

## The power of faking it

Written by Darryn Lowe

Tuesday, 17 March 2009 00:00 - Last Updated Tuesday, 17 March 2009 10:08

---

One of the most powerful features of Mac OS X is not a Mac OS X specific feature but one that has been around for almost as long as \*NIX operating systems but is one of the most under-utilised features of Mac OS X. It's power makes for some interesting ways of forcing software to conform to your way of storing files. This magical feature is called Aliases.

In layman's terms an alias is a shortcut but it does more than just create an icon to a file or application. The best way to describe an alias is to think of a railway junction. When a train is coming to said junction it will default to a single path, however throw a switch and the train is diverted to another path. That's the function of an alias. The beauty of an alias is that if an application such as iWeb, iPhoto, Bento, etc requires a dedicated path for a file you can create an alias in that directory to point to a file in a separate directory or even external drive or network path. This way you can keep files stored in a central location and have a number of machines connecting to the same file.

The caveat is that if you try to run the applications that require that file on multiple machines at the same time you run the risk of corrupting that file. Thankfully though many applications can lock a file so that it can't be edited by more than one person at a time.

Aliases don't have to be specific to file paths either. Back in my days on the Amiga I used to create a file that set up aliases for commands so for instance instead of typing:

*clear*

to clear the screen in a terminal I could create an alias whereby I only needed to type:

*clr*

This can be used to make switches in commands easier as well. For example, to do a directory listing in \*NIX based operating systems you'd type:

## The power of faking it

Written by Darryn Lowe

Tuesday, 17 March 2009 00:00 - Last Updated Tuesday, 17 March 2009 10:08

---

*ls*

However, if you want to see hidden files and in a listing with recursion you'd use this:

*ls -R -la {directory}*

This is fine but long-winded if you have to do this often. However, by editing a file you can create an alias that replaces that command and switching to simply:

*dir*

It's safe to use "dir" because it is not an internal \*NIX command but don't use that for Windows because "dir" is an internal command.

As you can see aliases are quite a handy and powerful tool. Mac OS X makes it very easy to set up file aliases as well. As I mentioned in this post:

[lweb 09 import and export is there one?](#)

Setting up an alias is really nothing more than dragging a file/folder while holding down Option and Command together and plonking it in the required path.

So if you want multiple machines using a single iPhoto or iTunes library then use aliases because it's a whole lot easier than trying to keep the data all synched.