How to Contribute Code to ROOT

==============================

Thank you for your interest in contributing to ROOT! We strongly welcome and appreciate such contributions!

This short guide aims to provide hints and pointers to making the process as quick and painless as possible.

ROOT Source Code

----------------

The source code for ROOT is kept in [GitHub](http://github.com/root-project/root) and follows mainly the pull-request model;

the primary branch for development is `master`.

Visit [this page](https://root.cern/for\_developers/creating\_pr) for more details on how to

create pull requests.

Once a PR is created, a member of the ROOT team will review it as quickly as possible. If you are familiar with the

ROOT community, it may be beneficial to add a suggested reviewer to the PR in order to get quicker attention.

The ROOT codebase roughly follows the [Taligent naming conventions](https://root.cern.ch/coding-conventions). Please

mind the coding conventions, as this is a simple item for reviewers to otherwise get stuck on.

Testing

-------

While there are several levels of automated tests that will be triggered before a PR is integrated into `master`, we

strongly encourage new developers to run the tests \_before\_ submitting new code. [Please follow the testing docs](https://root.cern/for\_developers/run\_the\_tests/)

for more information.

Continuous Integration

----------------------

ROOT currently has two automated CI tests that are used for pull requests:

- \*Build and test\*: When known contributors submit a PR, a Jenkins-based CI workflow will kick off; a project member will need to initiate the build; the results will be posted to the pull request. For project members, interacting with the Jenkins instance is [documented here](https://github.com/phsft-bot/build-configuration/blob/master/README.md). Be aware that the build platforms and tests done for each PR are a subset of those done for nightly builds.

- \*Formatting check\*: We utilize a Travis-CI-based build to automatically check that a PR follows known ROOT coding conventions. This is done with the `clang-format` tool. If coding violations are found, `clang-format` will generate a suggested patch file; we suggest utilizing this output directly

- \*Simple Static Analysis\*: The Travis-CI build will run a simple set of static analyses using `clang-tidy`.

Typically, every PR must pass all steps; we will ask you to fix any issues that arise. Committers have the ability to override this requirement.