

How to remove files that are listed in the .gitignore but still on the repository?

237 I have some files in my repository that should be ignored, i added them to the .gitignore but, of course, they are not removed from my repository.

So my question is, is there a magic command or script using filter-branch that can rewrite my history and remove all these files easily? Or simply a command that will create a commit that will remove them ?



git ignore gitignore

139

edited Mar 11 '14 at 17:37



Robert Harvey ♦

151k 36 287 425

asked Nov 24 '12 at 13:22



Intrepid

12k 5 43 59

3 Duplicate of [.gitignore file not ignoring](#) – user456814 May 25 '14 at 19:46

Similar to [stackoverflow.com/questions/1274057/...](#) – testing Apr 6 '16 at 11:00

1 Possible duplicate of [How to make Git "forget" about a file that was tracked but is now in .gitignore?](#) – Steven M. Vascellaro Nov 8 '17 at 16:52

WARNING: While this will not remove the physical file from your local, it will remove the files from other developers machines on next git pull. [How to make Git "forget" about a file that was tracked but is now in .gitignore?](#) – Kris Roofe Apr 12 at 7:14

5 Answers

You can remove them from the repository manually:

341

```
git rm --cached file1 file2 dir/file3
```

Or, if you have a lot of files:

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

But this doesn't seem to work in Git Bash on Windows. It produces an error message. The following works better:

```
git ls-files -i --exclude-from=.gitignore | xargs git rm --cached
```

Regarding rewriting the whole history without these files, I highly doubt there's an *automatic* way to do it. And we all know that rewriting the history is bad, don't we? :)

edited Jan 15 '14 at 19:30

answered Nov 24 '12 at 13:34



[Samy Dindane](#)

12.9k 3 31 45

1 Unfortunately the Git Bash on Windows command doesn't work with paths that contain spaces – [Nate Bundy](#) Apr 23 '14 at 18:50

@NateBundy if you're referring to the fact that `xargs` won't work with spaces, most command line utilities get around that by using special flags so that whitespace won't matter. Off the top of my head I don't remember what the flags are for `git ls-files` and `xargs` (I think it might be `-0` for `xargs`), but you can look them up. – [user456814](#) May 25 '14 at 19:50

13 @Cupcake thanks. `git ls-files -i -z --exclude-from=.gitignore | xargs -0 git rm --cached` seems to do the trick – [Nate Bundy](#) Jun 16 '14 at 8:52

3 I got stuck with the same problem on windows. I'm using powershell, and in my case I got the third command tipped by @samy-dindane and changed it into a foreach. It became this `git ls-files -i --exclude-from=.gitignore | %{git rm --cached $_}` – [digaomatias](#) Jul 26 '14 at 21:28

wrong, wrong, and wrong. :(sorry no insight, yeah git needs to add a feature that doesn't make this particular thing - which totally sucks ass and eats up tons of coding time - not suck so bad – [StackAttack](#) Oct 10 '17 at 23:46



An easier way that works on any OS is to do

269



```
git rm -r --cached .
git add .
git commit -m "Removing all files in .gitignore"
```

You basically readd all files, except the ones in the .gitignore

answered Dec 23 '15 at 11:58



[gtatr](#)

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

How does this effect other users on the repo executing a *pull* - as I read it the folder we dont want to track anymore is deleted??? How can we prevent this - we just want to untrack – [snh_nl](#) Nov 3 '17 at 10:36

1 This should be accepted answer. – [Michael Ecklund](#) May 30 at 13:38



As the files in **.gitignore** are not being tracked, you can use the **git clean** command to recursively remove files that are not under version control.

113



Use `git clean -xdn` to perform a dry run and see what will be removed.

Then use `git clean -xdf` to execute it.

Basically, `git clean -h` or `man git-clean` (in unix) will give you help.

Be aware that this command will also remove **new files** that are not in the staging area.

Hope it helps.

edited Sep 21 '16 at 18:50



[Brian Topping](#)

2,777 23 31

answered Apr 12 '16 at 12:45



[weiz](#)

1,475 1 7 5

2 This should be the accepted answer. It actually works (the accepted answer didn't work for me, on macOS), and it's way cleaner. – [Roger Oba](#) Sep 21 '16 at 17:33

16 This answer isn't applicable - the OP says the files in `.gitignore` are being tracked. – [Ken Williams](#) Dec 15 '16 at 23:15

9 BEWARE! This permanently deletes all untracked files, rather than just removing from the branch – [Emmanuel](#) Mar 6 '18 at 6:38

the `git clean -xdn` is a dry run which won't delete. the next one will. – [JohnZaj](#) Dec 18 '18 at 22:30


3 -1: This is a highly misleading answer - the original poster wanted to remove from the repository, not remove the files completely. I was this close to deleting a load of dynamic files required by my IDE but not required to be in the repo. – [Auspice](#) Dec 30 '18 at 14:11



I did a very straightforward solution by manipulating the output of the `.gitignore` statement with `sed`:

```
cat .gitignore | sed '/^#.* / d' | sed '/^\s*$/ d' | sed 's/^/git rm -r /' | bash
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

- 
1. print the .gitignore file
 2. remove all comments from the print
 3. delete all empty lines
 4. add 'git rm -r ' to the start of the line
 5. execute every line.

answered Sep 29 '17 at 17:54

[Scott Crossen](#)

41 1


3 sed is straightforward? – [IgorGanapolsky](#) Sep 13 '18 at 10:52



The git will ignore the files matched .gitignore pattern after you add it to .gitignore.

-2

But the files already existed in repository will be still in.



use `git rm files_ignored; git commit -m 'rm no use files'` to delete ignored files.

answered Nov 24 '12 at 13:31

[pensz](#)

1,484 9 15

1 There are a lot of them, is there a way to delete them without having to specifying their names? – [Intrepidd](#) Nov 24 '12 at 13:36
