

Contributing

The Secure Device Onboard projects welcomes all contributions. Prospective contributors are encouraged to read the project Code of Conduct and Contributors Guide

License and Copyrights¶

Contributions must be submitted under the Apache License 2.0. Include the Apache License header or SPDX header in every source file you contribute. See the following for specific guidance:

- <https://www.apache.org/licenses/LICENSE-2.0#apply>
- <https://spdx.org/licenses/Apache-2.0.html>

Note that any 3rd-party dependencies required by your code to execute must be available under the Apache 2.0 license. Exceptions should be discussed in advance by submitting an issue for review.

A copyright notice is permitted but not required. A copyright notice should appear at the top of all project artifacts (code files, configuration files, documentation, etc.). The copyright notice should identify the contributing entity (person or corporation).

Note that when subsequent contributors make substantive changes to a file, they may also optionally add a copyright notice for themselves (person or corporation), but should preserve unmodified any existing copyright notices.

Code Contributions¶

If you are contemplating a proposing a major contribution to the Secure Device Onboard project, please first submit an issue outlining your proposed changes or contribution, or contact the development community via Slack or mailing list. This will allow the community to provide feedback on the proposal and ensure that your proposal is in alignment with project goals and processes.

Major contributions are contributions that meet any of the following criteria:

- Change the application architecture
- Require new repositories be added to the GitHub Secure Device Onboard project, or require existing repositories to be re-organized
- Change, add or remove code across multiple repositories

You are welcome to submit incremental contributions directly by way of pull request in the project GitHub repository. You are also welcome to raise an issue, or discuss the contribution in Slack or via mailing list.

Incremental contributions are contributions typically meet the following following criteria:

- Do not change the architecture in any way
- Do not require the addition, removal, or reorganization of any project repository
- Affect only a single or limited number of repositories

Code Style¶

The Secure Device Onboard project has adopted Google Style Guide for Java for all code implemented in the Java programming language. **Note** that Java code style is enforced by build verification test run at the time pull requests are submitted using the [checkstyle tool](#). Pull requests that do not pass the style check will not be approved for merge.

The Secure Device Onboard project has adopted the Linux Kernel Style (with a few exceptions) for the Client SDK. Check with a client SDK developer if you have questions about style in this component.

Commit Process¶

This is the recommended process for submitting pull requests:

- Fork the repository and make your changes in a feature branch
- Include unit and integration tests for any new features, as well as updates to existing tests
- Ensure that existing unit and integration tests run successfully. Instructions on how to run unit tests can be found in the readme in each component's respective repository.
- Ensure that coding style checks pass. Instructions on how to run coding style checks can be found in the readme in each components respective repository.

A pull request can contain a single commit or multiple commits. The most important guideline is that a single commit should map to a single fix or enhancement. Here are some example scenarios:

- If a pull request adds a feature but also fixes two bugs, the pull request should have three commits: one commit for the feature change and two commits for the bug fixes.
- If a pull request is opened with five commits that contain changes to fix a single issue, the pull request should be rebased to a single commit.
- If a pull request is opened with several commits, where the first commit fixes one issue and the rest fix a separate issue, the pull request should be rebased to two commits (one for each issue).

Note: Your pull request should be rebased against the current master branch. Do not merge the current master branch in with your topic branch. Do not use the Update Branch button provided by GitHub on the pull request page.

Commit Message¶

Include a commit message. Commit messages should follow common Git conventions, such as using the imperative mood, separate subject lines, and a line length of 72 characters. These guidelines and more are documented here: <https://chris.beams.io/posts/git-commit/#seven-rules>¶

Signed-off-by¶

Each commit must include a "Signed-off-by" line in the commit message (git commit -s). This sign-off indicates that you agree the commit satisfies the Developer Certificate of Origin (DCO).

Commit Email Address¶

Your commit email address must match your GitHub email address. For more information, see <https://help.github.com/articles/setting-your-commit-email-address-in-git/>

Merge Requirements¶

A pull request will not be merged until it has passed these status checks:

- The build must pass all automated tests executed by the Jenkins build server.
- The pull request must be approved by at least two reviewers without any outstanding requests for changes

Reporting Issues¶

Report issues using each component's respective github repository.