How to contribute to GENIVI

- Code Contribution
- Patch process
  - Pull request through GitHub
  - Patches sent through GENIVI mailing lists
- New Project Contribution
- SPDX license tags
- Bug fixing
- Documentation
- Email mailing lists

GENIVI is an Open Source project and follows Open Source "best practices." This means that it is easy to contribute to GENIVI in a number of ways. This public wiki page goes into some detail on the various methods in which your contribution can be best formatted for easy acceptance. Not following specific requirements may lead to proposals being rejected. Please refer to the GENIVI contribution guidelines for more information.

Code Contribution

One can contribute a **new** project (see New Project Contribution below) or to an **existing** GENIVI project. There are a number of active projects available in GitHub. Contributing code to one of these projects is as easy as cloning the repo and

- sending a pull request through GitHub (preferred process)
- if you cannot access GitHub, sending a patch to the mailing list which is a similar, if not identical, process found in other open projects.

Note that you’ll have to abide by the license that the code is released under, but most of the time that is the Mozilla Public License 2.0 which is a Free Software license, so it is easy to work with. Take the time to understand the license and that will make your contribution process easier.

It’s important that you ensure you have permission to submit code to the GENIVI project. If your company is a GENIVI member and has signed the CLA, check with your company for okay first. Non-members can submit code by following the Open Source process.

For GENIVI Development Platform (GDP), there are additional details: GDP Contribution Process

Patch process

1. Each patch must contain only 1 logical change and hence smaller the patch, the better. Among other benefits, it’s easier to review.
2. Patches that have been tested against the target are likely to build on the maintainers machine, pro tip: test your patch
3. In some cases, like your first commit to GENIVI, you’ll need to include a "developer certificate of origin."
4. Requirement: for commit messages, follow these rules to the best of your abilities.
5. Requirement: if you’re from a company that has a signed CLA with GENIVI, you’ll need to be on the CLA contributors list or a maintainer will not be able to accept your patch (CLA FAQ)
6. Requirement: include a "Signed-off-by:" line. Use git format-patch with the -s flag for example

For most projects we recommend Rules for Commit Messages. For GDP the rules are required.

The goal with this checklist is make patches easier to contribute by providing a minimum of what is expected. Different project maintainers deal with patches differently so contact the project’s mailing list for their patch acceptance policy.

Pull request through GitHub

For GENIVI Development Platform (GDP) please refer to the more detailed GDP Contribution Process.

Pull Request is the preferred process to submit code to GENIVI. The process looks like this;

1. Use the standard GitHub process for creating pull requests.
   a. Please provide some form of explanation for what the patch does and why it is needed.
   b. The mailing list is often a good place to discuss patches but GitHub also provides good code review tools
2. The requirements above also apply; if you’re from a company that has a signed CLA with GENIVI, sign your commits (see Patch process)

See also: Using Pull Requests (GitHub official documentation) and Contributing to Open Source on GitHub (GitHub Guides).

Patches sent through GENIVI mailing lists

The preferred submission process is through GitHub PR but it is also possible to submit patches through the appropriate mailing list. GENIVI projects use the following best practices for submitting patches through mailing lists:

1. Start the subject line with the tag: [PATCH]. Multiple patches part of the same group are usually numbered, e.g. 2/10 (meaning second patch out of 10), and if needed patches are versioned. Here is an example from LKML
2. Please provide some form of explanation for what the patch does and why it is needed. The mailing list is often a good place to discuss patches.
3. Use at least diff -u to create your patch, different lists may prefer different formats so it's good to check beforehand.
4. Attach your patch as an attachment to your email, some mailers mangle white space.
5. Adding the patch in the email for inlined discussion is a prerogative of the maintainer.

See also: Git Email Setup for tips on setting up email for working with patches in a corporate environment. You can also view the Linux kernel patch process which GENIVI has tried to follow when designing its policy.

New Project Contribution

You can contribute a project to be hosted under GENIVI's umbrella, if your project meets the criteria specified here. The process is straightforward and is outlined here:

1. Fill in the required information in GENIVI's project proposal form. A Google form is provided for convenience.
2. Mail the project to GENIVI's Open Source projects list: oss-projects@mail.genivi.org
3. A project approval process occurs inside GENIVI's OSS office.

After those steps are complete GENIVI begins the code launching process.

1. Code scanning is done in conjunction with the License Review Team in FOSSology: http://fossology.genivi.org/
2. Documentation on the accepted component is delivered and hosted on GENIVI wiki and web site.
3. Additional infrastructure is created based on need -- issue tracking, document hosting, git repositories, mailing list.
4. Individuals are assigned to maintain the given infrastructure used to support components, i.e. the git repo, the documents, the issues on a per project basis.

SPDX license tags

GENIVI and many companies require the use of SPDX license tags to identify individual files which are part of a larger component. This is done to ensure compliance with all FOSS licenses as well as to provide help in determining provenance. Please use a tag exactly like this in your source code, please note whitespace;

SPDX-license-identifier: MPL-2.0

For more information see GENIVI’s Public Licensing Policy.

Bug fixing

Any non-trivial project has bugs. GENIVI hosts an open issue tracker which is used by some projects to hold new features and other easy tasks for those new to the project. Here is a list of issues tagged "new feature" or "improvement" for example: list of "improvement" and "new feature" issues.

Documentation

Good documentation is key to the success of any project and GENIVI is no exception. GENIVI strives to have good documentation and it can often be found in the GENIVI project pages. If you find issues or have improvements to the documentation, send a change to the relevant mailing list. I'm certain it will be welcomed. This wiki too holds a lot of documentation both in the form of tutorials and in the form of requirements. Looking into this wiki and using it as a resource in combination with the code repositories makes and effective set of tools to move a project forward and creates valuable artifacts which remain open.

Anyone can edit this wiki, all you have to do is register.

Email mailing lists

GENIVI has a number of public mailing lists. The discussion there is about technical aspects of GENIVI's projects. Since the discussions can get long and hard to follow, its preferred that email be sent as plain text. In addition, threads are easier to follow on high-traffic lists when there is a subject tag. For example, if you're writing about the GENIVI Demo Platform its preferred that you use a [GDP] tag in the subject line. Certain threads also become very long and with signatures and previous replies they can get stopped in the mail server due to their size. For this reason its preferred you inline your comments, trim your posts, and refrain from signatures on the public mailing lists.