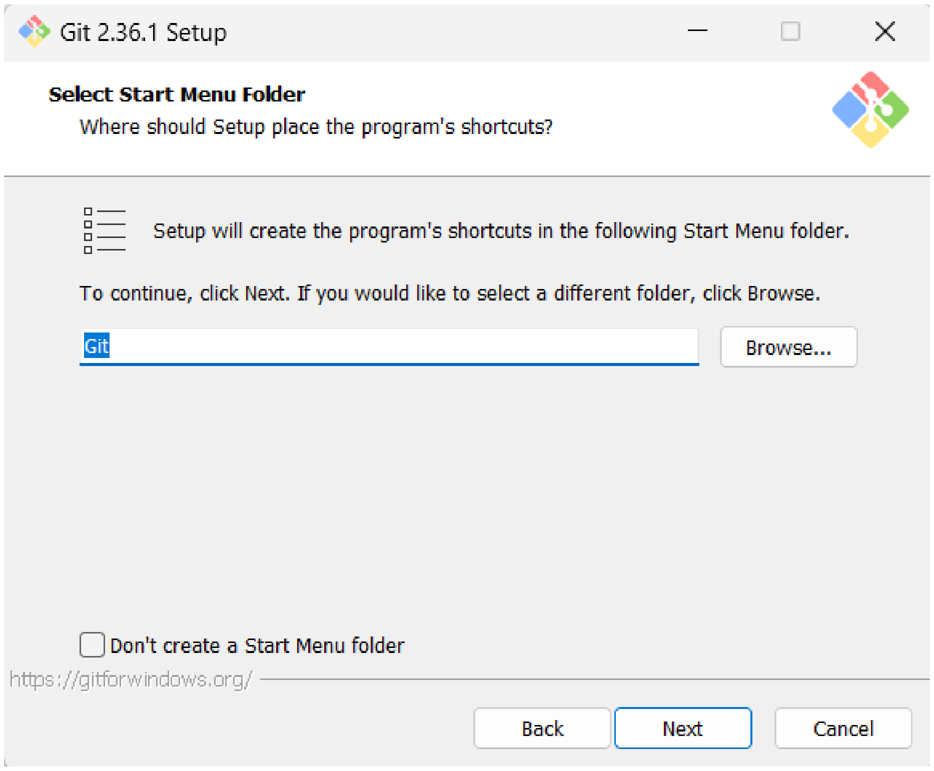
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- [✓] 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 263.9 MB of disk space.

https://gitforwindows.org/

Select start menu folder


Git 2.36.1 Setup

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.

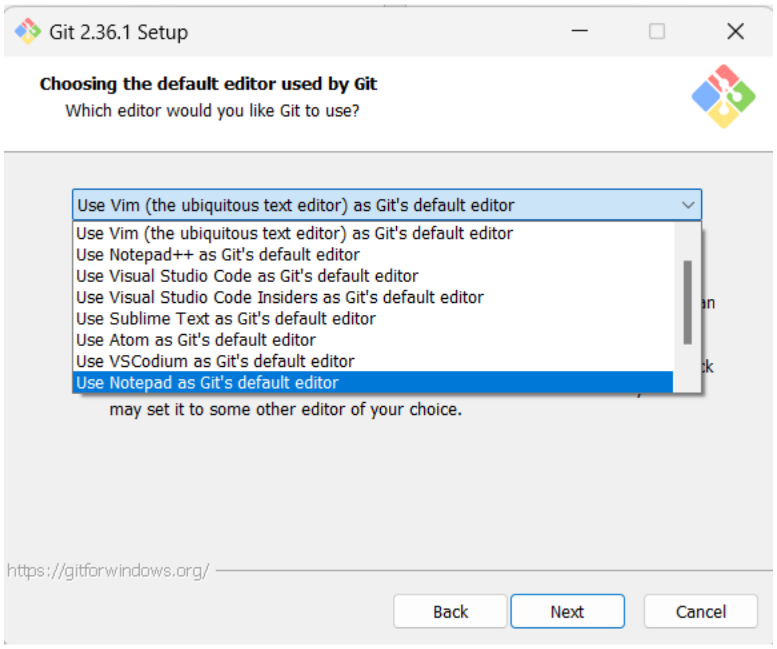
Git

- [ ] Don't create a Start Menu folder

https://gitforwindows.org/

Step 3: Select default text editor

Text editors are programs used for editing text files, by default git shows Vim as default text editor, but it is for linux systems, select Notepad or Notepad++ as per your choice (Notepad++ needs to be installed separately whereas Notepad is supplied with windows)



Git 2.36.1 Setup

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

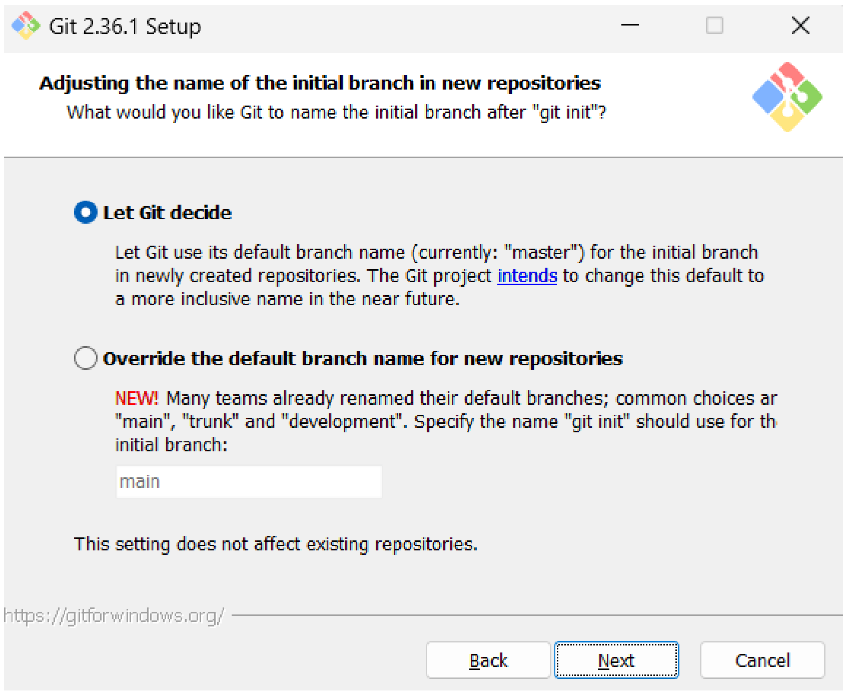
- Use Vim (the ubiquitous text editor) as Git's default editor
- Use Notepad++ as Git's default editor
- Use Visual Studio Code as Git's default editor
- Use Visual Studio Code Insiders as Git's default editor
- Use Sublime Text as Git's default editor
- Use Atom as Git's default editor
- Use VSCodium as Git's default editor
- Use Notepad as Git's default editor

may set it to some other editor of your choice.

https://gitforwindows.org/

Step 4: Deciding name of the Initial branch in the repository

By default, it is “master” leave it as it is unless you want to change it



Git 2.36.1 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

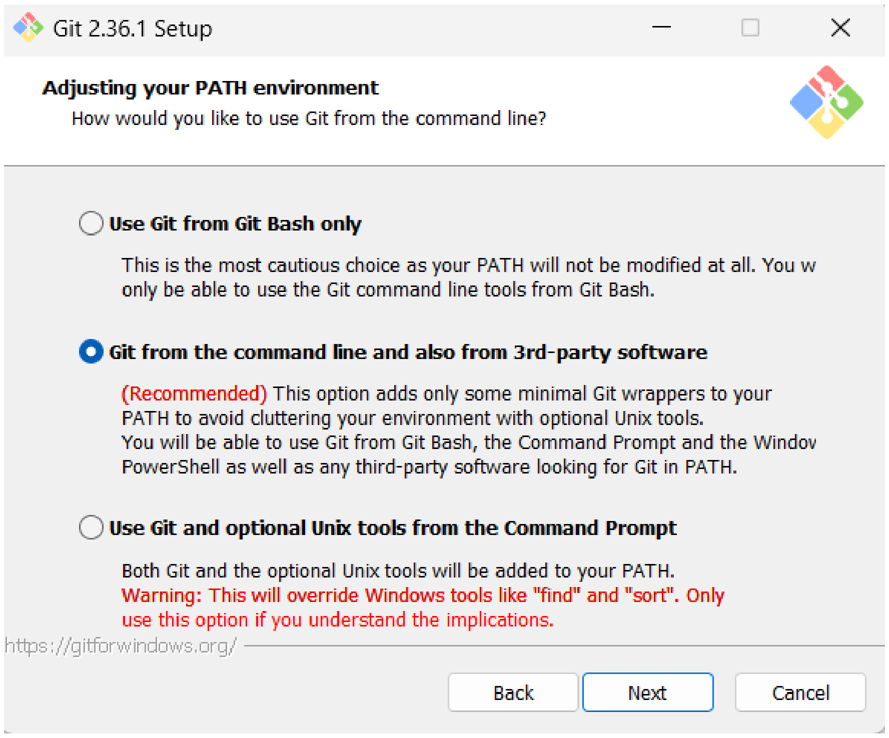
NEW! Many teams already renamed their default branches; common choices ar "main", "trunk" and "development". Specify the name "git init" should use for th initial branch:

main

This setting does not affect existing repositories.

https://gitforwindows.org/

__Step 5: Adjusting PATH environments__



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.

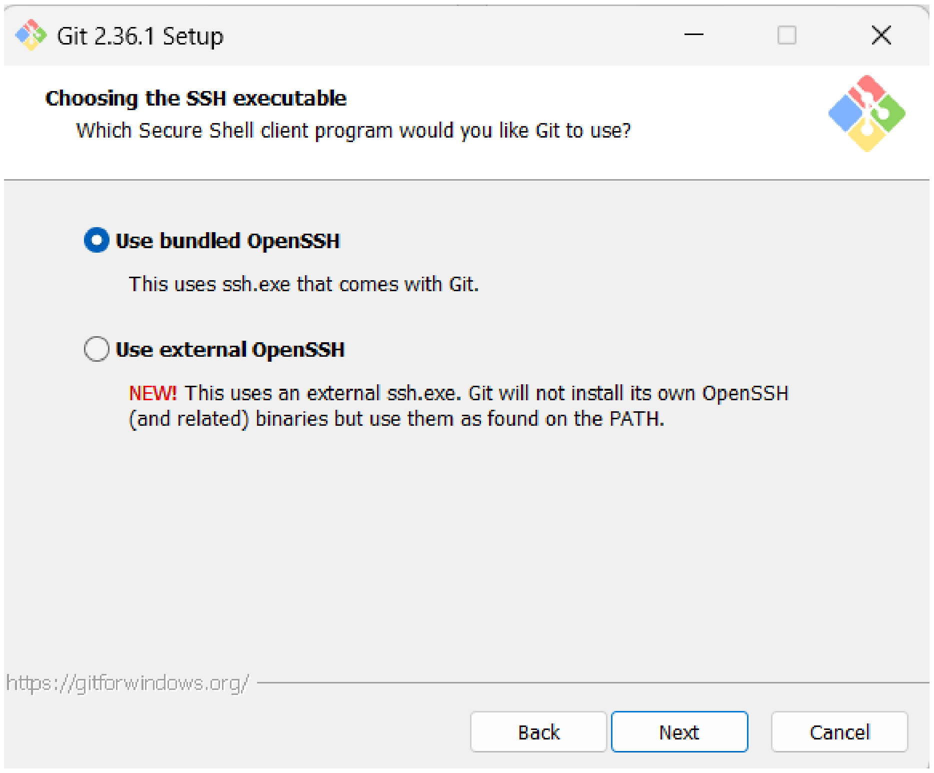
● **Git from the command line and also from 3rd-party software**

(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.

__Step 6: Selecting SSH executable for the git and repository__



Choosing the SSH executable

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

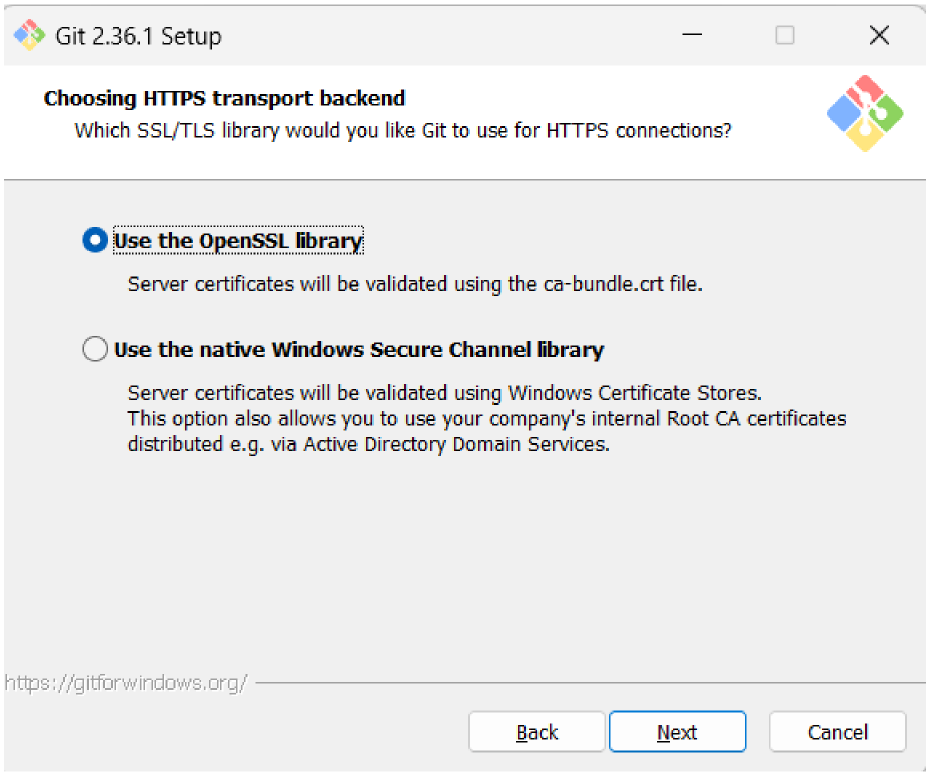
● **Use bundled OpenSSH**

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.

Step 7: Selecting HTTPS transport backend


Git 2.36.1 Setup

Choosing HTTPS transport backend

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

[✓ **Use the OpenSSL library**]

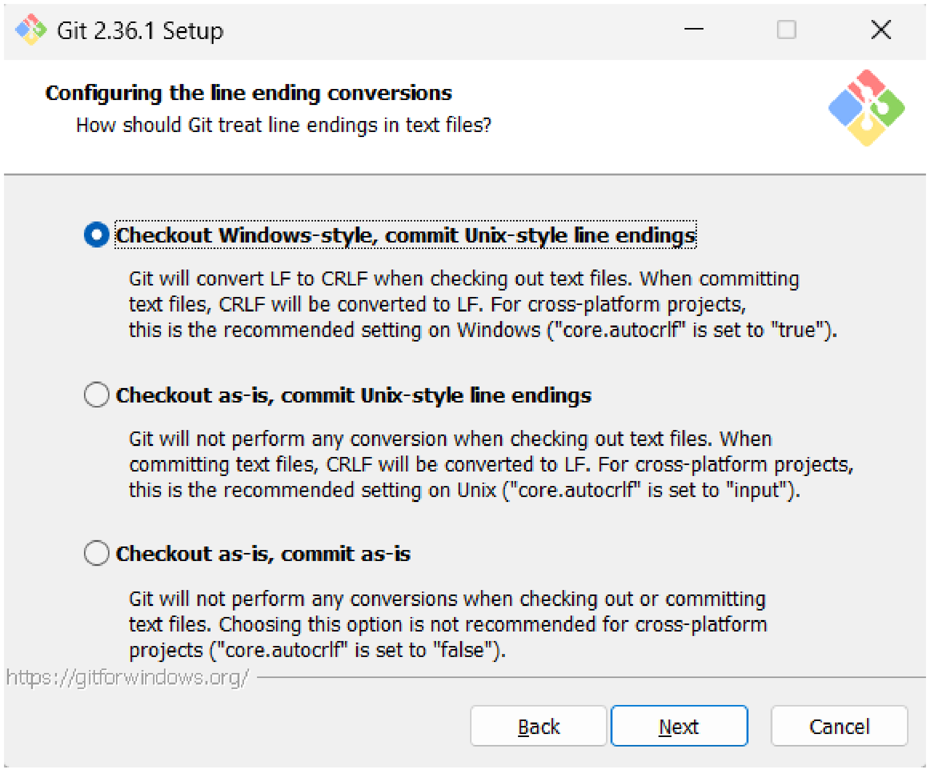
Server certificates will be validated using the ca-bundle.crt file.

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/

Step 8: Configuring the line ending conversions


Git 2.36.1 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

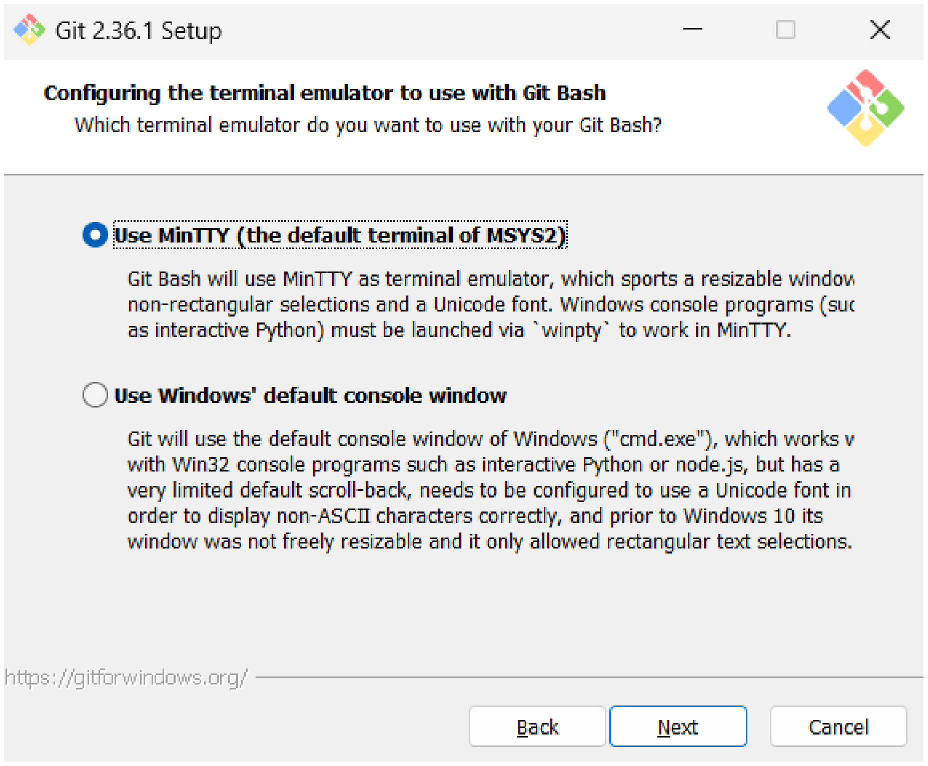
Git will not perform any conversion when checking out text files. When
committing text files, CRLF will be converted to LF. For cross-platform projects,
this is the recommended setting on Unix ("core.autocrlf" is set to "input").

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/

Step 9: Configuring the terminal emulator



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

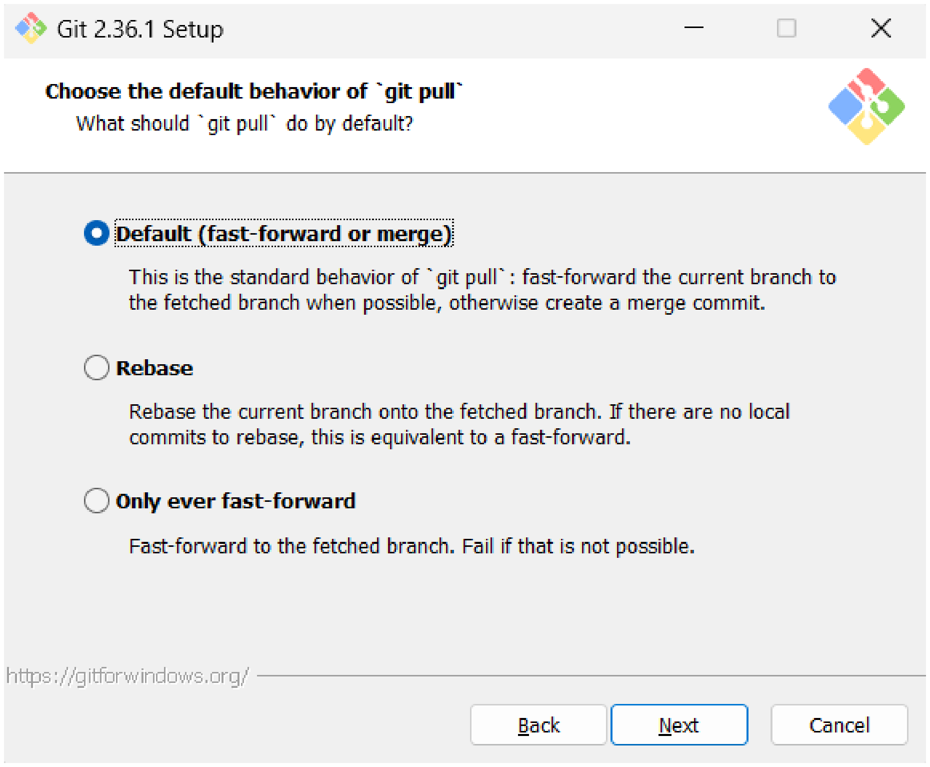
[✓ **Use MinTTY (the default terminal of MSYS2)**]

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.

Step 10: Chose the default behavior of 'git'



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.

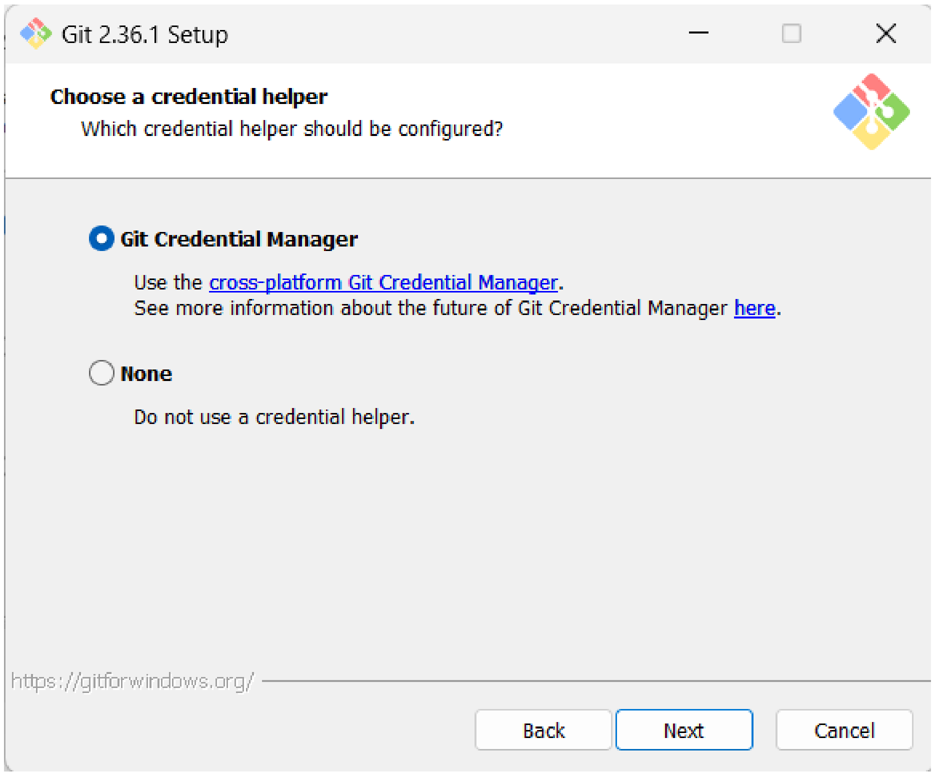
Rebase

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.

Step 11: Choose credential manger



Git 2.36.1 Setup

Choose a credential helper

Which credential helper should be configured?

[● Git Credential Manager]

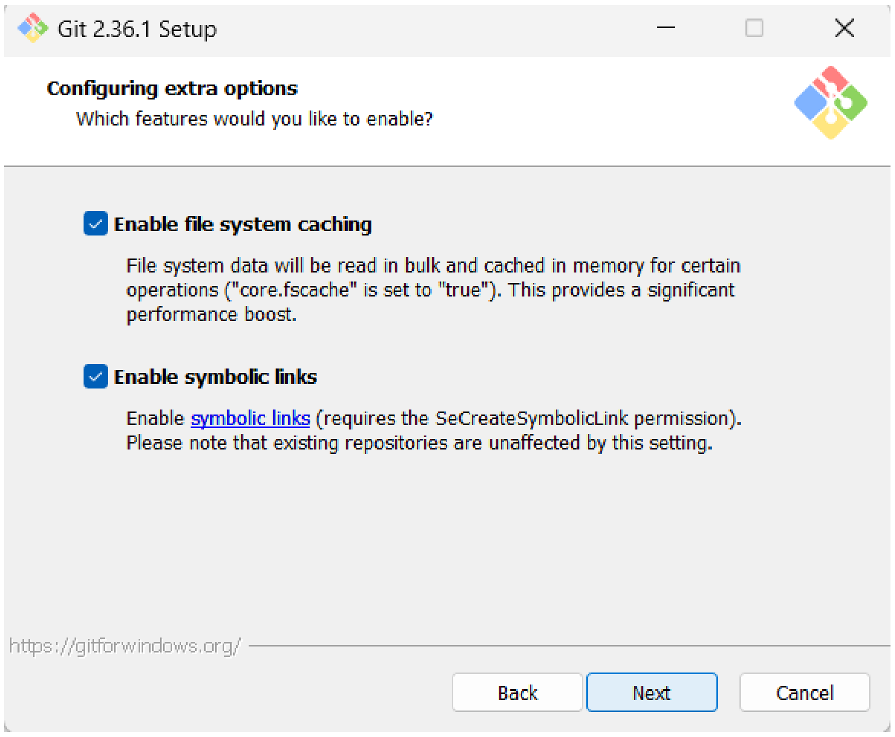
Use the cross-platform Git Credential Manager.
See more information about the future of Git Credential Manager here.

○ None

Do not use a credential helper.

https://gitforwindows.org/

Step 12: Configuring extra options



Git 2.36.1 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.

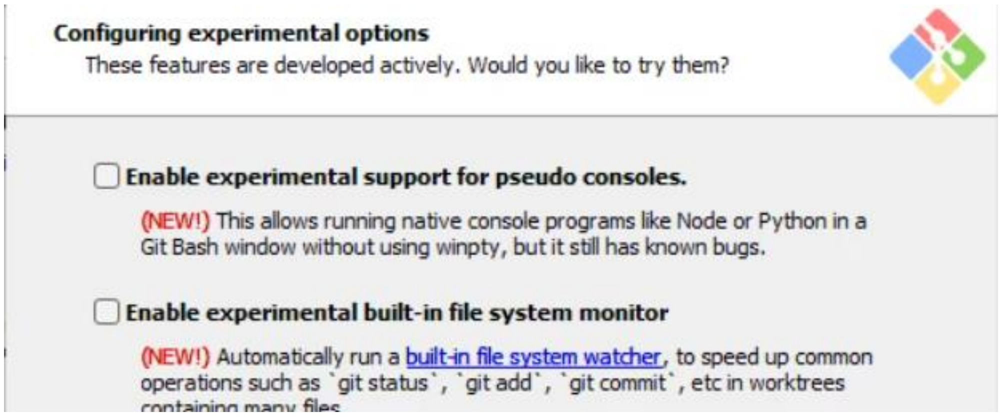
[✓] **Enable symbolic links**

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

https://gitforwindows.org/

Step 13: Configuring experimental options

Leave them unchecked, unless you want to use them



Configuring experimental options

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

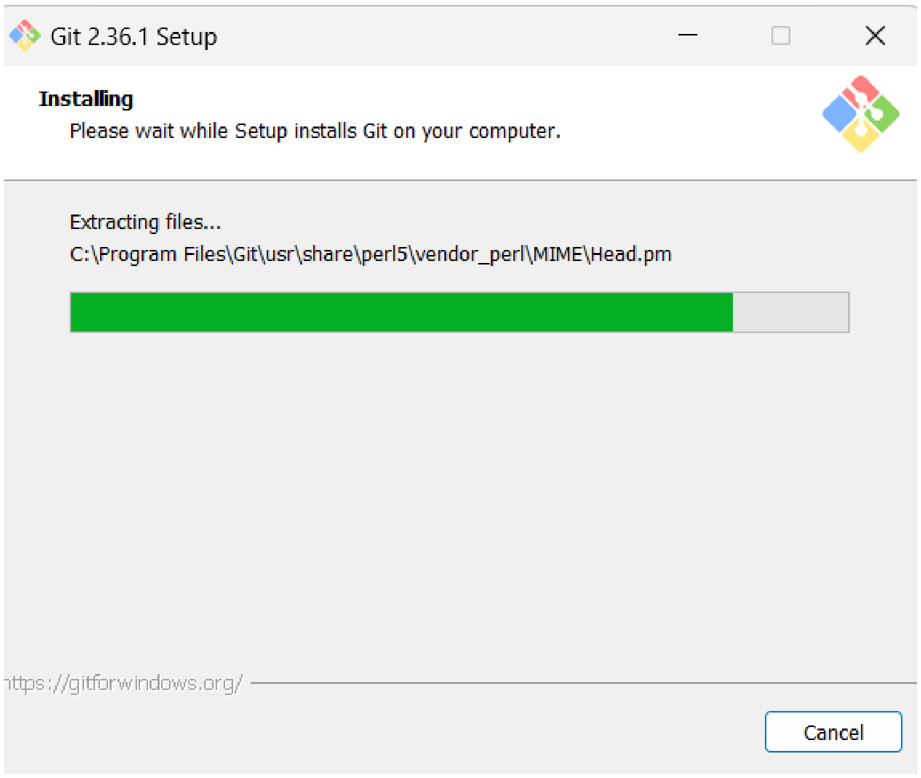
☐ **Enable experimental support for pseudo consoles.**

(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

Step 14: Click on Install, and let the installation be complete



Git 2.36.1 Setup

Installing

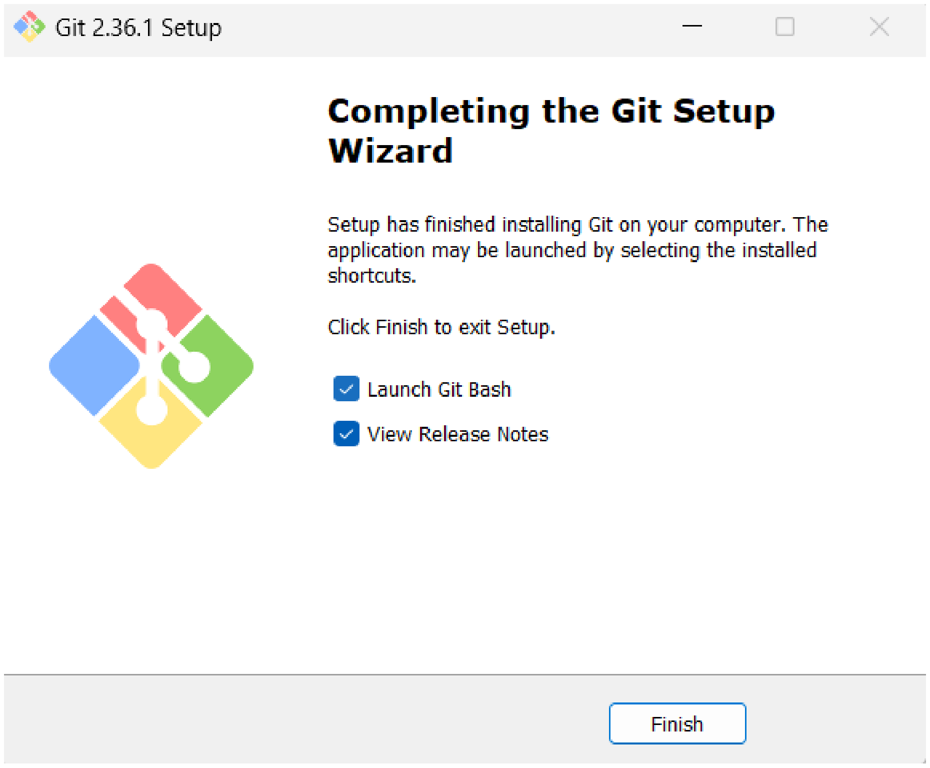
Please wait while Setup installs Git on your computer.

Extracting files...

C:\Program Files\Git\usr\share\perl5\vendor_perl\MIME\Head.pm

https://gitforwindows.org/

Step 15: Click finish to complete the setup


Git 2.36.1 Setup

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.

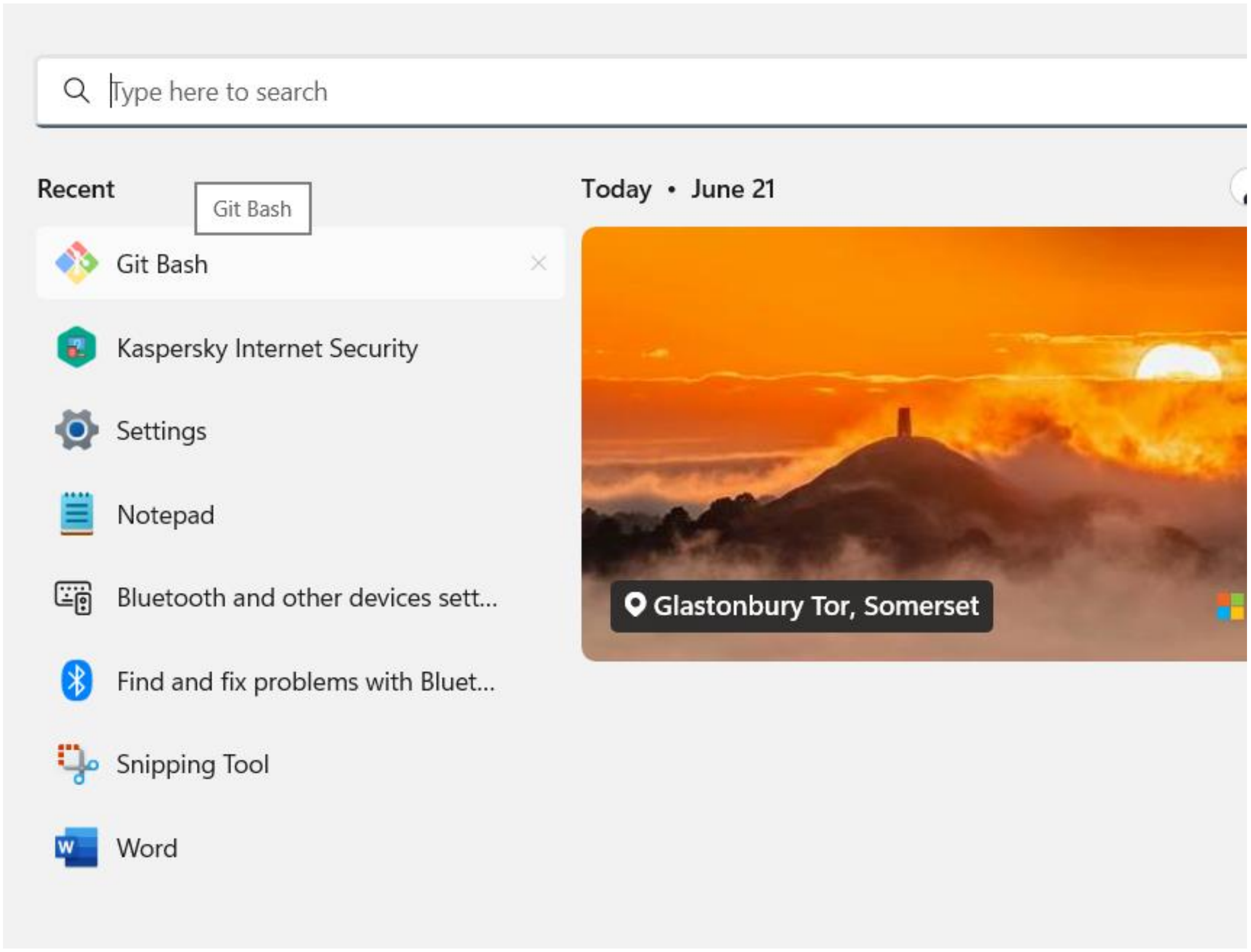
[✓] Launch Git Bash

[✓] View Release Notes

This completes the installation of the Git. Git can be started in command line interface as well as in GUI mode, we will explain here, both.

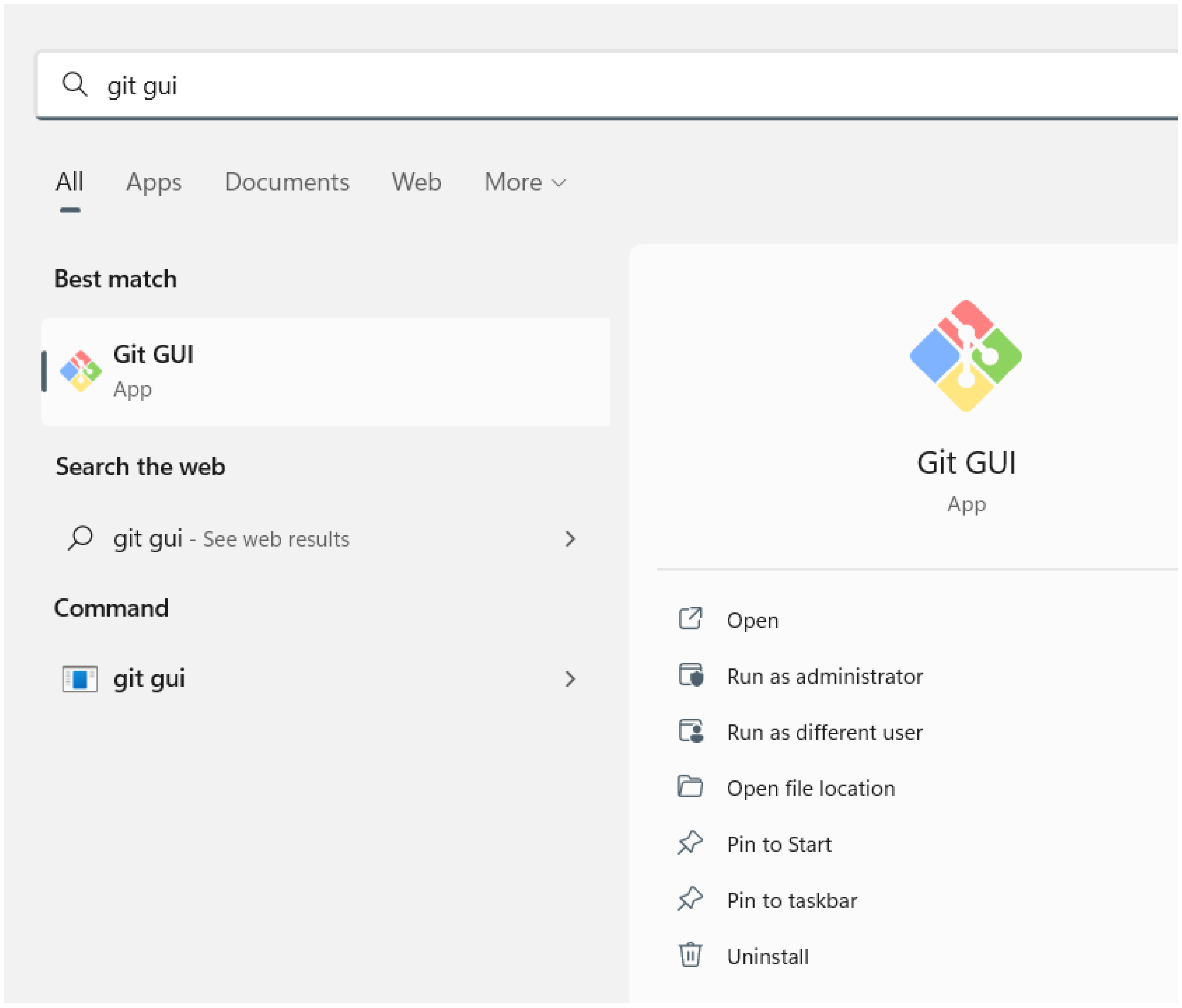
Launching Git in Command Line Interface

Click on Windows start menu and type git bash and press enter, alternatively, you may also click on the icon appearing.


Type here to search
Recent
Git Bash
Git Bash
Kaspersky Internet Security
Settings
Notepad
Bluetooth and other devices sett...
Find and fix problems with Bluet...
Snipping Tool
Word
Today • June 21
Glastonbury Tor, Somerset

Launching Git in Graphical User Interface

To launch git in GUI, write at Windows Start menu, git gui and press *Enter*, or alternatively, click at git gui icon if shown.



Working with the Local Repositories:

1. Setting a Local Repository

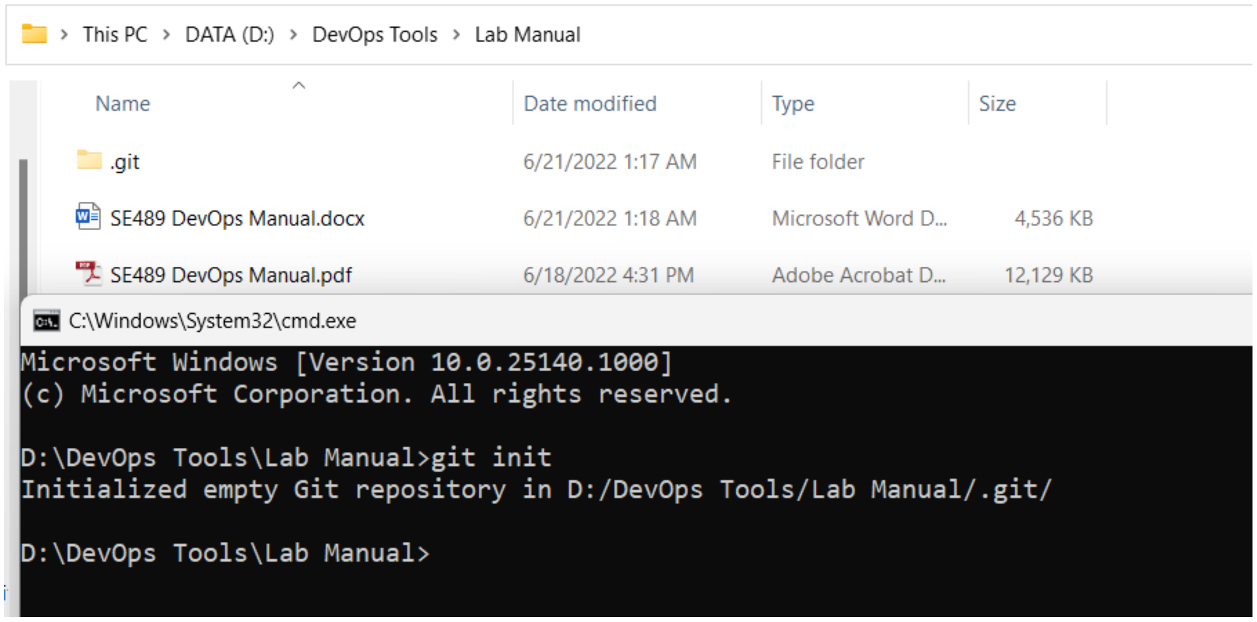
   Local Repositories → Local working directory/folder

   when git bash is called upon from start menu, it starts at **c:\Users\<username>, however if you want to create repository at some other location, there is a work around.

   Navigate to the folder you want to make local repository with windows explorer, and once at the required location, at the Address bar, write cmd and press Enter, command window will open at the same location. In our case it is *d:\DevApp Tools\Lab Manual*.

   When Command Window appears, write **git init** and press *Enter*

   Git make current folder, local repository, you will see a folder with **.git** name appears in the current folder, this means that you have successfully made current folder as local repository in git.


This PC > DATA (D:) > DevOps Tools > Lab Manual

Name	Date modified	Type	Size
.git	6/21/2022 1:17 AM	File folder	
SE489 DevOps Manual.docx	6/21/2022 1:18 AM	Microsoft Word D...	4,536 KB
SE489 DevOps Manual.pdf	6/18/2022 4:31 PM	Adobe Acrobat D...	12,129 KB

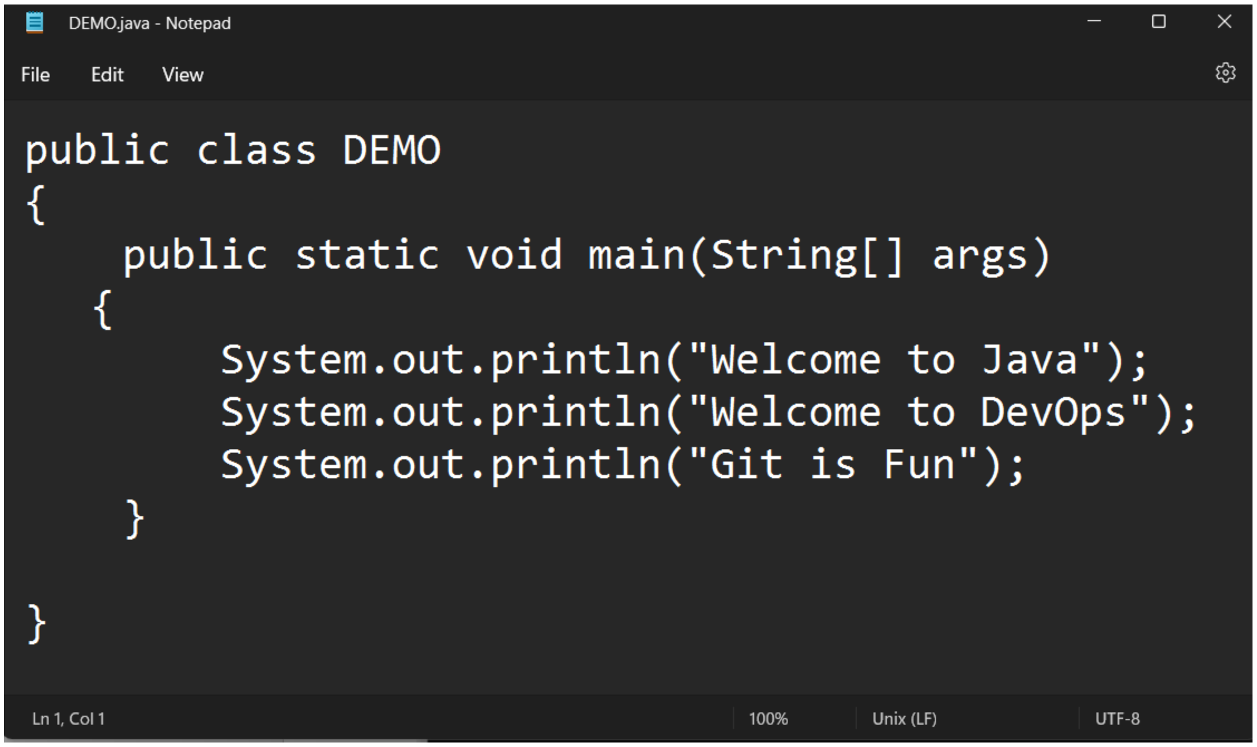
C:\Windows\System32\cmd.exe

```
Microsoft Windows [Version 10.0.25140.1000]
(c) Microsoft Corporation. All rights reserved.

D:\DevOps Tools\Lab Manual>git init
Initialized empty Git repository in D:/DevOps Tools/Lab Manual/.git/

D:\DevOps Tools\Lab Manual>
```

Now create a sample java program file with three print statements and save it.



```java
public class DEMO
{
     public static void main(String[] args)
    {
           System.out.println("Welcome to Java");
           System.out.println("Welcome to DevOps");
           System.out.println("Git is Fun");
     }


}
```

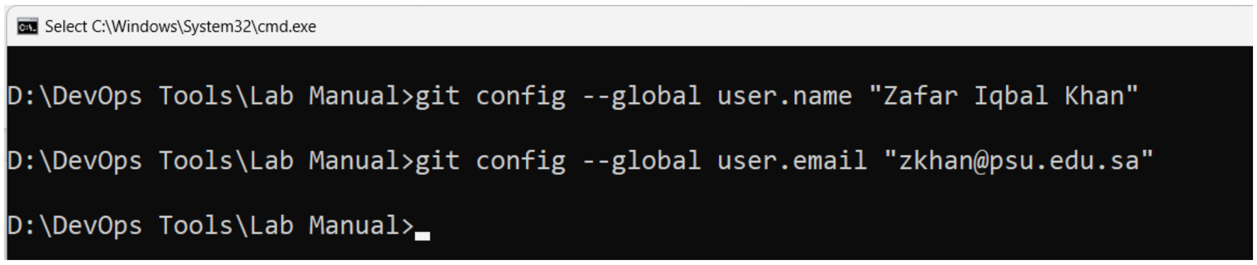
2. Configuring git with username and user email
   Before files and edits can be made and maintained, we need to configure git with username and user email id, so as git can track records precisely, what changes have been made by whom.

Although, individual repositories can be configured separately, it is wise to make configuration global if single user is going to use them all.

git configure --global user.name <username>

and

git configure -- global user.email <user email>



Select C:\Windows\System32\cmd.exe

```
D:\DevOps Tools\Lab Manual>git config --global user.name "Zafar Iqbal Khan"

D:\DevOps Tools\Lab Manual>git config --global user.email "zkhan@psu.edu.sa"

D:\DevOps Tools\Lab Manual>_
```

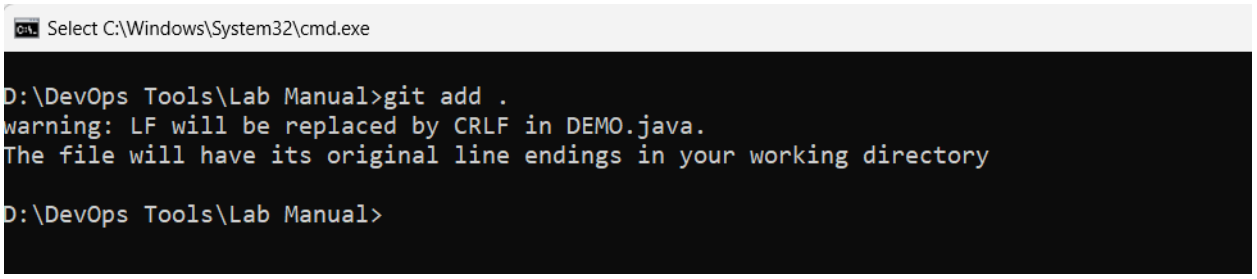
3. Adding files to the staging area

   To control the versioning of the file, git has provisions of ***Local repository →Staging Area→Global Repository***

   Now let's push this file (DEMO.java) to the **Staging area**.

   At command window write **git add .**

   Some messages like below will appear



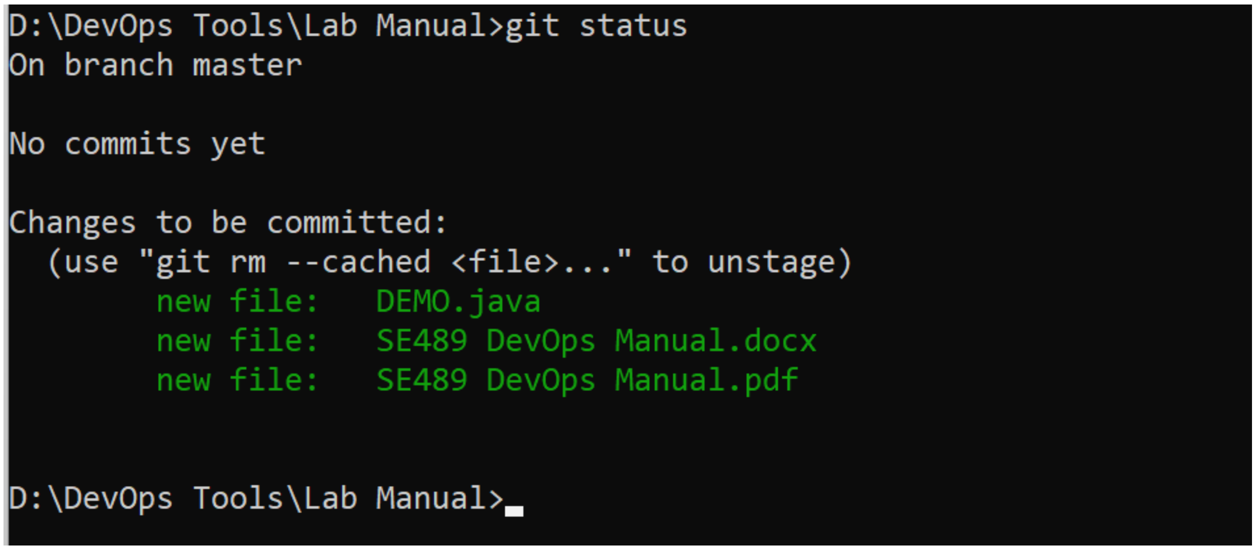
Select C:\Windows\System32\cmd.exe

```
D:\DevOps Tools\Lab Manual>git add .
warning: LF will be replaced by CRLF in DEMO.java.
The file will have its original line endings in your working directory

D:\DevOps Tools\Lab Manual>
```

4. Checking status of the repository (git status)

   To check the status of the repository, **git status** is the command



```
D:\DevOps Tools\Lab Manual>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   DEMO.java
        new file:   SE489 DevOps Manual.docx
        new file:   SE489 DevOps Manual.pdf


D:\DevOps Tools\Lab Manual>
```

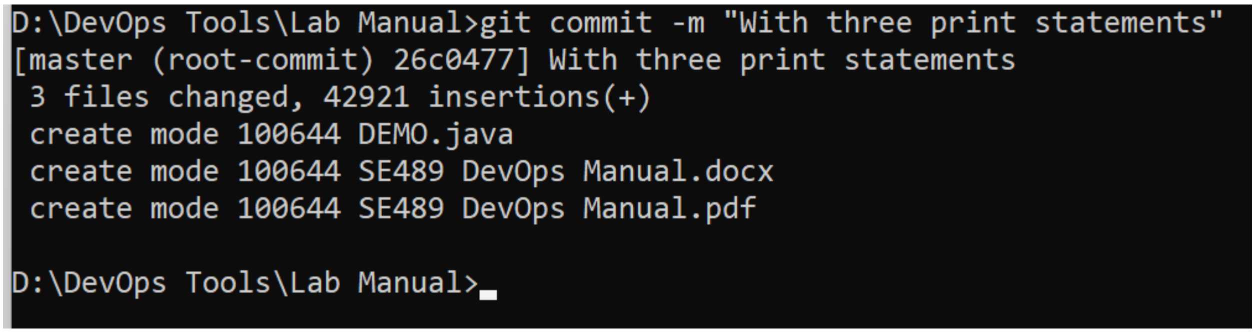
Obviously, from the output no commits have been performed, and all files are in the staging area.

5. Performing Commits (git commits)

   To make changes permanent, we commit these files, commands used for this is

   git commit -m "<marker message>"

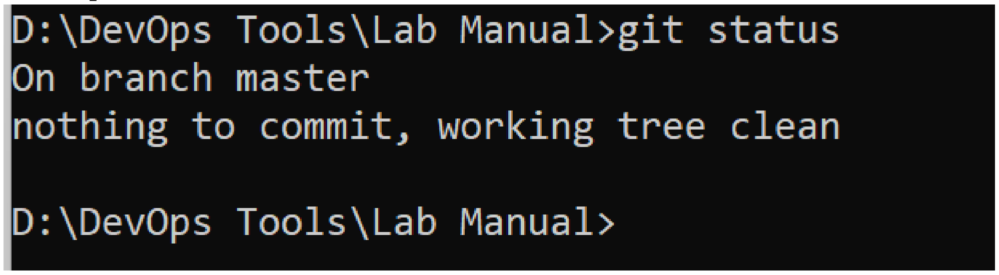
   marker message → these are the messages we write to easily identify different versions of the files, if we don't provide messages here, git will ask for one later with popups, a commit without message will be aborted.



```
D:\DevOps Tools\Lab Manual>git commit -m "With three print statements"
[master (root-commit) 26c0477] With three print statements
 3 files changed, 42921 insertions(+)
 create mode 100644 DEMO.java
 create mode 100644 SE489 DevOps Manual.docx
 create mode 100644 SE489 DevOps Manual.pdf

D:\DevOps Tools\Lab Manual>
```

6. Now check status again (git status)
   It will show following output, master is the default name of the main branch of the project development



```
D:\DevOps Tools\Lab Manual>git status
On branch master
nothing to commit, working tree clean

D:\DevOps Tools\Lab Manual>
```