

<code>git add -i</code>	<code>hg record</code>	Requires the RecordExtension to be enabled.
<code>git add <file></code> <code>git reset HEAD <file></code>	Not necessary in Mercurial (see Shelve below for partial commit support).	
<code>git add <new_file></code>	<code>hg add <new_file></code>	Only equivalent when <code><new_file></code> is not tracked.
<code>git am <mbox></code>	<code>hg mimport -m <mbox></code>	Requires the MboxExtension and the MqExtension . Imports patches to mq.
<code>git branch -a</code>	<code>hg branches</code>	
<code>git checkout <commit></code>	<code>hg update -c <cset></code>	git checks and reloads (accidentally) removed files
<code>git checkout <rev> -- <file(s)></code>	<code>hg revert [-r <rev>] <file(s)></code>	
<code>git cherry-pick <commit></code>	<code>hg transplant <cset></code> <code>hg graft <csets></code>	Transplant requires the TransplantExtension . Graft is available in 2.0 and higher.
<code>git clean</code> or <code>git status --porcelain sed -r 's:\?\?\s(.*):\1:g' xargs rm</code>	<code>hg purge</code> or <code>hg status -un xargs rm</code>	<code>purge</code> requires the PurgeExtension . In Windows you might need to add <code>sed 's:\\\:/:g'</code> before piping <code>xargs rm</code> , otherwise the inverted slash in Windows paths will be interpreted as an escape
<code>git commit --amend</code>	<code>hg commit --amend</code>	
<code>git config --global</code>	Edit <code>~/.hgrc</code> section "[ui]", key "username", value "First Last < mail@example.org >"	
<code>git fetch</code>	<code>hg pull</code>	
<code>git fetch && git log HEAD..origin</code>	<code>hg incoming</code>	<code>git fetch</code> keeps the changesets while <code>hg incoming</code> (without <code>--bundle foo</code>) discards them. Use <code>git pull</code> (will fetch further changes) or <code>git merge origin</code> to update the working directory
<code>git format-patch <commits></code> and <code>git send-mail</code>	<code>hg email -r <csets></code>	Requires the PatchbombExtension .
<code>git log</code>	<code>hg log</code>	
<code>git log -n</code>	<code>hg log --limit n</code>	
<code>git log origin..HEAD,</code> <code>git log origin/foobranch..HEAD</code>	<code>hg outgoing</code>	
<code>git ls-files</code>	<code>hg manifest</code>	
<code>git ls-remote <url> HEAD</code>	<code>hg identify <url></code>	
<code>git merge</code>	<code>hg merge</code>	<code>git merge</code> is capable of octopus merges, while mercurial merge prefers multiple merges
<code>git pull</code>	<code>hg fetch</code> <code>hg pull -u</code>	The <code>fetch</code> command is more similar but requires the FetchExtension to be enabled.
<code>git push</code>	<code>hg push -r .</code>	By default, git only pushes the current branch.
<code>git rebase --interactive</code>	<code>hg histedit <base cset></code>	Requires the HisteditExtension . In core since version 2.3
<code>git rebase <upstream></code>	<code>hg rebase -d <cset></code>	Requires the RebaseExtension .
<code>git remote add -f remotename url</code>	Edit <code>.hg/hgrc</code> and add the line <code>'remotename = url'</code> under section <code>[paths]</code> ; see below for getting changesets	
<code>git remote update remotename</code>	<code>hg pull remotename</code>	When <code>remotename</code> is omitted in Git, all remotes are updated. In Mercurial, the default remote is refreshed.
<code>git reset --hard</code>	<code>hg revert -a --no-backup</code>	
<code>git reset --hard HEAD~1</code>	<code>hg strip -r .</code>	
<code>git rev-parse HEAD</code>	<code>hg identify</code>	
<code>git revert <commit></code>	<code>hg backout <cset></code>	
<code>git show hash:file</code>	<code>hg cat -r rev file</code>	<code>Hg status --change <rev></code>
<code>git show rev</code>	<code>hg export rev</code>	
<code>git stash</code>	<code>hg shelve</code>	Requires the ShelveExtension or the AtticExtension .
<code>git status</code>	<code>hg outgoing</code> <code>hg status</code>	

Credit to <https://web.archive.org/web/20150204065617/http://mercurial.selenic.com/wiki/GitConcepts> for most of the table :)