## TensorRT OSS Contribution Rules

#### Issue Tracking

\* All enhancement, bugfix, or change requests must begin with the creation of a [TensorRT Issue Request](https://github.com/nvidia/TensorRT/issues).

\* The issue request must be reviewed by TensorRT engineers and approved prior to code review.

#### Coding Guidelines

- All source code contributions must strictly adhere to the [TensorRT Coding Guidelines](CODING-GUIDELINES.md).

- In addition, please follow the existing conventions in the relevant file, submodule, module, and project when you add new code or when you extend/fix existing functionality.

- To maintain consistency in code formatting and style, you should also run `clang-format` on the modified sources with the provided configuration file. This applies TensorRT code formatting rules to:

- class, function/method, and variable/field naming

- comment style

- indentation

- line length

- Format git changes:

```bash

# Commit ID is optional - if unspecified, run format on staged changes.

git-clang-format --style file [commit ID/reference]

```

- Format individual source files:

```bash

# -style=file : Obtain the formatting rules from .clang-format

# -i : In-place modification of the processed file

clang-format -style=file -i -fallback-style=none <file(s) to process>

```

- Format entire codebase (for project maintainers only):

```bash

find samples plugin -iname \*.h -o -iname \*.c -o -iname \*.cpp -o -iname \*.hpp \

| xargs clang-format -style=file -i -fallback-style=none

```

- Avoid introducing unnecessary complexity into existing code so that maintainability and readability are preserved.

- Try to keep pull requests (PRs) as concise as possible:

- Avoid committing commented-out code.

- Wherever possible, each PR should address a single concern. If there are several otherwise-unrelated things that should be fixed to reach a desired endpoint, our recommendation is to open several PRs and indicate the dependencies in the description. The more complex the changes are in a single PR, the more time it will take to review those changes.

- Write commit titles using imperative mood and [these rules](https://chris.beams.io/posts/git-commit/), and reference the Issue number corresponding to the PR. Following is the recommended format for commit texts:

```

#<Issue Number> - <Commit Title>

<Commit Body>

```

- Ensure that the build log is clean, meaning no warnings or errors should be present.

- Ensure that all `sample\_\*` tests pass prior to submitting your code.

- All OSS components must contain accompanying documentation (READMEs) describing the functionality, dependencies, and known issues.

- See `README.md` for existing samples and plugins for reference.

- All OSS components must have an accompanying test.

- If introducing a new component, such as a plugin, provide a test sample to verify the functionality.

- To add or disable functionality:

- Add a CMake option with a default value that matches the existing behavior.

- Where entire files can be included/excluded based on the value of this option, selectively include/exclude the relevant files from compilation by modifying `CMakeLists.txt` rather than using `#if` guards around the entire body of each file.

- Where the functionality involves minor changes to existing files, use `#if` guards.

- Make sure that you can contribute your work to open source (no license and/or patent conflict is introduced by your code). You will need to [`sign`](#signing-your-work) your commit.

- Thanks in advance for your patience as we review your contributions; we do appreciate them!

#### Pull Requests

Developer workflow for code contributions is as follows:

1. Developers must first [fork](https://help.github.com/en/articles/fork-a-repo) the [upstream](https://github.com/nvidia/TensorRT) TensorRT OSS repository.

2. Git clone the forked repository and push changes to the personal fork.

```bash

git clone https://github.com/YOUR\_USERNAME/YOUR\_FORK.git TensorRT

# Checkout the targeted branch and commit changes

# Push the commits to a branch on the fork (remote).

git push -u origin <local-branch>:<remote-branch>

```

3. Once the code changes are staged on the fork and ready for review, a [Pull Request](https://help.github.com/en/articles/about-pull-requests) (PR) can be [requested](https://help.github.com/en/articles/creating-a-pull-request) to merge the changes from a branch of the fork into a selected branch of upstream.

\* Exercise caution when selecting the source and target branches for the PR.

Note that versioned releases of TensorRT OSS are posted to `release/` branches of the upstream repo.

\* Creation of a PR creation kicks off the code review process.

\* Atleast one TensorRT engineer will be assigned for the review.

\* While under review, mark your PRs as work-in-progress by prefixing the PR title with [WIP].

4. Since there is no CI/CD process in place yet, the PR will be accepted and the corresponding issue closed only after adequate testing has been completed, manually, by the developer and/or TensorRT engineer reviewing the code.

#### Signing Your Work

\* We require that all contributors "sign-off" on their commits. This certifies that the contribution is your original work, or you have rights to submit it under the same license, or a compatible license.

\* Any contribution which contains commits that are not Signed-Off will not be accepted.

\* To sign off on a commit you simply use the `--signoff` (or `-s`) option when committing your changes:

```bash

$ git commit -s -m "Add cool feature."

```

This will append the following to your commit message:

```

Signed-off-by: Your Name <your@email.com>

```

\* Full text of the DCO:

```

Developer Certificate of Origin

Version 1.1

Copyright (C) 2004, 2006 The Linux Foundation and its contributors.

1 Letterman Drive

Suite D4700

San Francisco, CA, 94129

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

```

```

Developer's Certificate of Origin 1.1

By making a contribution to this project, I certify that:

(a) The contribution was created in whole or in part by me and I have the right to submit it under the open source license indicated in the file; or

(b) The contribution is based upon previous work that, to the best of my knowledge, is covered under an appropriate open source license and I have the right under that license to submit that work with modifications, whether created in whole or in part by me, under the same open source license (unless I am permitted to submit under a different license), as indicated in the file; or

(c) The contribution was provided directly to me by some other person who certified (a), (b) or (c) and I have not modified it.

(d) I understand and agree that this project and the contribution are public and that a record of the contribution (including all personal information I submit with it, including my sign-off) is maintained indefinitely and may be redistributed consistent with this project or the open source license(s) involved.

```