HOW TO CONTRIBUTE TO OpenSSL

Please visit our Getting Started page for other ideas about how to contribute.

Development is done on GitHub in the openssl/openssl repository.

To request new features or report bugs, please open an issue on GitHub

To submit a patch, please open a pull request on GitHub. If you are thinking of making a large contribution, open an issue for it before starting work, to get comments from the community. Someone may be already working on the same thing or there may be reasons why that feature isn't implemented.

To make it easier to review and accept your pull request, please follow these guidelines:

1. Anything other than a trivial contribution requires a Contributor License Agreement (CLA), giving us permission to use your code. If your contribution is too small to require a CLA (e.g. fixing a spelling mistake), place the text "`CLA: trivial`" on a line by itself separated by an empty line from the rest of the commit message. It is not sufficient to only place the text in the GitHub pull request description.

   To amend a missing "`CLA: trivial`" line after submission, do the following:

   ```
   git commit --amend
   [add the line, save and quit the editor]
   git push -f
   ```

2. All source files should start with the following text (with appropriate comment characters at the start of each line and the year(s) updated):

   ```
   Copyright 20xx-20yy The OpenSSL Project Authors. All Rights Reserved.

   Licensed under the Apache License 2.0 (the "License").  You may not use
   this file except in compliance with the License.  You can obtain a copy
   in the file LICENSE in the source distribution or at
   https://www.openssl.org/source/license.html
   ```

3. Patches should be as current as possible; expect to have to rebase often. We do not accept merge commits, you will have to remove them (usually by rebasing) before it will be acceptable.

4. Patches should follow our coding style and compile without warnings. Where `gcc` or `clang` is available you should use the `--strict-warnings` `Configure` option. OpenSSL compiles on many varied platforms: try to ensure you only use portable features. Clean builds via GitHub Actions and AppVeyor are required, and they are started automatically whenever a PR is created or updated.

5. When at all possible, patches should include tests. These can either be added to an existing test, or completely new. Please see test/README.md

for information on the test framework.

6. New features or changed functionality must include documentation. Please look at the "pod" files in doc/man[1357] for examples of our style. Run "make doc-nits" to make sure that your documentation changes are clean.

7. For user visible changes (API changes, behaviour changes, ...), consider adding a note in CHANGES.md. This could be a summarising description of the change, and could explain the grander details. Have a look through existing entries for inspiration. Please note that this is NOT simply a copy of git-log one-liners. Also note that security fixes get an entry in CHANGES.md. This file helps users get more in depth information of what comes with a specific release without having to sift through the higher noise ratio in git-log.

8. For larger or more important user visible changes, as well as security fixes, please add a line in NEWS.md. On exception, it might be worth adding a multi-line entry (such as the entry that announces all the types that became opaque with OpenSSL 1.1.0). This file helps users get a very quick summary of what comes with a specific release, to see if an upgrade is worth the effort.

9. Guidelines how to integrate error output of new crypto library modules can be found in crypto/err/README.md.