**WHAT ARE ENVIRONMENT VARIABLES IN LINUX?**

Linux environment variables act as placeholders for information stored within the system that passes data to programs launched in shells or subshells.

**Why are Environment Variables Valuable for System Administration?**

Admins have the ability to modify environment variables to fit personal or larger group needs of users within their environments. As you’ll notice below, admins can alter the hostname, command-line prompt, coloring in shells for text, and various other environment variables to better suit user preference.

**Commands for Environment Variables**

1. ***env***– The command lists all of the environment variables in the shell.
2. ***printenv***– The command prints all (if no environment variable is specified) of environment variables and definitions of the current environment.
3. ***set***– The command assigns or defines an environment variable.
4. ***unset*** – The command deletes the environment variable.
5. ***export***– The command exports the value of the newly assigned environment variable.

**How to Define Environment Variables**

After seeing the list of present environment variables on your system, you can modify or redefine them. Use the variable’s name, an equals sign (=), and enclose the new definition in double quotes (“”). See the example below.

HOSTNAME=”PizzaHeaven”

Then employ the **export**command with the variable name to export the data to new programs or subshells for use.

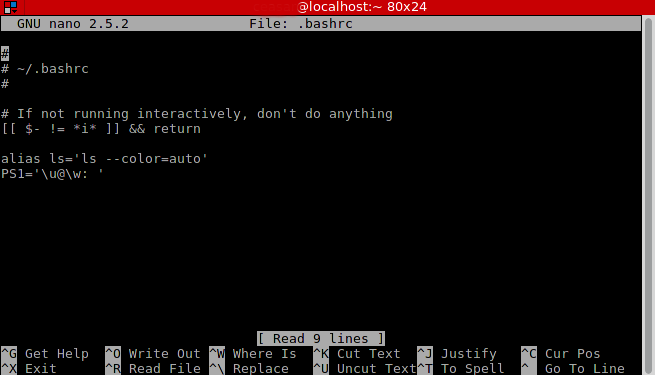
**Pay Attention to the Dollar Sign ($)**

On the command-line and in scripts, the dollar sign ($) precedes environment variables. When redefining variables, do not use the dollar sign.

**Persistent and Nonpersistent Environment Variables**

When modifying environment variables in your current shell, those variables remain nonpersistent. The changes stay temporary and vanish once you log out of the shell.

You can modify the variables to stay persistent by editing the bash configuration files. After you logout of the current shell, those changes remain intact and permanent for the user(s) or groups.

Edit the **.bashrc** or **~/.bash\_profile**configuration files to create persistent environment variables. See the image below as an example of the configuration file in the nano text editor.[](http://phoenixts.com/wp-content/uploads/2016/02/bashrcfile.png?__hstc=753710.7ec11815e7b45fa698b7ef0b641f42f1.1531151288095.1531151288095.1531151288095.1&__hssc=753710.1.1531151288095&__hsfp=2848525548)

**COMMON ENVIRONMENT VARIABLES**

Become familiar with common Linux environment variables by referring to the cheat sheet below (you can save the image).

How to permanently export a variable in Linux?

I am running RHEL6, and I have exported an environment variable like this:

export DISPLAY=:0

That variable is lost when the CMD is closed. How do I permanently add this so that this variable value always exists with a particular user?

**Ans:**

You have to edit three files to set a permanent environment variable as follow:

* ~/.bashrc

When you open any terminal window this file will be run. Therefore, if you wish to have a permanent environment variable in all of your terminal windows you have to add the following line at the end of this file:   
*export DISPLAY=0*

* ~/.profile

Same as **bashrc** you have to put the mentioned command line at the end of this file to have your environment variable in the every log in of your OS. 

* /etc/environment

If you want your environment variable in every windows or application ( not just terminal window ) you have to edit this file. Add the following command at the end of this file:  
*DISPLAY=0*

Note that in this file you do not have to write *export* command

Normally you have to restart your computer to apply this changes. But you can apply changes in **bashrc** and **profile** by these commands:

$ source ~/.bashrc  
$ source ~/.profile

But for **/etc/environemnt** you have no choice but restarting ( as far as I know )

On Ubuntu systems, use the following locations:

1. System-wide persistent variables in the format of JAVA\_PATH=/usr/local/java store in
2. /etc/environment
3. System-wide persistent variables that reference variables such as  
   export PATH="$JAVA\_PATH:$PATH" store in
4. /etc/.bashrc
5. User specific persistent variables in the format of PATH DEFAULT=/usr/bin:usr/local/bin store in

~/.pam\_environment

# How to set environment (PATH) variable permanently in Linux

There is a time when every Linux Administrator gets stuck at a point of his career when he/she has to set a custom path or any other environment variable permanently in the Linux machine.  
  
Suppose you have added a new path to the PATH variable using the shell but you might observe that every time you switch terminal the PATH variable does not works.

**Solution**  
To make this issue to be resolved permanently you need to add the variable inside .bashrc or .bash\_profile file inside the home folder of the user.  
  
For example, you want to add a PATH variable for root user so you need to add the path inside ~/.bashrc or ~/.bash\_profile  
  
Now the confusion comes which file should we place the variable or inside both the files?

**Difference between .bashrc and .bash\_profile**

Every time you login to a Linux (Red Hat) machine .bash\_profile file is executed  
**but**  
In case you are already logged in and you open a new terminal then .bashrc file is executed  
  
So, basically you can put the environment variable inside any of the two files. As per me I would advice you to put the same inside .bash\_profile.

**WHY?**

Have a look at .bash\_profile file  
# less ~/.bash\_profile  
# Get the aliases and functions  
if [ -f ~/.bashrc ]; then  
        . ~/.bashrc  
fi  
  
# User specific environment and startup programs  
  
PATH=$PATH:$HOME/bin  
  
export PATH  
export PATH=$PATH:/usr/local/samba/bin

You can see in the highlighted part in blue in the above part, every time .bash\_profile is executed it also runs .bashrc along with it. As you can see I have added an extra PATH variable for my samba so that I do not need to set it every time I log in.

Sorry for the long post but just thought to clarify in the best possible way from my side.