

# GettingInvolved

## Introduction

Apache Commons is a little unusual in the open source world because of its focus on libraries rather than an actual product. So people here generally get involved because their project uses a commons library and they find a bug, they are frustrated by a lack of documentation, or they need a new feature. Libraries are tricky things to work on because of a very large emphasis on backwards compatibility, stability and minimizing size; lots of discussion and testing often goes on before patches are accepted.

## Contributing to a commons project you use

If you are currently working on a project that makes use of a commons component and you would like to help, that is great news.

Patches for documentation are always very welcome and are usually committed pretty fast.

Patches for bugs, when provided together with a unit test that demonstrates the problem, are also very welcome. Patches that don't have unit tests are much slower to be processed as commons takes unit testing very seriously, and any code change submitted without a unit test generally means one of the maintainers needs to write the unit test - which is never a lot of fun.

Patches for new features may take quite a while to be processed; be patient.

Helping out on the user list is also a great way to provide assistance. The less time the project maintainers need to spend helping users with questions the more development they can do. And showing your competence by helping with tricky user questions will certainly make committers more interested in any patches you may offer later.

## Contributing your time

If you are just generally interested in contributing your time to a worthwhile project, some project that is actually creating a *product* would be an easier place to start than any of the commons components. If you have an interest in content management systems, you might like to look at the work on graffiti (see incubator.apache.org). If you have an interest in building web applications you might like to look at tapestry or Cocoon or MyFaces. If you have an interest in databases you might like to have a look at derby. Or maybe you're frustrated with the limitations of bugzilla? Then find a java project that implements a bug tracker and help with that.

Look for a project that's already got a reasonably active maintainer base; that's the easiest sort of project to get involved in as there are lots of people around to review any contributions you make and offer feedback (or better still, to commit your patches!). Fixes for bugs that are bothering \*you\* are most likely to be accepted; documentation updates and enhanced unit tests are also usually pretty welcome. Patches for arbitrary bugs you found listed in JIRA are likely to be looked at a little more sceptically as there's no evidence you actually have experience in that area. Offers to implement new features "because it looks like this might be useful" are treated even more cautiously - unless the project is implementing some kind of specification, and the new feature is just to complete compliance with the spec.

Eventually, if that project uses a commons library and you find a bug or need a new feature you might find yourself contributing to commons...

As mentioned above, helping out on the user list is also a great way to provide assistance. This may well mean delving into the code or writing test code to try out different scenarios - which will improve your coding skills and knowledge too.

## Dealing with dormant or dead commons components

Sometimes people find that having posted an email or created a JIRA entry about a particular commons project there is just no reply.

This may just mean that the development community for that component is small and the people involved are away or very busy. A polite reminder every week or so is acceptable.

However it sometimes happens that there are no maintainers of the project left, and no existing commons committer is interested in dedicating time to working on that component. In this situation there is unfortunately very little that an "outsider" can do about it; without an existing committer willing to review and commit patches the project just won't go anywhere no matter how many good patch submissions are waiting in the bugtracking system. And the commons community can't grant committer rights to people it doesn't know, as it can't let just anyone put out releases under the apache name, even when there is considerable interest in the user community for fixes and enhancements.

If you are an experienced open-source developer with a track record in other open-source groups then please post your details to the commons user or dev mailing list; an exception might be made in your case.

Otherwise you might just have to use an alternative project, or create a "fork" of the current commons project on sourceforge or similar. Where a commons component is dead/dormant, having a user create a fork is considered quite acceptable; in fact if the fork's community flourishes and the developers show good open-source spirit and ability then there is a very good chance that the project would be welcome to merge back into commons if the developers should choose to do so in the future.

## Checking project activity level

Anyone got any good ideas for how to monitor a commons component's "activity level"? Running "svn log" on the root directory of the project to see the number of recent patches is one reasonable way. Normally, looking at the email volume would be a good idea but I don't know any mail archives or forums that make it easy to see the volume of [projectname] emails over the last month or so.

