

## Reviewing tests

**i** Document development, review and version history

### ***Development and Review***

	Name	Date
Authored/Revised by	XXXXXXXXXX	xxxx-xx-xx
Reviewed by	YYYYYYYYYY	xxxx-xx-xx
Released by	ZZZZZZZZZZ	xxxx-xx-xx

### ***Version History***

Version	Date	Author	Summary of Changes
0.1	xxxx-xx-xx	XXXXXXXXXX	Initial draft

Once someone has prepared one or more tests for functions in a package and submitted them to be incorporated into the platform via a pull request. They should then be reviewed by another member of the platform, ideally from another unit, to check that the tests are programmed and documented appropriately.



## Who reviews?

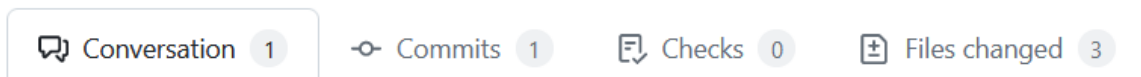
All pull requests are by default automatically assigned to one individual per unit. These individuals should agree among themselves who can perform the review, potentially nominating someone else from their unit. Those that will not be performing the review can be removed from the list of assignees.

### Note

If you need a review urgently, reach out to someone by other means (e.g. email) and arrange that they perform the review for you.

## Performing the review

The review is performed within the pull request on GitHub. There are four main tabs the pull request screen on GitHub:



- The conversation tab is for discussions of general points about the pull request.
- The commits tab lists the individual commits that make up the pull request. For our purposes, this is rarely of use.
- The checks tab shows the results of automated checks that are run on the pull request. As we have no automated checks running for this repository, this tab is also not useful.
- The files changed tab shows the changes that have been made in the pull request. This is where the review is performed.

The files changes lists all changes in all files modified during the pull request. The reviewer should look at each file in turn, checking that the changes are appropriate and that the code is well written and documented. The reviewer should also check that the tests are appropriate and that they test the correct things. Also, ensure that the details of the test in `info.txt` match the tests that were actually performed.

Where there are general points to be made, these should be made in the conversation tab. For specific points, the reviewer can comment on the specific line of code in the files tab.

1. hover over the line to be commented on
2. click on the + that appears
3. type the comment in the box that appears

The screenshot shows a code review interface. At the top, a green header bar contains the text "3 + library(testthat)". Below this is a toolbar with tabs for "Write" and "Preview", followed by icons for adding content, bold, italic, code, link, list, and more options. A large text area with a blue border contains the placeholder text "Leave a comment". Below the text area, there are two status indicators: "Markdown is supported" and "Paste, drop, or click to add files". At the bottom, there are three buttons: "Cancel", "Add single comment", and "Start a review".

4. click on Start a review to submit the comment

To indicate that you have checked a particular file and that it is suitable, you can click on the Viewed box on the top right of each file.

The very top of the page has a box for finalising the review.

0 / 3 files viewed

Review in codespace

Review changes

Finish your review

Write

Preview

H

B

I

≡

<>

🔗

≡

≡

≡

🔗

@

🗨️

↩️

Leave a comment

Markdown is supported

Paste, drop, or click to add files

☒ **Comment**  
Submit general feedback without explicit approval.

☐ **Approve**  
Submit feedback and approve merging these changes.

☐ **Request changes**  
Submit feedback that must be addressed before merging.

Submit review

If there are no issues, the review can be marked with approve. If you have suggestions or require modifications, you can mark it as request changes. If you have questions, you can mark it as comment or request changes, whichever is most appropriate, and enter your questions in the box.

## Incorporating the tests in the repository

Once the tests have been reviewed and found to be suitable and appropriately documented, the pull request needs to be merged into the repository. Each CTU has at least one nominated individual that can perform a merge (typically the same individuals distributing reviews). This individual should check that the review has been approved and then merge the pull request using the green button at the bottom of the conversation tab, followed by the “confirm merge” button that appears afterwards.