

How to Launch GUI Applications from the Terminal

Feb 1, 2007 - 29 Comments

We all know how to launch applications from the GUI with a double-click on the icon or clicking on the app in the Dock, and there are numerous ways to do so, and they're all relatively speedy. If you spend a decent amount of time with the command line though, it's nice to be able to launch Mac apps directly from there as well. Also, the Terminal has a fair share of applications that run in text based mode, but maybe you wanted to edit a text file in the OS X GUI app TextWrangler rather than the text based nano or vim.



We're going to demonstrate how to launch any graphical Mac app from the command line of OS X, including how to open specific files from the command line with a GUI app, and how to edit and open those files with root access if it's necessary.

Opening Mac OS X Applications from the Command Line

The Terminal command to launch OS X gui apps is appropriately called 'open' and here is how it works at it's most simple:

```
open -a ApplicationName
```

That will open the defined app named "ApplicationName".

But open is much more powerful than that. If you just type 'open' at the command prompt, you'll return the basic help file with details on how to properly use the command with a variety of flags and syntax. While the open command exists in all versions of Mac OS X, the abilities vary somewhat depending on what version of OS X the Mac is running. Nonetheless, in modern releases this is what you'll see:

```
$ open
Usage: open [-e] [-t] [-f] [-W] [-R] [-n] [-g] [-h] [-b ] [-a ]
[filenames] [--args arguments]
Help: Open opens files from a shell.
By default, opens each file using the default application for that
file.
If the file is in the form of a URL, the file will be opened as a
URL.
Options:
-a Opens with the specified application.
```

Subscribe to OSXDaily



Like 82K

Follow @OSXDaily

Tips & Tricks

- Try a (Very Broken) Dark Mode on MacOS High Sierra
- How to Find Where Homebrew Packages Are Installed on Mac
- Use Live Photo Effects to Capture Firework Action Photos with iPhone or iPad
- How to Reduce "System" Storage Size on iPhone or iPad
- How to Change the Safari Download Location on Mac OS

News

- MacOS High Sierra 10.13.6 Released, Security Update 2018-004 for Sierra & El Capitan too
- iOS 11.4.1 Update Released for iPhone and iPad [IPSW Download Links]
- Jailbreak iOS 11.2 – iOS 11.3.1 with Electra
- MacOS Mojave Beta 3 Download Released for Testing
- iOS 12 Beta 3 Download Available Now for Testing

iPhone / iPad

- iOS 11.4.1 Update Released for iPhone and iPad [IPSW Download Links]
- Jailbreak iOS 11.2 – iOS 11.3.1 with Electra
- Use Live Photo Effects to Capture Firework Action Photos with iPhone or iPad
- iOS 12 Beta 3 Download Available Now for Testing

```
-b Opens with the specified application bundle identifier.  
-e Opens with TextEdit.  
-t Opens with default text editor.  
-f Reads input from standard input and opens with TextEdit.  
-F --fresh Launches the app fresh, that is, without restoring  
windows. Saved persistent state is lost, excluding Untitled  
documents.  
-R, --reveal Selects in the Finder instead of opening.  
-W, --wait-apps Blocks until the used applications are closed (even  
if they were already running).  
--args All remaining arguments are passed in argv to the  
application's main() function instead of opened.  
-n, --new Open a new instance of the application even if one is  
already running.  
-j, --hide Launches the app hidden.  
-g, --background Does not bring the application to the foreground.  
-h, --header Searches header file locations for headers matching the  
given filenames, and opens them.
```

- How to Reduce "System" Storage Size on iPhone or iPad

Mac

- MacOS High Sierra 10.13.6 Released, Security Update 2018-004 for Sierra & El Capitan too

- Get Retro with a Classic Mac Finder Clone for Modern MacOS

- Try a (Very Broken) Dark Mode on MacOS High Sierra

- How to Find Where Homebrew Packages Are Installed on Mac

- MacOS Mojave Beta 3 Download Released for Testing

In other words, example simple command syntax could look like the following, opening "ApplicationName" with the file located at the path '/file/to/open':

```
open -a ApplicationName  
/file/to/open
```

You'll note you don't need the full path to the application name, but you would need the full path to a file name.

The usage is likely self explanatory to those who have experience in the command line environment, but for those who are new to the Terminal, don't be too confused, it is easy to use and we'll explain. For example, if you want to edit /etc/motd with TextWrangler to change your Message of the Day, but you hate the command line editors nano and vi, here is what you'd type:

```
$ open -a TextWrangler /etc/motd
```

Now you can edit these files in the familiar GUI. open is smart enough to know that when you apply the -a flag, you are launching an application so you don't need to type in its full path. Obviously, it'll still need the full path to the file you're editing though.

There are many other usages for the open command rather than just editing text files, so use your imagination and get creative. open could be particularly useful to system administrators who utilize it in a shell script, perhaps to launch a specific GUI application at a scheduled time.

Also worth noting is that if you are launching an application with spaces in its name, you'll want to add a backslash after each word, opening Adobe Photoshop CS would look like this:

```
$ open -a Adobe\ Photoshop\ CS
```

Launching GUI Apps as root from the Command Line

You can even open files with sudo by using the open command if you need to edit a file as root, for example:

```
sudo open -a TextEdit /tmp/magicfile
```

This will launch the target file into the desired application as root user, giving full root privileges to edit and modify the file, which is quite helpful for editing many system files. Of course, don't modify any system file if you don't know what you're doing.

Creating Shell Aliases for Frequently Launched GUI Apps

So it's kind of a pain in the butt to type a full command repeatedly, or to type out all that out over and over again, right? Well let's make it easier by assigning an alias to an application that gets frequently launched. We'll take the aforementioned Adobe Photoshop app as an example since the file name is lengthy, so here's how we'll do this with the Mac OS X default Bash shell:

First launch the profile or .bash_profile into a text editor:

```
$ nano .profile
```

or

```
$ open -e .profile
```

Ignoring whatever else may be in this file (it could be empty also), add the following to a new line:

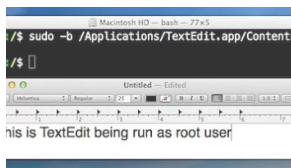
```
alias photoshop="open -a Adobe\ Photoshop\ CS"
```

This creates an alias, so that the "open -a Adobe\ Photoshop CS" command is now shortened to simply 'photoshop'. Save .profile, and you're on your way! You can use the alias command in conjunction with open for virtually anything, just be sure to pick an alias to a command that doesn't already exist.

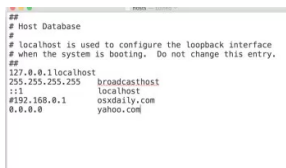


The open command is really handy as you can see, if you have any other great uses for it in OS X, be sure to let us know in the comments.

Related



[How to Run GUI Apps as root in Mac OS X](#)



[The Mac Hosts File: How to Modify /etc/hosts in OS X with TextEdit](#)



[Command Line MP3 Player in Mac OS X](#)