

Gitignore not working

Asked 10 years, 8 months ago Modified 1 month ago Viewed 1.1m times



My `.gitignore` file isn't working for some reason, and no amount of Googling has been able to fix it. Here is what I have:

1064



```
*.apk
*.ap_
*.dex
*.class
**/bin/
**/gen/
.gradle/
build/
local.properties
**/proguard/
*.log
```

It's in the directory `master`, which is my git repo. I'm running Git 1.8.4.2 because I'm on a MacBook running OSX 10.8.6.

`git` `gitignore`

Share Edit Follow

edited Jul 3, 2015 at 14:10



[edwardmlyte](#)

16.6k ● 23 ● 61 ● 83

asked Aug 21, 2014 at 21:34



[user3280133](#)

10.8k ● 3 ● 16 ● 15

- 13 1. what files are indexed although they shouldn't? 2. did you add the `.gitignore` after you added these files to the repo? – [Ahmad](#) Aug 21, 2014 at 21:36
- 3 @Ahmad In multiple projects the `/bin/` folder still shows when I do a `git status`. I haven't changed those files since I added the `.gitignore` – [user3280133](#) Aug 21, 2014 at 21:38
- 3 "since I added the `.gitignore`" - so you added the `.gitignore` after adding the files? – [Ahmad](#) Aug 21, 2014 at 21:40
- 4 possible duplicate of [.gitignore not working](#) – [Liam](#) Dec 19, 2014 at 15:04
- 5 Possible duplicate of [How to make Git "forget" about a file that was tracked but is now in .gitignore?](#) – [Lore](#) Mar 8, 2018 at 9:00

23 Answers

Sorted by: Highest score (default)



The files/folder in your version control will not just delete themselves just because you added them to the `.gitignore`. They are already in the repository and you have to remove them.

2468

You can just do that with this:

**Remember to commit everything you've changed before you do this!**

```
git rm -rf --cached .
git add .
```



Get updates on
questions and
answers



This removes all files from the repository, including the files in your `.gitignore` file, at any time respecting the rules in your `.gitignore` file.

Share Edit Follow

edited Feb 25, 2021 at 15:44

answered Aug 21, 2014 at 21:46

**Ahmad****72.8k** ● 18 ● 116 ● 138

- 7 or you could just try getting reverting the files/folders that you added to `.gitignore` using `git checkout folder_to_ignore/*` – Zahra Feb 7, 2017 at 19:22 ✎
- 3 does not work for me when I push the changes to github. Says Everything is up to date !!!! – Dr. Younes Henni Nov 4, 2017 at 14:24
- 10 I think it's better `git add` a 'git ignore' command to simplify this. Thus we could just use `git ignore path/file` at anytime. – Tyl Feb 19, 2018 at 5:44 ✎
- 25 Or use `git rm -rf --cached foldername` and `git add foldername` if you want to remove files from the folder `foldername`. That way you don't have to commit all other files – cwallenwein Jun 15, 2021 at 8:12 ✎
- 3 It would be helpful if an explanation of what `-rf` does is included within the answer. – MasayoMusic Jun 29, 2021 at 1:07

**425**

To untrack a single file that has already been added/initialized to your repository, i.e., stop tracking the file but not delete it from your system use: `git rm --cached filename`

To untrack every file that is now in your `.gitignore`:

First commit any outstanding code changes, and then, run this command:

```
git rm -r --cached .
```

This removes any changed files from the index(staging area), then just run:

```
git add .
```

Commit it:

```
git commit -m ".gitignore is now working"
```

Share Edit Follow

edited Dec 19, 2014 at 15:05

answered Oct 1, 2014 at 9:01

**Liam****29.8k** ● 28 ● 139 ● 203**Nicolas****5,829** ● 3 ● 21 ● 20

- 4 Be sure to merge your branches back to master or move your pointer because when I tried this I lost both of my feature branches. Fortunately I did not lose any work, all of my most recent updates from the working branch were reset to the master branch but depending on your setup this behavior may be undesirable. – [Sledge](#) Jun 22, 2018 at 14:33
- How would you go about using this command to remove a folder out of the cache instead of a file?
– [Michiel J Otto](#) May 7, 2020 at 12:07
- 7 He did say to first commit any outstanding code changes... – [rollin340](#) Oct 13, 2020 at 16:36
- 8 `git rm --cached filename` worked like a charm! Thanks! – [Caio Mar](#) Nov 15, 2020 at 18:42
- 2 This didn't work for me – [Fernando Torres](#) Aug 18, 2022 at 5:15



130



If you want to ignore a **specific file** or **a few files** in Git, follow these steps:

Step 1: Add the file or directory to .gitignore

Edit your .gitignore file and add the filename or directory path you want to ignore.

For example in .gitignore file do

```
# Ignore a specific file
/path/to/file.ext

# Ignore a specific directory
/path/to/directory/
```

Step 2: Remove the file/directory from Git tracking

If the file or directory was previously tracked by Git, you need to remove it from the index while keeping it locally.

To ignore a specific file:

```
git rm --cached filename # Example: ./foo/bar/abc.min.js
```

To ignore a directory:

```
git rm -r --cached directoryname # Example: ./foo/bar
```

Note: `--cached` removes the file from Git tracking but keeps it in your working directory.

Step 3: Commit the changes

Now, add and commit the updated .gitignore file and the removed files from Git tracking:

```
git add .gitignore
git commit -m "Ignore specific files in Git"
```

Note: Why not use `git rm -r --cached .`? Running `git rm -r --cached .` removes all tracked files and requires you to add everything back, which clutters your Git history with unnecessary changes. Instead, removing only the required files keeps your history clean.

[Share](#) [Edit](#) [Follow](#)

edited Mar 25 at 2:50

answered Oct 5, 2020 at 10:10

[sultanmyrza](#)

5,514 ● 1 ● 35 ● 30

10 Also in **Step2** if you have a directory instead of a file you can use: `git rm -r --cached directoryname` – [Jafar Amini](#) Dec 3, 2021 at 11:16



74



After going down a bit of a rabbit hole trying to follow the answers to this question (maybe because I had to do this in a visual studio project), I found the easier path was to

1. Cut and paste the file(s) I no longer want to track into a temporary location
2. Commit the "deletion" of those files
3. Commit a modification of the `.gitignore` to exclude the files I had temporarily moved
4. Move the files back into the folder.

I found this to be the most straight forward way to go about it (at least in a visual studio, or I would assume other IDE-heavy based environment like Android Studio), without accidentally shooting myself in the foot with a pretty pervasive `git rm -rf --cached .`, after which the visual studio project I was working on didn't load.

[Share](#) [Edit](#) [Follow](#)

edited Feb 8, 2023 at 22:26

answered Oct 26, 2016 at 17:10

[Garrett Simpson](#)

1,619 ● 2 ● 18 ● 18

7 I quite like this approach - seems less destructive, even if it's not considered "proper". – [theyetiman](#) Feb 22, 2017 at 11:01

1 I've often used this when dealing with less experienced developers that commit files, then wonder why their modified gitignore doesn't work. This is definitely the easiest and safest way of dealing with it. – [Jag](#) Apr 25, 2023 at 14:23

1 Had the same in Android Studio and your steps fixed it. Seems git related. – [Harmen](#) Jan 8, 2024 at 14:21



In cmd window use below git command,

32

`git rm --cached filename`

explanation:

`git-rm` - Remove files from the working tree and from the index`--cached` Use this option to unstage and remove paths only from the index. Working tree files, whether modified or not, will be left alone.`--from` <https://git-scm.com/docs/git-rm>

Share Edit Follow

answered Jul 30, 2021 at 11:21



user16557645

this solution is really useful if you just want to remove 1 file (eg. setting file) from remote repository, but you still want to keep it in your local repository. – [Wild Teddy](#) Jan 7, 2022 at 6:57



31

I used something to generate common `.gitignore` for me and I ran into this. After reading @Ozesh answer I opened in VS Code because it has a nice indicator at bottom right showing type of line endings. It was LF so I converted to CRLF as suggested but no dice.



Then I looked next to the line endings and noticed it was saved using UTF16. So I resaved using UTF8 encoding and voila, it worked. I didn't think the CRLF mattered so I changed it back to LF to be sure and it still worked.



Of course this wasn't OPs issue since he had already committed the files so they were already indexed, but thought I'd share in case someone else stumbles across this.

TLDR; If you haven't already committed the files and `.gitignore` still isn't being respected then check file encoding and, make sure its UTF8 and if that doesn't work then maybe try messing with line endings.

Share Edit Follow

answered Oct 2, 2019 at 19:02

[Andrew Wynham](#)

2,398 ● 22 ● 26

5 UTF16 was the culprit for me. Opened VS code > opened `.gitignore` > opened the command prompt > "Change file Encoding" > "UTF-8" > Save file – [Michael](#) Jan 3, 2020 at 17:57

2 Created files in PowerShell using `"blah blah" > file.txt (echo)`, but files were UTF16. Fix: `$PSDefaultParameterValues['Out-File:Encoding'] = 'utf8'` – [JoePC](#) Sep 30, 2020 at 23:58

Created files in PowerShell using `"blah blah" > .gitignore` ie echo,, but in Notepad++ files were identified as `UCS-2 LE BOM`. Using Notepad++ to change encoding to `UTF-8` worked for me. – [john v kumpf](#) Aug 26, 2024 at 17:47

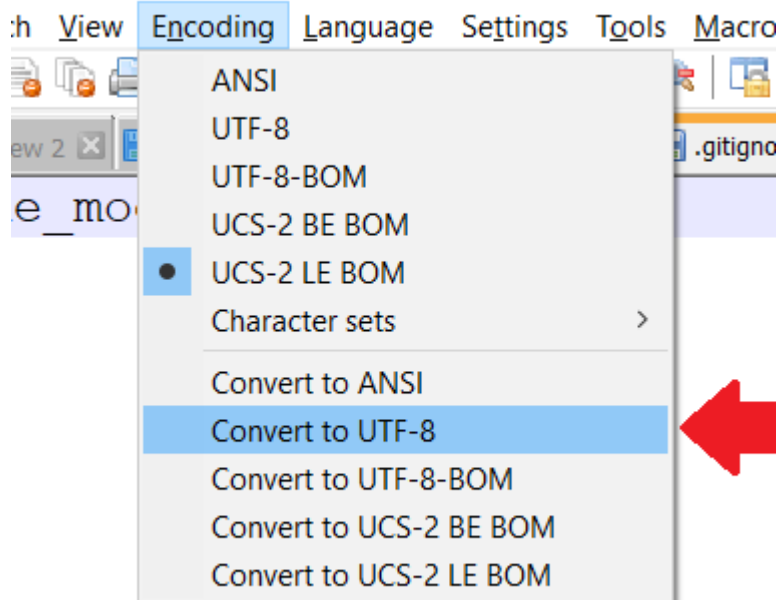
This worked for me, great answer! – [Abimbola Esuruoso](#) Dec 14, 2024 at 0:58

I had done `echo node_modules >> .gitignore` and it didn't work.

26

for some reason, the terminal from `vscode` saves the file in `UCS-2 LE BOM` and git doesn't seem to accept that.

I opened and converted the file to `UTF-8` using `Notepad++`



It Works now.

I think they need to fix this since `echo "filetoignore" >> .gitignore` actually seems a handy thing to do

Share Edit Follow

edited Apr 11, 2021 at 12:00

answered Mar 29, 2021 at 11:08



Abraham

15.9k ● 11 ● 85 ● 121

Uhh. This is a headache. I will just let the node_modules be in peace – [Gilbert](#) Jul 6, 2021 at 14:01

2 @Gilboot that folder will mess up your entire experience if you include it in your repository. It will take too long hour to push changes, and the size of your repository will be unbearably huge, especially the number of files. making this change will save your days. if you don't have notepad++ you can simply open the file with normal notepad and use SaveAs, then to the left of the "Save" button, you can set the encoding to "UTF-8" and save the .gitignore file. this will make it work too – [Abraham](#) Jul 6, 2021 at 14:33

1 Had the same problem. I recreated the .gitignore file with regular Notepad and saved as UTF-8. – [Migisha](#) Jul 28, 2021 at 4:21

1 Same problem. I deleted/recreated the file using vscode main editor and it worked. Probably could have just changed the file format though. – [RoboticRenaissance](#) Aug 22, 2021 at 18:34

1 Thank you!! I was scratching my head a lot before finding your answer! It worked like a charm! – [OCarneiro](#) Jul 10, 2023 at 13:31



In my case it was **a blank space at the beginning of the file** which showed up clearly when I opened the file in Notepad, wasn't obvious in Visual Studio Code.

11

[Share](#) [Edit](#) [Follow](#)

answered Jun 21, 2017 at 9:47



S Saad

131 ● 1 ● 4



there is an option to show whitespace in vscode to help with this, also helpful in files where whitespace matters, i.e. spaces vs tabs in python/makefiles – [ted](#) May 15, 2024 at 20:56



I solved my problem doing the following:

8

First of all, I am a windows user, but i have faced similar issue. So, I am posting my solution here.



There is one simple reason why sometimes the .gitignore doesn't work like it is supposed to. It is due to the EOL conversion behavior.



Here is a quick fix for that

Edit > EOL Conversion > Windows Format > Save

You can **blame your text editor settings** for that.

For example:

As i am a windows developer, I typically use **Notepad++** for editing my text unlike Vim users.

So what happens is, when i open my .gitignore file using Notepad++, it looks something like this:

```
## Ignore Visual Studio temporary files, build results, and
## files generated by popular Visual Studio add-ons.
##
## Get latest from
https://github.com/github/gitignore/blob/master/VisualStudio.gitignore

# See https://help.github.com/ignore-files/ for more about ignoring files.

# User-specific files
*.suo
*.user
*.userosscache
*.sln.docstates
*.dll
*.force
# User-specific files (MonoDevelop/Xamarin Studio)
*.userprefs
```

If i open the same file using the default Notepad, this is what i get

```
## Ignore Visual Studio temporary files, build results, and ## files generated by
popular Visual Studio add-ons. ## ## Get latest from
https://github.com/github/gitignore/blob/master/VisualStudio.gitignore # See
https://help.github.com/ignore-files/ for more about ignoring files. # User-specific
files *.suo *.user *.userosscache
```

So, you might have already guessed by looking at the output. Everything in the .gitignore has become a one liner, and since there is a ## in the start, it acts as if everything is commented.

The way to fix this is simple: Just open your .gitignore file with Notepad++ , then do the following

Edit > EOL Conversion > Windows Format > Save

The next time you open the same file with the windows default notepad, everything should be properly formatted. Try it and see if this works for you.

Share Edit Follow

edited Mar 28, 2018 at 11:44

answered Mar 28, 2018 at 11:38



Ozesh

7,014 ● 1 ● 29 ● 24

- 1 I had to do the above in combination with the accepted answer to make things work properly. Really sneaky problem, since .gitignore looks ok in VS editor, Notepad++ etc at first glance, but apparently the Windows implementation of Git can't handle Unix line breaks in a Windows environment. – Anders Sep 29, 2020 at 14:25



5

In my case whitespaces at the end of the lines of .gitignore was the cause. So watch out for whitespaces in the .gitignore!

Share Edit Follow

answered Nov 29, 2016 at 19:11



026

409 ● 4 ● 5



5

Also, comments have to be on their own line. They can't be put after an entry. So this won't work:

```
/node_modules # DON'T COMMENT HERE (since nullifies entire line)
```

But this will work:

```
# fine to comment here
/node_modules
```



Share Edit Follow

answered Apr 3, 2020 at 3:22



Ben Pritchard

548 ● 7 ● 14



4

Someone might find this useful. The problem for me was the **format**. For some reason, and without having made any changes, git didn't like the format of the contents of the file.

Directories in the `.gitignore` were like this:



```
directory1/*  
directory2/*  
directory3/*
```

When I changed the format to

```
directory1/  
directory2/  
directory3/
```

the directories were again ignored and the only thing I needed to add/commit/push was the `.gitignore`.

Share Edit Follow

answered Aug 3, 2021 at 16:33



Vasiliki

183 ● 10

That was exactly my case. The `.gitignore` was working just fine until today. Maybe some git update messed with the asterisks. – [samueldc](#) May 22, 2024 at 16:02



I know this has a lot of answers already, but the way I ended up at this question was as follows...

2



I had done `'node_modules' > .gitignore` before an `npm install` in a new repo and it wasn't until I changed the `.gitignore` (added `'**/node_modules', '/node_modules', etc`), and then went `git status`, that I realized it was in a Binary format, instead of a text format, and thus git didn't recognize it.



I solved my problem by deleting the `.gitignore` file and re-adding it in my editor.

Share Edit Follow

answered Aug 22, 2021 at 18:35



[RoboticRenaissance](#)

1,197 ● 1 ● 10 ● 18

This fixed it. For some reason the file created with `node_modules > .gitignore` did not work. Re-creating the `.gitignore` file with a text editor solved the issue. Weird! – [rotgers](#) Dec 17, 2023 at 18:56



Does `git reset --hard` work for anyone? I am not saying this is a good solution, it just seemed to work first time I tried.

2



Share Edit Follow

edited Dec 7, 2016 at 5:43



[DimaSan](#)

12.7k ● 15 ● 71 ● 80

answered Dec 6, 2016 at 21:47



[user1889992](#)

375 ● 3 ● 7



2 Please ignore my comment above. Someone at work did something so wrong in Git, it caused duplicate folders and I was not aware for quite a while. Thanks. – [user1889992](#) Dec 7, 2016 at 14:01



Adding my bit as this is a popular question.

2



I couldn't place `.history` directory inside `.gitignore` because no matter what combo I tried, it just didn't work. Windows keeps generating new files upon every save and I don't want to see these at all.



```

On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   configs.sh-example
        modified:   server-install.sh

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .history/configs_20200527173612.sh-example
        .history/configs_20200527173706.sh-example
        .history/configs_20200527173714.sh
        .history/server-install_20200527173749.sh
        .history/server-install_20200527173803.sh
        .history/server-install_20200527173832.sh

no changes added to commit (use "git add" and/or "git commit -a")

```

But then I realized, this is just my personal development environment on my machine. Things like .history or .vscode are specific for me so it would be weird if everyone included their own .gitignore entries based on what IDE or OS they are using.

So this worked for me, **just append ".history" to .git/info/exclude**

```
echo ".history" >> .git/info/exclude
```

Share Edit Follow

answered May 27, 2020 at 17:11



Daniel Katz

2,418 ● 3 ● 28 ● 27



1

There are lot of answers on stackoverflow but what worked for me is not altogether in any particular one, I am jotting all the steps one by one that needs to be done here in a simple manner :



Pre --- Take a backup of repo just in case. (I did not but may make you feel safer)



1. Make sure whether this is your first commit or earlier these files were add and thus have become cached.
2. If yes , then git rm -r --cached . and git add . (do not miss the dot)
3. If it still isn't working , open file in notepad

Go to File -> Save as Change the encoding to ANSI. (Some answers mistakenly say UTF-8 , but it doesn't work and wastes time needlessly)

```
Again git rm -r --cached .
and git add . (do not miss the dot)
```

```
Now do Git status and verify
what we wanted has happened
```

Share Edit Follow

answered Jun 25, 2021 at 13:25



user3251882

970 ● 1 ● 11 ● 21



I'm a beginner and I had the same problem last night (on Windows). I discovered that your .gitignore file must not have .txt extension.

1



Here is how I resolved it.



1. Open any directory and go to menu bar, click view, and then check 'file name extensions'
2. After that you edit the name of your .gitignore file by deleting .txt extension.
3. As result you'll have a text file .gitignore without .txt extension.

Share Edit Follow

answered Nov 3, 2022 at 8:59



Youssef Rajoul

44 ● 3



I had a similar problem. I wanted to ignore 'backup' folder. I've removed it from the project directory, pushed the changes to Github and then copied the the folder back again again. This time it was ignored. I also suggest using GitHub Desktop it makes life much easier!

0



Share Edit Follow

answered Nov 20, 2024 at 13:27



Chadee Fouad

2,958 ● 2 ● 28 ● 33



1- create new branch from last update branch 2- manuel delete caches files 3- update .gitignore + push to remote without caches 4- now generate again the caches files no problem

0



Share Edit Follow

answered May 29, 2023 at 11:28



Abdallah Mahmoud

947 ● 1 ● 9 ● 11



My problem was that I first created .gitignore on a Mac but was now working on a Windows machine. On Macs you use "/" in your file path. On Windows you use "\".

0



So in order for my .gitignore file to work I had be consistent in what symbol I used. In this case I had to use "/" for every file path I wanted to add, even if when copied it was like this: "file\path".



Lost multiple hours around this silliness (it really bugged me lol).

Share Edit Follow

edited May 3, 2021 at 11:21

answered May 3, 2021 at 11:15



Dharman ♦

33.5k ● 27 ● 101 ● 149



Wasonic

37 ● 8

For what it's worth, Windows will accept '/'. – Jeter-work Apr 11, 2024 at 22:05



0

For me, the .gitignore is not being detected due to tab spaces in front of the files listed in the .gitignore file. Please ensure the content of the .gitignore file is formatted correctly.

Share Edit Follow

answered Jan 22 at 7:36

MUHAMMAD SHAHID
RAFI C P

1,269 ● 2 ● 16 ● 31



0

in my case

i used git-bash and write this command:



```
echo [Enter_your_file_name.extension] >> .gitignore
```



then perform push in repo and it is working fine

Share Edit Follow

answered Apr 11, 2021 at 10:05



damanpreet singh

119 ● 2 ● 5

2 Funny, that's how I broke mine. – RoboticRenaissance Aug 22, 2021 at 18:30

@RoboticRenaissance is something wrong with this? Please let me inform.. thank you :)

– damanpreet singh Sep 18, 2021 at 7:08



-1

well my mistake was confusing .ginignore with .idea/.gitignore i was doing everything with .idea/.gitignore and no change would be applied to git repository. hope this helps the people doing the same mistake as i did.

Share Edit Follow

answered Jan 10, 2022 at 7:07



Sep

371 ● 5 ● 13





Highly active question. Earn 10 reputation (not counting the [association bonus](#)) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.

Start asking to get answers

Find the answer to your question by asking.

[Ask question](#)

Explore related questions

[git](#)

[gitignore](#)

See similar questions with these tags.