

## Rebasing a Branch to a Specific Commit

---

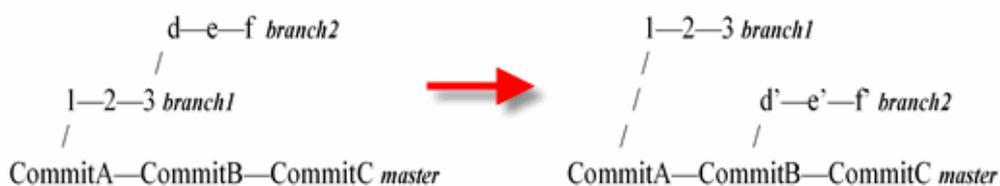
Suppose you have a branch `branch1` based on `master`. While you are working in `branch1`, some updates have been committed to `master`. For some reason, you do not want to rebase `branch1` to the current HEAD of `master` but to commit `CommitB`. Moreover, you do not want to apply commit 1 but want to start with commit 2.

The diagram below illustrates the described rebasing operation.



Transplanting a branch to a branch different from its base is also supported. Suppose your `branch2` is based on `branch1`, which, in its turn, is based on `master`.

The diagram below illustrates rebasing `branch2` to a specific commit in `master`.



## To rebase a branch to a specific commit

1. [Initiate the rebase](#) procedure.
2. Configure the rebase mode:
  - To [interactively](#) handle the list of commits to apply, select the **Interactive** check box.
  - To have IntelliJ IDEA try to recreate merges instead of ignoring them, select the **Preserve Merges** check box.

This option is available only in the interactive mode.

3. Define the contents of the **Onto** and **From** drop-down lists. By default, tags and remote branches are not available from them. To expand the lists with tags or remote branches, select the **Show Tags** and **Show Remote Branches** check boxes respectively.
4. From the **Onto** drop-down list, select the required branch and specify the commit to move the base to. Type the desired commit hash or use an expression, for example, of the following structure:

```
<branch>~<number of commits backwards between the latest commit (HEAD) and the required commit
```

Refer to the Git [commit naming](#) conventions for details.

If you select a tag, no commit specification is required.

5. Click the **Validate** button and view which files are affected in the specified commit in the **Paths affected in commit...** dialog box that opens.
6. From the **From** drop-down list, select the current branch and specify the commit, starting from which you want to apply commits on top of the new base. Type the desired commit hash or use an expression, for example, of the following structure:

```
<branch>~<number of commits backwards between the latest commit (HEAD) and the required commit
```

Refer to the Git [commit naming](#) conventions for details.

To apply all commits, leave the field empty.

7. Specify the merge procedure to be applied, when necessary:
  - To perform all merges manually, select the **Do not use merging strategies** check box.
  - To have IntelliJ IDEA apply one of the available merging methods, select the relevant option from the **Merge Strategy** drop-down list. See the [Rebase Branches](#) dialog box reference for description of available options.
8. Click the **Rebase** button and proceed depending on the chosen rebase mode:
  - If you have specified the [interactive](#) mode, configure the list of commits to apply in the [Rebasing Commits](#) dialog box, that opens.
  - If you have specified the automatic mode, view the rebase log in the [Version Control](#) tool window, [resolve conflicts](#) that arise, and [resume](#) rebasing

## See Also

### Concepts:

- [Version Control with IntelliJ IDEA](#)

### Procedures:

- [Rebasing Branches](#)
- [Applying a Branch Entirely on Top of Master](#)
- [Interactive Rebase](#)

**Reference:**

- [Version Control Reference](#)
- [Rebase Branches Dialog](#)
- [Rebasing Commits Dialog](#)

**Web Resources:**

- [Developer Community](#) 