

## Git Branches in Multirooted Projects

---

Sometimes you may need to have [several local Git repositories](#) within a single project. Such multirooted project can be set up in the following ways:

- The repositories are *independent*, they can be included in the project either as submodules or just as nested roots. One of these repositories can be intended for the project development itself while the others store the libraries used.

Such repositories are controlled [asynchronously](#), that is, commit, push, and other operations can be applied to them independently, just as it is done in projects with one Git repository.

- The repositories are *mutually connected*. Such repositories most often need to be controlled synchronously, that is, commit, push, and other operations should be applied to all these repositories at once. For such projects IntelliJ IDEA also allows you to control branches [synchronously](#). This means that all branch operations on several repositories are performed simultaneously as if it were a single repository. In case an operation fails in at least one of the repositories, IntelliJ IDEA [rolls it back](#) in the repositories where the operation has passed successfully.

In this topic:

- [Asynchronous Branch Control](#)
- [Applying a Git operation to a branch asynchronously \(in one of the repositories\)](#)
- [Synchronous Branch Control](#)
- [Disabling synchronous branch control on and off](#)
- [Enabling synchronous branch control on and off](#)
- [Rolling back a successful operation if it fails in some of the repositories](#)

### Asynchronous Branch Control

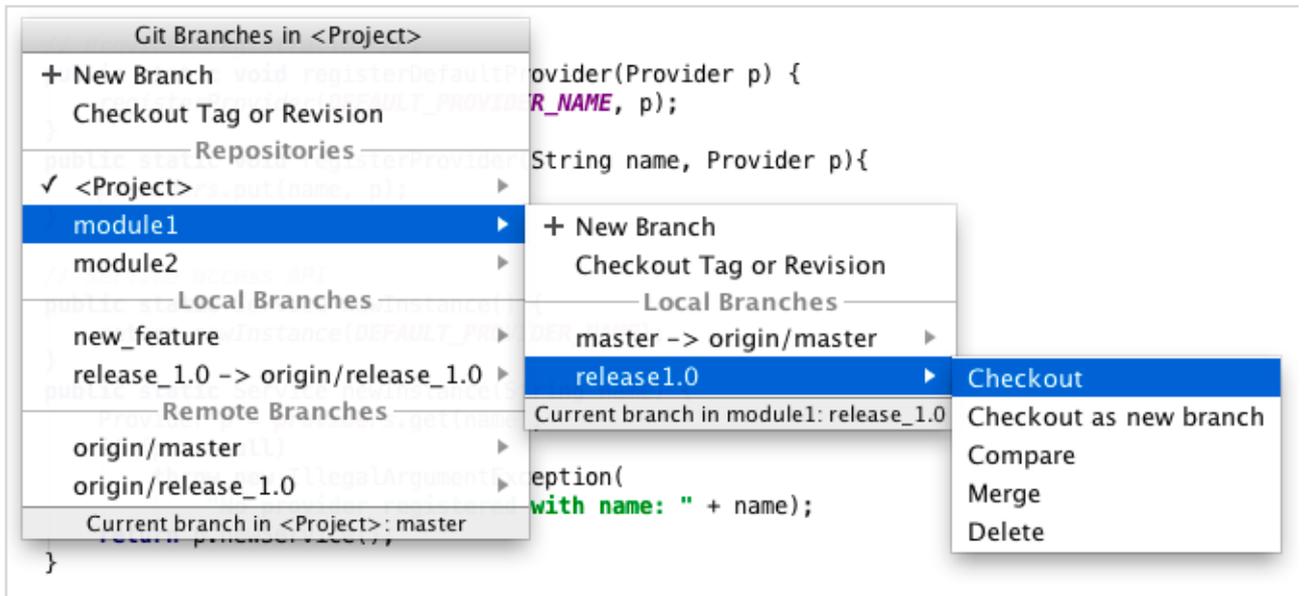
*Asynchronous* control is applied to the branches in a multirooted project when all local Git repositories in the project are *independent*. Commit, push, and other operations to the branches in the repositories are applied independently, just as it is done in projects with one Git repository.

## To apply a Git operation to a branch asynchronously (in one of the repositories)

1. Open the [Git Branches](#) popup menu.
2. In the **Git Branches** popup menu that opens, choose the branch to apply the required action to. The popup menu shows a list of branches in the current repository and a list of all the repositories in the project.

The current repository is detected in the following way:

- If you have explicitly selected a file or a folder in the **Project** tool window or elsewhere, IntelliJ IDEA treats the root under which the file is located as *current repository*.
- If you used the **VCS | Git | Git Branches** command on the main menu, IntelliJ IDEA treats the root of the file currently opened in the editor as *current repository*.



Do one of the following:

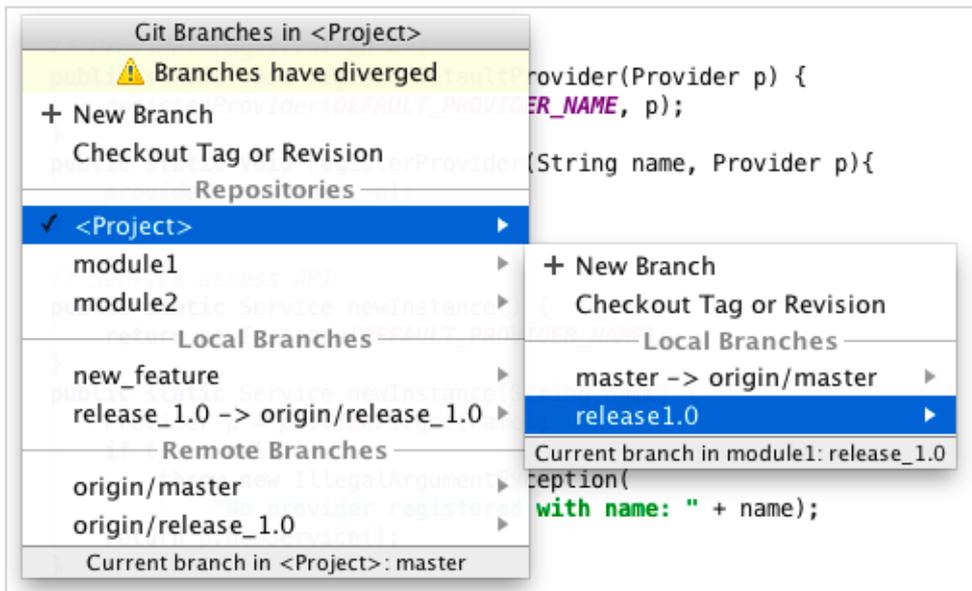
- To update a branch in the current repository, select the branch in the list.
  - To access another repository, select it under the **Repositories** group, and then select the relevant branch in the subordinate popup menu that opens.
3. Click the branch and choose the required [action](#) from the popup menu that opens.

## Synchronous Branch Control

In a multirooted project when several *mutually connected* repositories, it is often helpful to apply *synchronous* branch control, which involves the following:

- All branch operations on several repositories are performed simultaneously as if it were a single repository. This applies to all actions including [checkout](#), [merge](#), and [delete](#). If you invoke the [compare](#) action, IntelliJ IDEA shows a dialog box where you are asked to choose the repository to compare the branch with.
- The **Git Branches** popup lists only those local and remote branches that are common for all the repositories in the project.
- Creating a new branch from the popup will create it on all the repositories.

- You may still control branches individually for each repository by using the repository selector in the **Git Branches** popup menu. If you happen to checkout a branch only in one of the repositories, you'll get the warning that branches have diverged:



### To disable synchronous branch control, do one of the following

When you [invoke the Git Branches](#) popup menu for the first time, IntelliJ IDEA examines the repositories registered in the project. If they all are on branches with same names, IntelliJ IDEA displays a popup menu that proposes to control branches synchronously.

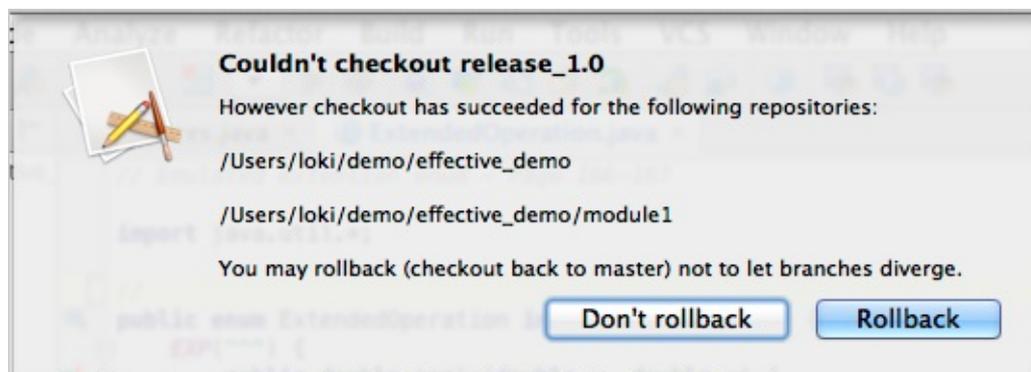
- Click the **Disable** link in the information popup window.
- [Open the project settings](#), and on the **Git** page under the **Version Control** node, clear the **Control branches from different roots synchronously** check box.

### To enable synchronous branch control

- [Open the project settings](#), and on the **Git** page under the **Version Control** node, select the **Control branches from different roots synchronously** check box.

### Rolling back a successful operation if it fails in some of the repositories

If you checkout a branch on several roots and the operation fails for some of the roots but has already succeeded for the others, the branches diverge. For example, the `master` branch is checked out in one repository while the `feature` branch is checked out in the other. To prevent this, IntelliJ IDEA offers to rollback the successful checkout operation, so the the previous checked out branch is checked out again.



Besides checkout, IntelliJ IDEA suggests to rollback merge and other branch operations.

### See Also

Procedures:

- [Managing Branches](#)

**Reference:**

- [Version Control Reference](#)
- [Checkout Dialog](#)
- [Merge Branches Dialog](#)

**External Links:**

- <http://blogs.jetbrains.com/idea/2012/08/git-branches-for-multi-root-projects/#more-6754> 

**Web Resources:**

- [Developer Community](#) 