How to permanently set $PATH on Linux?

I'm trying to add a directory to my path so it will always be in my Linux path. I've tried:

```
export PATH=$PATH:/path/to/dir
```

This works, however each time I exit the terminal and start a new terminal instance, this path is lost, and I need to run the export command again.

How can I do it so this will be set permanently?

linux bash unix

asked Feb 1 '13 at 0:57
Click Upvote
77.9k 205 478 647

19 Answers

You need to add it to your `~/.profile` file.

```
export PATH=$PATH:/path/to/dir
```

Depending on what you're doing, you also may want to symlink to binaries:

```
cd /usr/bin
sudo ln -s /path/to/binary binary-name
```

edited Sep 4 '14 at 17:42
Erick Robertson
21.1k 7 56 89

answered Feb 1 '13 at 1:01
mpowered
5,191 1 7 16

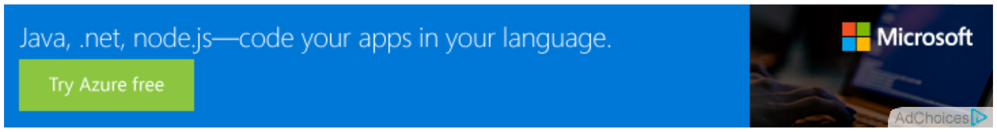
9 A couple of questions. 1) Shouldn't there be a colon between `$PATH` and `/usr/bin` . 2) Should `/usr/bin` even be there. 3) Shouldn't you rather use `/usr/local/bin` ? – Batandwa Jan 11 '14 at 0:16

116 *Please note*: it's often considered a security hole to leave a trailing colon at the end of your bash PATH because it makes it so that bash looks in the current directory if it can't find the executable it's looking for. Users who find this post looking for more information should be advised of this. – erewok Jan 14 '14 at 0:16

28 @AdamRobertson It is unsafe- consider the scenario when you unpack a tarball, then `cd` to the directory you unpacked it in, then run `ls` ---and then realize that the tarball had a malicious program called `ls` in it. – ikdc Feb 27 '14 at 0:39

9 For me it was .bash_profile, not .profile. Seems this is different for everyone. – donquixote Apr 9 '14 at 1:08

7 I think I significantly improved the quality of this answer, and addressed a few issues which other users brought up. Every path export, or every command which adjusts the path, should always make sure to separate an existing path with a colon. Leading or trailing colons should never be used, and the current directory should never be in the path. – Erick Robertson Sep 4 '14 at 17:43



I can't believe nobody mentioned `/etc/environment` file. It's sole purpose is to store Environment Variables. Originally the $PATH variable is defined here. This is a paste from my `/etc/environment` file:

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/l
```

So you can just open up this file as root and add whatever you want.

For Immediate results, Run (try as normal user *and* root):

```
source /etc/environment && export PATH
```

UPDATE:

If you use `zsh` (a.k.a Z Shell), add this line right after the comments in `/etc/zsh/zshenv` :

```
source /etc/environment
```

*I encountered this little quirk on Ubuntu 15.10, but if your **zsh** is not getting the correct **PATH**, this could be why*

edited Apr 3 '16 at 21:59 answered May 27 '14 at 16:27
Zeus77
2,202 1 9 15

5 Not all system have /etc/environment – user3439968 Nov 14 '14 at 17:20

6 FWIW `$PATH` is also defined in `/etc/profile` in Arch Linux. – Sparhawk Feb 24 '15 at 1:57

3 @e-sushi I am actually shocked at that. I'm on ubuntu 14.04.1 myself. and I can promise you the file came built in. – Zeus77 Feb 28 '15 at 12:18

2 After trying every suggestion under the sun but /etc/environment, and having them all NOT work, i finally stumbled across this. I am also on Ubuntu 14.04 and this is the only one that actually changed the PATH variable after reboots. – Chockomonkey Mar 24 '15 at 16:21

2 User should restart the PC after updating environment file. – Harish_N Mar 24 '16 at 17:50

There are multiple ways to do it. The actual solution depends on the purpose.

The variable values are usually stored in either a list of assignments or a shell script that is run at the start of the system or user session. In case of the shell script you must use a specific shell syntax.

System wide

1. `/etc/environment` List of unique assignments. Perfect for adding system-wide directories like `/usr/local/something/bin` to `PATH` variable or defining `JAVA_HOME` .
2. `/etc/xprofile` Shell script executed while starting X Window System session. This is run for every user that logs into X Window System. It is a good choice for `PATH` entries that are valid for every user like `/usr/local/something/bin` . The file is included by other script so use POSIX shell syntax not the syntax of your user shell.
3. `/etc/profile` and `/etc/profile.d/*` Shell script. This is a good choice for shell-only systems. Those files are read only by shells.
4. `/etc/<shell>.<shell>rc` . Shell script. This is a poor choice because it is single shell specific.

User session

1. `~/.pam_environment` . List of unique assignments. Loaded by PAM at the start of every user session irrelevant if it is an X Window System session or shell. You cannot reference other variable including `HOME` or `PATH` so it has limited use.
2. `~/.xprofile` Shell script. This is executed when the user logs into X Window System system. The variables defined here are visible to every X application. Perfect choice for extending `PATH` with values such as `~/bin` or `~/go/bin` or defining user specific

`GOPATH` or `NPM_HOME` . The file is included by other script so use POSIX shell syntax not the syntax of your user shell. Your graphical text editor or IDE started by shortcut will see those values.

3. `~/.profile` Shell script. It will be visible only for programs started from terminal or terminal emulator. It is a good choice for shell-only systems.
4. `~/.<shell>rc` . Shell script. This is a poor choice because it is single shell specific.

Distribution specific documentation

- Ubuntu
- archlinux

edited Jan 15 '16 at 13:43 answered Nov 16 '14 at 21:29
Grzegorz Żur
22.6k 8 58 76

Thank you for the detailed answer, this should be higher up. Maybe `.bash_profile` should be added to the list as well? – James Ko Nov 11 '16 at 23:12

@JamesKo that was number 4 – Zeus77 Nov 14 '16 at 4:11

Put the `export` declaration in `~/.bashrc` . My .bashrc contains this:

```
export PATH=/var/lib/gems/1.8/bin:/home/fraxtil/.bin:$PATH
```

answered Feb 1 '13 at 0:59
Fraxtil
2,046 1 13 28

7 restart needed? – Click Upvote Feb 1 '13 at 1:01

2 Worked when I put this in the `.profile` ', didn't find `.bashrc` – Click Upvote Feb 1 '13 at 1:10

It might be dependent on the exact system; I'm not sure exactly what conditions determine which file is executed. Glad the problem was solved, though. – Fraxtil Feb 1 '13 at 2:10

9 @Click Upvote You need to do `source ~/.bashrc` to to reload `.bashrc` configuration. Then it will work – BigSack Apr 6 '14 at 6:02

3 The `export` keyword is only needed if `PATH` is not already flagged as an environment variable -- which it almost unconditionally will be. Simply `PATH=/var/lib/gems/1.8/bin:/home/fraxtil/.bin:$PATH` would have the same effect. – Charles Duffy Oct 4 '14 at 17:20

You may set `$PATH` permanently in 2 ways.

1. To set path for particular user : You may need to make the entry in `.bash_profile` in home directory in the user.

   e.g in my case I will set java path in tomcat user profile

   ```
   [tomcat]$ echo "export PATH=$PATH:/path/to/dir" >> /home/tomcat/.bash_profile
   ```

2. To set common path for ALL system users, you may need to set path like this :

   ```
   [root~]# echo "export PATH=$PATH:/path/to/dir" >> /etc/profile
   ```

edited Feb 25 '15 at 16:54
e-sushi
8,477 8 24 52

answered Jan 3 '14 at 11:35
Mohit M
473 3 11

3 Is the file named `/etc/profiles` with an `s` on your distro? Mine has no `s` . I think you have a typo. – Chris Johnson Oct 16 '14 at 13:36

3 You probably want to escape the $ you are writing to the profile file. e.g. echo "export PATH=\$PATH:/path/to/dir" >> /etc/profile, that way you actually append to the variable when that script runs rather than setting it to a literal value based on it's value at the time of executing this initial command. – BuvinJ Jan 20 '16 at 15:12

You can add that line to your console config file (e.g. .bashrc) , or to .profile

answered Feb 1 '13 at 0:59
aqua

2,699 16 33

1 I have neither of those files in `/home/(username)` – Click Upvote Feb 1 '13 at 1:03

1 @ClickUpvote: What shell do you use? (And files that start with a dot are hidden, you need something like `ls -a` to see them.) – David Schwartz Feb 1 '13 at 1:05

I see `.profile` in there on a second look. Worked now, ty. – Click Upvote Feb 1 '13 at 1:10

Incase you don't have any of those files (bashrc or profile) you can manually create them and they will automatically be used – Zeus77 Sep 7 '14 at 1:20

You can use on Centos or RHEL for local user:

```
echo $"export PATH=\$PATH:$(pwd)" >> ~/.bash_profile
```

This add the current directory(or you can use other directory) to the PATH, this make it permanent but take effect at the next user logon.

If you don't want do a re-logon, then can use:

```
source ~/.bash_profile
```

That reload the `# User specific environment and startup programs` this comment is present in `.bash_profile`

answered Oct 21 '16 at 4:11
Daniel Antonio Nuñez Carhuayo
722 10 13

---

the files where you add the export command depends if you are in login-mode or non-login-mode.

if you are in login-mode, the files you are looking for is either /etc/bash or /etc/bash.bashrc

if you are in non-login-mode, you are looking for the file /.profile or for the files within the directory /.profiles.d

the files mentioned above if where the system variables are.

answered Nov 5 '13 at 13:35
Dikinha
31 1

Add to `/etc/profile.d` folder script `[name_of_script].sh` with line: `export PATH=$PATH:/dir`. Every script within `/etc/profile.d` folder is automaticaly executed by `/etc/profile` on login.

answered Apr 10 '15 at 12:12
Lurii
1,283 11 16

It's recommended way of how to customize your environment – Lurii Apr 10 '15 at 12:14

1 This is only if you want the settings to be system-wide, which is probably not the most common use case. Most people want (or should want) the path to be set locally, because most users/roles are doing contextually different operations, and the fewer assumptions you make, the better. – mpowered Apr 10 '15 at 20:25

@mpowered, yeah, this is only for system-wide. If you want localy change PATH you should add the same export in ~/.profile or ~/.bashrc. Here you should consider that login shells read ~/.profile and interactive shells read ~/.bashrc. This is very important because ssh for example does not do an login, therefore ~/.profile will not be read. Several distibution like suse source ~/.bashrc in /etc/profile. But it's not common for all linux' – Lurii Apr 15 '15 at 9:49

Zues77 has the right idea. The OP didn't say "how can i hack my way through this". OP wanted to know how to permanently append to $PATH:

```
sudo nano /etc/profile
```

This is where it is set for everything and is the best place to change it for all things needing $PATH

answered Sep 25 '15 at 2:26
Joe D
21 1

You can also set permanently, editing one of these files:

`/etc/profile` (for all users)

`~/.bash_profile` (for actual user)

`~/.bash_login` (for actual user)

`~/.profile` (for actual user)

You can also use `/etc/environment` to set a permanent PATH environment variable, but **it does not support variable expansion**.

Extracted from: http://www.sysadmit.com/2016/06/linux-anadir-ruta-al-path.html

answered Jun 29 '16 at 8:59
Delucaramos
21 1

I stumbled across this question yesterday when searching for a way to add a folder containing my own scripts to the PATH - and was surprised to find out that my own `~/.profile` file (on Linux Mint 18.1) already contained this:

```
# set PATH so it includes user's private bin if it exists
if [ -d "$HOME/bin" ] ; then
    PATH="$HOME/bin:$PATH"
fi
```

Thus, all I had to do was create the folder `~/bin` and put my scripts there.

answered Mar 4 at 11:09
RobertG
663 6 28

---

the best simple way is the following line:
`PATH="<directory you want to include>:$PATH"`
in your .bashrc file in home directory.
It will not get reset even if you close the terminal or reboot your PC. Its permanent

edited Oct 4 '14 at 16:38 answered Oct 18 '13 at 17:00
edward torvalds
237 2 17

8 This tells us nothing about how to make it permanent. – arman Apr 27 '14 at 4:54

@quant if you do what is said, it will set your settings permanently. it will work even if you close the terminal. – edward torvalds Oct 4 '14 at 16:37

My answer is in reference to the setting-up of `go-lang` on `Ubuntu linux/amd64` .I have faced the same trouble of setting the path of environment variables (`GOPATH` and `GOBIN`), losing it on terminal exit and rebuilding it using the `source <file_name>` every time.The mistake was to put the path (`GOPATH` and `GOBIN`) in `~/.bash_profile` folder. After wasting a few good hours, I found that the solution was to put `GOPATH` and `GOBIN` in `~/.bash_rc` file in the manner:

```
export GOPATH=$HOME/go
export GOBIN=$GOPATH/bin
export PATH=$PATH:$GOPATH:$GOBIN
```

and doing so, the go installation worked fine and there were no path losses.

EDIT 1: The reason with which this issue can be related is that settings for non-login shells like your ubuntu terminal or gnome-terminal where we run the go code are taken from `~./bash_rc` file and the settings for login shells are taken from `~/.bash_profile` file, and from `~/.profile` file if `~/.bash_profile` file is unreachable.

edited May 4 at 21:38

answered May 4 at 21:13
Abhiroj Panwar
180 14

Permanently add PATH variable

Global:

```
echo "export PATH=$PATH:/new/path/variable" >> /etc/profile
```

Local(for user only):

```
echo "export PATH=$PATH:/new/path/variable" >> ~/.profile
```

For **global** restart. For **local** relogin.

Example

Before:

```
$ cat /etc/profile

#!/bin/sh

export PATH=/usr/bin:/usr/sbin:/bin:/sbin
```

After:

```
$ cat /etc/profile

#!/bin/sh

export PATH=/usr/bin:/usr/sbin:/bin:/sbin
export PATH=/usr/bin:/usr/sbin:/bin:/sbin:/new/path/variable
```

Alternatively you can just edit profile:

```
$ cat /etc/profile

#!/bin/sh

export PATH=/usr/bin:/usr/sbin:/bin:/sbin:/new/path/variable
```

Another way(thanks gniourf_gniourf):

```
echo 'PATH=$PATH:/new/path/variable' >> /etc/profile
```

> You shouldn't use double quotes here! echo 'export PATH=$PATH:/new/path/variable'... And by the way, the export keyword is very likely useless as the PATH variable is very likely already marked as exported. – **gniourf_gniourf**

edited Nov 14 '14 at 18:19

answered Nov 14 '14 at 17:35
user3439968
1,719 9 11

1 Nope. You shouldn't use double quotes here! `echo 'export PATH=$PATH:/new/path/variable'` ... And by the way, the `export` keyword is very likely useless as the `PATH` variable is very likely already marked as exported. – gniourf_gniourf Nov 14 '14 at 17:48

Nope, you should use double quotes because $PATH in single quotes not interpolated. And BTW export also useful. – user3439968 Nov 14 '14 at 17:58

I got it. You can use double quotes or use single quotes, because $PATH interpolated when the echo executed or interpolate when /etc/profile execute. – user3439968 Nov 14 '14 at 18:14

1 @user3439968 actually, Double quotes will cause a lot of issues if you were to append to $PATH from multiple files. Consider: when you use double quotes, $PATH gets translated to a static string with all the previously defined PATH directories. say you append `/usr/local` to it using `~/.bashrc` . now if you intend to append `/opt/bin` to the same variable using `/etc/bash.bashrc` ; $PATH will translate to the same static string, as a result $PATH will be **replaced** instead of appended to... It will be a matter of system's preference to one file over another – Zeus77 Jan 27 '15 at 16:20

---

one way to add permanent path, which worked for me, is: cd /etc/profile.d touch custom.sh vi custom.sh export PATH=$PATH:/path according to your setting/ restart your computer and here we go path will there permanently cheers.

answered May 28 '16 at 3:19
user6393373

1

I think the most elegant way is:

1.add this in *~./bashrc* file

```
if [ -d "new-path" ]; then
  PATH=$PATH:new-path
fi
```

2.source *~/.bashrc*

(Ubuntu)

answered Jul 17 '16 at 2:50
saks
6 1

I've just encountered the same issue,and paste the reason that PATH get always reset when relogin.After a lot of searching works,I finally realize it's a custom system issue when i see this code in my /etc/passwd file:

```
root:x:0:0:root:/root:/usr/sbin/nologin
```

It means this custom system disable the bash shell of root.
If you have tried every solution but get nothing helpful information.And you use a custom system,you could take a look at my possible reason.

answered Dec 5 '16 at 10:08
Does
125 9

It can be directly added by using the following command:

echo 'export PATH=$PATH:/new/directory' >> ~/.zshrc source ~/.zshrc

answered Jul 11 '16 at 11:31
Anoop Nagabhushan
1

2 The question is labeled `bash` , so that is not very helpful. – Laurenz Albe Jul 11 '16 at 11:38

protected by tripleee Jul 28 '16 at 9:15

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 reputation on this site (the association bonus does not count).

Would you like to answer one of these unanswered questions instead?