

# Mac Tools and Tips

## App Launching

Launchpad — built-in — Shows a display of all your apps as icons; launch by clicking on them. I think you can trigger Launchpad (by default) using the “Pinch with thumb and three fingers” trackpad gesture (toggle this gesture in System Settings -> Trackpad -> More Gestures)

There is also a Launchpad launcher in the Dock (by default?) that looks like a grid of colorful squares.



Spotlight — built-in — The Spotlight service is basically the Mac’s built-in search tool. It can be activated in many ways, but it can be by a useful app launcher when activated using the default keyboard shortcut command + space (⌘+Spc). That will open a floating search bar; start typing the name of the app you want to open, and it will find it; hit return to open the app.

This idea of hitting a keyboard shortcut like ⌘+Spc is very popular, and there are a bunch of third party apps that can be set up to replace Spotlight and do similar things. These date back a long time. Quicksilver was a really popular launcher in the 2000s; it still exists, but it has been surpassed in power and popularity by several similar apps (see for example, lists of these kinds of apps: [a](#), [b](#), [c](#)). The one that seems to be the most popular in the past few years is Alfred. A newer entry into this space is Raycast, which is free, seems to be growing in popularity, and is pretty powerful. These apps do way more than just launch other apps; they can do more general searches, run scripts, provide clipboard management, manage macros, and lots more.

### My current recommendation:

Start with ⌘+Spc to activate Spotlight. If you like the way it works, explore Alfred or Raycast as more powerful upgrades. I currently use Raycast.

## The terminal & shell

The built-in Terminal app is perfectly fine. For most users, I’d say they should use Terminal and not worry about replacing it.

There are “x11” windows that are opened using XQuartz. I think it is still true that you need to install XQuartz to allow running GUI-based X-window applications from remote machines. These terminals are kind of, well, primitive. I’ve done some searching, and I honestly could not find a reason to use xterm over a modern terminal application.

As far as I can tell, the main alternative to the built-in Terminal is iTerm2. Despite the name, it is currently on version 3.4, and is actively developed. iTerm2 brings many features to the table that make it more powerful than Terminal. One that I really like is the ease of splitting the window into multiple “panes” that are each a terminal session, and the deep integration with tmux, which means that you can also split an ssh session into multiple terminals.

Other, less popular, options: Warp (pretty cool, actually), Alacritty, and others and more others.

**My current recommendation: I think iTerm2 is the way to go.**

New Macs ship with `zsh` being the default. When I got my last MacBook Pro in 2022 (I think), I decided to give it a try. Overall, I have to say that it is fine. For most of the last two decades I have used `bash`, and `zsh` seems like a slightly better `bash`. For most users, I don't think there's any need to worry about the default shell that runs in your Mac's terminal. Maybe — just maybe — it comes up when dealing with tools like Homebrew or Conda, but I haven't had more trouble with those with `zsh` than `bash`.

If you want to squeeze more juice from your shell experience, I recommend playing with the [Oh My Zsh](#). It is a framework for managing the `zsh` configuration. It's driven by community submissions, and provides a large number of customizations. Some of them are really amazing.

Oh My Zsh has been so popular that similar frameworks have been developed for other shells, including [Oh My Bash](#).

If you are feeling adventurous, a not-very-popular-but-pretty-cool alternative shell is [fish shell](#). I've used it quite a bit, and I tend to like it better than `bash` or `zsh`, but since it is Mac-only, I haven't committed to it fully. I find that it is easier to customize and write functions in `fish` than the older shells. It also has [Oh My Fish](#), of course.

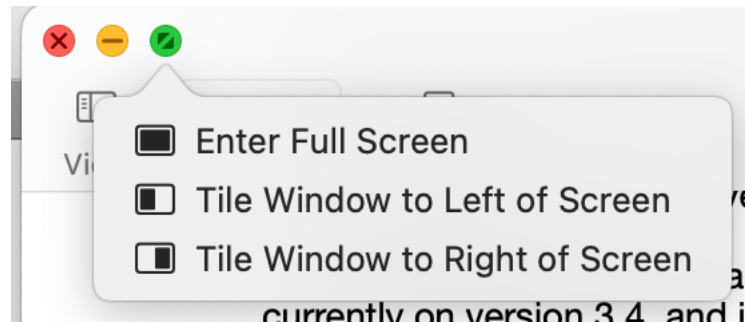
Feeling rebellious? <https://xon.sh/>

**My current recommendation: Stick with the default `zsh`, and play around with Oh My Zsh.**

## Window Management

This is a can of worms waiting to be opened. There are so many options that I don't even know all the names for them. Window management, from all indications, is a deeply personal and epistemic topic.

There are [built-in options](#) that probably do more than you realize. You know you can move a window around with the mouse, resize by dragging edges/corners, and you probably have some idea what happens when you press the yellow or green circles in the upper left. But there's more. The green circle these days provides more: hover over it and it lets you enter full screen or tile the window to the left or right. If you also press option, you can zoom the screen or move the window to the left or right. “Tiling” the window to left or right is a way to essentially split the screen, you are prompted to choose the window you want to tile to the other side. With the option-pressed version, the current window just moves to the side.



Also note that pressing option while resizing a window modifies the scaling of the window.

Control-Up (^+↑) opens [Mission Control](#), allowing you to see your open windows and [Spaces](#). Spaces are Apple's term for virtual desktops. Switch between Spaces with ^+(left,right) arrow.

Apple provides another window management approach with [Stage Manager](#). This one is relatively new to the Mac. It keeps sets of windows in “stages” along the left side of the screen,

allowing you to quickly switch between sets of windows. Toggle Stage Manager using the Control Center menu bar item.

One indispensable keyboard shortcut for me is command-tab (⌘+tab). It activates the built-in app switcher, allowing you to quickly see what apps are open and cycle through them. When engaged, keep pressing tab to cycle to the right through the apps, or use the arrow keys (while keeping ⌘ pressed).

Similar to ⌘+tab is ⌘+~ (command-tilde). This is similar, but cycle through open windows of a single app, bringing each to the front. Note that if you are using Stage Manager, ⌘+~ will cycle through all the windows in the active stage.

Okay... and then there are many, many third party options for window management.

Most of those app launchers that are mentioned above have some kind of window management built in. For example, Raycast provides a bunch of simple commands to arrange windows. When I press ⌘+space and type “left half” and hit return, the current window is moved and resized to take the left half of the current screen. There are a bunch of similar commands. Alfred provides analogous features, I think.

Other popular utilities that allow various amounts of window management:

- [better touch tool](#)
- [Moom](#)
- [Keyboard maestro](#)
- [Rectangle](#)

There are also systems that dynamically tile windows. Here’s a video that uses open source tools to set up such a tiling system with nice keyboard shortcuts: [LINK](#)

## Window arrangement with external displays

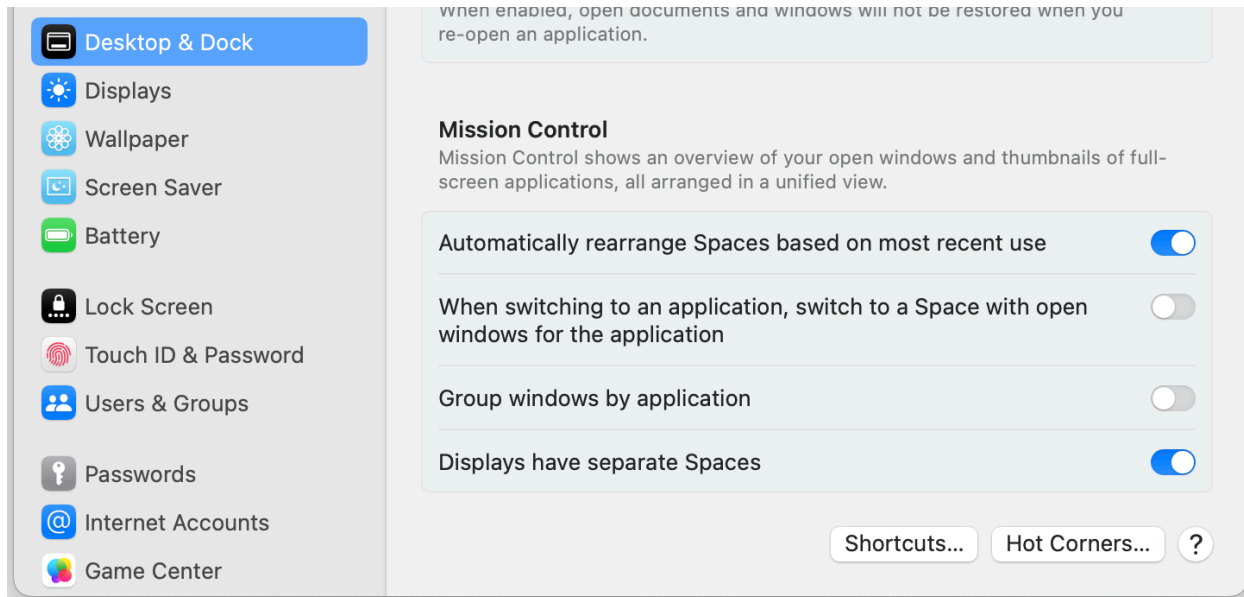
This seems to be a pain point for many users. The issue is that the computer is a laptop that is going between home and work. At work, say, the Mac is connected to an external display. The desired behavior is that apps would get rearranged to match the previous external-display arrangement.

The first to try is the built in window memory feature. Go to System Settings -> Desktop & Dock go to the bottom to the “Mission Control” section and activate the options “Automatically rearrange Spaces based on most recent use” and “Displays have separate Spaces”.

This seems to work to move open windows back to where they belong on the external display. This method *does not re-open* applications and windows that have been closed. If the desired behavior is to detect when an external display is connected and then perform a series of actions (opening apps, rearranging windows, etc.), that’s a different problem and has a more complicated solution. A first step toward a solution is to note that right-clicking on an app in the Dock allows you to “assign” the app to a specific Space (see [SE thread](#)).

Triggering an action when an external display is connected seems to be somewhat complicated (e.g., [Reddit thread](#)). I have found (but not used yet) an app called [EventScripts](#) that is supposed to be able to trigger scripts/shortcuts/etc. Combining this as a trigger with [Shortcuts](#) or similar set of actions to be run is probably the way to go. For example, if [EventScripts](#) can detect when an external display is connected and run a Shortcut that itself

steps through the process of opening several apps and arranging them, that seems like it covers >90% of the desired functionality. It might take some effort to get it to work, but once it is done, you probably don't need to worry about it much.



## Package Managers

Over the years, Mac package managers have come and gone. There was [fink](#), and then it was [MacPorts](#), and these days [Homebrew](#) is the clear winner. They all do the same thing: provide a simplified interface for installing unix-like applications. That is, they save us from having to build everything from source and manage all the dependencies on our own. While they provide a great convenience, these systems always come with some caveats. First, they are a little bit black-boxy, and you won't always know exactly what they are doing. Second, they are not official tools, so somethings break when Apple changes something (OS updates can sometimes mess things up, but less so in the last 5ish years). Third, sometimes things go wrong, and it can be hard to diagnose and recover if an installation really gets messed up. Even so, I use [Homebrew](#) for installing a few basic utilities and I rarely run into major problems.

Note: Apparently [fink](#) and [MacPorts](#) are still active, but I would not use them at this time just because [Homebrew](#) seems vastly more popular.

Also note, [Conda](#) (and now [Mamba](#)) are very similar in spirit to [Homebrew](#). Really they are both package managers that can install pre-built binaries on the Mac. They don't always play nicely together, but if you are careful to try to avoid having them interact, it seems that they can coexist just fine. Don't cross the streams!

## Clipboard Managers

Clipboard managers solve the age-old problem of copying something, eventually copying something else, and then wanting to paste that first thing again. You don't want to have to go back and find it and copy it again, even if the source of the thing is still available. The clipboard can have a history, and if you have a tool to access the entries of that history, then you don't need to copy again.

If you're using Raycast or Alfred or some similar app, they supply clipboard management tools. For example, I press  $\mathbb{A}$ +space to open the Raycast interface, start typing "Clipboard History" (I just type "c" and it comes up as the first entry), press Return, and I have a (filterable) list of my past copies to select and paste. Keyboard Maestro also offers clipboard history (but I think the Raycast version is a better interface).

There are also dedicated clipboard managers. These are small utility apps that are really just concerned with offering clipboard management. I think Pastebot is probably the most popular one, but there are others: Maccy, Copy 'Em, Jumpcut, Paste, and a zillion others.

## Special Case: Paste on Middle Mouse Button Click

Many Linux users have missed being able to middle-click to paste after moving to MacOS (examples: [2009](#), [2011](#), [2014](#), [2023](#)).



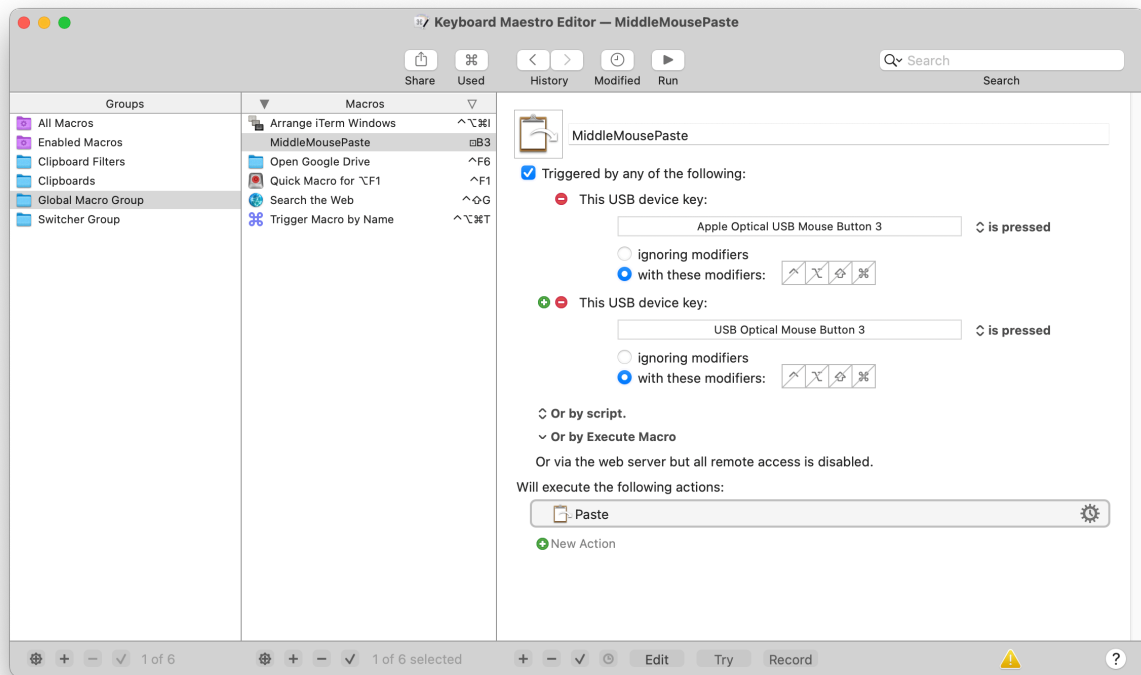
I have been trying to figure out if there's a "right way" to achieve this behavior. It's somewhat more complicated than you'd expect. Some of the responses in those linked discussions seem to indicate that the fundamental issue is that MacOS doesn't have good support for the 3-button mouse because Apple doesn't make such a mouse. Historically the official Apple mouse only had one button (but has had  $\geq 2$  buttons since 2005).

So what's a recovering Linux user to do?

I dug out a cheap, wired, 3-button AmazonBasics mouse for testing. When I clicked the scroll wheel (which I assume is button 3, i.e., middle button) it actually activated my App Switcher (same as  $\mathbb{A}$ +tab). I couldn't find any MacOS settings that allow me to change what that button does. As it turns out, I think this might be a setting from years ago when I had an Apple Mighty Mouse. Luckily I have some hoarder-like tendencies, and I still have that mouse! I found it, connected it, and some of the mouse settings did actually change. Amazingly, MacOS still has specific settings for the Mighty Mouse, and that little trackball does indeed trigger the App Switcher, and I was able to simply switch it to "Button 3". With the Mighty Mouse, I wasn't really able to get my new "Button 3" to do anything useful, but when I switched back to the cheap Amazon mouse, I could.

Here's what I did. Using Keyboard Maestro, I set up a new macro that I called "MiddleMousePaste" (see screenshot below). It triggers on USB device key, and I was able to enter the button 3 press with the Amazon mouse. On Button 3 press, I just have it execute "Paste." This works. Once I confirmed that, I went back to the Mighty Mouse to figure out why that didn't work with this macro, and discovered that it gets recognized as a different kind of device. I was able to add another trigger for "Apple Optical USB Mouse Button 3," and then it works just like the Amazon mouse.





I think Better Touch Tool would be able to set a similar macro.

During research for this, I also saw “[Mac Mouse Fix](#),” and that might be another option. Some of the discussion linked above also recommend [Karabinier-Elements](#).

## Additional Resources

---

### Podcasts

- [Mac Power Users](#)
- [Automators](#)

---

### Websites / Blogs

- <https://macautomationtips.com/>
- <https://macosxautomation.com/>