

# Branching

This page describes the process of creating stdcxx release branches, and merging changes between them.

1. Each release branch reflects the minor version of the library and contains changes for all micro releases for that minor version. The name of the branch is `branches/<major>.<minor>.x` where `<major>` and `<minor>` are the major and minor version numbers, respectively, for the release. The letter `x` denotes all micro versions. For example, [branches/4.2.x](#) is a branch for all of 4.2.0, 4.2.1, 4.2.2, and so on. Similarly, [branches/4.3.x](#) is a branch for all of 4.3.0, 4.3.1, etc.
2. `trunk` corresponds to the next major version in development. When it comes time to start the release process for the next major version, a new branch for the version is created. The name of the branch follows the naming convention listed above. For example, when it comes time to release version 5.0.0, `branches/5.0.x` will be created off of `trunk`. At that point, `trunk` becomes the future 6.0.0.
3. Changes are first committed to the branch for the lowest appropriate version number and then merged upward. Specifically, given the two branches, `4.2.x` and `4.3.x`, and `trunk`, changes that are both backward and forward compatible are first made on `branches/4.2.x` and then merged to `branches/4.3.x` and then to `trunk`. Changes that are forward but not backward binary compatible are first made on `branches/4.3.x` and then merged to `trunk`. Incompatible changes are made only on `trunk`.
4. Merges between branches are done often, no less frequently than once a month, to minimize the tedium involved and the risk of introducing mistakes or missing important changes.
5. The `ChangeLog` for each merge must mention the revision numbers of all changes merged and contain the corresponding `ChangeLog` entry for each. An example of a such a `ChangeLog` entry is this [commit](#).
6. When merging a change that itself resulted from a merge from another branch, only the original revision number and `ChangeLog` entry should be mentioned in the new `ChangeLog` entry.