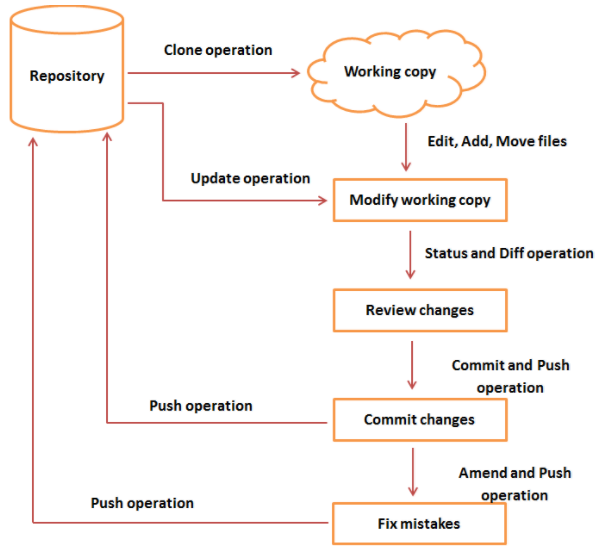
**GIT (Version Control)**

Git is a software that allows you to keep track of changes made to a project over time to time.

Git works by recording the changes you make to your project, storing those changes, then allowing you to reference them as needed.

Git is decentralized which means that it doesn’t depend on a central server to keep old version of your files instead it works fully locally by storing the data as folder on you hard-drive which we call a repository and we can store it online which makes it easy for multiple people to work on it. This is what websites like Github and Bitbucket are used for.

**Git Life Cycle**

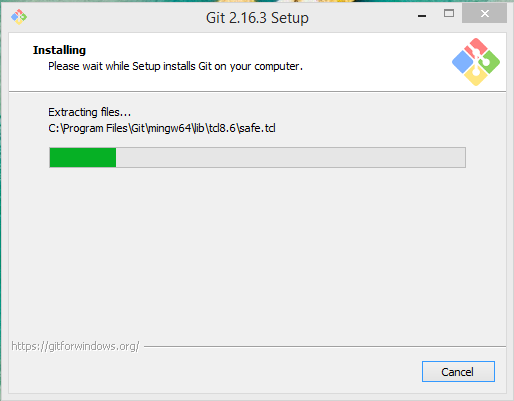
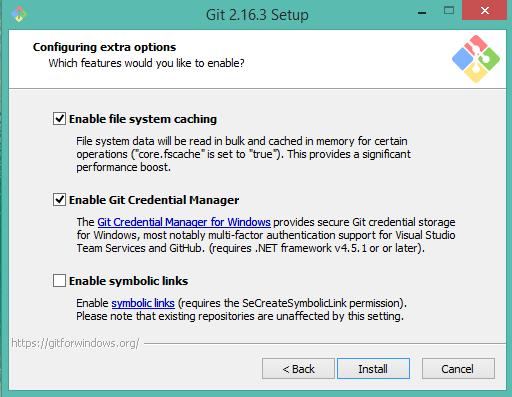
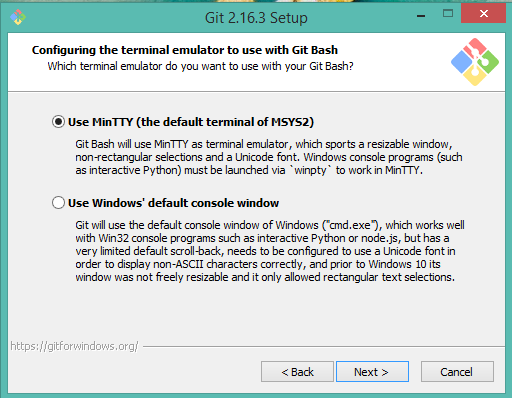
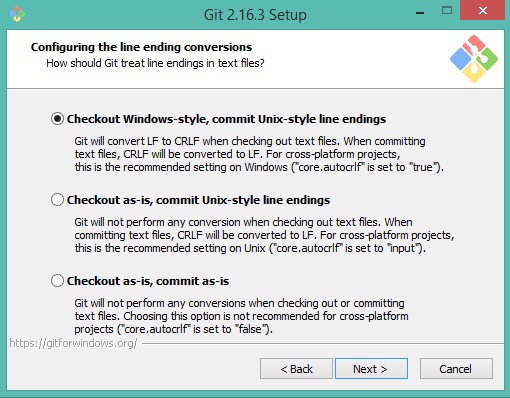
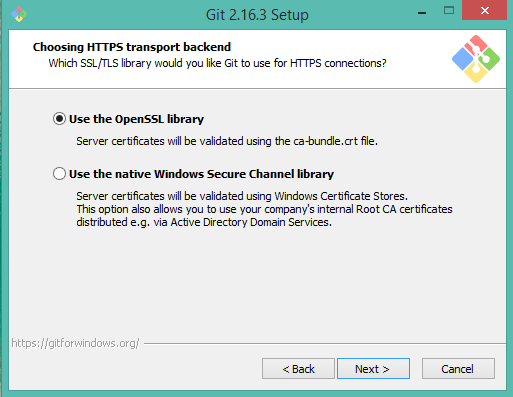
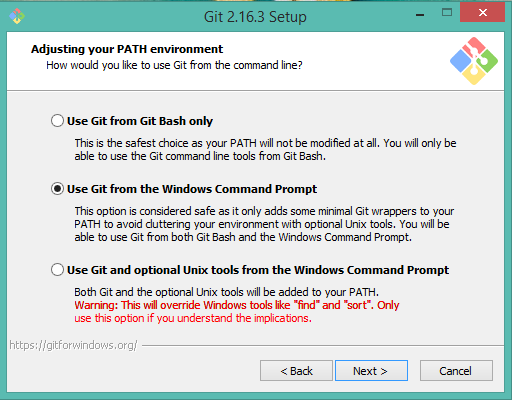
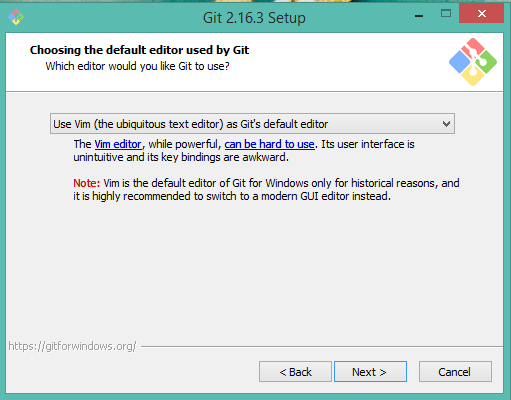
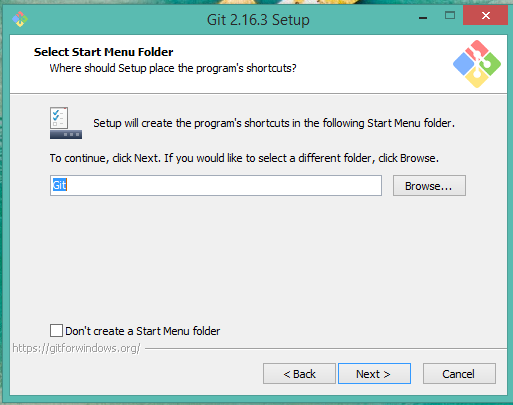
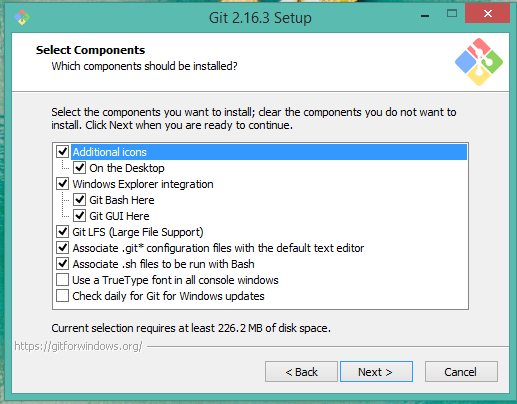
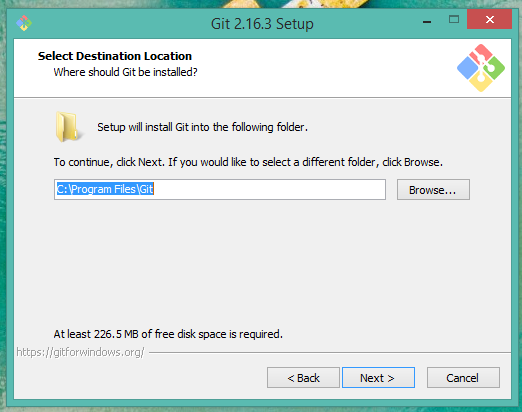
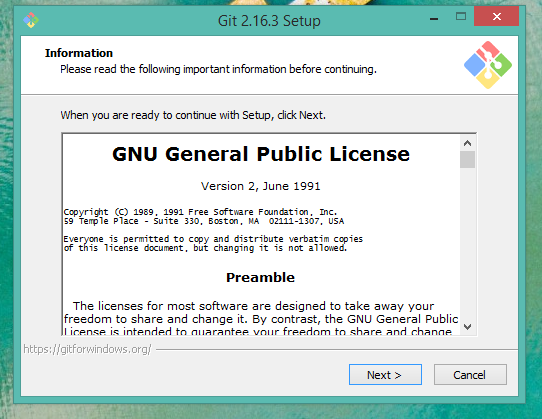
****

Installing Git

Go to this URL and download: <https://gitforwindows.org/>

It offers both a GUI client and a BASH command line emulator needed for Git.

Steps shown below:



**Step 1. Configuring Git:**

Now that we've installed git on our computer, we will need to add some quick configurations. There are a lot of options that can be fiddled with, but we are going to set up the most important ones: our username and email. Open a terminal and run these commands:

$ git config --global user.name "My Name"

$ git config --global user.email myEmail@example.com

Every action we do in Git will now have a stamp with our name and address on it. This way user always knows who did what and everything is way more organized.

**Step 2. Creating a new Repository:**

To set up a new repository, Open Git terminal, navigate to our project directory and run git init (initializes a new repository) command.

It enables git for this particular folder and creates a hidden .git directory where the repository history and configuration will be stored.

The command line responds with something along the lines of:

Initialized empty Git repository in /home/user/Desktop/git\_exercise/.git/

This means that our repository has been successfully created but is still empty.

**Step 3. Checking the status:**

To know about the current state of the repository/project: is everything up to date, what's new, what's changed, and so on.

Running git status in our newly created repository should return the following:

$ git status

On branch master

Initial commit

Untracked files:

(use "git add ..." to include in what will be committed)

hello.txt

The returned message states that *hello.txt* is untracked. This means that the file is new and Git doesn’t know yet if it should keep track of the changes happening to that file or just ignores it. To acknowledge the new file, we need to stage it.

**Working Directory:** where we will be doing our all the work creating, editing, deleting and organizing files etc.

**Staging Area:** where you list changes you make to working directory.

**Repository:** A Repository: where Git permanently stores those changes as different versions of the project

Git Commands:

1. Git log

From the terminal, log a list of your commits.

In the output, notice:

* A 40-character code, called a *SHA*, which uniquely identifies the commit. This appears in orange text.
* The commit author (you!)
* The date and time of the commit
* The commit message

1. Git log

From the terminal, log a list of your commits.

In the output, notice:

* A 40-character code, called a *SHA*, which uniquely identifies the commit. This appears in orange text.
* The commit author (you!)
* The date and time of the commit
* The commit message