Contents

[How to Install Git? 4](#_Toc447782836)

[How to configure GIT ? 6](#_Toc447782837)

[How to add the remote for new Repository ? 7](#_Toc447782838)

[Shortcuts : How to create Files ? 7](#_Toc447782839)

[How to unset the configurations ? 7](#_Toc447782840)

[How to create a local Repository ? 7](#_Toc447782841)

[How to Delete the Local Repository ? 7](#_Toc447782842)

[How to add a Remote in Local ? 7](#_Toc447782843)

[How to Track the file changes? 7](#_Toc447782844)

[How to add a file to the Git Repository ? 8](#_Toc447782845)

[How to add the updated files ? 8](#_Toc447782846)

[Commit : How to commit the file to the repository ? 8](#_Toc447782847)

[Commit : How to commit and add the file in one command? 8](#_Toc447782848)

[Commit : Recovering Deleted commits: 8](#_Toc447782849)

[How to check the list of changes made to the branch ? 9](#_Toc447782850)

[How to check the history of changes ? 9](#_Toc447782851)

[To determine how many commits are made to the repository ? 11](#_Toc447782852)

[What is HEAD ? 11](#_Toc447782853)

[How to check the differences between the different commited version ? 11](#_Toc447782854)

[How to remove files using Git ? 11](#_Toc447782855)

[How to Rename File names? 11](#_Toc447782856)

[How to undo the changes made ? 11](#_Toc447782857)

[How to remove the current version and add the previous version as the latest ? 12](#_Toc447782858)

[How to Clean the working Copy ? 12](#_Toc447782859)

[Ignoring Files with .gitignore ? 12](#_Toc447782860)

[Clone : How to clone the repository to local ? 13](#_Toc447782861)

[How to View Commits ? 13](#_Toc447782862)

[How to view list of changes made in a commit ? 13](#_Toc447782863)

[Git Remote: 14](#_Toc447782864)

[Git protocols: 15](#_Toc447782865)

[Branches - How to view Branches ? 15](#_Toc447782866)

[Tags - How to View Tags? 15](#_Toc447782867)

[What is Origin ? 16](#_Toc447782868)

[Unstage – how to unstage a staged file ? 16](#_Toc447782869)

[How to add the local repository to Remote? 16](#_Toc447782870)

[Pull - How to get the updates made to the repository? 16](#_Toc447782871)

[Push – How to push the changes to the remote repository ? 17](#_Toc447782872)

[Push – How to push the branch to the remote? 18](#_Toc447782873)

[How to remove the origin ? 18](#_Toc447782874)

[Tags – Creating and verifying Tags? 18](#_Toc447782875)

[Tags – Verifying tag : 19](#_Toc447782876)

[Tag – Pushing tags to Remote : 19](#_Toc447782877)

[Branch : Visualizing Branches: 19](#_Toc447782878)

[Alias: setting in GIT: 19](#_Toc447782879)

[Branch: Creating Local Branch: 19](#_Toc447782880)

[Branch : How to switch between branches? 19](#_Toc447782881)

[Branch : Renaming branches: 20](#_Toc447782882)

[Branch: Deleting a Branch 20](#_Toc447782883)

[BRANCHES VS TAGS : 20](#_Toc447782884)

[Branch : Merging Branches : 20](#_Toc447782885)

[Branch: MergeTool: 21](#_Toc447782886)

[Branch : How to create a branch from a tag ? 21](#_Toc447782887)

[Stash : Stashing Changes: 21](#_Toc447782888)

[Usage of stash: 21](#_Toc447782889)

[Stash : How to apply the saved Stash ? 22](#_Toc447782890)

[Stash : How to drop a stash ? 22](#_Toc447782891)

[Stash: How to remove the Stashed Changes ? 22](#_Toc447782892)

[Stash : Creating a branch from Stash. 22](#_Toc447782893)

[Branch : Rebase: 22](#_Toc447782894)

[Merged without Rebase: 23](#_Toc447782895)

[Merged with Rebase: 23](#_Toc447782896)

[Rebase Workflow: 24](#_Toc447782897)

[Cherry Picking Changes: 25](#_Toc447782898)

[How to clear the console ? 26](#_Toc447782899)

[Understanding the log graph: 26](#_Toc447782900)

[To check? 26](#_Toc447782901)

[Commands: 27](#_Toc447782902)

# How to Install Git?

Step 1 :

<https://git-scm.com/download/win>

Go to the above link and download the latest version as per the OS.

Step 2:



Step 3:



Step 4:



# How to configure GIT ?

**To set the user name for the GIT globally:**

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git config --global user.name "Dilip Sundarraj"

$ git config --global –list:

This is used to list global variables for GIT in the system.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git config --global --list

user.name=Dilip Sundarraj

**To set the email:**

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git config --global user.email [dilip.thelip@gmail.com](mailto:dilip.thelip@gmail.com)

**Auto** **Correct**:

The below command actually auto corrects if you mistype something.

It waits for one tenth of a second and auto corrects it. As per the below command it waits for 0.1 seconds and auto corrects it.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git config --global help.autocorrect 1

Eg.,

We mistyped statsu but it waited for 0.1 seconds and auto corrects it.

$ git statsu

WARNING: You called a Git command named 'statsu', which does not exist.

Continuing under the assumption that you meant 'status'

in 0.1 seconds automatically...

On branch master

Initial commit

Untracked files:

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

How to Install GIT.docx

~$w to Install GIT.docx

nothing added to commit but untracked files present (use "git add" to track)

**Color.ui:**

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git config --global color.ui auto

**Crlf : (Carriage return line feed)**

**Read this is in GIt document.**

**For windows it is preferred to set it as true.**

**For Mac and Unix it is preferred to set it as false.**

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git config --global core.autocrlf true

**To display the config:**

cat .git/config

vim .git/config

# How to add the remote for new Repository ?

Follow the instructions in the below link.

<https://help.github.com/articles/adding-an-existing-project-to-github-using-the-command-line/>

# Shortcuts : How to create Files ?

echo “Hello Again”> Readme.txt -> This will create the Readme.txt file with the Hello Again

touch file1.txt file2.txt -> This command creates two files.

## How to unset the configurations ?

Git config –unset user.name -> This command unsets the configuration variable from the configuration list.

# How to create a local Repository ?

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git init

# How to Delete the Local Repository ?

rm -rf .git

# How to add a Remote in Local ?

git remote add origin https://github.com/user/repo.git

# How to Track the file changes?

Git status -> this command will tell you what are the untracked files in the repository.

Untracked files:

This tells the files mentioned are not in the repository and the files are not tracked by Git.

$ git status

On branch master

Initial commit

Untracked files:

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

How to Install GIT.docx

README.txt

~$w to Install GIT.docx

~WRL1961.tmp

nothing added to commit but untracked files present (use "git add" to track)

# How to add a file to the Git Repository ?

$ git add README.txt

# How to add the updated files ?

git add -u

# Commit : How to commit the file to the repository ?

$ git commit -m 'added readme'

# Commit : How to commit and add the file in one command?

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git commit -am 'addinf and commiting together'

[master 4a02c31] addinf and commiting together

4 files changed, 1 insertion(+), 1 deletion(-)

delete mode 100644 ~WRL1961.tmp

# Commit : Recovering Deleted commits:

Reflog will give you the list of commits that are done to the repository.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (feature2)

$ git reflog

712a714 HEAD@{0}: commit: aaa

b0de99c HEAD@{1}: checkout: moving from bug1234 to feature2

b0de99c HEAD@{2}: checkout: moving from master to feature1

ffba29b HEAD@{3}: commit: aaa

459db56 HEAD@{4}: checkout: moving from feature1 to master

b0de99c HEAD@{5}: commit: ddd

0df90cd HEAD@{6}: commit: adding the README from feature1 branch

459db56 HEAD@{7}: checkout: moving from master to feature1

459db56 HEAD@{8}: checkout: moving from master to master

459db56 HEAD@{9}: checkout: moving from feature1 to master

459db56 HEAD@{10}: checkout: moving from master to feature1

459db56 HEAD@{11}: commit (merge): Merged the README.txt changes

4b57b2b HEAD@{12}: commit: changed made in llocal

6aeb570 HEAD@{13}: commit: adding the word document

3031966 HEAD@{14}: pull: Fast-forward

b2fd4bb HEAD@{15}: commit (merge): merged the changes

42f7371 HEAD@{16}: commit: lkdfjlsfj

006c7df HEAD@{17}: commit: modified Read me txt

c8b51bd HEAD@{18}: commit: added

48fbde4 HEAD@{19}: commit: added the file

a65f207 HEAD@{20}: commit: dddd

4a02c31 HEAD@{21}: commit: addinf and commiting together

7b6f9c4 HEAD@{22}: commit: Editing the README to push to remote

3ab3e2c HEAD@{23}: pull: Fast-forward

0480afd HEAD@{24}: merge origin/master: Fast-forward

04698df HEAD@{25}: commit: added .gitignore file

b474a11 HEAD@{26}: commit: adding the file\_updated.txt

8b85e6f HEAD@{27}: reset: moving to HEAD~1

3345106 HEAD@{28}: commit: updated the file txt file

8b85e6f HEAD@{29}: commit: added the renamed file

99aa1cb HEAD@{30}: commit: added a new file

430ccaa HEAD@{31}: commit: added the file

12739b9 HEAD@{32}: commit: added and modified files

405ea2a HEAD@{33}: commit (initial): added readme

The below command will bring back the branch. Ideally what it does is that it takes back the commit and apply it to the branch.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (feature2)

$ git branch bug1234 b0de99c

# How to check the list of changes made to the branch ?

DHC+z001qgd@2QBZP12 MINGW64 /c/Dilip/Study/GIT (feature3)

$ git show feature3

# How to check the history of changes ?

Approach 1:

$ git log

commit 405ea2a3d6e2202194c15a910539876af48be7e9

Author: Dilip.Sundarraj <dilip.thelip@gmail.com>

Date: Fri Nov 20 13:33:32 2015 -0600

added readme

Approach 2:

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/Git\_PluralSight/Remote\_Repository/gitTutorial (master)

$ git log --oneline

ec3d3a3 Merge pull request #2 from dilipthelip/news

c3cd16f added the news module

b64a57e Merge pull request #1 from dilipthelip/feature1

79d674c added new module

4370fcf added the file inside the folder

ef46c74 added this change

89d4a6b made changes

2da6825 added index.html

24dba1f Initial commit

Approach 3:

This will give you the graph option.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/Git\_PluralSight/Remote\_Repository/gitTutorial (master)

$ git log --graph

\* commit ec3d3a37c064c4b9d3a54e38b59638340c9b147b

|\ Merge: b64a57e c3cd16f

| | Author: dilipthelip <dilip.thelip@gmail.com>

| | Date: Wed Aug 12 07:13:55 2015 -0500

| |

| | Merge pull request #2 from dilipthelip/news

| |

| | added the news module

| |

| \* commit c3cd16f9d48a23133124d56b15224c84d7c72d8b

|/ Author: dilipthelip <dilip.thelip@gmail.com>

| Date: Wed Aug 12 06:46:45 2015 -0500

|

| added the news module

|

\* commit b64a57e4400d727e12a1d354a180903978bf3b2f

|\ Merge: 4370fcf 79d674c

| | Author: dilipthelip <dilip.thelip@gmail.com>

| | Date: Wed Aug 12 06:42:29 2015 -0500

| |

| | Merge pull request #1 from dilipthelip/feature1

| |

| | sasadadsadadded new module

| |

| \* commit 79d674c4ab5ecbea9418c6aec05599fef61ea140

|/ Author: dilipthelip <dilip.thelip@gmail.com>

| Date: Wed Aug 12 06:31:13 2015 -0500

|

| added new module

|

\* commit 4370fcfaf2a5091ac62414d9780fb390a49c424d

| Author: dilipthelip <dilip.thelip@gmail.com>

| Date: Tue Aug 11 17:35:08 2015 -0500

|

| added the file inside the folder

|

\* commit ef46c7448391b00f6a0b804377218ada0c04f279

| Author: dilipthelip <dilip.thelip@gmail.com>

| Date: Tue Aug 11 16:37:01 2015 -0500

|

| added this change

|

\* commit 89d4a6b25f34d98096aeb8f00f6c52f9c2c7472d

| Author: dilipthelip <dilip.thelip@gmail.com>

| Date: Tue Aug 11 16:13:38 2015 -0500

|

| made changes

|

\* commit 2da682525ce742870bd943616a742bb4e28e2a91

| Author: dilipthelip <dilip.thelip@gmail.com>

| Date: Tue Aug 11 15:49:16 2015 -0500

|

| added index.html

|

\* commit 24dba1f387d594f9ac46df9c21d86f012c34e3f3

Author: dilipthelip <dilip.thelip@gmail.com>

Date: Tue Aug 11 15:19:24 2015 -0500

Initial commit

Approach 4:

This is another view of the log. If there are multiple people involved in checking in the files then it will be listed down below.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/Git\_PluralSight/Remote\_Repository/gitTutorial (master)

$ git shortlog

dilipthelip (9):

Initial commit

added index.html

made changes

added this change

added the file inside the folder

added new module

Merge pull request #1 from dilipthelip/feature1

added the news module

Merge pull request #2 from dilipthelip/news

Approach 5:

git shortlog -sne

S -> summary

N ->order them by Numerically decreasing

E ->include email in the result

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/Git\_PluralSight/Remote\_Repository/gitTutorial (master)

$ git shortlog -sne

WARNING: You called a Git command named 'shorlog', which does not exist.

Continuing under the assumption that you meant 'shortlog'

in 0.1 seconds automatically...

9 dilipthelip <dilip.thelip@gmail.com>

## To determine how many commits are made to the repository ?

The below command will tell you how many commits are made to the repository ?

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/Git\_PluralSight/Remote\_Repository/gitTutorial (master)

$ git log --oneline | wc -l

9

# What is HEAD ?

The latest commit is known as HEAD.

Head~1 -> This is the version is the prior to the latest commit.

# How to check the differences between the different commited version ?

HEAD -> This is the latest committed version

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git diff HEAD~1..HEAD

# How to remove files using Git ?

Approach 1:

To remove the file from git and local system

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git rm file1.txt

Approach 2:

To remove the file from git and not from local system.

git rm --cached file1.txt

# How to Rename File names?

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ mv file file\_updated.txt

# How to undo the changes made ?

Approach 1:

To undo the changes of one particular file

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git checkout file\_updated.txt

# How to remove the current version and add the previous version as the latest ?

**Approach 1:**

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git reset --soft HEAD~1

This command removes the current HEAD version from the repository and updates it with the previous version.

**Approach 2:**

If you want to undo changes of multiple files then

Git reset –hard

# How to Clean the working Copy ?

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git clean

fatal: clean.requireForce defaults to true and neither -i, -n, nor -f given; refusing to clean

git clean –n:

This will give you the message would remove the stray files

git clean –f:

This will remove the unstaged files from the working directory

# Ignoring Files with .gitignore ?

Add the .gitignore file in to the root path of the repository.

.gitignore file is a txt file.

Eg.,

I have added a logs folder to the root path.

$ git status

On branch master

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: How to Install GIT.docx

Untracked files:

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

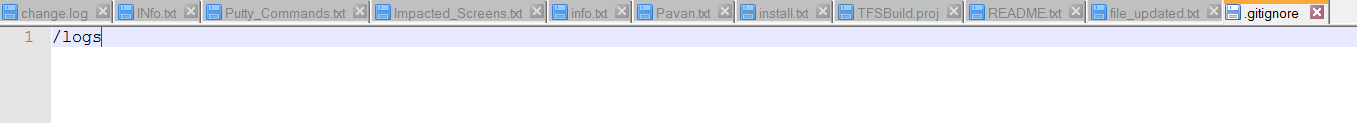
logs/

~WRL1250.tmp

~WRL4044.tmp

no changes added to commit (use "git add" and/or "git commit -a")

I do not want to be considered as a tracked file. Add the below line in the .gitignore file.



After adding the /logs in the ignore file the log folder is not displayed in the untracked files section after you run the git status command.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git status

On branch master

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: How to Install GIT.docx

Untracked files:

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

.gitignore

~WRL1250.tmp

~WRL4044.tmp

no changes added to commit (use "git add" and/or "git commit -a")

## Files Already in the folder :

If you have files already in the folder and checked in to the github then please delete the files in the folder:

Step 1:

Delete the files in the folder and push the changes to the environment.

Step 2:

Lets say Ibatis and Mybatis is the folder that you want to ignore. Please add the below in the

Ibatis/

Mybatis/

Working Remotely with GIT:

# Clone : How to clone the repository to local ?

git clone <URL>

# How to View Commits ?

This will tell you what is the latest commit that has been made to the repository.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/Git\_PluralSight/Remote\_Repository/gitTutorial (master)

$ git show HEAD

commit ec3d3a37c064c4b9d3a54e38b59638340c9b147b

Merge: b64a57e c3cd16f

Author: dilipthelip <dilip.thelip@gmail.com>

Date: Wed Aug 12 07:13:55 2015 -0500

Merge pull request #2 from dilipthelip/news

added the news module

# How to view list of changes made in a commit ?

Follow the below steps:

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/Git\_PluralSight/Remote\_Repository/gitTutorial (master)

$ git log --oneline

ec3d3a3 Merge pull request #2 from dilipthelip/news

c3cd16f added the news module

b64a57e Merge pull request #1 from dilipthelip/feature1

79d674c added new module

4370fcf added the file inside the folder

ef46c74 added this change

89d4a6b made changes

2da6825 added index.html

24dba1f Initial commit

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/Git\_PluralSight/Remote\_Repository/gitTutorial (master)

$ git show 24dba1f

commit 24dba1f387d594f9ac46df9c21d86f012c34e3f3

Author: dilipthelip <dilip.thelip@gmail.com>

Date: Tue Aug 11 15:19:24 2015 -0500

Initial commit

**diff --git a/README.md b/README.md**

**new file mode 100644**

**index 0000000..27e4350**

**--- /dev/null**

**+++ b/README.md**

@@ -0,0 +1 @@

+# gitTutorial

# Git Remote:

This tells “Git” from where the code is fetched and pushed to.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/Git\_PluralSight/Remote\_Repository/gitTutorial (master)

$ git remote

origin

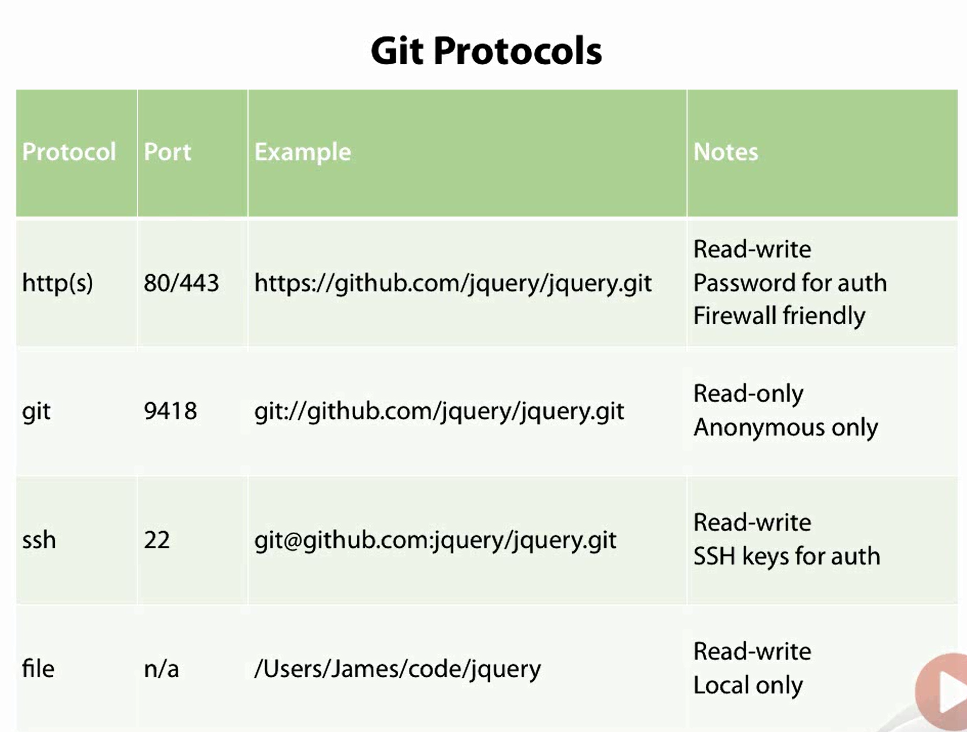
DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/Git\_PluralSight/Remote\_Repository/gitTutorial (master)

$ git remote -v

origin https://github.com/dilipthelip/gitTutorial.git (fetch)

origin https://github.com/dilipthelip/gitTutorial.git (push)

# Git protocols:



# Branches - How to view Branches ?

git branch -> This command will list of branches created for the repository.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/Git\_PluralSight/Remote\_Repository/gitTutorial (master)

$ git branch

\* master

# Tags - How to View Tags?

Tags -> Tags are known points in your code base where we can often tag versions.

The below tag will list down the number of tags available for the repository

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/Git\_PluralSight/Remote\_Repository/gitTutorial (master)

$ git tag

# What is Origin ?

**origin** is not the remote repository name. It is an alias given to the URL pointing to the default remote repository.

# How to Delete Untracked file:

Dry Run:

The below command will show what will happen after we run the command

git clean -f -n

## To Delete Files :

This will delete the files.

git clean –f

## To Delete Directories:

git clean –f -d

# Unstage – how to unstage a staged file ?

DHC+z001qgd@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git rm --cached How\_to\_Install\_GIT\_And\_Commands.docx

# How to unstage multiple files after add ?

### Approach 1:

**Git reset** -> This will remove the files from the index and have the changes intact.

DHC+z001qgd@2QBZP12 MINGW64 /c/Dilip/Study/MicroServices/workspace (master)

$ git reset

warning: LF will be replaced by CRLF in CustomerService/src/main/java/com/customerService/CustomerServiceApplication.java.

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

Unstaged changes after reset:

M CustomerService/src/main/java/com/customerService/CustomerServiceApplication.java

M CustomerService/src/main/java/com/customerService/businesslogic/CustomerRegistrar.java

### Approach 2:

**git reset --hard HEAD** ->This will remove the files from the index and revert the changes that are added as part of the index.

DHC+z001qgd@2QBZP12 MINGW64 /c/Dilip/Study/MicroServices/workspace (master)

$ git reset --hard HEAD

HEAD is now at 6ec4aa3 added the unstaged fil

# How to add the local repository to Remote?

Create the repository in github and get the clone url

<https://github.com/dilipthelip/git_pluralsight>

Approach 1:

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git remote add origin <https://github.com/dilipthelip/git_pluralsight.git>

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git push -u origin master

Approach 2: (SSh version)

# How to see the list of changes in a commit ?

git show <commit>

# Pull - How to get the updates made to the repository?

Approach 1:

This will pull down any changes made to the remote repository.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git fetch

The below command will tell you the changes that has been made to the branch remotely by someone.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git log origin/master

How to get that in to our local working copy?

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git merge origin/master

Updating 04698df..0480afd

Fast-forward

README.txt | 4 +++-

1 file changed, 3 insertions(+), 1 deletion(-)

Approach 2:

The below command gets the updates from the remote branch and updates the local repository.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git pull

remote: Counting objects: 3, done.

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

remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0

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

From https://github.com/dilipthelip/git\_pluralsight

0480afd..3ab3e2c master -> origin/master

Updating 0480afd..3ab3e2c

Fast-forward

README.txt | 1 +

1 file changed, 1 insertion(+)

**Scenario 1: (changes not committed)**

Changes not committed in local.

Git pull will always fail

**Scenario 2:**

Changes committed to the repository but not pushed to remote.

Git pull will automerge the changed lines in the file like below.

<<<<<<< HEAD

Changing it in local.

=======

Changing it in remote.

>>>>>>> 91903b3cf7072c0e816a441e5ddf7e128aa999d2

# Push – How to push the changes to the remote repository ?

warning: push.default is unset; its implicit value has changed in

Git 2.0 from 'matching' to 'simple'. To squelch this message

and maintain the traditional behavior, use:

git config --global push.default matching

To squelch this message and adopt the new behavior now, use:

git config --global push.default simple

Simple -> Pushes only the current branch.This is the preferable option.

Matching -> It will push all the branches to the remote branch and would merge them.

<http://stackoverflow.com/questions/21839651/git-what-is-the-difference-between-push-default-matching-and-simple>

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git commit -am 'addinf and commiting together'

[master 4a02c31] addinf and commiting together

4 files changed, 1 insertion(+), 1 deletion(-)

delete mode 100644 ~WRL1961.tmp

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git push

If changes are not pushed then we need to set the working directory to point to the correct branch.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git branch --set-upstream-to=origin/master master

Provide the credentials it will automatically push it to the branch

# Push – How to push the branch to the remote?

This will push the branch to the remote.

DHC+z001qgd@2QBZP12 MINGW64 /c/Dilip/Study/GIT (feature3)

$ git push --set-upstream origin feature3

# How to remove the origin ?

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git remote rm origin

# Tags – Creating and verifying Tags?

This is basically used to tag a commit point.

Regardless of whatever happens the tag version v1.0 always points to the commit point when the tag was created.

Approach 1:

DHC+z001qgd@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git tag v1.0

Approach 2:

In the below version it provides a tag with a message.

DHC+z001qgd@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git tag v1.0\_message -m 'tag with version'

Approach 3:

Signing the tag – meaning some person signed and confirmed this is the official 1.0 release.

Follow the below link and set up few stuff

<http://stackoverflow.com/questions/12061645/generating-a-gpg-key-for-git-tagging>

Please re read it again. It did not work.

# Tags – Verifying tag :

-v is used to verify tag.

DHC+z001qgd@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git tag -v v1.0\_message

object b2fd4bbea33c34818ab88a08298ae8129271f311

type commit

tag v1.0\_message

tagger dilipthelip <dilip.thelip@gmail.com> 1448551517 -0600

tag with version

error: no signature found

error: could not verify the tag 'v1.0\_message'

# Tag – Pushing tags to Remote :

DHC+z001qgd@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git push –tags

This will push the tag to the remote.

# How to tag an old commit ?

git tag -a v1.2 9fceb02 -m "Message here"

# Tag – How to delete a tag?

git tag -d 1.2

# Branch : Visualizing Branches:

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master|MERGING)

$ git log --graph --oneline --all --decorate

# Alias: setting in GIT:

This get alias lga a defined value like below.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master|MERGING)

$ git config --global alias.lga "log --graph --oneline --all --decorate"

Alias : TO view it in git config.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master|MERGING)

$ cat ~/.gitconfig

# Branch: Creating Local Branch:

The below command creates a local branch

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git branch feature1

# Branch : How to switch between branches?

Approach 1:

“git checkout” will make you switch between branches.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git checkout feature1

This will create the branch “feature2” and checkout to that branch.

Approach 2:

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (bug1234)

$ git checkout -b feature2

# Branch : Renaming branches:

-m - > This command is used to rename a branch.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (feature1)

$ git branch -m feature1 bug1234

# Branch: Deleting a Branch

The below command will help me delete a branch.

Approach 1:

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (feature2)

$ git branch -d bug1234

Deleted branch bug1234 (was b0de99c).

Approach 2:

Force delete a branch. Use “D” instead of “d”.

git branch –D bug3345

# BRANCHES VS TAGS :

|  |  |
| --- | --- |
| Branches | Tags |
| This will create branch and follow the commits made to the branch. | This is like a reference which points to one commit. |
|  |  |

# Branch : Merging Branches :

How to merge branch changes to a master.

Switch to master branch.

git merge <Target Branch Name>

Approach 1: (Fast Forward)

DHC+z001qgd@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git merge mergebranch

Updating 03413ef..f7db93b

Fast-forward

README.txt | 7 ++-----

1 file changed, 2 insertions(+), 5 deletions(-)

This merges the changes from the “mergebranch” to the master branch.

Basically it changes the master to point to the mergebranch.

Delete the branch that got merged.

DHC+z001qgd@2QBZP12 MINGW64 /c/Dilip/Study/GIT (master)

$ git branch -d mergebranch

Approach 2: (Dealing with confilicts):

Approach 3:

Merge the specified branch into the current branch, but always generate a merge commit.

git merge --no-ff <branch>

# Branch: MergeTool:

# Branch : How to create a branch from a tag ?

git branch feature3 v1.0

# Stash : Stashing Changes:

A stash is a holding are for your pending changes.

## Usage of stash:

Stash can be mainly used in scenarios where you are working on something but suddenly there is a production issue which you need to work on during that scenario you can stash the changes you were working and start working on the production issue.

This will save your work. This is a holding area for the pending changes.

This command will remove the changes made to the branch and move it to the previous commited version.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (feature1)

$ git stash

The below command will list you the saved stash.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (bug1234)

$ git stash list

stash@{0}: WIP on bug1234: b0de99c ddd

# Stash : How to apply the saved Stash ?

Approach 1:

The below command will apply the stashed changes to the branch after you switch to the branch you stashed initially.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (feature1)

$ git stash apply

Approach 2:

The below command will remove the stash from the stash list and apply it to the working copy.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (feature1)

$ git stash pop

# Stash : How to drop a stash ?

This will drop the stash which is at the top.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (feature1)

$ git stash drop

Dropped refs/stash@{0} (668b5147ce2502855d76665d91e5f64e59a3d682)

# Stash: How to remove the Stashed Changes ?

Approach 1 :

The below git command will remove the changes that was stashed.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (feature1)

$ git reset --hard HEAD

# Stash : Creating a branch from Stash.

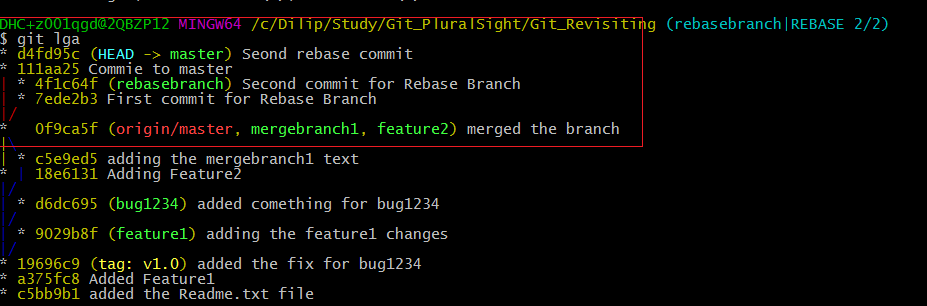
1. This will remove the stash from the stash list and
2. Apply the changes to the new branch
3. Checkouts to the new branch.

DHC+Z001QGD@2QBZP12 MINGW64 /c/Dilip/Study/GIT (feature1)

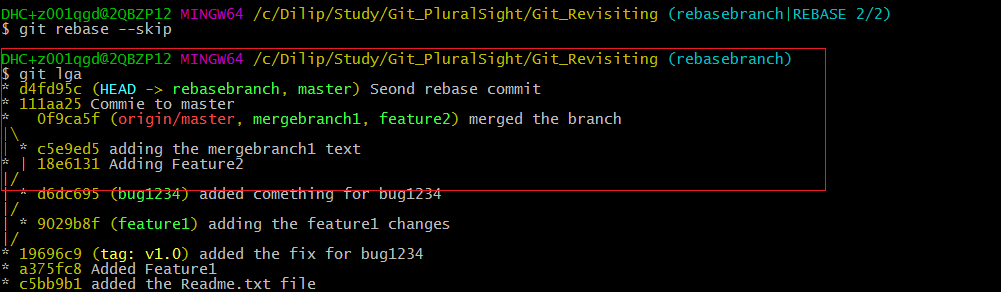
$ git stash branch 'feature1\_additional'

# Branch : Rebase:

Before :



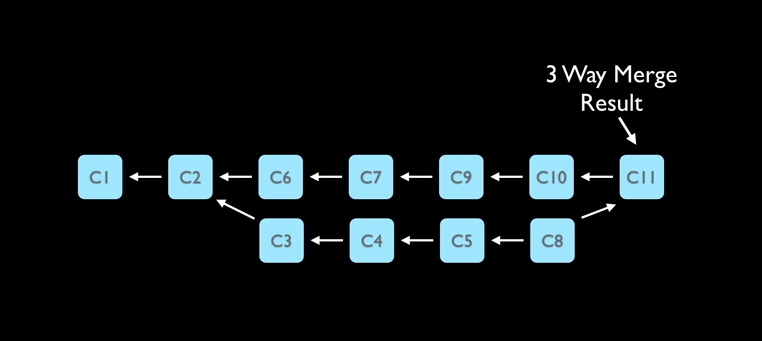
After:



By comparing those two the graph structure has been changed.

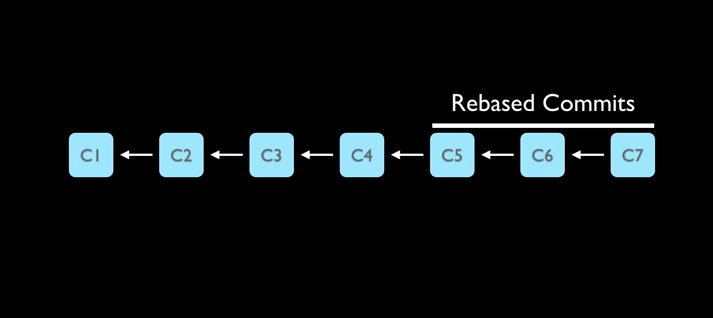
Comparison merge between without rebase and with rebase:

## Merged without Rebase:



## Merged with Rebase:

It does not look like work was done on a branch.



## Rebase Workflow:

Commit **a7a7acf** is away from the master branch before rebase.

Step1:

Switch to the master branch.

Step 2:

git rebase master

Step 3:

If there are no conflicts then it will merge the branch and the commit history will be looking different from the previous one.

DHC+z001qgd@2QBZP12 MINGW64 /c/Dilip/Study/Git\_PluralSight/Git\_Revisiting (master)

$ git lga

\* da52aca (HEAD -> master) cherrypicking changes1

| \* a7a7acf (cherrypickbranch) cherrypicking changes2

| \* 3b0295b cherrypicking changes1

|/

\* d4fd95c (rebasebranch) Seond rebase commit

\* 111aa25 Commie to master

\* 0f9ca5f (origin/master, mergebranch1, feature2) merged the branch

|\

| \* c5e9ed5 adding the mergebranch1 text

\* | 18e6131 Adding Feature2

|/

| \* d6dc695 (bug1234) added comething for bug1234

|/

| \* 9029b8f (feature1) adding the feature1 changes

|/

\* 19696c9 (tag: v1.0) added the fix for bug1234

\* a375fc8 Added Feature1

\* c5bb9b1 added the Readme.txt file

DHC+z001qgd@2QBZP12 MINGW64 /c/Dilip/Study/Git\_PluralSight/Git\_Revisiting (master)

$ git checkout cherrypickbranch

Switched to branch 'cherrypickbranch'

DHC+z001qgd@2QBZP12 MINGW64 /c/Dilip/Study/Git\_PluralSight/Git\_Revisiting (cherrypickbranch)

$ git rebase master

First, rewinding head to replay your work on top of it...

Applying: cherrypicking changes2

DHC+z001qgd@2QBZP12 MINGW64 /c/Dilip/Study/Git\_PluralSight/Git\_Revisiting (cherrypickbranch)

$ git lga

\* 5917650 (HEAD -> cherrypickbranch) cherrypicking changes2

\* da52aca (master) cherrypicking changes1

\* d4fd95c (rebasebranch) Seond rebase commit

\* 111aa25 Commie to master

\* 0f9ca5f (origin/master, mergebranch1, feature2) merged the branch

|\

| \* c5e9ed5 adding the mergebranch1 text

\* | 18e6131 Adding Feature2

|/

| \* d6dc695 (bug1234) added comething for bug1234

|/

| \* 9029b8f (feature1) adding the feature1 changes

|/

\* 19696c9 (tag: v1.0) added the fix for bug1234

\* a375fc8 Added Feature1

\* c5bb9b1 added the Readme.txt file

DHC+z001qgd@2QBZP12 MINGW64 /c/Dilip/Study/Git\_PluralSight/Git\_Revisiting (cherrypickbranch)

$

# Cherry Picking Changes:

Step 1:

Create a branch out of the master

Step 2:

Make two commits to the branch

Step 3:

You need only the first commit to be merged in to the master branch.

Step 4:

Switch to master branch and use the below command. It does the addition and committing the transaction.

DHC+z001qgd@2QBZP12 MINGW64 /c/Dilip/Study/Git\_PluralSight/Git\_Revisiting (master)

$ git cherry-pick 3b0295b

[master da52aca] cherrypicking changes1

Date: Mon Dec 28 08:44:12 2015 -0600

1 file changed, 1 insertion(+)

# How to clear the console ?

Clear

# Understanding the log graph:

HEAD is the latest commit on the branch.

\* 18e6131 (HEAD -> feature2) Adding Feature2

The above line tells the feature2 is ahead of the below branches.

19696c9 (tag: v1.0, origin/master, master, bug1234) added the fix for bug1234

The above line tells that tag v1.0,origin/master,master,bug1234 are in sync with each other.

DHC+z001qgd@2QBZP12 MINGW64 /c/Dilip/Study/Git\_PluralSight/Git\_Revisiting (feature2)

$ git lga

\* 18e6131 (HEAD -> feature2) Adding Feature2

\* 19696c9 (tag: v1.0, origin/master, master, bug1234) added the fix for bug1234

\* a375fc8 (feature1) Added Feature1

\* c5bb9b1 added the Readme.txt file

# To check?

1. How to remove the initialized repository ?
2. How to handle conflicts?

Scenario 1:

1. Modify the file in Git hub
2. Modify the same file in local and do not commit or stage the file.
3. Git pull request

This will not let you update the file in the local. Git will notify you to push the change or stage the change to the GIT repository.

Scenario 2:

1. Add a new file in local.
2. Modify a different file in git hub
3. Pull request.
4. How to push changes from branch to master?

This is completed. Check the “Push – How to push the branch to the remote?”

1. How to merge two branches ?

# Commands:

Cat:

Cat command is used to read one or more files and display it in the standard output.