

Understanding a little more about `/etc/profile` and `/etc/bashrc`

By Benjamin Cane

Archived: 2026-04-05 14:47:14 UTC

The Wayback Machine -

<https://web.archive.org/web/20220316014323/http://bencane.com/2013/09/16/understanding-a-little-more-about-etcprofile-and-etcbashrc/>

Recently I was working on an issue where an application was not retaining the `umask` setting set in the root users profile or `/etc/profile`. After looking into the issue a bit it seemed that the application in question only applied the `umask` setting that was set in `/etc/bashrc` and would not even accept the values being the applications own start scripts.

After doing a bit of researched I learned a little bit more about what exactly these files do, the differences between them and when they are executed. I figured this would be a good thing to share as it is not a topic that comes up very often.

What is `/etc/profile` used for?

If you have been using Linux for a while you are probably familiar with the `.profile` or `.bash_profile` files in your home directory. These files are used to set environmental items for a users shell. Items such as `umask`, and variables such as `PS1` or `PATH`.

The `/etc/profile` file is not very different however it is used to set system wide environmental variables on users shells. The variables are sometimes the same ones that are in the `.bash_profile`, however this file is used to set an initial `PATH` or `PS1` for all shell users of the system.

`/etc/profile.d`

In addition to the setting environmental items the `/etc/profile` will execute the scripts within `/etc/profile.d/*.sh`. If you plan on setting your own system wide environmental variables it is recommended to place your configuration in a shell script within `/etc/profile.d`.

What is `/etc/bashrc` used for?

Like `.bash_profile` you will also commonly see a `.bashrc` file in your home directory. This file is meant for setting command aliases and functions used by bash shell users.

Just like the `/etc/profile` is the system wide version of `.bash_profile`. The `/etc/bashrc` for Red Hat and `/etc/bash.bashrc` in Ubuntu is the system wide version of `.bashrc`.

Interestingly enough in the Red Hat implementation the `/etc/bashrc` also executes the shell scripts within `/etc/profile.d` but only if the users shell is a Interactive Shell (aka Login Shell)

When are these files used?

The difference between when these two files are executed are dependent on the type of login being performed. In Linux you can have two types of login shells, Interactive Shells and Non-Interactive Shells. An Interactive shell is used where a user can interact with the shell, i.e. your typical bash prompt. Whereas a non-Interactive shell is used when a user cannot interact with the shell, i.e. a bash scripts execution.

The difference is simple, the `/etc/profile` is executed only for interactive shells and the `/etc/bashrc` is executed for both interactive and non-interactive shells. In fact in Ubuntu the `/etc/profile` calls the `/etc/bashrc` directly.

Interactive Shell vs Non-Interactive Shell

To show an example of an interactive shell vs a non-interactive shell I will add a variable into both `/etc/profile` and `/etc/bash.bashrc` on my Ubuntu system.

`/etc/profile`

```
# grep TEST /etc/profile
export TESTPROFILE=1
```

`/etc/bash.bashrc`

```
# grep TEST /etc/bash.bashrc
export TESTBASHRC=1
```

Interactive Shell

The below example is showing an interactive shell, in this case both the `/etc/profile` and `/etc/bash.bashrc` was executed.

```
# su -
# env | grep TEST
TESTBASHRC=1
TESTPROFILE=1
```

Non-Interactive Shell

In this example we are running a command through SSH that is non-interactive; because this is a non-interactive shell only the `/etc/bash.bashrc` file is executed.

```
# ssh localhost "env | grep TEST"  
root@localhost's password:  
TESTBASHRC=1
```

Conclusion

In my case the applications child processes are not recognizing the umask value set in `/etc/profile` but do recognize the value in `/etc/bashrc` . This tells me that the subprocess is starting as a non-interactive shell. While the suggested route of modifying environmental variables is to add a shell script into `/etc/profile.d` in my case it is better to set the umask value in the `/etc/bashrc` .

Source: <https://web.archive.org/web/20220316014323/http://bencane.com/2013/09/16/understanding-a-little-more-about-etcprofile-and-etcbashrc/>