

# Contributing Code



## Incomplete

This page needs your help to finish.

Community members can contribute to the project in many different ways, not all of which involve writing code; your careful reports of problems, your suggestions of new features or improvements, and even an occasional word of appreciation are all very useful and appreciated.

If, however, you are willing and able to go to the next level and actually do development, then that's terrific too.

## Get Connected

1. Get an [account at SIPfoundry](#)
2. Subscribe to *and regularly read* the [developer list](#) (it's a good idea to also keep an eye on the [users list](#)).

## Get an Issue

If you have a problem or feature that you'd like to work on, that's great. If there is already an issue for it in the tracker, then send a note to the dev list to let people know that you're going to get started on it.

If you don't have a particular issue in mind, check the tracker - any issue not assigned to someone is fair game.

## Get To Work

When you have a design approach in mind, send a detailed note to the developer list describing it. Pay particular attention to describing:

- What changes may be required in which components
- What data your change requires from configuration; especially note any new configuration will be needed
- What data from SIP messages your change uses to make decisions
- Any interoperability issues with external systems
- Any external dependencies that are introduced, and especially **the licenses for any other open source**

There are a number of pages under this one on how to check out the code and get a development environment working.

## Test your changes

You can greatly improve the chances that your contribution will be accepted, and significantly reduce the time the process takes, by doing as much of the checking as possible before you submit it. In particular:

- Check that it follows the established conventions (see elsewhere in the Developer documentation)
- Some components have specific regression testing requirements that are not executed as part of the unit tests in the standard build.
- Regression test as much of the system as possible to make sure that anything even remotely connected to your change is still backwards compatible.
- Ensure that an rpm build on the CentOS versions that are the primary binary distributions for the project work; pay particular attention to availability of any dependencies (see TBD).

## No Contributor Agreement Required!

You may be asked to verify you own copyright to you code for significant contributions but under most circumstances, there are no forms to sign.

## Submit a Pull Request on GitHub

Submit a Pull Request (PR). If nothing happens within a week or so, feel free to also post a note to the developers list, just to make sure that your patch gets noticed. A pointer to a ticket or the list archive of the design discussion thread, and any changes you made while actually implementing and testing are also very useful.

How soon your patch will get reviewed by a committer depends on many factors - feel free to ping the list now and again.