<!--

Copyright (c) 2017, 2020 IBM Corp. and others

This program and the accompanying materials are made available under

the terms of the Eclipse Public License 2.0 which accompanies this

distribution and is available at https://www.eclipse.org/legal/epl-2.0/

or the Apache License, Version 2.0 which accompanies this distribution and

is available at https://www.apache.org/licenses/LICENSE-2.0.

This Source Code may also be made available under the following

Secondary Licenses when the conditions for such availability set

forth in the Eclipse Public License, v. 2.0 are satisfied: GNU

General Public License, version 2 with the GNU Classpath

Exception [1] and GNU General Public License, version 2 with the

OpenJDK Assembly Exception [2].

[1] https://www.gnu.org/software/classpath/license.html

[2] http://openjdk.java.net/legal/assembly-exception.html

SPDX-License-Identifier: EPL-2.0 OR Apache-2.0 OR GPL-2.0 WITH Classpath-exception-2.0 OR LicenseRef-GPL-2.0 WITH Assembly-exception

-->

# Contributing to Eclipse OpenJ9

Thank you for your interest in Eclipse OpenJ9!

We welcome and encourage all kinds of contributions to the project, not only

code. This includes bug reports, user experience feedback, assistance in

reproducing issues and more. Contributions to the website

(https://github.com/eclipse/openj9-website), to the user documentation

(https://github.com/eclipse/openj9-docs), to the system verification tests

(https://github.com/eclipse/openj9-systemtest), or to Eclipse OMR

(https://github.com/eclipse/omr), which is an integral part of OpenJ9 are all

also welcome.

## Submitting a contribution to OpenJ9

You can propose contributions by sending pull requests (PRs) through GitHub.

Following these guidelines will help us merge your pull requests smoothly:

1. Your pull request is an opportunity to explain both what changes you'd like

pulled in, but also \_why\_ you'd like them added. Providing clarity on why

you want changes makes it easier to accept, and provides valuable context to

review.

2. Follow the commit guidelines found below.

3. We encourage you to open a pull request early, and mark it as "Work In

Progress", by prefixing the PR title with "WIP". This allows feedback to

start early, and helps create a better end product. Committers will wait

until after you've removed the WIP prefix to merge your changes.

4. If your contribution introduces an external change that requires an update

to the [user documentation](https://www.eclipse.org/openj9/docs/), add the

label `doc:externals` and open an [issue](https://github.com/eclipse/openj9-docs/issues/new?template=new-documentation-change.md)

at the user documentation repository. Examples of an external change include

a new command line option, a change in behavior, or a restriction.

5. Please carefully read and adhere to the legal considerations and

copyright/license requirements outlined below.

## Building and testing

In order to build OpenJ9, see the [build instructions](https://github.com/eclipse/openj9/tree/master/doc/build-instructions).

Once the build system is prepared, building consists of a few simple steps. If

building the original source fails, check the [level of the compiler](https://eclipse.github.io/openj9-docs/openj9\_support/)

being used.

There are a large number of test cases that are run automatically as part of

the Eclipse OpenJ9 build and test pipeline. The tests can be triggered by

committers from pull requests. You can see the latest results on the

[Eclipse OpenJ9 Jenkins instance](https://ci.eclipse.org/openj9/).

The tests can also be run manually on your own machine, refer to the

[OpenJ9 test quick start guide](https://github.com/eclipse/openj9/blob/master/test/README.md).

## Commit Guidelines

The first line describes the change made. It is written in the imperative mood,

and should say what happens when the patch is applied. Keep it short and

simple. The first line should be less than 70 characters, where reasonable,

and should be written in sentence case preferably not ending in a period.

Leave a blank line between the first line and the message body.

The body should be wrapped at 72 characters, where reasonable.

Include as much information in your commit as possible. You may want to include

designs and rationale, examples and code, or issues and next steps. Prefer

copying resources into the body of the commit over providing external links.

Structure large commit messages with headers, references etc. Remember, however,

that the commit message is always going to be rendered in plain text.

Please add `[skip ci]` to the commit message when the change doesn't require a

compilation, such as documentation only changes, to avoid unnecessarily wasting

the project's build resources.

When a commit has related issues or commits, explain the relation in the message

body. When appropriate, use the keywords described in the following help article

to automatically close issues.

https://help.github.com/articles/closing-issues-using-keywords/

For example:

```

Correct race in frobnicator

This patch eliminates the race condition in issue #1234.

Fixes: #1234

```

Sign off on your commit in the footer. By doing this, you assert original

authorship of the commit and that you are permitted to contribute it. This can

be automatically added to your commit by passing `-s` to `git commit`, or by

manually adding the following line to the footer of the commit.

```

Signed-off-by: Full Name <email>

```

Remember, if a blank line is found anywhere after the `Signed-off-by` line, the

`Signed-off-by:` will be considered outside of the footer, and will fail the

automated Signed-off-by validation. The email used to sign off the commit must

be the same, including case-sensitivity, as the one used to sign the Eclipse ECA,

or your commit will fail IP validation.

It is important that you read and understand the legal considerations found

below when signing off or contributing any commit.

### Example commits

Here is an example of a \*good\* commit:

```

Update and expand the commit guidelines

Elaborate on the style guidelines for commit messages. These new

style guidelines reflect the conversation found in #124.

The guidelines are changed to:

- Provide guidance on how to write a good first line.

- Elaborate on formatting requirements.

- Relax the advice on using issues for nontrivial commits.

- Move issue references from the first line to the message footer.

- Encourage contributors to put more information into the commit

message.

Closes: #124

Signed-off-by: Robert Young <rwy0717@gmail.com>

```

The first line is meaningful and imperative. The body contains enough

information that the reader understands the why and how of the commit, and its

relation to any issues. The issue is properly tagged and the commit is signed

off.

The following is a \*bad\* commit:

```

FIX #124: Changing a couple random things in CONTRIBUTING.md.

Also, there are some bug fixes in the thread library.

```

The commit rolls unrelated changes together in a very bad way. There is not

enough information for the commit message to be useful. The first line is not

meaningful or imperative. The message is not formatted correctly, the issue is

improperly referenced, and the commit is not signed off by the author.

### Other resources for writing good commits

- http://chris.beams.io/posts/git-commit/

## Legal considerations

Please read the [Eclipse Foundation policy on accepting contributions via Git](http://wiki.eclipse.org/Development\_Resources/Contributing\_via\_Git).

Your contribution cannot be accepted unless you have a signed [ECA - Eclipse Foundation Contributor Agreement](http://www.eclipse.org/legal/ECA.php) in place. If you have an active signed Eclipse CLA

([the CLA was updated by the Eclipse Foundation to become the ECA in August 2016](https://mmilinkov.wordpress.com/2016/08/15/contributor-agreement-update/)),

then that signed CLA is sufficient. You will have to sign the ECA once your CLA expires.

Here is the checklist for contributions to be \_acceptable\_:

1. [Create an account at Eclipse](https://dev.eclipse.org/site\_login/createaccount.php).

2. Add your GitHub user name in your account settings.

3. [Log into the project's portal](https://projects.eclipse.org/) and sign the ["Eclipse ECA"](https://projects.eclipse.org/user/sign/cla).

4. Ensure that you [\_sign-off\_](https://wiki.eclipse.org/Development\_Resources/Contributing\_via\_Git#Signing\_off\_on\_a\_commit) your Git commits.

5. Ensure that you use the \_same\_ email address as your Eclipse account in commits.

6. Include the appropriate copyright notice and license at the top of each file.

Your signing of the ECA will be verified by a webservice called 'ip-validation'

that checks the email address that signed-off on your commits has signed the

ECA. \*\*Note\*\*: This service is case-sensitive, so ensure the email that signed

the ECA and that signed-off on your commits is the same, down to the case.

### Copyright Notice and Licensing Requirements

\*\*It is the responsibility of each contributor to obtain legal advice, and

to ensure that their contributions fulfill the legal requirements of their

organization. This document is not legal advice.\*\*

Eclipse OpenJ9 is dual-licensed under the Eclipse Public License v2.0 and the

Apache License v2.0. Any previously unlicensed contribution should be released

under the same license.

\* If you wish to contribute code under a different license, you must consult

with a project lead before contributing.

\* For any scenario not covered by this document, please discuss the copyright

notice and licensing requirements with a project before contributing.

The template for the copyright notice and dual-license is as follows:

```

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Copyright (c) %s, %s IBM Corp. and others

\*

\* This program and the accompanying materials are made available under

\* the terms of the Eclipse Public License 2.0 which accompanies this

\* distribution and is available at https://www.eclipse.org/legal/epl-2.0/

\* or the Apache License, Version 2.0 which accompanies this distribution and

\* is available at https://www.apache.org/licenses/LICENSE-2.0.

\*

\* This Source Code may also be made available under the following

\* Secondary Licenses when the conditions for such availability set

\* forth in the Eclipse Public License, v. 2.0 are satisfied: GNU

\* General Public License, version 2 with the GNU Classpath

\* Exception [1] and GNU General Public License, version 2 with the

\* OpenJDK Assembly Exception [2].

\*

\* [1] https://www.gnu.org/software/classpath/license.html

\* [2] http://openjdk.java.net/legal/assembly-exception.html

\*

\* SPDX-License-Identifier: EPL-2.0 OR Apache-2.0 OR GPL-2.0 WITH Classpath-exception-2.0 OR LicenseRef-GPL-2.0 WITH Assembly-exception

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

```