

Ignore files that have already been committed to a Git repository [duplicate]

Asked 14 years, 8 months ago Modified 4 months ago Viewed 922k times



2832



This question already has answers here:

[How do I make Git forget about a file that was tracked, but is now in .gitignore?](#) (35 answers)

Closed 6 years ago.

I have an already initialized Git repository that I added a `.gitignore` file to. How can I refresh the file index so the files I want ignored get ignored?

`git` `caching` `version-control` `gitignore` `git-rm`

Share Improve this question Follow

edited May 25, 2018 at 23:17



[dreftymac](#)

31.9k ● 26 ● 122 ● 185

asked Jul 16, 2009 at 19:26



[trobrock](#)

46.9k ● 11 ● 40 ● 47

21 Answers

Sorted by: Highest score (default)



4706



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"
```

To undo `git rm --cached filename`, use `git add filename`.

Make sure to commit all your important changes before running `git add .` Otherwise, you will lose any changes to other files.

Please be careful, when you push this to a repository and pull from somewhere else into a state where those files are still tracked, the files will be DELETED

Share Improve this answer Follow

edited Oct 23, 2021 at 14:53

community wiki
14 revs, 14 users 20%
takeshin

294 be aware to commit all your changes before, otherwise you will loose control on all the changed files – [Hoang Pham](#) Jan 13, 2010 at 14:48

9 @TravisWebb You would have to make sure you set up .gitignore first. Also I tend not to remove all files from the index, only the ones I need to by using Fileglobs such as *.o – [Jason S](#) Dec 27, 2011 at 20:41

9 git rm -r --cached . removed way more files than was in the git ignore for me :(:(. It removed files in directories that I don't even have listed – [Dean Hiller](#) Aug 14, 2012 at 20:07

117 Great answer but `git rm --cached filename` seems a little less drastic imho.. – [Jim Morrison](#) Oct 21, 2013 at 20:37

- 26 @JimMorrison you miss the point. if you have a large project with a complicated .gitignore (such as a C# project in Visual Studio) figuring out each individual file to remove is tedious. these three simple commands fixes everything painlessly. – [longneck](#) Nov 13, 2013 at 13:52



733



If you are trying to ignore changes to a file that's already tracked in the repository (e.g., a dev.properties file that you would need to change for your local environment but you would never want to check in these changes), then what you want to do is:

```
git update-index --assume-unchanged <file>
```

If you want to start tracking changes again:

```
git update-index --no-assume-unchanged <file>
```

See [git-update-index\(1\) Manual Page](#).

Also have a look at the `skip-worktree` and `no-skip-worktree` options for update-index if you need this to persist past a git-reset ([via](#))

Update: Here's a convenient alias for seeing which files are currently "ignored" (--assume-unchanged) in your local workspace

```
git config --global alias.ignored = !git ls-files -v | grep "^[[:lower:]]"
```

Share Improve this answer Follow

edited Feb 28, 2023 at 10:16



Alexandre Rocha

73 ● 7

answered Jul 6, 2012 at 17:09



dyodji

7,584 ● 2 ● 17 ● 17

- 19 This is genius! Brilliantly useful. Do you know if there's a way to get a list of all the 'on hold' files? – [artfulrobot](#) Feb 5, 2013 at 13:06

- 15 This'll work for you: `git ls-files -v` If the character printed is lower-case, the file is marked assume-unchanged. see:

stackoverflow.com/a/2363495/677381 and: git-scm.com/docs/git-ls-files – [dyodji](#) Feb 6, 2013 at 1:33 ✎

- 5 Here's my slightly more verbose version of the `ignored` alias, as it appears in my `~/.gitconfig` file: `ignored = !git ls-files -v $(git rev-parse --show-toplevel) | (grep '^[[:lower:]]' || echo 'None ignored.') && echo '\nIgnore changes with: git update-index --assume-`

unchanged <file> \nor track again with: `git update-index --no-assume-unchanged <file>`' The toplevel part of it makes sure it searches the entire repository. – [Chris](#) Feb 10, 2015 at 6:22

3 Unfortunately `--assume-unchanged` doesn't work with `git stash`: the changes are reverted during `git stash` and not reapplied during `git stash pop`. See [this question](#). – [Scott Weldon](#) Apr 27, 2016 at 17:39

7 Not sure if the syntax is different on mac but I had to modify the alias slightly `git config --global alias.hidden '!git ls-files -v | grep "^[[:lower:]]"'` – [Crhstian](#) Mar 13, 2018 at 16:14



384



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

Share Improve this answer Follow

answered Sep 14, 2010 at 23:55



[pagetribе](#)

15.1k ● 3 ● 25 ● 18

10 This was the perfect way to remove the couple of files I'd added, committed, but later realized didn't need to be tracked. After adding those files to `.gitignore`, I was able to do this and untrack them perfectly. – [Andrew Larned](#) Apr 8, 2011 at 16:17

9 Tip: for example if you add `*.config` to your `.gitignore`, you can do `git rm --cached *.config` to stop tracking all `*.config` files. – [dav_i](#) Mar 6, 2013 at 13:30

66 Also note doing this will delete the files from other repos you push to, even though it stays on your system. If you're just ignoring changes to files and don't want to delete them from other user's repos try using `git update-index --assume-unchanged file.name` – [dav_i](#) Mar 8, 2013 at 10:15

1 IMO, this is *the correct answer*. Wiki answer works-kinda, but is awfully heavy-handed, especially given the question. (I'd fix the wiki, but I'm not sure how. "Soon!" ;) – [Olie](#) Jul 17, 2014 at 3:20

@Olie be *extremely* careful with this "solution". This *will* remove the file in the repo, and anyone who pulls this change will have the file removed too. The best solution would be to `assume-unchanged`, as @dav_i said, or `--skip-worktree` as an alternative. – [Sebastianb](#) Sep 12, 2017 at 20:37

Yes - `.gitignore` system only ignores files not currently under version control from git.



90



I.e. if you've already added a file called `test.txt` using `git-add`, then adding `test.txt` to `.gitignore` will still cause changes to `test.txt` to be tracked.

You would have to `git rm test.txt` first and commit that change. Only then will changes to `test.txt` be ignored.

Share Improve this answer Follow

edited Jan 2, 2017 at 17:44



[pjvleeuwen](#)

4,305 ● 1 ● 20 ● 31

answered Jul 18, 2009 at 8:15



[Antony Stubbs](#)

13.4k ● 5 ● 37 ● 40

13 This isn't entirely true, it is possible to ignore changes in a tracked file... see my answer: stackoverflow.com/a/11366713/677381 – [dyodji](#) Jan 3, 2013 at 22:08

`git update-index --assume-unchanged <file>` and `git rm --cached <file>` was not actually working for me. I have done `git rm <file>` and then created new file after that `<file>` is successfully ignored. I am using git version 1.8.1 -- If that was the issue. – [mujaffars](#) Feb 24, 2016 at 6:24 ✎

your syntax is wrong here. it's `git rm test.txt` and here's a link to a more comprehensive answer [stackoverflow.com/questions/12661306/...](https://stackoverflow.com/questions/12661306/) – [Linnea Huxford](#) Dec 24, 2016 at 20:13 ✎

This is the correct answer if your `.gitignore` is not working for some specific files. The accepted answer will cause a commit for all of your files. – [devfaysal](#) Feb 9, 2020 at 3:39



57



Remove trailing whitespace in .gitignore

Also, make sure you have no trailing whitespace in your `.gitignore`. I got to this question because I was searching for an answer, then I had a funny feeling I should open the editor instead of just cat'ing `.gitignore`. Removed a single extra space from the end and **poof** it works now :)

Share Improve this answer Follow

answered Feb 2, 2012 at 15:59



[MikeJansen](#)

3,436 ● 3 ● 27 ● 40

4 I had exactly the same problem, :P. I also got to this question because of that. Good thing you have this documented here. +1 – [Hindol](#) Jul 11, 2012 at 5:08

3 If like me you use vi to quickly edit `.gitignore` use `':set list'` to show whitespace. – [Sam Giles](#) Mar 12, 2013 at 16:35

1 This happened to me when I did a `echo node_modules >> .gitignore` (at least on windows) – [Khôi](#) May 3, 2014 at 20:16

For weeks I was frustrated with this until I saw your whitespace post. Thanks, fixed my problem. – [Blazes](#) Oct 8, 2015 at 22:01



I followed these steps

51

```
git rm -r --cached .  
git add .  
git reset HEAD
```



After that, git delete all files (*.swp in my case) that should be ignoring.



Share Improve this answer Follow

edited Nov 18, 2023 at 3:43



[Nimantha](#)

6,310 ● 6 ● 30 ● 72

answered Jul 15, 2011 at 7:22



[Orlando](#)

9,464 ● 3 ● 57 ● 54

3 be careful with that one, as it assumes you want to add all files except what's ignored, and usually that's not the case – [Roeek](#) Jan 19, 2017 at 22:32

1 Awesomeness! +1 – [Codetard](#) Oct 14, 2019 at 20:35



Complex answers everywhere!

44

Just use the following



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



It will remove the files you are trying to ignore from the origin and not from the master on your computer!

After that just commit and push!

[Share](#) [Improve this answer](#) [Follow](#)

answered Jul 19, 2017 at 5:38

[Ahmad Awais](#)

35.1k ● 5 ● 77 ● 57

This was only half of the solution. Also needed a 2nd command: `git add .` Credit to: dylanwooters.wordpress.com/2015/10/24/... – Ben Oct 7, 2022 at 16:34



43



If you want to stop tracking file without deleting the file from your local system, which I prefer for ignoring `config/database.yml` file. Simply try:

```
git rm --cached config/database.yml
# this will delete your file from git history but not from your local system.
```

now, add this file to `.gitignore` file and commit the changes. And from now on, any changes made to `config/database.yml` will not get tracked by git.

```
$ echo config/database.yml >> .gitignore
```

[Share](#) [Improve this answer](#) [Follow](#)

edited Nov 18, 2023 at 3:44

[Nimantha](#)

6,310 ● 6 ● 30 ● 72

answered Mar 2, 2014 at 9:00

[przbadu](#)

5,929 ● 6 ● 43 ● 69



31



To remove just a few specific files from being tracked:

```
git update-index --assume-unchanged path/to/file
```

If ever you want to start tracking it again:

```
git update-index --no-assume-unchanged path/to/file
```

[Share](#) [Improve this answer](#) [Follow](#)

edited May 23, 2014 at 15:27

answered Aug 21, 2013 at 14:20



jasilva

750 ● 3 ● 18 ● 45



Mark Salvatore

628 ● 1 ● 7 ● 13

▲ As dav_i says, in order to keep the file in repo and yet removing it from changes without creating an extra commit you can use:

29

```
git update-index --assume-unchanged filename
```

[Share](#) [Improve this answer](#) [Follow](#)

answered Apr 17, 2015 at 7:21



Iman Mohamadi

6,711 ● 3 ● 36 ● 33

5 Undo : `git update-index --no-assume-unchanged filename` – [Chemical Programmer](#) Nov 24, 2015 at 12:01

1 what if you want to do that for all files in a folder in one go? – [bg17aw](#) Dec 2, 2016 at 12:54



28

1. Move the file out of the git-controlled directory
2. Check the removal into git
3. Move the file back into the git-controlled directory



After moving the file back, git will ignore it.



Works with directories too!

[Share](#) [Improve this answer](#) [Follow](#)

edited Nov 18, 2023 at 3:43

answered Nov 15, 2015 at 4:03



Nimantha

6,310 ● 6 ● 30 ● 72



Hunter S

1,293 ● 1 ● 16 ● 26

▲ Not knowing quite what the 'answer' command did, I ran it, much to my dismay. It recursively removes every file from your git repo.

25

Stackoverflow to the rescue... [How to revert a "git rm -r ."?](#)

▼ `git reset HEAD`



Did the trick, since I had uncommitted local files that I didn't want to overwrite.



Share Improve this answer Follow

edited May 23, 2017 at 12:26

answered Apr 12, 2011 at 20:39



Community Bot
1 ● 1



averydev
5,737 ● 2 ● 36 ● 35

3 The `git rm -r --cached .` didn't work for me. Git was still claiming an my textmate project file was not being tracked even though `.tmproj` is in my global ignore file. Resetting my local repro like this worked, though. Actually I added the 'hard' option as in `git reset --hard HEAD`. That should have nearly the same effect in this case. – [IAmNaN](#) May 16, 2012 at 19:33

3 Be careful with the `--hard` flag. It *will* throw out any uncommitted changes without a warning! – [Mikko Rantalainen](#) Mar 1, 2013 at 13:02

▲ Put the `.gitignore` file into your repository root **not** in `.git` folder.

25

Share Improve this answer Follow

edited Nov 19, 2023 at 13:43

answered Jun 18, 2013 at 21:08



Dharman ♦
32k ● 25 ● 91 ● 139



Olga
1,670 ● 1 ● 22 ● 32






▲ If the files are already in version control you need to remove them manually.

Share Improve this answer Follow

answered Jul 16, 2009 at 19:28

23


Aragorn
 843 ● 12 ● 25

- 2 I tried `git rm --cached` and `git reset HEAD` both tools I'm fairly familiar with and just could get it from the repo. Success came from first `rm --cached`, then actually manually deleting it, committing the delete, then recreating it manually. And it's gone. – [doublejosh](#) Apr 17, 2012 at 23:17 
- 2 This worked for me like so: `rm foo/bar && git add -u && git commit -m "removed foo/bar" && git push`. Then running `touch foo/bar && git status` will show the file is now properly ignored. – [AWrightIV](#) Feb 21, 2014 at 3:03 

21

Another problem I had was I placed an inline comment.

```
tmp/*    # ignore my tmp folder (this doesn't work)
```

This works


```
# ignore my tmp folder
tmp/
```

Share Improve this answer Follow

edited Nov 18, 2023 at 3:41

answered Dec 6, 2011 at 13:14


Nimanthan
 6,310 ● 6 ● 30 ● 72


kindahero
 5,827 ● 3 ● 26 ● 33

16

Thanks to your answer, I was able to write this little one-liner to improve it. I ran it on my `.gitignore` and repo, and had no issues, but if anybody sees any glaring problems, please comment. This should `git rm -r --cached` from `.gitignore`:

```
cat $(git rev-parse --show-toplevel)/.gitIgnore | sed "s/\\$//" | grep -v "^#" | xargs -L 1 -I {} find $(git rev-parse --show-
```



```
toplevel) -name "{}" | xargs -L 1 git rm -r --cached
```



Note that you'll get a lot of `fatal: pathspec '<pathspec>' did not match any files`. That's just for the files which haven't been modified.



Share Improve this answer Follow

edited Jun 19, 2020 at 14:45



Dan Loewenherz

11k ● 7 ● 51 ● 82

answered Jul 3, 2012 at 17:25



umop

2,142 ● 2 ● 18 ● 22

Is the sed `s///` supposed to be `s/$/` ? Also, what's the point of the sed and grep commands? I'm guessing it's comment filtering from the gitignore?

– Fake Name Aug 17, 2016 at 17:27



11

I have found a weird problem with .gitignore. Everything was in place and seemed correct. The only reason why my .gitignore was "ignored" was, that the line-ending was in Mac-Format (`\r`). So after saving the file with the correct line-ending (in vi using `:set ff=unix`) everything worked like a charm!



Share Improve this answer Follow



answered Mar 2, 2012 at 9:56



Johannes Vetter

111 ● 1 ● 2

1 If anyone has problems with .gitignore after creating the file in windows notepad, there is more information here:

stackoverflow.com/questions/11451535/gitignore-not-working/... – Matt Parkins Dec 4, 2012 at 11:16

The .gitignore format is each line is either a comment (starting with a `#`) or the whole line (including any whitespace) is full filename pattern. If you have `\r` mixed into the line, `git` will ignore only files that end up with `\r` (you can create those if you want!). See `man gitignore` for details, it's worth reading.

– Mikko Rantalainen Mar 1, 2013 at 13:05



10

One other problem not mentioned here is if you've created your .gitignore in Windows notepad it can look like gibberish on other platforms as I found out. The key is to make sure you the encoding is set to ANSI in notepad, (or make the file on linux as I did).

From my answer here: <https://stackoverflow.com/a/11451916/406592>

[Share](#) [Improve this answer](#) [Follow](#)

edited May 23, 2017 at 10:31

answered May 21, 2013 at 9:43



Community Bot

1 • 1



Matt Parkins

24.5k • 9 • 51 • 60



If you need to stop tracking a lot of ignored files, you can combine some commands:

9

```
git ls-files -i --exclude-standard | xargs -L1 git rm --cached
```



This would stop tracking the ignored files. If you want to actually remove files from filesystem, do not use the `--cached` option. You can also specify a folder to limit the search, such as:



```
git ls-files -i --exclude-standard -- ${FOLDER} | xargs -L1 git rm
```

[Share](#) [Improve this answer](#) [Follow](#)

answered Apr 20, 2017 at 6:18



Etherealone

3,528 • 2 • 37 • 56

2 This should be the accepted answer. `git rm -r --cached .` removes lot of other things! – [Shital Shah](#) Jan 2, 2019 at 18:13



One thing to also keep in mind if `.gitignore` does not seem to be ignoring untracked files is that you should not have comments on the same line as the ignores. So this is okay

8

```
# ignore all foo.txt, foo.markdown, foo.dat, etc.
foo*
```



But this will not work:



```
foo* # ignore all foo.txt, foo.markdown, foo.dat, etc.
```

`.gitignore` interprets the latter case as "ignore files named `"foo*" # ignore all foo.txt, foo.markdown, foo.dat, etc."`, which, of course, you don't have.

Share Improve this answer Follow

answered Feb 19, 2016 at 22:52



[Gabriel Perdue](#)

1,573 ● 2 ● 15 ● 24



8

`www/archives/*`



Share Improve this answer Follow



edited Nov 18, 2023 at 3:42



[Nimantha](#)

6,310 ● 6 ● 30 ● 72

answered Mar 9, 2013 at 6:36



[ndemoreau](#)

3,859 ● 4 ● 43 ● 55



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.