

'<Branch>' is already checked out at '</other/location>' in git worktrees



I started using git worktrees. It seems to work, but I'm getting this error when attempting to check out a branch in the cloned worktree:

19

fatal: '<branch>' is already checked out at '</other/location>'



How do I get around this without deleting the `.git/worktrees` directory?



3

[git](#)[git-worktree](#)

asked Jan 9 '17 at 9:59



[Harald Nordgren](#)

5,858 2 19 44

Then `git worktree move` can be of interest (with Git 2.17+, Q2 2018): see [my answer below](#). – VonC Mar 17 '18 at 0:25

5 Answers



Git won't let you check out the same branch twice, because if you do, and then go to one of the two work-trees and make a new commit, you'll set yourself up for misery when you go back to the *other* work-tree.

18

If you have actually removed the other work-tree, simply run `git worktree prune` to make Git realize this. If you have *not* actually removed the other work-tree, don't check it out twice: it's no fun.



answered Jan 9 '17 at 11:00



[toreK](#)

212k 21 270 356

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google



- 1 I'd still say "don't do it". You can get the same effect without the risk: check out a new branch pointing to the same commit as the existing work-tree. Make your new commit on the new branch, and push the new branch to the upstream server under the old name: `git push <remote> <newbranch>:<oldname>`. Now you can `git fetch && git merge --ff-only` (or anything similar or equivalent) to pick up the changes in the other work-tree. — [torek](#)
Jan 9 '17 at 11:07
-

▲ How do I get around this without deleting the `.git/worktrees` directory?

2

You will have an easier time with Git 2.17+ (Q2 2018), since " `git worktree` " has learned ' `move` ' and ' `remove` ' subcommands.

▼ See [commit 7f19def](#) (04 Mar 2018) by [Eric Sunshine](#) ([sunshineco](#)).

See [commit ee6763a](#), [commit cc73385](#), [commit 78d986b](#), [commit c64a8d2](#), [commit 9f792bb](#), [commit 9c620fc](#) (12 Feb 2018), and [commit 4ddddc1](#) (24 Jan 2018) by [Nguyễn Thái Ngọc Duy](#) ([pclouds](#)).
(Merged by [Junio C Hamano](#) — [gitster](#) — in [commit bd0f794](#), 14 Mar 2018)

In your case, you could *move* the existing worktree to the place you now wants it (when attempting to create a new worktree for the same branch).

`worktree move` : **new command**

This command allows to relocate linked worktrees.
Main worktree cannot (yet) be moved.

And:

`worktree move` : **refuse to move worktrees with submodules**

Submodules contains `.git` files with relative paths.
After a `worktree move` , these files need to be updated or they may point to nowhere.

This is a bandage patch to make sure " `worktree move` " don't break people's worktrees by accident.
When `.git` file update code is in place, this `validate_no_submodules()` could be removed.

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH

 Google

Facebook 

edited Jan 23 at 23:35

answered Mar 17 '18 at 0:24



VonC

873k 315 2822
3377

Since you cannot checkout twice in both worktree and the original repository. How about checkout original repo to somewhere else before you checkout the worktree?

1



```
git -C </other/location> checkout <branch>~1
git -C <worktree> checkout <branch>
```

answered Sep 17 '18 at 1:53



Ramius

61 6



Simply go to the `worktree` directory of your desired branch, and it automatically `checkout` for you.

0



In my case, I have two long-running `worktree` that means two relevant branches beside the `master` .

```
$git branch
master # base stuff here
version-silver # some normal features
version-gold # some better features
```

There is one repository, but I have 3 separate folders beside each other for each branch above. And make the common changes in `master` . then merge it with both other versions.

Specific changes of each version will go in the corresponding folder as well, and the works on each project are isolated and IDE wouldn't be confused.

.. .. .

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google



▲ Note that this also happens if your `$pwd` has links in it. `git` should probably `readlink -f` on the `$pwd` prior to checking.

0

▼ Edit: Or this may indeed be because I missed to call `git worktree prune` . Now it's working.

edited Sep 4 '18 at 8:12

answered Sep 4 '18 at 7:46



[Sebastian Graf](#)

1,780 1 17 29

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH

 Google

Facebook 