**Check version**

$git –version

Git version 1.7.4

**Configuration**

Config file is under home directory (.gitconfig) global

System /etc/gitconfig

Change global setting(current user)

$git config --global user.name “First Last”

$git config --global user.email [firstlast@123.com](mailto:firstlast@123.com)

$git config --global color.ui true

Change system setting (all users) user must have administrator privilege

$sudo git config --system alias.st status

$sudo git config –system alias.ci commit

Edit config file

$cd /path/to/my/workspace/demo

$git config –e

$git config –e –global

$git config –e –system

Read value

$git config <section>.<key>

[core]’

Bare = false

$git config core.bare

False

Write value

$git config <section>.<key> <value>

$git config a.b something

[a]

B = something

$git config x.y.z others

[x “y”]

Z = others

**Use git config to change other INI file(config)**

$GIT\_CONFIG=test.ini git config a.b.c.d “hello, world”

Read

$GIT\_CONFIG=test.ini git config a.b.c.d

Hello, world

Delete setting

$git config –unset –global user.name

**Git initialization(init)**

$cd /path/to/my/workspace

$mkdir demo

$cd demo

$git init

Initializaed empty Git repository in /path/to/my/workspace/demo/.git/

If git version is higher than 1.6.5

$cd /path/to/my/workspace

$git init demo

Initializaed empty Git repository in /path/to/my/workspace/demo/.git/

$cd demo

**Add file to be commited(add)**

$git add <file name>

**Commit file to version control system(commit)**

$git commit –m “initialized”

*-m*

Add updates comments

Allow submit empty file

--allow-empty

$git commit –allow-empty

--*amend*

Fix the submission info

*--reset-author*

Fix author id, date

**Search files(grep)**

$git grep “search content”

**Check version status(status)**

Trace search path

$strace –e ‘trace=file’ git status

$git status

-s

Use simple format

$git status –s

M filename

M means modified. If M is displayed in col 1, this file is modified between Repository and Stage(2 and 3).

If M is displayed in col 2, this file is modified between work copy and stage(1 and 2).

Repository (3)

Stage (2)

Work Copy (1)

**Find .git location**

Find .git folder location

$cd /path/to/my/workspace/demo/

$mkdir –p a/b/c

$cd /path/to/my/workspace/demo/a/b/c

$git rev-parse –show-toplevel

/path/to/my/workspace/demo

$git rev-parse –show-prefix

a/b/c

$git rev-parse –show-cdup

../../../

**Log info(log)**

–pretty=fuller

Use full format

$git log –pretty=fuller

--pretty=oneline

Use simple format

$git log –pretty=oneline

–stat

Display file change history

$cd /path/to/my/workspace/demo

$git log –stat

Commit 40 bit commit id

Author: xxxx xxx<email>

Date:

**Compare file(diff)**

$git diff

Diff –git a/xxxxx.filename b/xxxxx.filename

Index

--- a/filename

+++ b/filename

HEAD

$git diff HEAD

Stage (2)

Work Copy (1)

Repository (3)

No paramter

$git diff

Compare the differences between work copy and stage (1 and 2)

HEAD

$git diff HEAD

Compare the differences between work copy and repository (1 and 3)

--cached, --staged

$git diff –cached or –staged

Compare the differences between stage and repository(2 and 3)

**Push file**

$git push origin master

**Withdraw the changes(checkout)**

$git checkout – welcome.txt

$git checkout . (all files)

All change in work copy will be gone.

**Clean up file(clean)**

$cd /path/to/my/workspace/demo

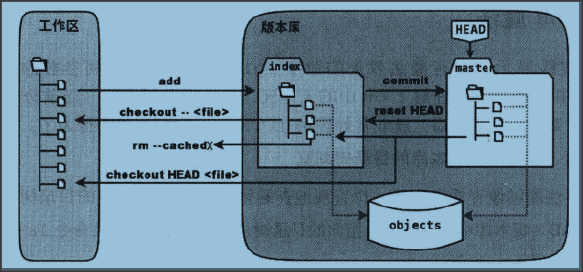
$git clean –fd

$git checkout .

Clean up the files in work copy which haven’t been added to stage

Replace the rest of files with stage.

**Unstage file(reset or checkout)**



**Git add(add file from work copy to stage)**

**Git checkout --<file> (work copy will be replaced by stage)**

**Git checkout HEAD. Or Git checkout HEAD <file> (work copy and stage file will be replaced by repository – HEAD)**

**Git commit(submit file from stage to repository - HEAD)**

**Git reset HEAD(stage will be replaced by repository –HEAD)**

**Git rm –cached <file> (the file will be deleted from stage)**