

Mac Command Line Interface Tips - Creating Symbolic Links

Posted At : October 30, 2009 9:36 AM | Posted By : Bob Silverberg

Related Categories: OS X, bash

Although I'm just starting to discover the power of the CLI, I have used it in the past for creating symbolic links, which come in handy in a number of scenarios.

What is a Symbolic Link?

A **symbolic link**, or symlink as it's often called, is a special kind of file that points to and acts like another file or folder. You can think of it as kind of like a shortcut. Your Mac will treat it like a file or folder, and it therefore allows you to pretend to have the same file or folder in more than one location. When you look at a file listing in Finder, symlinks appear with a little curved arrow in the lower left-hand corner of the icon, and the **kind** column reports that the file is an **Alias**, but it actually isn't.

What's the Difference Between an Alias and a Symbolic Link?

An alias is similar to a symlink in that it is also a file that points to and acts like another file or folder. There are, to my knowledge, two main differences between an alias and a symlink:

1. A symlink simply stores the path to the target, so if you move the target the symlink will no longer be pointing at the actual target - it will be pointing at the old location of the target. Aliases, on the other hand, store more information than simply the path to the target; information about the identity of the target. This means that you can move or rename the target and the alias will still point to it.
2. Symlinks exist at the shell level. Any symlinks that you create can be utilized both by the OS X GUI, and also by any shell commands. Aliases can only be interpreted through the GUI, otherwise they are treated like actual files. For example, let's say I create a symlink called linkA and an alias called aliasA, both of which point to a text file elsewhere on my system. Then I open up a terminal window and use the bash **cat** command, which outputs the contents of a file to the terminal, to look at each file. The command "cat linkA" will display the contents of the text file, whereas the command "cat aliasA" will display the actual contents of the aliasA file, which appears mostly as garbage.

Because of that second difference, I prefer to use symlinks. I'm honestly not sure how other software deals with aliases, but I figure it's safer to just use symlinks.

How Do I Create a Symbolic Link?

Creating symlinks is extremely easy. You just use the bash **ln** command. Typing "man ln" at a bash prompt will display the usage of the command. The version that I generally use is:

```
ln -s -i -v target linkname
```

where

- -s means that I want to create a symbolic link
- -i means that I will be prompted if the symlink already exists
- -v means that I'll see a message confirming that the symlink was created
- target is the path to the file or folder that I want to point to
- linkname is the path and name of the symlink file that I want to create

For example, to create a symlink to the file /Users/username/Documents/test.txt, and place that symlink in my home directory, I would use this command:

```
ln -s -i -v /Users/username/Documents/test.txt ~/test.txt
```

What Are Symbolic Links Good For?

I'm sure that there are a ton of uses for symlinks. Two that I have found are:

1. Placing source code in one location, e.g., /Users/username/Documents/sites/mySite, and then creating a symlink in your wwwroot folder that points to your source. This allows your web server to serve your content, but you can still keep it wherever you like.
2. Placing system files under source control. I'm planning another post about this and a cool technique that I picked up from GitHub, but in a nutshell, if you have a file that needs to be located somewhere, but you want the "real" file to reside elsewhere (for example, so it can be in a Git repository), symlinks provide a solution.

What other uses have people found for symlinks?