

# How to merge a branch back to icon-mpim:master

1. rebase your branch on the current master
2. open a (draft) merge request (MR)
3. initiate the necessary testing and code review
4. upon passing the required tests and reviews (see below) update and finalize merge request

## Opening merge requests

- when opening a MR use 'Draft:' at the beginning of your MR-title, this should only be changed after the code passes the tests and code review.
- the description of the merge request is the default commit message of your MR. Please provide a meaningful text.

## Code review

Before being merged all changes have to pass a technical and (for features and bug corrections) a content review. Features are changes that affect the result or the supported functionality of the model. For many obvious changes (i.e., to configuration files, or to clean-up code) a technical review will be sufficient. Features targeted for merging should be coordinated ahead of time, just to avoid going through the trouble of preparing a merge request for a feature that is not desired. Technical reviews can be assigned to any department member, but ideally people familiar with the coding standards (see `icon_standard.pdf` in the `doc/style` directory of the distribution). Code reviews (of either kind) **should elicit a response within two working days** by a reviewer deemed as competent by the Gatekeepers; if you don't know who might constitute a competent reviewer, coordinate their selection with your supervisor or the Gatekeeper or ask the Gatekeeper to select a reviewer. More elaborate code changes will require more elaborate review, but at the very least a review schedule should be established within the two (working) day period.

## Testing

The default form of testing consists of automated technical tests. These can be run automatically with buildbot and if they are passed, then that is sufficient for a proposed change to be ready to merge. Among the automated tests, one is called an update test. This will fail if the results have been changed (as would sometimes be expected when adding a feature, or fixing a bug). In case the update test fails:

- a new reference experiment has to be performed and compared to the existing reference data. If the differences are consistent with the intended changes this must be documented and filed for the content review.
- subject to a successful content review, buildbot can be used to generate new tolerance data and the update test should be repeated.

If other tests fail, the issues they raise need to be resolved before the code is ready to be merged.

## Finalizing the MR

Once all the tests and code reviews are successful, the branch must be rebased on the current state of the master. When rebasing no comment like 'source branch is xx commits behind target branch' may appear in your merge request. If a rebasing is necessary the technical tests will need to be redone and passed before finalizing the MR. Only now are you are ready to finalize the MR, to do so, remove the [Draft] in the title of the request and notify the Gatekeeper

## Role of the Gatekeepers

The gatekeepers manage the merge process. They decide if the merge requirements are satisfactorily met, and follow up on open requests to nudge the process forward, or close stale requests. They can decide to refer questionable MRs to the department leadership, and their decisions can likewise be appealed. Coordinating complex developments with the gatekeepers well in advance of their eventual MR is recommended.

From:  
<https://wiki.mpimet.mpg.de/> - **MPI Wiki**

Permanent link:  
[https://wiki.mpimet.mpg.de/doku.php?id=models:icon:development:necessary\\_conditions\\_to\\_merge\\_a\\_brach\\_back\\_to\\_master](https://wiki.mpimet.mpg.de/doku.php?id=models:icon:development:necessary_conditions_to_merge_a_brach_back_to_master)

Last update: **2022/11/09 10:25**

