Git

## Common procedures

How do you get help on a git command?

e.g.

git help status

How do I create a git repository?

In gitweb

Or use git init

How do I rename a repository?

Via the gitweb app, navigate to the project and click settings

How do you push local commits to a remote repository?

git push

How can I avoid typing username / pw all the time?

git config --global credential.helper "cache --timeout=3600"

How can you access the git UI?

export DISPLAY=:0.0

start Cygwin-X

C:\dev\tools\cygwin\bin\run.exe /usr/bin/bash.exe -l -c /usr/bin/startxwin.exe

gitk

git gui

How can you see a graphical display of branches?

Apply changes made in a branch to parent branch?

Combining commits

You want to combine several commits into a single commit

ONly do this if commits haven't been pushed to a public repository

fetch versus pull

git pull does git fetch followed by git merge

common git tasks

Create a repository

Either create a new local repository, or copy an existing one.

Get a copy of a repository

git clone file:///opt/git/recipes.git

Display contents of last commit

View local commits

See if there are any remote changes

TODO

Add a new file to the repository

git add

Remove a file from the repository

git rm

Switch to different branch

See what branch you are on

git branch

current branch will have a star next to it

Plumbing

Git is essentially an object key-value datastore. Given an object, git can store the object and return a key which can be used to later retrieve the object.

Keys

Keys are used for referencing stored content.

Repository structure

A git repository is physically represented by the contents of the .git folder

objects

folder is where content is stored

refs

holds files containing references to checksums (SHA-1's)

tags and heads

How do you view the current context of the index?

Commands

hash-object -

calculate checksum for content, optionally store the content.

doesn't store filename when storing file contents

cat-file - provide information about content referenced by given checksum

update-index

can be used to create / manipulate an index

added files will display in git status as new file staged for commit

What's the difference between updating an index and writing a tree?

A tree is a tree object representation of the staging area

Why write tree?

Creates a tree object eligible to be committed

stages content in the staging area

write-tree

create tree objects from the staging area

--prefix use to create a subdirectory

Tree is a representation of a snapshot

Any tree write affects git's representation of a tree structure

Differences between what is stored in git and what is stored in the working tree is interpretted as differences to be committed.

generates tree nodes in the object database

checkout

switch to a different branch

git checkout [branch]

merge

apply changes from one branch into another branch

git merge [branch]

creates a merge commit such that both branches end up with the same content

commit-tree

performs a commit

read-tree

load tree objects into the staging area

References

Aliases to commits

update-ref

How does HEAD relate to references?

References the latest commit in the current branch

More specifically, it's a symbolic reference to the branch that is checked out.

Branches are references in commit object SHA's contained in refs/heads

HEAD refers to a file in refs/heads

How do you determine what the current branch is?

See what HEAD points to.

git symbolic-ref HEAD

How does commit-tree konw what the parent commit is?

The commit referred to by HEAD

Branches

The default branch is master

Master is created when you run git init

To create a new branch

git branch [branchname]

To switch to a different branch

git checkout [branchname]

Concepts

Terms

index

Another name for the staging area

Source of commits

state of your next commit

A file in the git directory in which changes to be committed in the next commit are stored

Index is git's knowledge of files

Patches

Patches allow application of a single commit / set of commits to another line of development

Patch collaboration doesn't depend on a branch structure.

Object id

Identifier for a git object

inode

In a unix filesystem, data structure representing a filesystem object

Working directory

The directory structure which relates somehow to a git repository

On first creating a repository, the working directory / tree is the same as the repository.

As time goes on, the working directory may differ from the object database.

If you add a file to the working tree, git will detect it as an untracked file

To track a new file, you add it to the index.

To save the file, you write the index as a tree object and commit the tree object

A checked out version of a project as represented in the file system

Clean working directory

State of the working direectory in which the file contents match the version stored in the git directory and to which no changes are stored in the staging area

Working tree

The tree object which will be referenced by the next commit.

Commit

Represents additions to a local git database

A commit represents a composed snapshot

In git, a commit represents a snapshot of the content staged (All unchanged and staged content), metadata about the author's name and email, and reference(s)to the commit(s) that came directly before this commit.

A commit may reference multiple parents when it represents the merging of two or more branches.

Reference to a tree object

Concepts

Rebase

Similar to merge.

Integrates changes from one branch into another branch

Moves all commits from one branch to the tip of another branch

Cherry-pick

Checksum

A datum which represents in a small fashion a signature for a piece of content. Changes in the content, even small ones, will generate a different checksum.

In git, both files and directories can be checksummed

Git determines changes based on checksums, not file contents.

A reference to stored content

Object database

TODO

Tracked

A file is tracked if it is in the last snapshot.

Changes

A change may either be staged or unstaged

Snapshot

A picture of the state of all files at a moment in time

Somewhat similar to a revision in svn.

A snapshot is represented in git represents the entire state of the workspace as opposed to a revision in svn, which only represents diffs between the state of one of more files following a commit.

Two snapshots containing a file that doesn't change both point to the same file (it's not stored twice)

File state

A file managed by git can be in one of three states: Committed, modified, staged

Committed means the file is stored in the database and has no changes.

Modified means the file is stored in the database and has unstaged changes

Staged means the file is stored in the database and has staged changes

Repository

Where git stores snapshots

Repositories may be either local or remote

A remote repository is stored on a remote location

Remote repositories may be accessed via a variety of protocols

Remote

A reference to a remote repository

When listing remotes, git will tell you which are read only vs read/write by annotating information with fetch for read-allowed and push for write-allowed

Origin

A reference to the repository from which a cloned repository was copied.

Local

TODO

Branch

A thread of development separate from other threads of development

Pointer to the head of a line of work

Creates a new named reference pointing to the SHA at which the branch was created.

see update-ref

Branches may be either local or remote

Branches organize an entire set of development off a parent branch point.

Head

Symbolic reference to the current branch (reference in refs folder)

Remote Tracking branch

Another name for upstream tracking branch

A local branch which is tied to a remote branch

TRacks the state of the local branch relative to the remote branch

Head branch

The branch on which active development is occurring

Fetch

Retrieves all the information in a remote repository not contained in the local repository yet doesn't

Merge

Commit object

TODO

Blob

TODO

Revision

TODO

Your branch is ahead of origin/master...

Means

Remote tracking

Currently active branch

Merge commit

A commit in which changes from one branch are applied to another branch

Has more than one parent commit

fast-forward

Where a branch is created off a master, work is done on it, and the branch is merged back to the master. If no work was performed on master in the meantime, just make master point to the next commit represented by the branch's HEAD.

Terms

porcelain - refers to more user-friendly mechanisms

plumbing - refers to lower-level things which are combined to create porcelain