Git command	Hg command	Notes
`git pull`	`hg fetch` `hg pull -u`	The fetch command is more similar but requires the [FetchExtension](#) to be enabled.
`git fetch`	`hg pull`	
`git push`	`hg push -r .`	By default, git only pushes the current branch.
`git checkout <commit>`	`hg update -c <cset>`	git checks and reloads (accidentally) removed files
`git checkout [<rev>] -- <file(s)>`	`hg revert [-r <rev>] <file(s)>`	
`git reset --hard`	`hg revert -a --no-backup`	
`git reset --hard HEAD~1`	`hg strip -r .`	
`git revert <commit>`	`hg backout <cset>`	
`git add <new_file>`	`hg add <new_file>`	Only equivalent when `<new_file>` is not tracked.
`git add <file>` `git reset HEAD <file>`	—	Not necessary in Mercurial (see `shelve` below for partial commit support).
`git add -i`	`hg record`	Requires the [RecordExtension](#) to be enabled.
`git commit --amend`	`hg commit --amend`	
`git rebase --interactive`	`hg histedit <base cset>`	Requires the [HisteditExtension](#). In core since version 2.3
`git stash`	`hg shelve`	Requires the [ShelveExtension](#) or the [AtticExtension](#).
`git merge`	`hg merge`	git merge is capable of octopus merges, while mercurial merge prefers multiple merges
`git cherry-pick <commit>`	`hg transplant <cset>` `hg graft <csets>`	Transplant requires the [TransplantExtension](#). Graft is available in 2.0 and higher.
`git rebase <upstream>`	`hg rebase -d <cset>`	Requires the [RebaseExtension](#).
`git format-` `patch <commits>` and `git send-mail`	`hg email -r <csets>`	Requires the [PatchbombExtension](#).
`git am <mbox>`	`hg mimport -m <mbox>`	Requires the [MboxExtension](#) and the [MqExtension](#). Imports patches to mq.

`git describe`	`hg log -r . --`
	`template '{latesttag}-`
	`{latesttagdistance}-{node|short}\n'`	

`git describe rev`	`hg log -r rev --template '{latesttag}-{latesttagdistance}-{node|short}\n'`	
`git log origin..HEAD` `git log origin/foobranch..HEAD`	`hg outgoing`	
`git fetch && git log HEAD..origin`	`hg incoming`	`git fetch` keeps the changesets while `hg incoming` (without `--bundle foo`) discards them. Use `git pull` (will fetch further changes) or `git merge origin` to update the working directory
`git show rev`	`hg export rev`	
`git show hash:file`	`hg cat -r rev file`	
`git ls-files`	`hg manifest`	
`git log`	`hg log`	
`git log -n`	`hg log --limit n`	
`git ??`	`hg summary`	
`git status`	`hg outgoing` `hg status`	
`git remote add -f remotename url`	—	Edit .hg/hgrc and add the line 'remotename = url' under section '[paths]'; see below for getting changesets
`git remote update remotename`	`hg pull remotename`	When `remotename` is omitted in Git, all remotes are updated. In Mercurial, the default remote is refreshed.
`git branch -a`	`hg branches`	
`git config --global user.(name|email) ...`	—	Edit `~/.hgrc` section "[ui]", key "username", value "First Last < mail@example.org >"
`git clean` or `git status --porcelain|sed -r 's:\?\?\s(.*):\1:g'|xargs rm`	`hg purge` or `hg status -un|xargs rm`	`purge` requires the PurgeExtension In Windows you might need to add `sed 's:\\:/:g'` before piping `xargs rm`, otherwise the inverted slash in Windows paths will be interpreted as an escape