**GIT - Useful Commands**

**Version 2.1**

**September 16, 2016**

# Table of Contents:

[1 Table of Contents: 2](#_Toc532039561)

[2 Overview 2](#_Toc532039562)

[3 Abbreviations and terminology 2](#_Toc532039563)

[4 Git Simple Commands 3](#_Toc532039564)

[4.1 Add 3](#_Toc532039565)

[4.2 Branch 3](#_Toc532039566)

[4.3 Checkout 3](#_Toc532039567)

[4.4 Clone 3](#_Toc532039568)

[4.5 Config – basic 3](#_Toc532039569)

[4.6 Config – Beyond Compare 3](#_Toc532039570)

[4.7 Config – kdiff3 3](#_Toc532039571)

[4.8 Config – proxy for repo not on local network 4](#_Toc532039572)

[4.9 Config – unset 4](#_Toc532039573)

[4.10 “git config --Config – Verify 4](#_Toc532039574)

[4.11 Diff 4](#_Toc532039575)

[4.12 Fetch 4](#_Toc532039576)

[4.13 Git utilities 4](#_Toc532039577)

[4.14 Push 4](#_Toc532039578)

[4.15 Remote 4](#_Toc532039579)

[4.16 Tag 4](#_Toc532039580)

[5 Git Sequence of Commands 5](#_Toc532039581)

[5.1 Commit Changes 5](#_Toc532039582)

[5.2 Merge 5](#_Toc532039583)

[5.3 Mirror 1st Option 5](#_Toc532039584)

[5.4 Mirror 2nd Option 5](#_Toc532039585)

[5.5 Reset Local Branch 6](#_Toc532039586)

[5.6 Remove Branch from Your Remote 6](#_Toc532039587)

[6 Version History: 6](#_Toc532039588)

# Overview

This is simply a list of useful GIT commands.

# Abbreviations and terminology

* Git - is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency

# Git Simple Commands

## Add

* + “git add <file\_name>” – Adds file to index
  + “git add .” – adds all files to index

## Branch

* “git branch” – shows all branches on local, \* indicates current branch
* “git branch -a” – shows all branches on all remotes that have been fetched
* “git branch -a | grep [CWSID or "msXXXX"]” – filter branches based on remote name, msa or other text
* “git remote show <remote\_name> – show branch on a remote
* “git branch -m <new\_name>” – renames current branch
* “git branch -m <old\_name> <new\_name>” – renames other branch
* “git branch -d <local\_branch>” – deletes branch
* Delete remote branch – see section 5.4

## Checkout

* “git checkout <branch\_name>” – checks out existing branch
* “git checkout -b <new\_branch> <start\_point>” – creates a new branch starting branch or tag <start\_point>
* “git checkout <tag\_name>” – checks out existing tag, can not edit
* “git checkout <remote\_name/branch\_name>” – checks out remote branch without creating a local branch, can not edit

## Clone

* “git clone https://github.com/mkontz/Kontz-How-To.git” – clone repo

## Config – basic

* “git config --global user.name “John Doe”
* “git config --global user.email [Doe\_John@gmail.com](mailto:Doe_John@gmail.com)”

## Config – Beyond Compare

* “git config --global merge.tool bc3” – sets beyond compare 3 or newer as mergetool
* “git config --global mergetool.bc3.trustExitCode true”
* “git config --global diff.tool bc3” – sets beyond compare 3 or newer as difftool
* “git config --global difftool.bc3.trustExitCode true”

## Config – kdiff3

* “git config --global merge.tool kdiff3” – sets kdiff3 as mergetool
* “git config --global diff.tool kdiff3” – sets kdiff3 as mergetool

## Config – proxy for repo not on local network

* “git config --global http.proxy http://proxy.misc-company.com:80”
* “git config --global https.proxy https://proxy. misc-company.com:80”
* “git config --global https.sslverify false”
* “git config --global http.sslverify false”

## Config – unset

* “git config --global --unset <field>” – unsets/removes field

## “git config --Config – Verify

* “git config -l --global” show all global configs
* “git config --get <field>” – gets current field

## Diff

* “git difftool <other-branch/tag> <path\_relative\_to\_repo>/<file-name>” – compares file on HEAD to same file from branch or tag
* “git difftool <branch/tag-1> <branch/tag-1> < path\_relative\_to\_repo >/<file-name>” – compares file on two arbitrary branches/tags

## Fetch

* “git fetch –all” – fetch from all repos
* “git fetch <repo\_name>” – fetch from single repo

## Git utilities

* “git gui” – opens useful utility
* “gitk” – open useful utility

## Push

* “git push <repo\_name> HEAD” – push current branch, HEAD could be replace with other branch
* “git push <repo\_name> --all” – push all branch from current repo to <repo\_name>
* “git push <repo\_name> --tags” – push all tags from current repo to <repo\_name>

## Remote

* “git remote add <repo\_name> https://github.com/mkontz/Kontz-How-To.git”
* “git remote –v” – shows all remotes defined on local repo
* “git remote rm <remote\_name> – removes remote from local repo
* “git remote show <remote\_name> – show branch on a remote

## Tag

* “git tag” – shows all tags
* “git tag –a <new\_tag\_name> -m ‘comment’ ” – create a new tag
* “git push <remote\_name> <tag\_name” – push tag to remote

# Git Sequence of Commands

## Commit Changes

Option 1: (command line)

* Step 1: “git add .” – adds new files to index
  + Step 2: “git commit – a –m ‘your comments here’ ” – commits with comments

Option 2: (preferred)

* + Use “git gui” utility, preferred method since allows visual inspection of files and files included/changed

## Merge

Setup/Install Merge Tool in GIT

* + Step 1: Use Synaptic to install merge tool of choice
  + Step 2: config merge tool (see section 4.5)

Trivial Merge

* Step 1: commit files (see section 5.1)
* Step 2: “git merge <other\_branch>” – merges other branch into current branch

Merge – without automatic commit

* Step 1: commit files (see section 5.1)
* Step 2: “git merge <other\_branch> --no-commit --no-ff” – merges other branch into current branch

Non-trivial Merge

* Step 1: set/install if needed (see above)
* Attempt merge without automatic commit (see above)
* If trivial merge fails due to conflicts then type: “git mergetool”

## Mirror 1st Option

* Step 1: “git fetch <origin\_repo>” – fetches tags and branches from origin repo Step 1:
* Step 2: “git push <repo\_name> --all” – push all branch from current repo to <repo\_name>
* Step 3: “git push <repo\_name> --tags” – push all tags from current repo to <repo\_name>

## Mirror 2nd Option

* Step 1: “git fetch <origin\_repo>” – fetches tags and branches from origin repo
* Step 2: “git push <mirror\_repo> +refs/remotes/<origin\_repo>/\*:refs/heads/\* +refs/tags/\*:refs/tags/\*” – push all origin branches and tags to mirror repo

## Reset Local Branch

* Step 1: “git reset --hard HEAD” – reverts to previous commit, then do next two commands
  + Step 2: “git clean --df”
  + Step 3: “git checkout -- .”

## Remove Branch from Your Remote

* Step 1: “git branch –rd <your\_remote>/<branch\_name>” – removes local reference to remote branch
  + Step 2: “git push <your\_remote> --delete <branch\_name>” – removes branch from remote
  + Step 3: “git fetch <your\_remote>” – fetch remote
  + Step 4: Verify that remote branch is gone

# Version History:

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Description** |
| 1.0 | February 5, 2015 | Matt Kontz | Initial version compiled from matt’s notebook |
| 1.1 | February 17, 2015 | Matt Kontz | Added commit sequence and remote show repo |
| 1.2 | April 8, 2015 | Matt Kontz | Added directing on using and setup up kdiff3 for merging, updated commit section |
| 1.3 | April 10, 2015 | Matt Kontz | Added direction for merge without automatic commit and directions for diffing files on two branch |
| 1.4 | April 13, 2015 | Matt Kontz | Added direction on how to filter git branch all. Added direction on how to create and push a tag. |
| 1.5 | April 29, 2015 | Matt Kontz | Updated config section and added direction on delete a local branch. |
| 1.6 | July 13, 2015 | Matt Kontz | Checking out current tag |
| 1.7 | July 27, 2015 | Matt Kontz | Checking out remote branch |
| 1.8 | March 23, 2016 | Matt Kontz | Directions on setting git proxy settings and showing all global configs |
| 1.9 | July 15, 2016 | Matt Kontz | Added directions for adding Beyond Compare to git config |
| 2.0 | Sept. 16, 2016 | Matt Kontz | Added additional sequence for mirroring repo. |
| 2.1 | Dec 8, 2018 | Matt Kontz | Add to Kontz-How-To repo, misc. clean up |