# Contributing to Crosswalk

Thanks a lot for contributing to Crosswalk, you rock!

This page lists some guidelines for the contribution process to help you send

an awesome patch that gets reviewed and merged quickly.

[Crosswalk's website](https://crosswalk-project.org/contribute/) also has a

whole section dedicated to this, be sure to check it out.

## License

Crosswalk is licensed under the

[3-clause BSD license](http://opensource.org/licenses/BSD-3-Clause). When you

submit a patch, you agree to license your contribution to Intel under this

license.

## Submitting pull requests

\* If it applies, add the platform your change affects in both the commit

message and the pull request title. For example, \_"[Android] Update target

SDK to android-21"\_ or \_"[Windows] Add unit test for metadata element

handler"\_.

\* If you know them, be sure to mention the people you would like to review your

patch in a comment. You can check the `OWNERS` files in the tree to know the

best people to look at your changes.

\* Do not use your `master` branch in your pull request. This means you will not

be able to send multiple pull requests at the same time without changing all

of them at once, and they will also be changed when you update your branch to

track Crosswalk's latest changes. Instead, use a separate branch for each

pull request.

\* If someone asks you to make a change to your pull request, \*\*DO NOT\*\* close

it and open a new one. Instead, make the changes to your branch, amending and

rebasing when necessary, and then use `git push -f` to push the changes to

the same branch you have used in your existing pull request.

## Commit messages

\* Use the present tense ("Add feature" not "Added feature").

\* Use the imperative mood ("Move cursor to..." not "Moves cursor to...").

\* Limit the first line to 72 characters or less.

\* If your pull request fixes an open issue in

[our bug tracker](https://crosswalk-project.org/jira), please reference it in

your commit message as well as your pull request message, like this:

```

Fix something.

Yadda, yadda, yadda.

BUG=XWALK-123456

```

This way, the issue will be updated when the pull request is sent and, once

it is merged, the issue will be closed automatically.

If you simply want the issue to be updated but do \*\*not\*\* want it to be

closed, use a different construct in your message, like \_"Related to:

XWALK-123"\_.

## Coding style

\* C++: We follow

[Chromium's coding style](http://dev.chromium.org/developers/coding-style),

which basically mean's

[Google's](https://google-styleguide.googlecode.com/svn/trunk/cppguide.html).

Please take some time to get familiar with it, particularly with the amount

of space used for indentation and the position of asterisks and braces. If in

doubt, check the rest of the code around what you are changing.

\* Java: We also follow

[Chromium's guidelines](http://www.chromium.org/developers/coding-style/java),

which are very similar to

[Android's style guide](http://source.android.com/source/code-style.html), so

there should not be any big surprises.

\* Python: Just like

[Chromium's](http://dev.chromium.org/developers/coding-style), our Python

code mostly follows [PEP-8](https://www.python.org/dev/peps/pep-0008/), the

exceptions being the amount of space used for indentation (2 instead of 4)

and the use of `MixedCase` functions and methods instead of

`lower\_case\_ones`.

## Updating the release notes

\* When submitting a new feature, the author is welcome and encouraged to add a

short description to the release notes in the same PR that implements the

feature.

\* Latest before branching to beta the release notes are updated to reflect the

content up to that point.

\* Any notable bug fixes or changes done in the beta branch are updated in the

branched release notes.

\* When the release is promoted to the stable channel, the release date is

added to the release notes.