

HowToBecomeACommitter

How to Become a Committer on Apache Hadoop

We regularly get people asking "how they can become a Hadoop committer?"

It's a long process that effectively comes from being recognised as a good developer within the Hadoop development community.

- You have to be capable of writing code that works well. Not immediately -the JIRA-based development process is designed to iterate to provide good solutions with the insight and input of others.
- Being able to write good tests and documentation are also valued skills. You can contribute these without supplying production code alongside them.
- You have to be patient: put in hours, work up from small "newbie" patches to engagement in large features -often in collaboration with others.
- You have to be shown to work with the rest of the team. That's collaborative on-line development,

How do you manage these? By working on issues, starting small and then building up -and collaborating with others on their work.

- Start by using Hadoop and focus on the small patches that fix things that matter to you -based on your experience of using it
- Follow the [HowToContribute](#) documentation
- Look at the JIRAs others are filing -watch ones you are interested in and follow the process for getting it in. Don't be afraid to review and comment on other people's work.
- Review bits of the code that you are curious about, dive into the classes, step through bits of it with a debugger -get a feel for the system.

Don't

- Don't expect to be granted committer status just because you've filed a couple of patches. They can be merged into the codebase -after review. You'll get acknowledgement, and the fixes you need will get into later versions of Hadoop.
- Don't be afraid that just because English isn't your first language you'll be excluded from the process. It is a global project -English is used as the language, but it's not the first language of many of the developers, and nobody will worry about any mistakes.
- Don't try and do ambitious reworks of core bits until you are trusted to be able to write code that spans big parts of it, code that is testable and supportable. We're scared of "here is a giant patch" contributions.
- Don't expect to be immediately granted committer status -it's about contributions and trust. Trust in the quality of your code and insights, which comes from your extended participation in the development process.

Everyone in the project ~~even the committers~~ have to have their code reviewed by others before checking it in. Becoming a committer is not so much a reward for time put in but an acknowledgement that you are considered capable of reviewing and judging other people's work, to +1 their contributions and then perform the housekeeping of actually checking it in. That's actually a hard skill -and it's why committers tend to restrict their reviewing to the parts of the system which they understand well.

Now -get coding!

See also

1. [Inside Hadoop dev](#)