



[Create your free blog at Obolog!](#) or [Sign in](#)

[Report inadequate content](#)

- [Home](#)
- [Archives](#)
- [Tags](#)
- [Pictures](#)
- [Contact](#)
- [Subscribe](#)

Search:



[Harecoded](#)

Simple recipes for web developers

- 23

Me gusta
- Twittear
- G+

Merging two, three or more git repositories keeping the log history



If you ask anyone, merging several git repositories into a single one would usually be considered a bad strategy: as a rule of thumb, your code base should have several components isolated and then use a dependency manager (e.g: [Composer](#)) to bring them together.

Nevertheless, there are always many different scenarios and with them come different use cases, weird situations or just plain normal activities that would justify this procedure. In any case if you ever need to merge several repos into one, and still want to **keep the git history** this is an easy way to do it.

The following technique is not rocket science and it's based in the **simple addition of remotes and branch merging**. If you expected something fancier I am sorry to tell you you'll find this boring.

Assumptions

- You have two, three (or N) repositories you want to merge inside a single (new) one.
- All the commit history of the three will be kept and browseable in the new one
- You want to merge everything in an empty repo named repository-all-merged
- repository, repository2 and repository3 are the ones you want to glue together
- Every merged repo will end-up inside a new folder, wherever it is. If you do not want this, just skip the early steps.
- The master branch or any others will be present in the new repo as well.

Preparing the repositories

As we said, we are going to put every repo in a different folder inside the new one. This is what you need to do (you can copy sequentially the commands writing the right paths):

Create a new directory and clone everything inside. The final repo and the existing ones. E.g:

```
1. mkdir ~/all_repos_merged
2. cd all_repos_merged
3. git clone git@bitbucket.org:alombarte/repository-all-merged.git
4. git clone git@bitbucket.org:alombarte/repository.git
5. git clone git@bitbucket.org:alombarte/repository2.git
6. git clone git@bitbucket.org:alombarte/repository3.git
```

Now go to the first repository you want to import and create a new folder, and git-move all the content inside. The folder structure you create now is the folder you'll see in the merged repo. In the following example this first repo would appear under the path/to/repository folder.

```
1. cd repository/
2. mkdir -p path/to/repository
3. git mv -k * path/to/repository/
4. git commit -m "Merging repositories like a boss"
```

Note that the -k flag ignores errors such as moving the newly created folder to itself. Before moving on with the next operation you might want to check that you don't have in each repos conflicting files, such as the .gitignore. You can delete those files and restore them later or you just fix the conflicts when they arise (edit the file, git add and then git commit: conflict solved).

Now we prepare the second, third, forth... repository. Repeat the same N more times:

```
1. cd ../repository2/
2. mkdir -p path/to/repository2
3. git mv -k * path/to/repository2/
4. git commit -m "Merging repositories like a boss"
5.
6. cd ../repository3/
7. mkdir -p path/to/repository3
8. git mv -k * path/to/repository3/
9. git commit -m "Merging repositories like a boss"
```

CAUTION: You should never PUSH this changes to origin.

Commits are local in your computer and remember you want a new repo, not mess around with the existing ones.

Now the three repos are "ready". Meaning they are all inside a new folder and content is not mixed.

Merging all the repos together in the new one

Next steps are having the just prepared repositories available for importing code in the final repository. With git you can reference any other external repository with the "remotes".

Go to the final repo:

```
1. cd ../repository-all-merged/
```

Add all the remotes for merging named repository, repository2, repository3... Note that although being a "remote" this repos are the ones you cloned in your local computer:

```
1. git remote add repository ~/all_repos_merged/repository
2. git remote add repository2 ~/all_repos_merged/repository2/
3. git remote add repository3 ~/all_repos_merged/repository3/
4.
```

And this is the moment when we actually bring the changes to the final repo:

```
1. git checkout master
2. git pull repository master
3.
```

Now we have in the master branch all the changes of repository/master branch. Conflicts might arise at this point (specially if you didn't do anything with the gitignore), but fixing them is as easy as:

```
1. vi .gitignore
```

2. `git add .gitignore`
3. `git commit`

And repeat for every other repo:

1. `git pull repository2 master`
2. `git pull repository3 master`
3. `...`

Check that everything is in place:

1. `git log --oneline --graph --decorate --all`

The repository-all-merged contains now all the code with all the history of the master branches. If you for instance need to import more branches it would be as easy as:

1. `git checkout another_branch`
2. `git pull repositoryN another_branch`

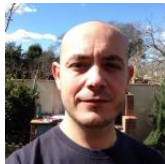
If this merged repo has a remote, you can clean and push it now:

1. `git remote rm repository`
2. `git remote rm repository2`
3. `git remote rm repository3`
4. `git push`

The rest of the repositories can be deleted, so you don't push by mistake later on.

🔖 [development git](#)

author:



[Albert](#)

📅 Date:

1/10/2014

💬 Comments

[Leave a comment](#) (3)

Related to Merging two, three or more git repositories keeping the log history

```

el) $
el) $ git chec
ck remote bran
ategy'
ture/quality-s

```

- Terminal tuning for Git developers in Mac
- Upload an existing Git repository to a remote GitHub, Bitbucket, Beanstalk...



- Migrating a Github repo to Bitbucket (or similar services)



- Organizing git branches in logical folders

Related tags

- php
- development
- git
- linux
- mac
- software
- trucos
- mysql
- svn
- tricks
- bash
- subversion
- terminal
- varnish
- vim

💬 Comments Merging two, three or more git repositories keeping the log history

hi Albert,

Thanks for sharing this.

I followed this method and I can see all the history on top-level folder of new repository but when I check history on a file or folder, all I see is the commit message used to import the old repo. Though the "blame" tab shows history of a file. Did you face this issue?



bhavin 21/04/2015 at 20:23

Where's the repository-all-merged.git come from in the preparation step? Is it a new one we shall create? The rest of the steps are confusing as well, I cannot easily follow what's happening.



tsomogyi 27/08/2015 at 15:55

With newer versions of git, you need to use the `--allow-unrelated-histories` flag.



JC 12/08/2016 at 04:06

💬 Leave your comment Merging two, three or more git repositories keeping the log history

📘 [Log in](#) to **Obolog**, or [create your free blog](#) if you are not registered yet.




1. Name



2. mail



3. Web


You





Your comment

1. con

2.



Your name

-  [Home](#)
-  [Archives](#)
-  [Tags](#)
-  [Pictures](#)
-  [Contact](#)
-  [Subscribe](#)

Search:



Harecoded - blog made with the free blogging platform **OboLog**.

[Create free blog at Obolog](#) | [Privacy & Terms of use](#)