# Contributing to Uncrustify

## How to contribute

There are lots of ways to contribute to Uncrustify:

- Report Issues

- Propose Features or Improvements

- Submit Pull Requests

## Making changes

\* Pull latest [master][master] and create a new branch:

- Branch name \_should\_ use lowercase, using `-` to separate words

and not `\_`. Other special characters \_should\_ be avoided.

(However, feel free to use option names verbatim in branch names;

`\_` \_should\_ be used when part of an option name.)

- A hierarchical structure \_may\_ be designated using `/`

(e.g. `area/topic`), although this is uncommon.

The last part of the name can be keywords like `bugfix`, `feature`,

`optim`, `docs`, `refactor`, `test`, etc.

- Branches \_should\_ be named after \_what\_ the change is about.

- Branches \_should not\_ be named after the issue number,

developer name, etc.

\* Organize your work:

- Specialize your branch to target only one thing.

Split your work in multiple branches if necessary.

- Make commits of logical units.

- Avoid "fix-up" commits.

Instead, rewrite your history so that the original commit is "clean".

- Try to write a [quality commit message][commits]:

+ Separate subject line from body with a blank line.

+ Limit subject line to 50 characters.

+ Capitalize the subject line.

+ Do not end the subject line with a period.

+ Use imperative present tense in the subject line.

A proper subject can complete the sentence

"If applied, this commit will, [subject]".

+ Wrap the body at 72 characters.

+ Include motivation for the change

and contrast its implementation with previous behavior.

Explain the \_what\_ and \_why\_ instead of \_how\_.

- If the git diff command, or the diff part of the git gui,

don't produce accurate output, it might be necessary to add

some lines to the ~/.gitconfig file:

[diff]

algorithm = patience

[gui]

diffopts = --patience

\* Add or update unit tests:

- All behavioral changes should come with a unit test that verifies

that the new feature or fixed issue works as expected.

- Consider improving existing tests if it makes sense to do so.

- Any unused test number may be used,

however it is preferred to keep related tests together, if possible.

- Read [Writing Tests][tests] for more details.

\* Polish your work:

- The code should be clean, with documentation where needed.

- The change must be complete (no upcoming fix-up commits).

- Functional changes should always be accompanied by tests (see above).

Changes without tests should explain why tests are not present.

(Changes that are non-functional, such as documentation changes,

can usually omit tests without justification.)

\* Prepare a Pull Request (PR):

- To reduce the likelihood of conflicts and test failures,

consider rebasing your work on top of latest master before creating a PR.

- Verify that your code is working properly

by running `ctest` in your build directory.

(Changes that fail CI will \_not\_ be merged.

Running the tests locally will help to avoid this.)

You can change the level of logging by changing the line 104 and 109

of the file tests/test\_uncrustify/test.py to another value.

- The PR title should represent \_what\_ is being changed

(a rephrasing of the branch name if set correctly).

- The PR description should document the \_why\_ the change needed to be done

and not \_how\_, which should be obvious by doing the code review.

[master]: https://github.com/uncrustify/uncrustify/tree/master

[commits]: https://chris.beams.io/posts/git-commit/

[tests]: https://github.com/uncrustify/uncrustify/wiki/Writing-Tests