

Patch Landing Process Summary

Brief summary for landing patch to Lustre:

1. File or find a [Lustre JIRA](#) ticket describing your change.
2. Test and commit your patch locally.
3. Verify patch follows [Lustre Coding Guidelines](#) with `git show | build/checkpatch.pl` - (and/or symlink the `contrib/git-hooks/commit-msg` and `.../prepare-commit-msg` scripts into `.git/hooks` in your local Git repository)
4. [Push your patch to Gerrit](#) for review, and Add at least two reviewers on the Gerrit page for this patch (preferably ones with experience in this area of code, use `contrib/scripts/get-maintainer.pl` for suggestions).
5. Verify the Gerrit change URL in the JIRA ticket (this should happen automatically today for the Lustre project, if missing you may have pushed with the wrong Jira ticket in the commit message).
6. Test your patch and upload test results to the [Maloo](#) test results database (this happens automatically for all patches in the `fs/lustre-release` project).
7. Attach additional test results appropriate for the patch being landed to the JIRA ticket and/or patch commit message (interoperability, performance, etc.) if necessary.
8. Add a comment in Gerrit with the Maloo link to indicate that the patch works, and set the `Verified: +1` flag (this is done automatically for standard test results).
9. Solicit and address review comments from patch reviewers until they have given a positive review (it is the patch submitter's responsibility to direct attention to a patch if they feel it isn't making progress)
10. Once the patch has a `Verified: +1` label from Jenkins and Maloo, and two separate `Code-Review: +1` labels from qualified reviewers, the branch gatekeeper will review the patch, confirm the test results, and submit it to the tree (subject to feature freeze, code freeze, etc.). See [Patch Status](#) page for current status of all in-flight patches.
11. If the submission failed due to patch conflict(s), the `wC Checkpatch` bot will mark the patch `Code-Review: -1` and/or the gatekeeper will ask you in Gerrit to rebase your patch to the head of target branch and repeat above steps 4 to 10.
12. If your patch is useful for other branch(es), please repeat above steps against corresponding branch(es) after the patch has landed to master.