export can't export variables permanently on Mac

I want to set a env variable globally, so I add this to `~/.bashrc` :

```
export MONGO_HOME=~/mongo
```

It works in this command console, but when I open another command console, it doesn't work; just after I type `source ~/.bashrc` it works, so I add this code to `~/.bash_profile` and `~/.profile` also, but same result.

Can anyone tell me why this happens and how can I save this variable permanently? I'm on Mac.

`osx` `bash`

edited Jan 27 '16 at 4:07
Benjamin W.
13.9k ● 11 ● 32 ● 51

asked Jan 27 '16 at 3:11
starkshang
3,515 ● 3 ● 13 ● 33

How did you open the terminal? – cuonglm Jan 27 '16 at 3:14

@cuonglm,right click command console icon,then click "new window",I also tried `command+N` ,same result. – starkshang Jan 27 '16 at 3:23

Strange, it should work. What `$SHELL` return? – cuonglm Jan 27 '16 at 5:00

@cuonglm,returns `/bin/bash` . – starkshang Jan 27 '16 at 5:06

Are you aware that , by default, `.bashrc` is not executed by scripts? – cdarke Jan 27 '16 at 8:19

2 Answers

Scott Crook's answer provides an effective solution.

Here's some background:

- On OS X, all Bash instances created by `Terminal.app` are *login* shells (unlike the instances created by terminals on most Linux systems).
- Interactive Bash login shells, at the user level, source the *first* of the following files that exists: `~/.bash_profile` , `~/.bash_login` , `~/.profile` ; as interactive *login* shells, however, they do *not* also source `~/.bashrc` .
- Thus, to ensure that `~/.bashrc` is loaded in *all* interactive shells, it should be sourced from `~/.bash_profile` , as Scott Crook's answer recommends.

Adding your `export` statements to `~/.bash_profile` *should* have taken effect in any shell tabs/windows opened thereafter. To troubleshoot, simply add an `echo "Loading <filename>"` statement to the beginning of the initialization file(s) of interest.

Note that even though you say *globally*, the above only takes effect:

- for *interactive* shell instances (notably not for shells created by applications behind the scenes).
- for the current user

To have environment-variable definitions take effect for *all* users, place them in `/etc/profile` , but not that, again, this will only take effect for *interactive* shells.

edited May 23 at 12:07

answered Jan 28 '16 at 7:12

1   thanks for your answer. – starkshang Jan 29 '16 at 5:44

but after I move the export statements to `bash_profile` it still not work. – starkshang Feb 4 '16 at 6:38

@FireSun: Have you tried adding an `echo` statement to `~/.bash_profile` to see if it is even loaded? Are you using the Mac's default terminal application, `Terminal.app` ? How are you opening a new terminal window? You've already confirmed that Bash is your default shell, via `$SHELL` . At this point I suggest you ask a *new* question and describe all troubleshooting steps you've tried and their outcome. – mklement0 Feb 4 '16 at 13:25 ✎


```
36    if (dev.isBored() || job.sucks()) {
37        searchJobs({flexibleHours: true, companyCulture: 100});
38    }
39    // A career site that's by developers, for developers.
```

The `.bash_profile` file in your home directory should contain the following code:

```
#!/bin/bash
test -f ~/.bashrc && source ~/.bashrc
```

Which does a quick check if your `~/.bashrc` file exists, and sources the file only if it exists. Ensure that bit of code is present, and then reopen your terminal application. Your variable `MONGO_HOME` should then be present.

edited Jan 28 '16 at 7:05   answered Jan 28 '16 at 6:30

                                              Scott Crooks
                                              168 ● 1 ● 2 ● 10

1   The shebang line is unnecessary and potentially confusing. `.bash_profile` should not be executed by a subshell, ever, as that would defeat its purpose. – tripleee Jan 29 '16 at 6:07 ✎