

ITK Git Reference

One-time Global Setup

| | | |
|-------------|--|--|
| SETUP USER | <code>git config --global user.name "Jane Doe"</code> | Use your full name. |
| INFORMATION | <code>git config --global user.email "jane@doe.org"</code> | |
| | <code>git config --global color.ui auto</code> | Enable color output in Git (optional). |

Create and Configure Your Repository

| | | |
|------------|---|--|
| CLONE | <code>git clone https://github.com/InsightSoftwareConsortium/ITK</code> | |
| REPOSITORY | <code>cd ITK</code> | |
| CONFIGURE | <code>./Utilities/SetupForDevelopment.sh</code> | Follow prompts, sets up local hooks, remotes, push URLs etc. |

Topic Branch Lifecycle

| | | |
|----------------------|--|---|
| UPDATE | <code>git status</code> <code>git checkout master</code> <code>git pull --rebase upstream master</code> | Review local modifications, delete, stash or commit them. |
| NEW TOPIC BRANCH | <code>git checkout -b my-topic upstream/master</code> | Create branch off master. |
| WRITE CODE | <code>vim MyExistingFile.cxx MyNewFile.cxx</code> | Use your favorite editor. |
| NEW COMMIT | <code>git add MyExistingFile.cxx MyNewFile.cxx</code> <code>git commit</code> | Stage changes to existing and new files. Commit changes locally. |
| AMEND COMMIT | <code>vim MyExistingFile.cxx</code> <code>git add MyExistingFile.cxx</code> <code>git commit --amend</code> | Edit the file. Stage changes. Amend previous commit locally. |
| ADD TESTING DATA | <code>git data-upload DataFile.png</code> <code># Add DATA{DataFile.png} to CMakeLists.txt</code> <code>git add -- DataFile.png.cid CMakeLists.txt</code> <code>git commit --amend</code> | Upload a testing data input or baseline image to content-link-upload.itk.org . Reference the content link in the CMake configuration. Add the content link and updated CMake configuration. |
| SUBMIT FOR REVIEW | <code>git prepush</code> <code>git review-push</code> | Check what will be pushed to GitHub. Push the topic to GitHub for review. Enter the URL returned in your browser, and review and open the pull request. |
| RESPOND TO REVIEW | <code>git rebase -i HEAD~3</code> <code>git prepush</code> <code>git review-push --force</code> | Modify local commits (3 back here). Check what will be pushed to Gerrit. Push the updated topic to Gerrit. |
| DELETE TOPIC | <code>git fetch upstream</code> <code>git branch -d my-topic</code> | Fetch the latest changes from origin. Delete topic, fails if not merged. |

Review

| | | |
|----------------------|---|--|
| PUSH FOR REVIEW | <code>git review-push</code> | Push the current topic branch for review. |
| REVIEW PULL REQUESTS | <code>git pr <pull-request-number></code> | Fetch and check out by number a pull request locally for review. |
| CLEAN PULL REQUESTS | <code>git pr-clean</code> | Clean locally fetched pull requests. |

General Guidelines

| | | |
|----------------|---|--|
| COMMIT MESSAGE | <code><short-description></code> <code><long description></code> <code>Issue #1234</code> | Should be less than 78 characters. Blank line after short description. Could be several paragraphs. GitHub issue number, if applicable. |
| TOPIC NAMING | <code>summary-of-topic</code> | Lower case, separated by dashes. |

For more information, see git-scm.com.