**Resolve Merge Conflicts with Binary Files**

Mar. 5, 2020

POSTED IN: [Snippets](https://www.hannaliebl.com/blog/categories/snippets) TAGS: [Git](https://www.hannaliebl.com/blog/tags/git)

In an effort to write more regularly, I’m going to do “snippets,” much smaller posts about something I learned or thought was useful enough to remember by writing it down publically. This is the first one!

I find this isn’t something that happens very often and so I forget how to do it: resolving a merge conflict with binary files. This has happened to me with image snapshots from tests. You update snapshots in your branch, but (assuming you are branched off of master), someone has updated master while you were on your branch and updated those same snapshots. You try to merge master into your branch and get the message:

warning: Cannot merge binary files: a-snapshot.jpg (HEAD vs. master)

Auto-merging a-snapshot.jpg

CONFLICT (content): Merge conflict in a-snapshot.jpg

Automatic merge failed; fix conflicts and then commit the result.

You can’t just go through the image file and take parts of it from their branch that you want and parts of it from your branch that you want like you would with other text-based files in Git. Instead, you need to decide which version you want: theirs (in this case, master) or yours.

**If you want their changes:**

git checkout master a-snapshot.jpg (replace master with whatever their branch is if it isn’t master.)

**If you want your changes:**

git add a-snapshot.jpg

Then, continue with your merge. Usually I just do git commit to trigger the default git merge message, unless there are other conflicts to resolve.

Sidenote: managing binary files in git is something I’m not super familiar with, but is something people may want to do, and likely demands some more custom solutions. [This article](https://opensource.com/life/16/8/how-manage-binary-blobs-git-part-7) describes some of those potential solutions.