

# Lab: (Command Line) Rebasing

Estimated time: 10 minutes

Note: This lab assumes that you are using a command line. If you would prefer to use Sourcetree, there are separate instructions.

In this lab, you will:

1. Rebase a topic branch.
2. Rebase and resolve a merge conflict.

## 1: Rebase a topic branch.

1. Create a local repository named `projectg`.
2. Create an initial commit that adds an empty `fileA.txt` file. The commit message should be "add fileA.txt".
3. Create a `feature1` branch.
4. Create a commit on the `feature1` branch. Add the string "feature 1 wip" to `fileA.txt`. The commit message should be "add feature 1 wip".
5. Checkout the `master` branch. Create a commit that adds an empty `fileB.txt` file. The commit message should be "add fileB.txt".
6. View your commit graph and notice that it is no longer linear. Be sure to use the `--all` option with `git log`.
7. Rebase the commit on the `feature1` branch to the tip of the `master` branch. Do this by first checking out the `feature1` branch. Then execute `git rebase master`.
8. View the commit graph. It should now be linear. Verify that you can see `fileB.txt` in your working tree of `feature1`.

Congratulations, you have rebased a topic branch.

## 2: Rebase and resolve a merge conflict.

1. Checkout the `master` branch.

2. Create a commit on the `master` branch. Add the string "feature 2" to `fileA.txt` . The commit message should be "add feature 2".
3. Rebase the `feature1` branch onto the tip of the `master` branch. This will involve resolving a merge conflict, because the `master` branch and the `feature1` branch both modified the same part of `fileA.txt` in different ways.
  - Checkout the `feature1` branch.
  - Execute `git rebase master` . Notice that there are merge conflicts.
  - Execute `git status` to view the status of the rebase.
  - Fix the content of `fileA.txt` so that "feature 1 wip" is before "feature 2".
  - Stage the fixed file.
  - Execute `git rebase --continue` .
4. View your commit graph. Verify that you again have a linear graph.
5. You will not use the `projectg` repository in future labs. You can delete it.

Congratulations, you have performed a rebase involving a merge conflict and have completed this lab.