**Git Command List**

**High-level commands (porcelain)**

**We separate the porcelain commands into the main commands and some ancillary user utilities.**

**Main porcelain commands**

git-add

Add file contents to the index

git-am

Apply a series of patches from a mailbox

git-archive

Create an archive of files from a named tree

git-bisect

Use binary search to find the commit that introduced a bug

git-branch

List, create, or delete branches

git-bundle

Move objects and refs by archive

git-checkout

Switch branches or restore working tree files

git-cherry-pick

Apply the changes introduced by some existing commits

git-citool

Graphical alternative to git-commit

git-clean

Remove untracked files from the working tree

git-clone

Clone a repository into a new directory

git-commit

Record changes to the repository

git-describe

Give an object a human readable name based on an available ref

git-diff

Show changes between commits, commit and working tree, etc

git-fetch

Download objects and refs from another repository

git-format-patch

Prepare patches for e-mail submission

git-gc

Cleanup unnecessary files and optimize the local repository

git-grep

Print lines matching a pattern

git-gui

A portable graphical interface to Git

git-init

Create an empty Git repository or reinitialize an existing one

gitk

The Git repository browser

git-log

Show commit logs

git-maintenance

Run tasks to optimize Git repository data

git-merge

Join two or more development histories together

git-mv

Move or rename a file, a directory, or a symlink

git-notes

Add or inspect object notes

git-pull

Fetch from and integrate with another repository or a local branch

git-push

Update remote refs along with associated objects

git-range-diff

Compare two commit ranges (e.g. two versions of a branch)

git-rebase

Reapply commits on top of another base tip

git-reset

Reset current HEAD to the specified state

git-restore

Restore working tree files

git-revert

Revert some existing commits

git-rm

Remove files from the working tree and from the index

git-shortlog

Summarize git log output

git-show

Show various types of objects

git-sparse-checkout

Initialize and modify the sparse-checkout

git-stash

Stash the changes in a dirty working directory away

git-status

Show the working tree status

git-submodule

Initialize, update or inspect submodules

git-switch

Switch branches

git-tag

Create, list, delete or verify a tag object signed with GPG

git-worktree

Manage multiple working trees

**Ancillary Commands**

**Manipulators:**

git-config

Get and set repository or global options

git-fast-export

Git data exporter

git-fast-import

Backend for fast Git data importers

git-filter-branch

Rewrite branches

git-mergetool

Run merge conflict resolution tools to resolve merge conflicts

git-pack-refs

Pack heads and tags for efficient repository access

git-prune

Prune all unreachable objects from the object database

git-reflog

Manage reflog information

git-remote

Manage set of tracked repositories

git-repack

Pack unpacked objects in a repository

git-replace

Create, list, delete refs to replace objects

**Interrogators:**

git-annotate

Annotate file lines with commit information

git-blame

Show what revision and author last modified each line of a file

git-bugreport

Collect information for user to file a bug report

git-count-objects

Count unpacked number of objects and their disk consumption

git-difftool

Show changes using common diff tools

git-fsck

Verifies the connectivity and validity of the objects in the database

git-help

Display help information about Git

git-instaweb

Instantly browse your working repository in gitweb

git-merge-tree

Show three-way merge without touching index

git-rerere

Reuse recorded resolution of conflicted merges

git-show-branch

Show branches and their commits

git-verify-commit

Check the GPG signature of commits

git-verify-tag

Check the GPG signature of tags

gitweb

Git web interface (web frontend to Git repositories)

git-whatchanged

Show logs with difference each commit introduces

**Interacting with Others**

These commands are to interact with foreign SCM and with other people via patch over e-mail.

git-archimport

Import a GNU Arch repository into Git

git-cvsexportcommit

Export a single commit to a CVS checkout

git-cvsimport

Salvage your data out of another SCM people love to hate

git-cvsserver

A CVS server emulator for Git

git-imap-send

Send a collection of patches from stdin to an IMAP folder

git-p4

Import from and submit to Perforce repositories

git-quiltimport

Applies a quilt patchset onto the current branch

git-request-pull

Generates a summary of pending changes

git-send-email

Send a collection of patches as emails

git-svn

Bidirectional operation between a Subversion repository and Git

Reset, restore and revert

There are three commands with similar names: git reset, git restore and git revert.

git-revert is about making a new commit that reverts the changes made by other commits.

git-restore is about restoring files in the working tree from either the index or another commit. This command does not update your branch. The command can also be used to restore files in the index from another commit.

git-reset is about updating your branch, moving the tip in order to add or remove commits from the branch. This operation changes the commit history.

git reset can also be used to restore the index, overlapping with git restore.

**Low-level commands (plumbing)**

**The interface (input, output, set of options and the semantics) to these low-level commands are meant to be a lot more stable than Porcelain level commands, because these commands are primarily for scripted use. The interface to Porcelain commands on the other hand are subject to change in order to improve the end user experience.**

**The following description divides the low-level commands into commands that manipulate objects (in the repository, index, and working tree), commands that interrogate and compare objects, and commands that move objects and references between repositories.**

**Manipulation commands**

git-apply

Apply a patch to files and/or to the index

git-checkout-index

Copy files from the index to the working tree

git-commit-graph

Write and verify Git commit-graph files

git-commit-tree

Create a new commit object

git-hash-object

Compute object ID and optionally creates a blob from a file

git-index-pack

Build pack index file for an existing packed archive

git-merge-file

Run a three-way file merge

git-merge-index

Run a merge for files needing merging

git-multi-pack-index

Write and verify multi-pack-indexes

git-mktag

Creates a tag object with extra validation

git-mktree

Build a tree-object from ls-tree formatted text

git-pack-objects

Create a packed archive of objects

git-prune-packed

Remove extra objects that are already in pack files

git-read-tree

Reads tree information into the index

git-symbolic-ref

Read, modify and delete symbolic refs

git-unpack-objects

Unpack objects from a packed archive

git-update-index

Register file contents in the working tree to the index

git-update-ref

Update the object name stored in a ref safely

git-write-tree

Create a tree object from the current index

**Interrogation commands**

git-cat-file

Provide content or type and size information for repository objects

git-cherry

Find commits yet to be applied to upstream

git-diff-files

Compares files in the working tree and the index

git-diff-index

Compare a tree to the working tree or index

git-diff-tree

Compares the content and mode of blobs found via two tree objects

git-for-each-ref

Output information on each ref

git-for-each-repo

Run a Git command on a list of repositories

git-get-tar-commit-id

Extract commit ID from an archive created using git-archive

git-ls-files

Show information about files in the index and the working tree

git-ls-remote

List references in a remote repository

git-ls-tree

List the contents of a tree object

git-merge-base

Find as good common ancestors as possible for a merge

git-name-rev

Find symbolic names for given revs

git-pack-redundant

Find redundant pack files

git-rev-list

Lists commit objects in reverse chronological order

git-rev-parse

Pick out and massage parameters

git-show-index

Show packed archive index

git-show-ref

List references in a local repository

git-unpack-file

Creates a temporary file with a blob’s contents

git-var

Show a Git logical variable

git-verify-pack

Validate packed Git archive files

In general, the interrogate commands do not touch the files in the working tree.

**Syncing repositories**

git-daemon

A really simple server for Git repositories

git-fetch-pack

Receive missing objects from another repository

git-http-backend

Server side implementation of Git over HTTP

git-send-pack

Push objects over Git protocol to another repository

git-update-server-info

Update auxiliary info file to help dumb servers

The following are helper commands used by the above; end users typically do not use them directly.

git-http-fetch

Download from a remote Git repository via HTTP

git-http-push

Push objects over HTTP/DAV to another repository

git-receive-pack

Receive what is pushed into the repository

git-shell

Restricted login shell for Git-only SSH access

git-upload-archive

Send archive back to git-archive

git-upload-pack

Send objects packed back to git-fetch-pack

**Internal helper commands**

These are internal helper commands used by other commands; end users typically do not use them directly.

git-check-attr

Display gitattributes information

git-check-ignore

Debug gitignore / exclude files

git-check-mailmap

Show canonical names and email addresses of contacts

git-check-ref-format

Ensures that a reference name is well formed

git-column

Display data in columns

git-credential

Retrieve and store user credentials

git-credential-cache

Helper to temporarily store passwords in memory

git-credential-store

Helper to store credentials on disk

git-fmt-merge-msg

Produce a merge commit message

git-interpret-trailers

Add or parse structured information in commit messages

git-mailinfo

Extracts patch and authorship from a single e-mail message

git-mailsplit

Simple UNIX mbox splitter program

git-merge-one-file

The standard helper program to use with git-merge-index

git-patch-id

Compute unique ID for a patch

git-sh-i18n

Git’s i18n setup code for shell scripts

git-sh-setup

Common Git shell script setup code

git-stripspace

Remove unnecessary whitespace