Contributing to libjpeg-turbo

=============================

libjpeg-turbo is a stable and mature product with a worldwide reach, it is an

ISO/ITU-T reference implementation of the JPEG standard, and its maintainer

does not earn a salary for maintaining it. Thus, not every code contribution

can or will be accepted into the libjpeg-turbo code base. In order to maximize

the chances that a code contribution is acceptable, please adhere to the

following guidelines:

1. If the code contribution is a bug fix, then please ensure that enough

information is provided so that the maintainer can readily reproduce and

document the bug. That information should include:

- A clear and concise description of the bug

- The steps and (if applicable) images necessary to reproduce the bug

- The compilers, operating systems, and CPUs with which the bug was

observed

- The versions of libjpeg-turbo with which the bug was observed

- If the bug is a regression, the specific commit that introduced the bug

(use `git bisect` to determine this)

2. If the code contribution will implement a major new feature, then please

contact the project maintainer (through a

[GitHub issue](https://github.com/libjpeg-turbo/libjpeg-turbo/issues/new),

[direct e-mail](https://libjpeg-turbo.org/About/Contact), or the

[libjpeg-turbo-devel mailing list](https://libjpeg-turbo.org/About/MailingLists))

prior to implementing the feature. In general, major new features that are

potentially disruptive to the quality of libjpeg-turbo are unlikely to be

accepted unless:

- The new feature is within the existing scope of libjpeg-turbo.

- The new feature has been thoroughly regression tested and benchmarked on

all of the supported platforms that are potentially affected by it.

- The new feature has been documented clearly and concisely in the change

log and (if applicable) the libjpeg and TurboJPEG API documentation and

man pages.

- The code implementing the new feature is formatted consistently with the

rest of the libjpeg-turbo code base (use

[checkstyle](https://github.com/libjpeg-turbo/checkstyle) to validate

this.)

- The new feature does not introduce new members into the exposed libjpeg

API structures (doing so would break backward ABI compatibility.)

- The new feature does not alter existing libjpeg-turbo usage or

development workflows.

- The code implementing the new feature is elegant, easily maintainable,

and causes the least possible amount of disruption to the libjpeg-turbo

code base.

- The new feature is based on the dev branch of the libjpeg-turbo

repository.

... OR ...

- Your organization is prepared to pay for the labor necessary to ensure

all of the above. Even the most well-written patches can still require

a significant amount of labor to clean up, test, and integrate. (See

above RE: the maintainer not earning a salary.)

Specific types of features that \*will not\* be accepted:

- Features that extend the scope of libjpeg-turbo to support image formats

other than DCT-based JPEG and JFIF

- Features that extend the scope of libjpeg-turbo to support image

processing algorithms that are not necessary for JPEG compression or

decompression

- Extensions to the JPEG format that have not been approved by the

appropriate standards bodies

- Non-trivial performance enhancements that have less than a 5% overall

impact on performance

3. If the code contribution is a build system enhancement, then please be

prepared to justify the enhancement. In general, build system enhancements

are unlikely to be accepted unless:

- The enhancement is potentially beneficial to a significant number of

upstream libjpeg-turbo users/developers. (If the enhancement is only

beneficial to a downstream project, then it does not belong here.)

- The enhancement has been tested with the following CMake versions:

- The earliest version of CMake that libjpeg-turbo currently supports

(refer to CMakeLists.txt)

- The most recent major version of CMake

- (if applicable) The earliest version of CMake with which the

enhancement can be used

- The enhancement has been tested on all of the major platforms (Mac,

Linux, Windows/Visual C++, Windows/MinGW) that are potentially affected

by it.

- The enhancement does not introduce new build system requirements or CMake

variables unless absolutely necessary.

- The enhancement does not alter existing libjpeg-turbo development

workflows.

Specific types of build system enhancements that \*will not\* be accepted:

- Enhancements that allow libjpeg-turbo to be built from a subdirectory

of a downstream repository. These enhancements are not maintainable in

the upstream libjpeg-turbo build system. Use the CMake

`ExternalProject\_Add()` function instead.

- Enhancements that introduce new (non-CMake-based) build systems