# Contributing

Glad to see you're interested in contributing to Reko. Being a small team, all help is appreciated.

## What can you do?

\* If you have an exciting idea for a feature, a question about how Reko works, or have found a bug in Reko,

please open an [issue](http://github.com/uxmal/reko/issues) so we know about it.

\* If you feel an urge to code, submit a pull request. Please make an effort to provide one or more unit tests

along with your code.

## Unit tests

When contributing code we request that you submit one or more unit tests that exercise

the code you added or modified. Because of the size and complexity of the Reko project, it is critical that

we manage changes with proper tests to ensure that no existing code is being broken.

We use [NUnit 2.6](http://www.nunit.org/) as our test framework.

To make the unit tests suite run as fast as possible (at the time of writing, there are 3462 unit tests)

they should avoid reading or writing large amounts of data from disk. A unit test should be as "quiet"

as possible. Avoid output to stdout/stderr, and avoid using `Debug.Write` and friends \*\*if the

unit test passes\*\*. If the unit test \*\*fails\*\*, diagnostic output to stdout or using `Debug.Write` is

encourage to aid in diagnosis.

## Submitting a pull request (PR)

Once the unit tests pass, there is a final step: run all the regression tests on the test subjects.

The test subjects are executable files, located in the `./subjects` folder of the Reko project tree.

Run the Python script `./subjects/regressionTests.py` to exercise all the test programs. The script will

run the command-line version of Reko on the subjects, which generates output files for each subject.

If your code changes resulted in the outputs also changing, you will need to review them to make sure that

all differences make sense, and that no regressions have occurred.

Once you've reviewed any changes in the `./subjects` tree and have committed any differences, you can

submit the PR.