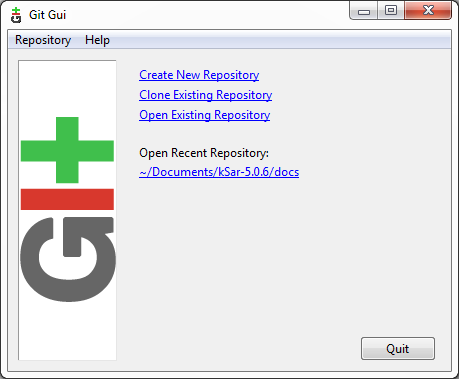
**Git Installation**

Download the Binaries and install GIT Setup.

* Go to C:\Program Files\Git\cmd location and invoke e binary “git-gui.exe”

1. **Create New Repository:**

You will create a new repository only when you have the original source code on your local machine, and you like to track the version locally. If the source code is located in a remote GIT repository, you should use “Clone Existing Repository” option that is explained in the next section.



To create the git repo, select the option “Create New Repository” and locate the git Repo path and click on **create** button.

This will create the repo in this directory and a hidden directory name “. git” is created inside this directory which contains various config directory.

Same can be created from CLI also. Create a directory for git repo and switch to that repo and fire the “git init” command. This will again create an invisible “.git“ directory and contains all the necessary configurations .

Once the Git Repository is created, Run the following command to create the identity of this git repo and this credentials must be same as that of github credentials so that we local repository may transfer file to remote repository.

**git config --global user.name "vishen31"**

**git config --global user.Email neeraj.vishen@gmail.com**

and to check the setting of Git Repo

**git config --list**

**git config --system --list**

**git config --global –list**

**Create Repository from remote (i.e, Central Repository)**

neeraj@Neeraj-Laptop MINGW64 /d/target-repo (master)

$ **git clone https://github.com/vishen31/Git-Test**

Cloning into 'Git-Test'...

remote: Counting objects: 13, done.

remote: Total 13 (delta 0), reused 0 (delta 0), pack-reused 13

Unpacking objects: 100% (13/13), done.

**Git add and Git commit**

$ echo "test content for git tutorial" >> CommitTest.txt

neeraj@Neeraj-Laptop MINGW64 /d/target-repo (master)

$ dir

CommitTest.txt Git-Test test1.txt

neeraj@Neeraj-Laptop MINGW64 /d/target-repo (master)

$ git add CommitTest.txt

warning: LF will be replaced by CRLF in CommitTest.txt.

The file will have its original line endings in your working directory.

neeraj@Neeraj-Laptop MINGW64 /d/target-repo (master)

$ git commit -m "added CommitTest.txt to the repo"

[master (root-commit) 2a8627e] added CommitTest.txt to the repo

2 files changed, 2 insertions(+)

create mode 100644 CommitTest.txt

create mode 100644 test1.txt

**Now this changes to remote directory when this repo is cloned of remote Repo (central repo)**

$ **git remote add origin https://github.com/vishen31/Git-Test**

fatal: remote origin already exists.

🡪 Above command will create a new connection to a remote repository but it seems connection is already created. This is created during the cloning only we got the above message. To check the connection, run the below command.

$ git remote

🡪 Origin -> When you clone a repository with git clone, it automatically creates a remote connection called origin pointing back to the cloned repository. This is useful for developers creating a local copy of a central repository, since it provides an easy way to pull upstream changes or publish local commits. This behavior is also why most Git-based projects call their central repository origin.

$ git push origin master

Counting objects: 3, done.

Delta compression using up to 4 threads.

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 435 bytes | 435.00 KiB/s, done.

Total 3 (delta 0), reused 0 (delta 0)

To https://github.com/vishen31/Git-Test

5973071..9c1dd64 master -> master

🡪 Pushing is how you transfer commits from your local repository to a remote repo and same can be easily performed by GUI .

In case of New Manual Repository creation at remote side, we need to manually create the repository in github and then clone the same at local repo . We can also create the repo at remote location by using cli, but this is lengthy process. In future we will create repo in github manually and clone the same at local machine.

* We can the multiple git local repo inside a git repo.
* A remote connection called origin is same for all the connection as there is one to one relation between local repo and remote repo .

Eg .

neeraj@Neeraj-Laptop MINGW64 /d/target-repo/**HelloWorld** (master)

$ git remote -v

origin https://github.com/vishen31/HelloWorld (fetch)

origin https://github.com/vishen31/HelloWorld (push)

$ cd ../Git-Test/

neeraj@Neeraj-Laptop MINGW64 /d/target-repo/**Git-Test** (master)

$ git remote -v

origin https://github.com/vishen31/Git-Test (fetch)

origin https://github.com/vishen31/Git-Test (push)

$ git push origin master

Counting objects: 3, done.

Delta compression using up to 4 threads.

Compressing objects: 100% (3/3), done.

Writing objects: 100% (3/3), 392 bytes | 392.00 KiB/s, done.

Total 3 (delta 2), reused 0 (delta 0)

remote: Resolving deltas: 100% (2/2), completed with 2 local objects.

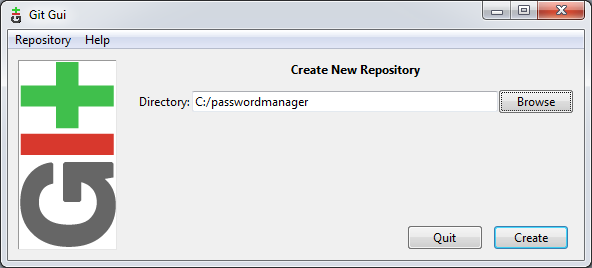
To https://github.com/vishen31/HelloWorld

cc2b19a..6798a25 master -> master

neeraj@Neeraj-Laptop MINGW64 /d/target-repo/HelloWorld (master)

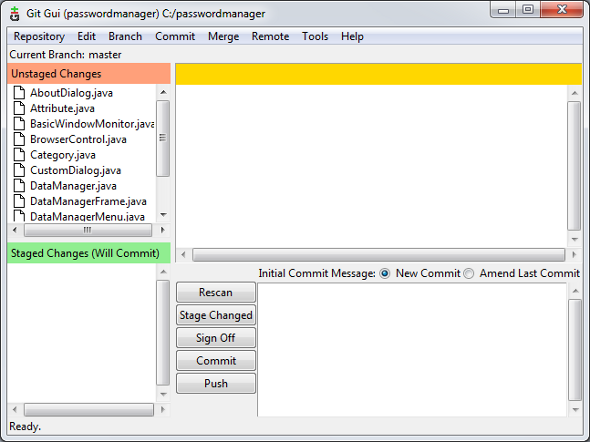
**2) Select the code Directory:**

Select the directory where the source code is located. In this example, the source code is located in “c:\passwordmanager”



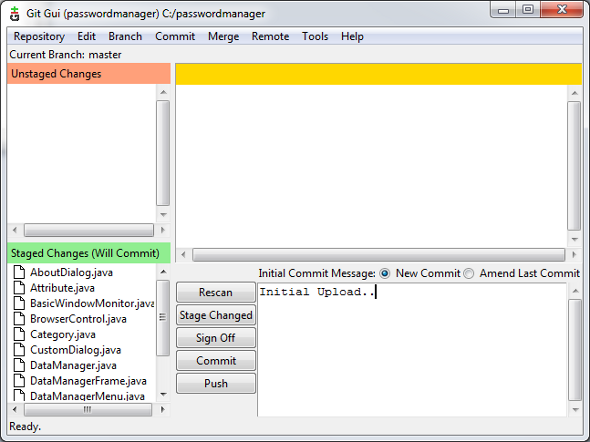
**3) Unstaged Changes:**

All the files located under “c:\passwordmanager” will be displayed under the “Unstaged Changes” section that is located on the top left corner.



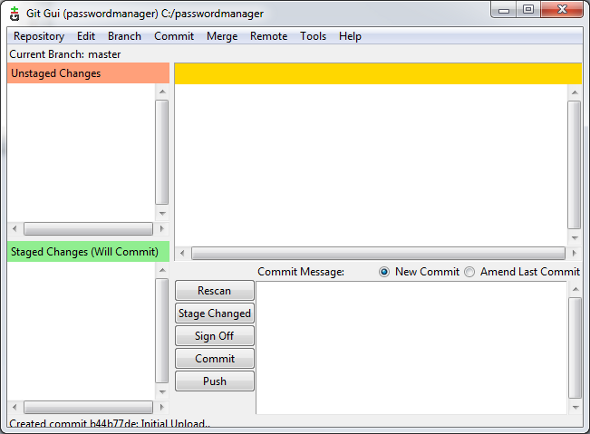
**4) Staged Changes**

Click on the “Stage Changed” button located at the bottom-middle section. This will stage all these files. Now, you’ll see the “Unstaged Changes” section will become empty, and all the files will be displayed under the “Staged Changes (Will Commit)” section that is located on the bottom left corner.



**5) Commit Changes**

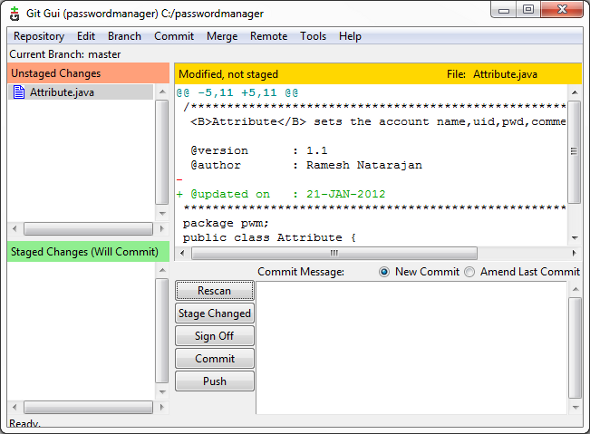
Enter a commit message on the big text box located on the bottom right corner, and click on “Commit” button. This will commit all the files from the “Staged Changes (Will Commit)” section. Once you commit, both the “Unstaged Changes” and “Staged Changes” section will be empty.



**6) Modify a file**

Whenever you modify a file from this local repository (in this example, c:\passwordmanager), you need to go to Git and stage the changes -> commit the change to the repository. For example, I modified the Attribute.java file located under c:\passwordmanager, and clicked on the “Rescan” button from the Git GUI, which will display only the changed file (for example: Attribute.java) in the “Unstaged Changes” section. You can click on this file, to see what lines were actually changed.

If you are happy with this change, click on “Stage Changed”, and then click on “Commit” to get this change committed to the local Git repository.



3. How to Connect to a Remote Git Repository

Most of the times, you’ll install Git GUI on Windows, so that you can download a remote repository located on a Linux server, and download the files to work on your local system.

In this case, you need to use the “Clone Remote Repository” item from the Git GUI main menu.

When you clone, Git pulls all the versions (not just the current version) of every file from the remote server.

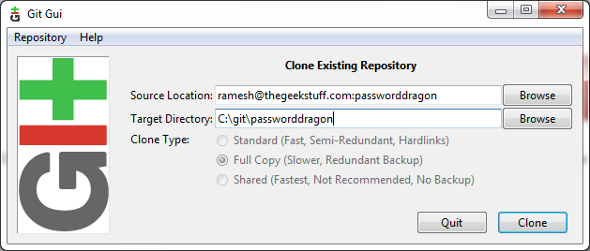
**Source Location:** This is the remote git URL. In most scenarios, you’ll be connecting to a remote Git repository using SSH.

Enter the source location in any one of the following format (Both are exactly the same):

* ssh://username@servername:gitproject
* username@servername:gitproject

In the following example, I’ve given ramesh@thegeekstuff.com:passworddragon, which will connect to thegeekstuff.com server using SSH protocol, and with username ramesh. Once connected, this will try to download the git project named ‘passworddragon’.

**Target Directory:** This is the directory on the local Windows machine, where the downloaded git project from the remote server should be stored.



Click on “Clone”, which will request you to accept the SSH key (if you are connecting to the remote git server for the first time), and prompt you to enter the password for the given username. At this stage, this will be downloading all the files from the remote Git repository and clone it as a local repository from where you can make any changes.

**GIT STATUS**

$ **git status**

On branch master

Initial commit

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: abc.txt

**Physically Modified the File abc.txt**

$ **git status**

On branch master

Initial commit

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: abc.txt

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

modified: abc.txt

nsingh@Tricorehyd-PC MINGW32 /c/Gits-Repo (master)

$ **git checkout -- abc.txt**

nsingh@Tricorehyd-PC MINGW32 /c/Gits-Repo (master)

$ git status

On branch master

Initial commit

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: abc.txt

**Finally, Changes are discards**

**Deleted the file Manually**

nsingh@Tricorehyd-PC MINGW32 /c/Gits-Repo (master)

$ git status

On branch master

Initial commit

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: abc.txt

Changes not staged for commit:

(use "git add/rm <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

deleted: abc.txt

$ **git checkout -- abc.txt**

nsingh@Tricorehyd-PC MINGW32 /c/Gits-Repo (master)

$ git status

On branch master

Initial commit

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: abc.txt

**Finally, file is restored**

nsingh@Tricorehyd-PC MINGW32 /c/Gits-Repo (master)

$ git status

On branch master

Initial commit

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: abc.txt

**Modified the abc.txt**

$ git status

On branch master

Initial commit

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: abc.txt

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

modified: abc.txt

nsingh@Tricorehyd-PC MINGW32 /c/Gits-Repo (master)

$ git add .

$ git status

On branch master

Initial commit

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: abc.txt