## Concepts

#### **Storage**

Most everything that git uses / manages, etc is stored in the .git directory.

What is in the .git directory?

.git

\objects

\refs

\heads

\tags

\info

HEAD

config

description

index

Explain contents of each…

config - project specific configuration

index - contains staging area information

Addressible?

git is an addressible database, in other words, stores information using an address

Content can be retrieved using it’s address

How is content stored in git?

In the objects directory

Is there anything git needs which isn’t in the .git directory?

TODO

Git content

Content stored in a git repository

Content can be added either using porcelain commands or plumbing commands

Content is made up of blob and tree objects

Git object

Object types are:

blob

similar to unix inode

A bunch of bytes which can be anything, a file, a picture, etc.

The basic storage unit in git

tree

similar to unix directory structure

can point to or include things

blob objects

other tree objects

associates a name with content and may contain multiple pieces of content

contains one or more tree entries

Tree entry may be either a checksum reference to a blob or another tree entry

represents a snapshotg

commit

Metadata about a commit operation

Pointer to tree object representing repository at point when commit was made

Parent commit

tag

Like a commit object except that refers to a commit instead of a tree.

Like a branch reference except that the commit referenced is static.

two types of tags

lightweight

reference to a commit that never moves

annotated

reference to a tag object

Does git manage file changes?

No, it manages content changes.

staged change

Change which has been marked for commit in your workspace.

head

A reference to the latest commit on each branch.

The tip of the currently checked out branch

#### **Remotes**

#### **Repositories**

What is the repository?

A collection of all the files that belong to a project, along with the history of the project's files.

What is a "bare" repository?

A repository with no working directory.

#### **Commits**

What is a commit?

A change in the history of a project

What is a parent commit?

TODO

How does commit relate to revision?

They are synonyms

How does one identify a commit?

By the 40 character hexadecimal string

In relation to other commits

origin..master - all commits to master since it deviated from origin

What is an SHA-1?

A type of secure hash algorithm

What is the meaning of the ~ symbol?

parent commit

HEAD~2 refers to 2 commits back in history

master?

The default branch in your local working copy

What is a remote-tracking branch?

A reference to branches present in the remote repository from which one cloned a particular branch.

origin?

The default name of the remote repository you want to publish to.

origin is typically the remote repository originally cloned from

Lots of commands

Many commands are simply special versions of the core commands.

For example, git shortlog is a special version of git log

#### **References**

What is a reference?

A reference is a named commit.

Why references?

Because it is hard working with SHA-1 values.

References make is easier to refer to a commit without having to remember a complicated SHA-1 value.

What are examples of references?

HEAD

MERGE\_HEAD

Branch names

Tags

Branch is a reference?

Yes, it is a pointer to the head of a line of work, in other words, the last commit on a line of work

How are references represented in the repository?

References are stored in .git/

HEAD references one of the files in .git/refs/heads

Branches are stored in .git/refs/heads

Tags are stored in .git/refs/tags

#### **Branching and merging**

What is a branch?

A line of development

What is meant by a branch referencing a start-point?

A reference to a commit from which a line of development started

What is auto-merge?

This is the default behavior of git.

What problems can occur when merging?

TODO

How do you switch branches?

git checkout {branchname}

Can you switch branches if there are unstaged changes in your working dir?

Yes

When you switch branches and there are unstaged changes in the working directory, what is the behavior?

They are similarly untracked in the new branch.

Why can’t you switch branches when there are outstanding changes in working dir or staging dir?

What types of merges are there?

Fast-forward merge

True merge

What is a true merge?

A merge that involves creation of a merge commit

What the possible outcomes of a merge?

If --no-commit and --no-ff and no conflicts, nothing happens

if --no-commit and no conflicts and no changes on parent branch since branch point, ff-merge is performed.

if no conflicts and true merge required, a merge commit is created

if conflicts, provides you a mechanism for manually resolving the conflicts so that the merge can be performed manually.

When a conflict is detected, how is this represented in git?

HEAD doesn’t change (points to the tip of the local branch)

MERGE\_HEAD is created and points to tip of origin branch

paths (files) which merge cleanly are updated both in the staging area and the working directory.

paths which have conflicts cause the following:

staging area is updated to include up to three versions of the path

1. version from common ancestor
2. version from HEAD
3. version from MERGE\_HEAD

working directory path is updated to represent the result of the merge program

How are merge conflicts resolved?

Use a graphical mergetool

Manually

???

What is an octopus merge?

TODO

Unmerged paths?

TODO

What kinds of conflicts are typical?

edit collision

the same line is changed in different lines of development

TODO

Once a path conflict is resolved, how is it committed?

#### **Applying branch changes to other branches**

Use either merge or rebase

merge joins two different branches

What is a merge?

The point at which two lines of development reconverge

#### **Fast-forward commit**

A merge in which a merge commit isn’t created

Only possible in situations where the current branch hasn’t changed between the time the other branch was created and the referenced commit.

Doesn’t actually create a commit, simply repoints the target branch to the commit referenced by the source.

#### R**evisions**

What is a revision?

A commit

How are revisions identified?

By SHA-1 has, or named identifier such as a branch or tagname

SHA-1 hashes are long. Is there an easier way to specify a hash?

use --abbrev-commit in various commands to retrieve a shortened version of the hash

Why are revisions referenced in commands?

When information indicated by a revision is required by a command.

What are some commands which make use of revisions?

diff

#### **Untracked files**

Files in the working directory but not yet added to the index

#### **detached head**

State in which the HEAD doesn’t point to a branch.

#### **garbage collection**

Process used by git to remove commits that aren’t referred to by either a branch or a tag.

merge conflicts

#### 

**Pull request**

Sends the owner of a repository a notification that you have something that they may like to include in the source.

**Local versus tracking branch**

local branches are viewable using git branch

remote tracking branches are viewable using git branch -r

**Remote versus local branch**

You can’t actually work on remote tracking branches

Local branch

branch which is only visible to a single user

stored in refs/heads

Remote branch

a former local branch which has been pushed to a remote repository and is visible to multiple people

Local tracking branch

A local copy of a remote branch

Remote tracking branch

A branch on your local machine which corresponds to the current state of that branch on the remote machine. Should not be edited.

Provides a view of the last known state of the branch in the remote repository.

Remote branches are stored in refs/remotes/<remote-name>/<branch-name>

remotes/origin/

remotes/origin/HEAD -> origin/master

remotes/origin/master

remotes/origin/origin/tags/

remotes/origin/origin/trunk