



Brett Marshall
@
Tweaksource.com

Mac OSX 10.7: (Un)Locking the Dock

The Dock in Mac OSX is attractive and fun. I have used similar docking/launching systems for years in Linux (Docky, WBar, Cairo, AWN) and found them to enhance my productivity while providing a bit of eye-candy. Thus, I have looked forward to the Dock in OSX, but have been disappointed to find a lack of configuration options.

Enter AppleScript.

I won't go into explaining AppleScript in detail or even touch on more than the basics of how to work with it in OSX. There are [plenty of good tutorials available for that](#). What I will do is share six scripts I have put together to provide a

decent enhancement to the existing options. The following scripts will allow you to:

1. Add "Spacers" to the Applications section (left side) of the Dock to aid in organization.
2. Add "Spacers" to the Documents section (right side) of the Dock to aid in organization.
3. Save a backup of the current Dock configuration.
4. Restore the Dock configuration to the saved state using the Save from script 3.
5. Lock the Dock to prevent accidental removal or shifting of Icons.
6. Unlock the Dock to allow modification.

The scripts are very simple, but illustrate the power of scripting on *nix systems. AppleScript can execute shell commands via [do shell script](#), and AppleScripts can be executed from the shell and much more. These scripts focus on executing shell commands from easily executable scripts.

DockAppSpace

```
do shell script "defaults write com.apple.dock persistent-apps -array-add
  '{tile-data={}; tile-type=\"spacer-tile\";}' && killall Dock"
```

DockOtherSpace

```
do shell script "defaults write com.apple.dock persistent-others -array-add
  '{tile-data={}; tile-type=\"spacer-tile\";}' && killall Dock"
```

DockSave

```
do shell script "cp $HOME/Library/Preferences/com.apple.dock.plist $HOME/
  Library/Preferences/com.apple.dock_.plist"
```

DockRestore

```
do shell script "cp $HOME/Library/Preferences/com.apple.dock_.plist $HOME/
  Library/Preferences/com.apple.dock.plist; killall Dock"
```

DockLock

```
do shell script "defaults write com.apple.dock contents-immutable -bool Yes;
  killall Dock"
```

DockUnlock

```
do shell script "defaults write com.apple.dock contents-immutable -bool No;
  killall Dock"
```

You will notice that the first two scripts end with '&& killall Dock' and the last four end with '; killall Dock'. I have provided two examples of legal syntaxes for chaining commands in the bash shell. Both the double-ampersand (&&) and the semicolon (;) essentially mean "and then do."

Command1 && Command2

and

Command1; Command2

are equivalent and both mean execute Command1 and then execute Command2.

To use these scripts, open the AppleScript Editor. You can launch it from Applications/Utilities, or use Command + Space to open Spotlight and type Apple and it should be the Top Hit.

First, let's save your current Dock configuration so you can restore it if you need to. Copy and Paste the **DockSave** script into the Editor. Save the file (Cmd + S) as *DockSave.scpt* and then Run (Cmd + R) the script. I made a Scripts folder in my home directory. Save it wherever you like.

Copy and Paste the **DockAppSpace** script into the Editor. Save the file as *DockAppSpace* and then Run the script. You should now have a space or gap between two icons on your Dock. This Spacer can be manipulated the same as the other icons. You can move it around, drag it off of the Dock to remove it, or whatever. Running the script multiple times creates multiple spacers.

The **DockOtherSpace** script does the same thing for the Documents (right) side of the Dock.



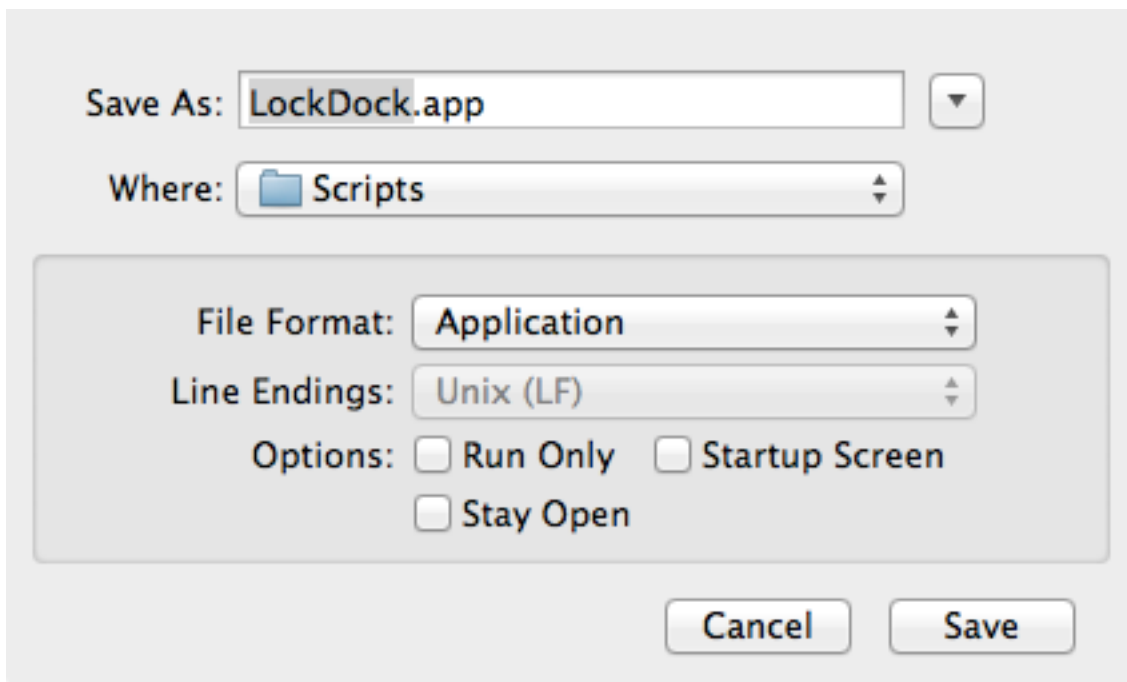
As you can see in the screenshot above, I have my icons organized into what I consider logical groups. At the far left are administration utilities such as Finder, System Preferences, Terminal, and Disk Utility client. Next are Internet apps like web browsers, mail, and FTP client. Then productivity apps like iWork, Text Editor, XCode, and Grab. Fourth are multimedia apps. At the far right are virtualization and remote access apps.

I find that this goes a long way towards helping me organize and make better use of the Dock.

The **DockRestore** script will (surprise...) restore your Dock to whatever state it was in when you last ran the **DockSave** script.

The **DockLock** and **DockUnlock** scripts lock and unlock the Dock respectively to deny or allow changes in the Dock state. This is what really got me started because I kept accidentally dragging Icons off of the Dock and “poofing” them away.

Now comes the fun part. You saved the scripts as .sct files. This is fine, but by changing the File Format in the drop-down box you can save the script as an Application with a .app extension.



Now the App can be added to your Applications Folder and run like any other app. This means you can, if you want, add it to the dock or desktop and execute it as an application. Double-clicking an .sct file will open it in the AppleScript Editor, which is probably not what you want. Using the .app is likely preferred.

If you'd like to change the icon for your app from the script icon to something else, you can. The file format for icons in OSX is .icns. There is a nifty little free app called [img2icns](#) which lets you drag-and-drop images to the window and convert them to .icns in a snap by clicking the .icns icon.



Once you have created an .icns, go to where you saved your .app file and Ctrl + click it. From the context menu select "Show Package Contents". Inside the resulting Contents folder are a.plist file (preferences), PkgInfo file, a MacOSX folder, and a Resources folder. Inside the Resource folder is an applet.icns. Replace this with your own .icns file. I renamed my icon applet.icns. It may not be needed.

It appears that Finder is notoriously bad at refreshing icons, so you have a couple of options if your new icon doesn't show up.

1. Open Terminal and execute 'killall Finder' (no quotes). Finder will die and immediately resurrect itself, at which point the icon cache will be reloaded.
2. Install or reproduce the tiny AppleScript app by Samuel Svensson called [Refresh Finder](#) which is discussed [here](#).

While it is nice to be able to have some control over my own machine, the clear importance to us as administrators is a 'window' into managing the Mac OSX environment with AppleScripts. A script like DockLock, when run as root, would prevent unprivileged users from removing icons from the Dock. Similarly, the .plist files found in /Library/Preferences afford control over System preferences, but that is for another time.