

Git Reference: The Fundamentals

Important (Frequent) Commands

Command	Effect
<code>module load git</code>	Load a recent version of Git on CHPC resources
<code>git help <i>command</i></code>	Read more about the operation <i>command</i>
<code>git init</code>	Create a Git repository in the current directory (if one isn't already present)
<code>git clone <i>source</i> [<i>destination</i>]</code>	Create a Git repository with the contents of <i>source</i> in the directory <i>destination</i> ; will inherit directory name if unspecified
<code>git remote add <i>name</i> <i>location</i></code>	Add a remote <i>name</i> that points to (the URL) <i>location</i>
<code>git config [--global] user.name "<i>name</i>"</code>	Change your name to <i>name</i> ; omit <i>--global</i> to change only in the current repository
<code>git config [--global] user.email "<i>email</i>"</code>	Change your email address to <i>email</i> ; omit <i>--global</i> to change only in the current repository; this is visible to all who can view the repository
<code>git status</code>	Display the status of the repository (modified, unmodified, and staged files)
<code>git add <i>file</i></code>	Add <i>file</i> (or a list of files separated by spaces) to the staging area
<code>git rm <i>file</i></code>	Remove <i>file</i> from the repository; add <i>--cached</i> to remove only from staging area
<code>git clean -i</code>	Remove (delete) untracked files; <i>-i</i> does this interactively; be careful
<code>git commit -m "<i>message</i>"</code>	Create a commit with description <i>message</i> (if <i>-m</i> is omitted, a text editor will open to accept a message)
<code>git pull <i>remote branch</i></code>	Pull <i>branch</i> from <i>remote</i> into the current branch
<code>git push <i>remote branch</i></code>	Push the current branch to <i>branch</i> of <i>remote</i>
<code>git checkout [-b] <i>branch</i></code>	Switch to <i>branch</i> ; if it does not exist, include <i>-b</i> to create it; this will discard changes that have not been committed
<code>git branch</code>	Display available branches, marking the current branch
<code>git branch -d <i>branch</i></code>	Delete <i>branch</i>
<code>git merge <i>branch</i></code>	Merge commits in <i>branch</i> into the current branch
<code>git diff</code>	Display the changes to files since the last commit; can be used to compare objects
<code>git log [--stat] [--summary]</code>	Display an overview of the project history: commit messages and authors
<code>git show <i>object</i></code>	Display information about <i>object</i> (a commit, tag, or tree, for instance)
<code>git tag -a <i>version</i> -m "<i>message</i>" [<i>commit</i>]</code>	Create a tag <i>version</i> (e.g. <i>v1.2</i>) with associated text <i>message</i> ; optionally apply to a specific <i>commit</i> checksum; omit arguments to view all tags
<code>git revert <i>commit</i></code>	Revert to <i>commit</i> by creating a new commit

Important Files

File	Purpose
<code>.gitignore</code>	Selectively ignore files (with patterns)
<code>README</code>	Explain the project and provide high-level documentation
<code>LICENSE</code>	Provide information about the license of the repository's contents
<code>CITATION</code>	Provide information about citing the project