

Lustre Development

Intel is committed to providing and supporting high quality Lustre releases. Intel hosts the current Lustre source code and welcome contributions.

Downloading Lustre Code

The main Lustre repository is browsable at [Lustre Git Web](#) or via Git clone from [git://git.whamcloud.com/fs/lustre-release.git](https://git.whamcloud.com/fs/lustre-release.git).

Building Lustre from Source

Lustre RPM are available from [Whamcloud's build server](#). However, if you need to build Lustre from source, you may find the [Building Lustre from Source](#) pages helpful.

Code contribution

Intel has an open code submission policy that does not require copyright sign-over.

Patch submission is similar to the Linux kernel and is detailed in the [Submitting Changes](#) page.

Coding formatting standards

Lustre code is written according to the the [Coding Guidelines](#).

Community tools

Intel provide technical expertise and development tools to enable well tested releases to be achieved in a open and convenient manner. The most common tools are listed below.

Lustre test suites

Lustre includes a number of test suites. These suites range from basic functionality in the form of `llmount.sh` thru to the advanced auster script with integrated reporting.

- [llmount.sh test script](#).
- Auster test suite: page coming soon.

Intel HPDD Tools Overview

Jira, Bug tracking

We're using [Jira](#) to track bugs and issues. You use the same account for both [Jira](#) and this Wiki. If you would like your account to be marked as a developer account then open a ticket in the LU project to request this

Gerrit, code review.

review.whamcloud.com

We are using Gerrit for our code inspections. You will need an OpenID to login to Gerrit, however we currently have not created our own OpenID service. For now please use a regular Gmail or Yahoo account, or another OpenID account if you have one. Once you have registered on Gerrit, you can add your email address to your Gerrit account.

Further reading:

- [Using Gerrit](#)

Jenkins, build.

build.whamcloud.com

Our build server is based on the Jenkins (formally Hudson) continuous integration platform, and it is currently creating CentOS/RHEL RPMs when changes are pushed to 1.8, 2.1, and master. It is also building packages when patches are submitted for inspection requests. Every patch that passes the Jenkins build process is automatically tested on our test cluster for at least 8-12 hours with a full range of Lustre regression tests.

Further reading:

- [Jenkins Build Slave Setup](#) is covered in a separate document.

Maloo

testing.whamcloud.com

Once tests have been run, the test results are uploaded to the Maloo test database. This includes basic PASS/FAIL/TIMEOUT/SKIP results for the hundreds of tests that were run, as well as the test environment, output and runtimes. In case of test failures, the console and test logs are also available, along with client and server debug logs and stack traces. The Maloo results database is searchable in a variety of ways.

Mailing lists and community

[Mailing Lists](#)

Organizations, [mailing lists](#) and [IRC](#) are available from the [Community Resources](#) page.

Wiki

wiki.whamcloud.com

You are already here. Please explore the wiki. Wikis need maintenance: if you see a spelling error or typo, please take a moment to correct it. If you find an documentation lacking please consider adding to the wiki. The page [Wiki Hints & Tricks](#) is a good place to start.

Third party tools

A separate page includes a list of [Third Party Tools](#)