

.bash_profile vs .bashrc

by *Josh Staiger*

WHEN working with Linux, Unix, and Mac OS X, I always forget which bash config file to edit when I want to set my PATH and other environmental variables for my shell. Should you edit `.bash_profile` or `.bashrc` in your home directory?

You can put configurations in either file, and you can create either if it doesn't exist. But why two different files? What is the difference?

According to the [bash man page](#), `.bash_profile` is executed for login shells, while `.bashrc` is executed for interactive non-login shells.

What is a login or non-login shell?

When you *login* (type username and password) via console, either sitting at the machine, or remotely via ssh: `.bash_profile` is executed to configure your shell before the initial command prompt.

But, if you've already logged into your machine and open a new terminal window (xterm) inside Gnome or KDE, then `.bashrc` is executed before the window command prompt. `.bashrc` is also run when you start a new bash instance by typing `/bin/bash` in a terminal.

Why two different files?

Say, you'd like to print some lengthy diagnostic information about your machine each time you login (load average, memory usage, current users, etc). You *only* want to see it on login, so you *only* want to place this in your `.bash_profile`. If you put it in your `.bashrc`, you'd see it every time you open a new terminal window.

Mac OS X — an exception

An exception to the terminal window guidelines is Mac OS X's Terminal.app, which runs a login shell by default for each new terminal window, calling `.bash_profile` instead of `.bashrc`. Other GUI terminal emulators may do the same, but most tend not to.

Recommendation

Most of the time you don't want to maintain two separate config files for login and non-login shells — when you set a `PATH`, you want it to apply to both. You can fix this by sourcing `.bashrc` from your `.bash_profile` file, then putting `PATH` and common settings in `.bashrc`.

To do this, add the following lines to `.bash_profile`:

```
if [ -f ~/.bashrc ]; then
    source ~/.bashrc
fi
```

Now when you login to your machine from a console `.bashrc` will be called.