

A lock file already exists in the repository, which blocks this operation from completing

Asked 2 years, 10 months ago Modified 5 months ago Viewed 66k times



I'm working on an ASP.NET Core project. I'm trying to `push` a commit to my branch using GitHub Desktop. But every time I try to commit, I keep getting this error:

27



A lock file already exists in the repository, which blocks this operation from completing.



Upon searching the internet, the only solution I've found is to delete `index.lock` file inside `.git` folder manually.

But when I delete this file, it keeps re-appearing every time I try to commit, making it impossible for me to commit. I have already tried to commit from Visual Studio but it also failed.

Can anyone help me?

[git](#) [git-commit](#) [github-desktop](#)

Share Follow

edited Mar 3, 2021 at 4:44



tripleee

179k

35

276

323

asked Mar 3, 2021 at 0:52



Chrys

271

1

3

4

- 3 It seems some other process is constantly creating the lock file. Try restarting your computer. Also try a new clone and check if it works in the new cloned repository. – [ElpieKay](#) Mar 3, 2021 at 2:36

Be careful to not just delete your clone in case you have stashed changes on a branch. You will lose them. Deleting `index.lock` as mentioned below should work. – [Akaisteph7](#) Jul 13, 2023 at 16:33

4 Answers

Sorted by: Highest score (default)



Just delete the file called `index.lock` which is in the `.git` folder. After you delete it, try to commit again.

43

Steps



Join Stack Overflow to find the best answer to your technical question, help others answer theirs.

Sign up with email



Sign up with Google

Sign up with GitHub

Sign up with Facebook



3. There will be a `.git` folder shown to you. Just go and delete the `index.lock` file from there and you are ready to commit again.

Share Follow

edited Jul 13, 2023 at 16:38

answered Dec 25, 2021 at 2:58



Akaisteph7

5,334 2 21 44



GigaTera

874 9 10

1 Easy, logical and summarized answer. Thank you. – Omrum Cetin Aug 10, 2022 at 22:05

This file should be `.git/index.lock`. It might not appear in your editor (e.g. VSCode) though.

– Akaisteph7 Jul 13, 2023 at 16:32



20



Git uses `index.lock` to know that some Git process—perhaps this one, if it just created `index.lock` itself, or perhaps some other Git process—is busy doing things with the repository that require locking Git's index. When a Git successfully creates this lock file, it then goes about its business, updating various internal databases, until it is done. It then *deletes* this lock file to indicate that it is done, and *other* Git commands can now get started, do their things, and finish.

If there is a *bug* in your Git system, it's possible for a Git command to create `index.lock`, start doing its work, and then crash, leaving the lock file behind. In this case, the correct thing to do is to upgrade to a non-buggy Git so that the problem does not recur, and also to remove the `index.lock` file. (You can do this in either order but if the bug remains, some Git command(s) may leave the bogus lock behind.) Due to a fairly extensive test suite, this kind of bug is rare in Git these days—in the bad old days, it was more common, but now it should hardly ever occur.

If your *computer itself crashes* (due to a non-Git bug, or the power failing, or whatever) while Git is in the middle of one of these operations, it's possible that Git never gets a chance to remove the `index.lock` file. In this case, you just need to remove the `index.lock` file manually. However, after a computer crash, *other* parts of Git's internal databases might have been damaged by the crash, so it's wise to run `git fsck`. The `git fsck` program prints a number of messages even on healthy systems so don't be too alarmed about various "dangling commit" or "dangling blob" messages: these are just informative. (In general and ideally, your computer itself should never crash at all and your power should never fail, and people should not have frozen to death in Texas in February 2021, but the world is not always ideal.)

(Windows systems sometimes see bad behavior from some antivirus software. This might go away on its own in some cases. I avoid Windows, so I have no specific recommendations here, other than "avoid Windows".)

Last, there's a problem that has become a common issue lately for some people because

Join Stack Overflow to find the best answer to your technical question, help others answer theirs.

Sign up with email



`index.lock` files or even damaging Git's internal databases. Don't do this! Keep the Git *repository* in an un-shared, local-disk area.

¹Well, at least superficially. There really are advantages, but personally I subscribe to the [Lamport definition of a distributed system: A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable.](#)

Share Follow

answered Mar 3, 2021 at 6:21



torek

459k

61

667

803



2

The `index.lock` file kept reappear most likely because either GitHub Desktop or another programme (e.g. Visual Studio) or process is still using it.

In rare scenarios, this could be a bug in your Git.



In case you cannot easily identify the culprit and prevent the reappearance behaviour, you should consider restarting your computer then delete the `index.lock` file before you launching GitHub Desktop and or your Visual Studio.



As an [alternative](#), you may consider to:

- Stash changes,
- Move to another branch,
- Move back,
- Apply stash, and
- Push again.

Share Follow

edited Jun 2, 2021 at 7:03

answered Jun 2, 2021 at 6:57



nyedidikeke

7,127

8

44

60



1

There is a folder called **HEAD.lock** inside your git folder. **Delete** and retry to commit. that is work for me

Share Follow

answered Nov 24, 2021 at 6:52



Tharindu Nilakshana

31

2



Join Stack Overflow to find the best answer to your technical question, help others answer theirs.

Sign up with email



Join Stack Overflow to find the best answer to your technical question, help others answer theirs.

Sign up with email

