

.gitignore and “The following untracked working tree files would be overwritten by checkout”

▲ So I added a folder to my .gitignore file.

766 Once I do a git status it tells me

▼ # On branch latest
nothing to commit (working directory clean)

★ 241 However, when I try to change branches I get the following:

```
My-MacBook-Pro:webapp marcamillion$ git checkout develop
error: The following untracked working tree files would be overwritten by checkout:
public/system/images/9/thumb/red-stripe.jpg
public/system/images/9/original/red-stripe.jpg
public/system/images/8/thumb/red-stripe-red.jpg
public/system/images/8/original/red-stripe-red.jpg
public/system/images/8/original/00-louis_c.k.-chewed_up-cover-2008.jpg
public/system/images/7/thumb/red-stripe-dark.jpg
public/system/images/7/original/red-stripe-dark.jpg
public/system/images/7/original/DSC07833.JPG
public/system/images/6/thumb/red-stripe-bw.jpg
public/system/images/6/original/website-logo.png
public/system/images/6/original/red-stripe-bw.jpg
public/system/images/5/thumb/Guy_Waving_Jamaican_Flag.jpg
public/system/images/5/original/logocompv-colored-squares-100px.png
public/system/images/5/original/Guy_Waving_Jamaican_Flag.jpg
public/system/images/4/thumb/DSC_0001.JPG
public/system/images/4/original/logo.png
public/system/images/4/original/DSC_0001.JPG
public/system/images/4/original/2-up.jpg
public/system/images/3/thumb/logo2.gif
public/system/images/3/original/logo2.gif
```

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

Email Sign Up

OR SIGN IN WITH



Facebook

Please move or remove them before you can switch branches.

Aborting

This is what my .gitignore file looks like:

```
.bundle
.DS_Store
db/*.sqlite3
log/*.log
tmp/**/*
public/system/images/*
public/system/avatars/*
```

How do I get this working so I can switch branches without deleting those files?

If I make a change, will it affect those files? In other words, if I came back to this branch afterwards would everything be perfect as up to my latest commit?

I don't want to lose those files, I just don't want them tracked.

[git](#) [git-merge](#) [gitignore](#)

edited Jul 1 at 10:35



Pavan Nagadiya

381 1 7

asked Feb 1 '11 at 1:39



marcammillion

10.7k 43 144 300

7 if you really don't care about these files: git checkout -f <branch> in My case the files are generated in the build process, so I couldn't care less – [Hobbamok](#) Sep 21 '18 at 11:29

Sometimes happens if you do 'git checkout' (without a branch name). To fix, do 'git checkout branchname' – [crafter](#) Nov 5 '18 at 14:32

Separate, but critically related question: why does this error occur at all? why can't git just switch between branches? – [ahnbizcad](#) Dec 8 '18 at 1:55

28 Answers

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

Email Sign Up

OR SIGN IN WITH

Google

Facebook

245

will just mark as removed the next time you commit. After the files are removed from the repo then the .gitignore will prevent them from being added again.



But you have another problem with your .gitignore, you are excessively using wildcards and its causing it to match less than you expect it to. Instead lets change the .gitignore and try this.

```
.bundle
.DS_Store
db/*.sqlite3
log/*.log
tmp/
public/system/images/
public/system/avatars/
```

edited Oct 11 '16 at 8:25



Fry

4,142 8 42 84

answered Feb 1 '11 at 1:59



Arrowmaster

8,016 2 23 23

-
- 2 Thanks....I removed all the files from the current branch and backed them up. Then switched branches and put them back. That worked. Also, thanks for the tip on the .gitignore though – [marcamillion](#) Feb 1 '11 at 2:36

@marcamillion: What do you mean by "that worked"? If the files were tracked on the branch you switched to, you've overwritten them with your versions, which could be different... – [Cascabel](#) Feb 1 '11 at 6:30

-
- 1 I was having an issue with a /build folder that doesn't need to be tracked. So I deleted the local folder, committed my .gitignore file, then checked out the other branch. That finally worked for me. – [Mike S.](#) Oct 12 '11 at 19:40

-
- 15 I think the first part is for the reverse of this particular error message. This error is stating the user is currently in a branch that doesn't have those JPG files tracked and the user is trying to move to one that does. So doing `git rm --cached` will not make a difference, those files don't exist in the current branch. For this error I think the user instead needs to follow @Greg Hewgill's answer - "move them out of working copy, switch branches, and move them back". – [studgeek](#) Jan 29 '13 at 22:20

-
- 4 How would one go about solving the your files would be overwritten with fatal: pathspec 'test/node_modules' did not match any files when I do `git rm -r --cache test/node_modules`? I can't pull because of the overwritten message and can't remove because git can't find them (they are there) – [HMR](#) Feb 6 '15 at 7:40

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

[Email Sign Up](#)

OR SIGN IN WITH


[Google](#)


I hit this message as well. In my case, I didn't want to keep the files, so this worked for me:

git 2.11 and newer

```
git clean -d -f .
```

older git

```
git clean -d -f ""
```

If you also want to remove files ignored by git, then execute the following command.

BE WARNED!!! THIS MOST PROBABLY DESTROYS YOUR PROJECT, USE ONLY IF YOU KNOW 100% WHAT YOU ARE DOING

git 2.11 and newer

```
git clean -d -fx .
```

older git

```
git clean -d -fx ""
```

<http://www.kernel.org/pub/software/scm/git/docs/git-clean.html>

- `-x` means ignored files are also removed as well as files unknown to git.
- `-d` means remove untracked directories in addition to untracked files.
- `-f` is required to force it to run.

edited Feb 26 at 10:58

answered Dec 2 '11 at 20:38

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

Email Sign Up

OR SIGN IN WITH



Google



Facebook

- 134 BE CAREFUL WHEN RUNNING git clean! – [Noel](#) May 1 '14 at 15:06
- 242 To avoid a facepalm, first run it with the dry-run option to see what it would do: `git clean -dfxn` or `git clean -dfx --dry-run` – [Dennis](#) Jun 26 '14 at 16:28 
- 70 Holy crap. This deletes all the config files on my xcode and now the project is turning into mac project. BE VERY CAREFUL WHEN RUNNING THIS COMMAND. I thought it would only remove it from git. – [tyegah123](#) Sep 29 '14 at 8:37
- 24 The `-x` option hurts me – [wener](#) Nov 12 '14 at 3:17



Warning: This will delete the local files that are not indexed

538

Just force it : `git checkout -f another-branch`

[edited Jan 22 at 21:38](#)



Aerin

4,751 7 46 73

[answered Jan 11 '13 at 17:40](#)



Régis

5,795 1 13 14

- 74 Warning: This will delete the local files that are not indexed. – [givanse](#) Jan 2 '14 at 20:23

git clean didn't work for me, but force was exactly what I needed. It was stuck on that branch and I just needed it to change branches. – [Simon The Cat](#) Nov 26 '14 at 20:03

- 6 I didn't want the file that was not indexed! +1 for you – [ryansstack](#) Mar 19 '15 at 22:54 

i got this error,,, :(error: pathspec 'mybranch' did not match any file(s) known to git. – [Budi Mulyo](#) Jan 24 at 5:51

This is the real answer. – [metamonkey](#) Feb 19 at 5:08



If you're on OS X, it may be because a file's name has had certain characters change case. Try setting the following config option:

135

`git config core.ignorecase true`

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

Email Sign Up

OR SIGN IN WITH

 Google

 Facebook

- 12 It worked on Windows as well, looks like this situation happened in the first place due to case change which GIT couldn't determine – [SagiLow](#) Jan 30 '15 at 13:08
- 3 This is exactly the problem I had, a file path was different by one letter case - Windows treats this as same but GIT does not which is the problem. – [Daniel Sokolowski](#) Aug 1 '15 at 3:19
- 1 My problem happened in checkout another branch in Windows 10, and only this work for me, Thanks – [Weijie Sun](#) Apr 11 '17 at 15:49
- awesome! this fix my problem when I was moving between tags – [William Añez](#) Aug 24 '17 at 18:24
- 1 It works also if you try to `git rebase` as well. Thanks. – [user3890355](#) Aug 9 '18 at 12:59

42

Git is telling you that it wants to create files (named `public/system/images/9/...` etc), but you already have existing files in that directory that *aren't* tracked by Git. Perhaps somebody else added those files to the Git repository, and this is the first time you have switched to that branch?

There's probably a reason why those files in your `develop` branch but not in your current branch. You may have to ask your collaborators why that is.

how do I get this working so I can switch branches without deleting those files?

You can't do it without making the files disappear somehow. You could rename `public` to `my_public` or something for now.

if I came back to this branch afterwards would everything be perfect as up to my latest commit?

If you commit your changes, Git won't lose them. If you don't commit your changes, then Git will try really hard *not* to overwrite work that you have done. That's what Git is warning you about in the first instance here (when you tried to switch branches).

answered Feb 1 '11 at 1:47



[Greg Hewgill](#)

697k 151 1034
1181

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

Email Sign Up

OR SIGN IN WITH



Google

Facebook

▲ This worked for me.

23

1. git fetch --all
2. git reset --hard origin/{branch_name}

edited Jan 8 '18 at 12:54



user7294900

27.6k 12 39 72

answered Jan 8 '18 at 12:36



Chamara Jayalath

231 2 4

3 Please add explanation on your solution. see stackoverflow.com/help/how-to-answer – user7294900 Jan 8 '18 at 12:54

Here's my take. Your local copy of the remote branch actually has all of the untracked files, somehow. You're checking that out to restore the untracked files it originally complained about. Now you can switch to other branches – [ahnbizcad](#) Dec 8 '18 at 1:24

1 For some reason, this is the only working solution for me. Thanks mate. – [Kiwad](#) Jan 29 at 4:04

Solved my problem. Thanks – [Devashis Kant](#) Mar 29 at 12:02

Many thanks, so simple! git reset –soft origin/develop . How I hate these merge conflicts editing. This command so nice and simple. – [nine9five](#) Jun 9 at 7:58

▲ There is a command for this delicate task (permanently deleting untracked files)

21

git clean -i

▼ Then `git pull` will do.

answered Sep 21 '16 at 8:37



Abhishek Goel

11.5k 6 71 58

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

Email Sign Up

OR SIGN IN WITH

G Google

Facebook

12

My solution ended up being pushing my branch to remote, cloning a new repo, then doing my merge in the new repo. Other people accessing the repo had to do the same.

Moral of the story: use a `.gitignore` file from inception.

answered Jan 9 '13 at 5:33



Kyle Clegg

29.9k 25 121 132

▲

For those who need something less far-reaching than [Scott Schafer's answer](#),

11

`git clean -f`

▼

will likely work. I *highly* suggest running

`git clean --dry-run`

first. That command will output a list of files that Git will remove if you run `git clean -f`, and might save you the pain of inadvertently removing something you didn't want to.

See [this Stack Overflow answer](#) or [the docs](#) for more information on `git clean`.

edited Feb 14 '18 at 19:01



Rory O'Kane

19.5k 8 67 110

answered Feb 14 '18 at 17:29



mc_kaiser

464 5 15

▲

If you want to quickly resolve this question, You can use this command:

9

`git checkout -f dev`

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

Email Sign Up

OR SIGN IN WITH

Google

Facebook

This helped me, when my issue was not related to `.gitignore` afaik. – [Nakilon](#) Aug 7 '16 at 2:40 

1 This answer is already posted above – [Dan](#) Aug 12 '16 at 20:29

error: pathspec 'dev' did not match any file(s) known to git. – [Black](#) Aug 29 '18 at 13:09

@Black 'dev' is branch name, put your branch instead – [Vinit Solanki](#) Oct 15 '18 at 7:21

8

I had the same problem when checking out to a branch based on an earlier commit. Git refused to checkout because of untracked files.

I've found a solution and I hope it will help you too.

Adding the affected directories to `.gitignore` and issuing `$ git rm -r --cached` on them is apparently not enough.

Assume you want to make a branch based an earlier commit K to test some stuff and come back to the current version. I would do it in the following steps:

1. Setup the untracked files: edit the `.gitignore` and apply `$ git rm -r --cached` on the files and directories you want the git to ignore. Add also the file `.gitignore` itself to `.gitignore` and don't forget to issue `$ git rm -r --cached .gitignore`. This will ensure the ignore behavior of git leaves the same in the earlier commits.
2. Commit the changes you just made:

```
$ git add -A  
$ git commit
```

3. Save the current log, otherwise you may get problems coming back to the current version

```
$ git log > ../git.log
```

4. Hard reset to the commit K

```
$ git reset --hard version_k
```

5. Create a branch based on the commit K

```
$ git branch commit_k_branch
```

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

Email Sign Up

OR SIGN IN WITH



Google



8. Checkout back into master again

```
$ git checkout master
```

9. Reset to the current Version again

```
$ git reset current_version OR $ git reset ORIG_HEAD
```

10. Now you can reset hard to the HEAD

```
git reset --hard HEAD
```

NOTE! Do not skip the next-to-last step (like e. g. `$ git reset --hard ORIG_HEAD`) otherwise the untracked files git complained above will get lost.

I also made sure the files git complained about were not deleted. I copied them to a text-file and issued the command `$ for i in $(cat ./test.txt); do ls -ahl $i; done`

If you checkout to the branch mentioned above again, do not forget to issue `$ git status` to ensure no unwanted changes appear.

edited Sep 24 '12 at 3:55



alex

352k 174 789 923

answered Feb 5 '12 at 17:58



Juri Sinitson

357 4 8



This happened to me on a **Windows 8** system, using Git from the command prompt. The rest of my team uses **TFS**, and I use Microsoft's **git-tf** to push/pull between TFS and my local Git repository.



The problem arose due to some **files that had been renamed only to change their case**. What appears to have happened was this:

- The files were checked in with mixed casing in their names.
- In a later commit, the file names were changed to all lower-case.
- git-tf initially got the files in mixed case.
- When the files were renamed to lower-case, git-tf didn't get the files because to Windows 8 those file names are equivalent.

[Join Stack Overflow](#) to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



- Then `git checkout` the latest version of the project, with the correct file casing.

edited May 30 '17 at 8:02

answered Aug 19 '14 at 13:25



Ryan Lundy

160k 31 164 197

+1 I was able to use `git log` on the current branch and branch to rebase to see when the commit occurred that changed the case; then I hacked around it... – sage Apr 10 '16 at 17:11

4

In my case, the problem was with the submodules. `master` was merged with another branch which added a new submodule to the project. The branch I was trying to checkout didn't have it, that's why git was complaining about untracked files and none of the other suggested solutions worked for me. I forced the checkout to my new branch, and pulled `master`.

- `git checkout -f my_branch`
- `git pull origin master`
- `git submodule update --init`

edited Apr 16 '18 at 21:48

answered Jan 23 '18 at 13:08



wisbucky

13.1k 4 69 60



Bruno Pinheiro

130 7

3

In my case `git rm --cached` didn't work. But i got it with a `git rebase`

edited Sep 2 '12 at 10:40

answered Jun 14 '12 at 13:47



applesnort

3 1



Hillkorn

493 3 10

I was also facing a similar issue and i tried all the solutions posted above but it didn't work

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

Email Sign Up

OR SIGN IN WITH



Google



Now whenever i switched to master it gave me an error

```
The following untracked working tree files would be overwritten by checkout
```

and then it aborted .

In order to solve this, i forcefully checked out master branch and then renamed my `onMusicUpdateListener.java` to `OnMusicUpdateListener.java`, committed it and then merged it with develop branch.

Then i updated my develop branch by merging it into master and now everything is back to normal and problem is solved.

answered Sep 1 '15 at 6:47



Sheraz Ahmad Khilji

5,644 6 37 81

I have run into similar issues before. To my understanding, case sensitivity issue seems to be a problem on windows only. I guess you are developing on windows? – [Ji_in_coding](#) Jun 13 at 17:58

These two functions(git rm --cached, git checkout -f another-branch) did NOT work for me.

3

Instead, I physically removed the file (in eclipse) as what Git tells you to do; *Please move or remove them before you can switch branches.*

and then I add/committed it.

and then I pulled and it worked!

answered Feb 8 '17 at 16:46



In-young Choung

305 1 3 18

[Join Stack Overflow](#) to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH

Google

Facebook

```
sudo chown -v -R usr-name:group-name folder-name
```

answered Apr 24 '15 at 20:10



Won Jun Bae

2,914 3 31 43

I also had the same issue as Won. I added a .gitignore to a folder that was already getting tracked. I deleted the file and then was able to do a git checkout. – [cbloss793](#) Oct 18 '16 at 21:27

2 files with the same name but different case might be the issue.

2 You can Delete one on these files or rename it. Ex:

Pdf.html.twig (The GOOD one)

pdf.html.twig (The one I deleted)

edited Nov 10 '16 at 12:29



Skatox

3,766 10 35 40

answered Nov 10 '16 at 12:12



Samuel Vicent

409 5 10

Move files, instead of delete

2 One way of avoiding deleting files is to move them instead. For example:

```
cd "`git rev-parse --show-toplevel`"
git checkout 2>&1 | while read f; do [ ! -e "$f" ] || mv "$f" "$f".bak; done
```

answered Mar 10 '17 at 6:16

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

[Email Sign Up](#)

OR SIGN IN WITH



Google



Facebook

If you have renamed a file locally and then do a `pull`, it will display that error message.

1

answered Oct 23 '17 at 8:50

lyuboslav kanev
185 1 6

How to get over this error in that case? This message also appears when switching branches after changing the case in a file name (MyFile => myfile).

– Bernhard Döbler Nov 22 '17 at 18:55

that's easy to solve, git is saying that you have the same files in both branches, therefore you have to delete the specific files from master branch and then you will be able to merge:

1

`git merge "your branch"`

I hope it works for you, I just solved my error. my error was:

error: The following untracked working tree files would be overwritten by merge: .vs/slnx.sqlite Please move or remove them before you merge. Aborting

Now it is working! In my case `.vs/slnx.sqlite` was generated by visual studio, I needed to close it before delete it.

edited Dec 28 '17 at 18:38

answered Dec 28 '17 at 18:24

jeirueda
33 1 5

In my case, I was seeing this error because I am using a popular open source CMS and the directory which was causing issues was the uploads directory which the CMS writes to.

0

So what it was saying is that there are files which you don't have, but which you can't get from versioning.

I'm grabbing all the files from the live site to my local, then I'll check this into the repo in the hope that this fixes the issue.

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

[Email Sign Up](#)

OR SIGN IN WITH



Google



Facebook

▲ Delete .gitignore file from appname/gen/ to solve this issue.

0

answered Jan 19 '16 at 9:32



vishnuc156

435 3 6

▲ I just went to the file system and deleted the file directly, then continued with git checkout and it worked.

0

I've had the problem occur several times and it may be related to developers doing delete, push, re-add, push or some such thing.

answered Dec 13 '17 at 21:27



Phil Carter

836 7 8

▲ Most of the answers consider deleting or removing the files, which is the easy way. But sometimes you don't want to get rid of the local files. But merge with a strategy, so git has solution for this too ;

0

git merge --strategy=ours master

answered Aug 2 '18 at 14:50



Erdinç Çorbaci

868 9 16

▲ Just delete the files or rename them.

0

e.g.

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

Email Sign Up

OR SIGN IN WITH

G Google

Facebook

```
ajax/product.php
Please move or remove them before you merge.
Aborting
Updating a04cbe7a..6aa8ead5
```

I had to rename/delete **ajax/product.php** and **ajax/produtPrice.php**.

Don't worry, git pull will bring them back. I suggest you to rename them instead of deleting, because you might loose some changes.

If this does not help, then you have to delete the whole Branch and create it again and then do `git pull origin remotebranch`

answered Aug 29 '18 at 13:23



Black

5,207 13 59 127

In order to save the modified files and to use the modified content later. I found this error while i try checking out a branch and when trying to rebase. Try Git stash

0

git stash

answered Nov 12 '18 at 9:47



MöHämëd İbräHüm

1 2

Check if any folder name having '/' or any special symbol then rename that folders. Then you just clone the repository to another location.

0

answered May 16 at 9:55



Codemaker

1,090 8 20

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

Email Sign Up

OR SIGN IN WITH

Google

Facebook

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

Email Sign Up

OR SIGN IN WITH

