

# Magnolia Git Cheat Sheet

- Configuration
  - Global Settings
  - Aliases
- Commit Code Changes
  - git add
  - git commit
- Display Changes and Log
- Branches, Merges and Tags
  - Branches
  - Merges
  - Tags
- Repositories Push, Fetch, Pull
  - Initialize Repository
  - Difference Fetch / Pull
  - Rebasing
- Undo and Recovery
  - Restore accidentally deleted files
  - Other Restore Commands
- Misc
  - Stashing
  - Other commands
- Configure Specific Repositories
  - SSH
  - Github Repository
  - Bitbucket Repository
- Git Resources

## Configuration

### Global Settings

Action	Command
<i>show current configuration</i>	<code>git config --global --list</code>
<i>command line colors</i>	<code>git config --global color.ui auto</code>
	<code>git config --global color.branch.auto</code>
<i>LF (LineFeed) on all platforms:</i>	<code>git config --global core.autocrlf input</code>
<i>use TextMate as default editor:</i>	<code>git config --global core.editor "mate -w"</code>
<i>use Sublime as default editor:</i>	<code>git config --global core.editor "subl -w"</code>

### Aliases

examples:

```
git config --global alias.s "status"  
git config --global alias.l "log --stat"  
git config --global alias.lpo "log --pretty=oneline --abbrev-commit --graph --decorate"
```

```
git config --global alias.sync "!git pull && git push"
```

(shell command begins with '!')

example configurations from `.gitconfig`

```
[alias]
s = status
lpo = log --pretty=oneline --abbrev-commit --graph --decorate
orphank = !gitk --all `git reflog | cut -c1-7`&
hist = log --pretty=format:@"%h %ad | %s%d [%an]" --graph --date=short
```

## Commit Code Changes

### git add

Action	Command
<i>split changes into several commits</i>	<b>git add . -p</b>
<i>accept deleted into staging</i>	<b>git add -u .</b>
<i>shortcut to accept all changes</i>	<b>git add -A .</b>

### git commit

Action	Command
<i>verbose display when committing</i>	<b>git commit -v</b>
<i>shortcut for commit message</i>	<b>git commit -a -m"This is my great message"</b>
<i>change last commit message while in staging</i>	<b>git commit --amend</b>

## Display Changes and Log

Action	Command
<i>show log</i>	<b>git log</b>
<i>last commit of current branch</i>	<b>git diff HEAD</b>
<i>display log with rename and copy</i>	<code>git log --stat --C (for extended display use --find-copies-harder--)</code>
<i>difference to a specific repository</i>	<b>git diff HEAD remote/name/master</b>
<i>difference to a branch</i>	<b>git diff master feature1</b>
<i>show branch</i>	<b>git show-branch</b>
<i>display executed commands</i>	<b>git reflog</b>
<i>view local repository in browser</i>	<b>git instaweb --httpd webrick</b>
<i>tool to visualize git</i>	<b>gitk --all&amp; (all branches)</b>

## Branches, Merges and Tags

### Branches

Action	Command
<i>show current branch</i>	<b>git branch</b>
<i>show all branches (including remote branches)</i>	<b>git branch -a</b>
<i>non-merged branches</i>	<b>git branch --no-merged</b>
<i>merged branches</i>	<b>git branch --merged</b>
<i>create branch</i>	<b>git branch name</b>
<i>go to a specific branch (also tags)</i>	<b>git checkout branchname</b>
	<b>git checkout master</b>
<i>shortcut for creating a branch and checking it out</i>	<b>git checkout -b branchname</b>
<i>push branch changes</i>	<b>git push -u pushable branchname</b>
<i>get branch from repository</i>	<b>git checkout remotes/origin/branchname</b>
<i>change branch number of steps</i>	<b>git checkout master^^</b> (go back two steps)

## Merges

**git checkout master**  
**git merge branchname**

## Tags

Action	Command
<i>show tags</i>	<b>git tag -l</b>
<i>create tag</i>	<b>git tag tagname</b>
<i>remove specific tag</i>	<b>git tag -d tagname</b>
<i>tag with message</i>	<b>git tag -a tagname</b>
<i>push tags to (remote) repository</i>	<b>git push --tags (origin master) / git push (origin master) tagname / git push --all origin master</b>
<i>remove tag from (remote) repository</i>	<b>git push repo :tagname</b>

## Repositories Push, Fetch, Pull

### Initialize Repository

create a directory and init with

```
git init
git add .
git commit
```

(git remote / git remote -v)

*add repository to configuration*

```
git remote add reponame https://schniebiwutz.git  
  
(goes to .git/config)
```

**git pull/fetch reponame**  
**git diff HEAD remotes/reponame/master**

## Difference Fetch / Pull

**git pull**: combination of **git fetch** and **git merge**, pulls and merges all changes to all your files

**git fetch**: fetches changes from the repository but does **NOT** update files changed locally

## Rebasing

Several commits can be combined to one using the **rebase** command. Use this to clean up the commit history before pushing changes to a remote repository.

**git checkout myfeaturebranch**  
**git rebase master**

## Undo and Recovery

### Restore accidentally deleted files

*display deleted files*

**git ls-files -d**

*restore the deleted files*

**git ls-files -d | xargs git checkout -**

### Other Restore Commands

**git reset HEAD first.java**

*revert to last commit / version*

**git reset --hard**  
**git reset --hard 78827**

*discard last commit but keep code*

**git reset --soft**

*discard local changes*

**git remote prune name (origin)**

*go back to a specific version*

**git revert 2F66a**

*clean commands*

**git clean -n**  
**git clean -fd**

*emergency rescue*

**git reflog**  
**git reset --hard 2727 (taken from reflog, if not already killed by GC)**

*edit last commit message*

**git commit --amend**

*add note*

**git notes add master^^^^**

## Misc

### Stashing

*stashing is only local*

**git stash**

**git stash list**

**git stash apply** (this applies and leaves the changes in the stack as opposed to "pop")

**git stash pop (back)**

**git stash branch newbranch**

**git stash branch newbranch stashname**

### Other commands

*check all hashes*

**git fsck**

*check who created a file*

**git blame filename**

## Configure Specific Repositories

### SSH

Take the content of the file

**.ssh/id-rsa.pub**

and paste it to the repository configuration.

**Test SSH connections:**

**Github: ssh git@github.com**

**Gitolite: ssh gitolite@localhost**

*get a verbose protocol by using*

**ssh -v**

### Github Repository

```
git remote add origin git@github.com:doofenschmerz/schniedel.git
git push origin master
```

### Bitbucket Repository

```
git remote add origin git@bitbucket.org:doofenschmerz/wutz.git
git push origin master
```

## Git Resources

- <http://delicious.com/matthew.mccullough/git>
- <http://bit.ly/gitlinks>
- <http://github.com/matthewmccullough/git-workshop>
- <http://bit.ly/gitbashshell>
- <http://bit.ly/gitcombinedshell> (bash + zsh)
- <http://git.springsource.org>
- Lars Vogella's Git tutorial
- A few Git tips you didn't know about
- Skipping locally updated files with Git
- Git aliases to ease cherry picking
- Git from the bottom up