How to fix homebrew permissions?

I have uninstalled and installed Homebrew 3 times now because it seems to never allow me to install anything as it denies me permissions at the end of most installations.

As an example I will post this libjpeg download scenario that I'm currently facing.

I try to install libjpeg and get:

```
$ brew install libjpeg
==> Downloading https://downloads.sf.net/project/machomebrew/Bottles/jpeg-
8d.mountain_lion.bottle.1.tar.gz
Already downloaded: /Library/Caches/Homebrew/jpeg-8d.mountain_lion.bottle.1.tar.gz
==> Pouring jpeg-8d.mountain_lion.bottle.1.tar.gz
Warning: Could not link jpeg. Unlinking...
Error: The brew link step did not complete successfully
The formula built, but is not symlinked into /usr/local
You can try again using `brew link jpeg'
Error: Permission denied - /usr/local/opt/jpeg
```

'brew link jpeg' results in

```
Error: Permission denied - /usr/local/opt/jpeg
```

Here is what my brew doctor reads

```
$ brew doctor
Warning: "config" scripts exist outside your system or Homebrew directories.
./configure scripts often look for *-config scripts to determine if
software packages are installed, and what additional flags to use when
compiling and linking.

Having additional scripts in your path can confuse software installed via
Homebrew if the config script overrides a system or Homebrew provided
script of the same name. We found the following "config" scripts:

/Library/Frameworks/Python.framework/Versions/2.7/bin/python-config
/Library/Frameworks/Python.framework/Versions/2.7/bin/python2-config
/Library/Frameworks/Python.framework/Versions/2.7/bin/python2.7-config
Warning: You have unlinked kegs in your Cellar
Leaving kegs unlinked can lead to build-trouble and cause brews that depend on
those kegs to fail to run properly once built. Run brew link on these:

jpeg
```

This permission issue has been making it impossible to use brew on anything and I would really appreciate any suggestions.

osx   homebrew   libjpeg

edited Mar 11 '15 at 10:47
yochannah
2,028 ● 5 ● 19 ● 38

asked May 8 '13 at 3:11
Saturnino
4,463 ● 3 ● 7 ● 6

3 Answers

I was able to solve the problem by using `chown` on the folder:

```
sudo chown -R "$USER":admin /usr/local
```

Also you'll (most probably) have to do the same on `/Library/Caches/Homebrew`:

```
sudo chown -R "$USER":admin /Library/Caches/Homebrew
```

Apparently I had used `sudo` before in a way that altered my folder permission on `/usr/local`, from here on forward all installations with brew have proven to be successful.

This answer comes courtesy of gitHub's homebrew issue tracker

edited Aug 14 '15 at 17:20

Xiang Ji
705 ● 8 ● 15

answered May 8 '13 at 21:26

Saturnino
**4,463** ● 3 ● 7 ● 6

---

12   Thanks for this. I also needed to run `sudo chown -R $USER:admin /Library/Caches/Homebrew` to get rid of my permissions issues. – alexpls Jun 15 '14 at 5:23

8   changing ownership of /usr/local to a specific user is not a solution. It is a terrible hack and a workaround *if* you have a single user system. But then you might as well just chown -R / $USER:$USER – fijiaaron Jan 23 '15 at 4:02 ✏

2   @fijiaaron So what would be a better solution? – haphazard Jun 16 '15 at 23:27

5   chowning /usr/local : completely insane solution. Dearly hoping this isn't actually the party line. – John Clements Sep 1 '15 at 22:48

2   it may be insane but it is what brew itself suggests: `$ brew update Error: The /usr/local directory is not writable. Even if this directory was writable when you installed Homebrew, other software may change permissions on this directory. Some versions of the "InstantOn" component of Airfoil are known to do this. You should probably change the ownership and permissions of /usr/local back to your user account. sudo chown -R $(whoami):admin /usr/local` – remcoder Oct 17 '15 at 18:49 ✏
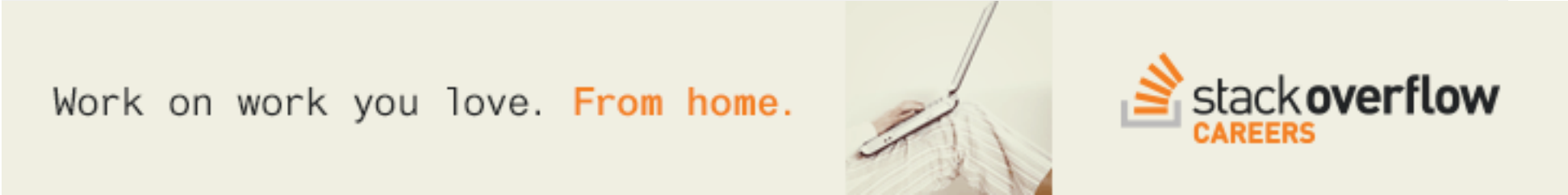
As a first option to whomever lands here like I did, follow whatever this suggests you to do:

```
brew doctor
```

It's the safest path, and amongst other things, it suggested me to:

```
sudo chown -R $(whoami) /usr/local
```

which solved that permissions issue.

The OP did just that and probably didn't get the above suggestion, but you might, and it's always better to start there first.

answered Dec 9 '15 at 9:32

Nycen
3,498 ● 26 ● 44

Command from top-voted answer not work for me.

It got output:

```
chown: /usr/{my_username}dmin: illegal user name
```

This command works fine (group for /usr/local was `admin` already):

```
sudo chown -R $USER /usr/local
```

answered Nov 22 '14 at 19:49

skywinder
9,600 ● 7 ● 55 ● 85

2 Add some quotes `sudo chown -R "$USER":admin /usr/local` —