

Running at startup: when to use a Login Item or a LaunchAgent/LaunchDaemon

Published: 2018-05-22 · Archived: 2026-04-05 23:03:10 UTC

There are two popular ways of getting software to run whenever your Mac starts up: it can be installed as a Login Item, or as a LaunchAgent or LaunchDaemon. This article looks at the differences between these mechanisms, so that you can decide which to use in each case, understand what can go wrong with them, and how to fix them.

launchd

macOS brought together the functions of a disparate group of Unix tools, including `(x)inetd`, `init`, and `watchdogd`, in a single service manager `launchd`, which is controlled by `launchctl`. During startup, once the kernel is running and required kernel extensions have been loaded, `launchd` is run with the process ID of 1, and it remains running until your Mac shuts down again.

Before you have logged in, `launchd` runs services and other components which are specified in Property List files in the LaunchAgents and LaunchDaemons folders in `/System/Library`, and in `/Library`. Those in `/System/Library` are all part of macOS, owned by Apple, and protected by SIP, but those in `/Library` include many installed by third party products. As they're run before the user logs in, they work for all users, so are global services.

Once you have logged in, `launchd` runs any services and other components specified in any LaunchAgents folders in `~/Library`. Those are user-specific.

These Property List files contain keyed settings which determine what `launchd` does with what. Although they can contain many other key-value pairs, two most important ones are **ProgramArguments** (or **Program**), which tells `launchd` what to run, and **RunAtLoad**, which determines whether `launchd` should run the service or app whenever your Mac starts up and `launchd` loads those agents and services.

Other keys determine whether the agent/service should be kept running at all times; if that is set, if it crashes or is otherwise terminated, `launchd` will automatically start it up again. That is important for background services which apps rely on to function.

You can use this `launchd` mechanism to run a GUI app as a LaunchAgent if you wish. To run an app such as Bailiff, for example, you would set the **ProgramArguments** to load the executable code within that app, say at `/Applications/Bailiff.app/Contents/MacOS/Bailiff`. Note that this is not the top level of the app bundle, which would be at `/Applications/Bailiff.app`

`launchd` is best suited to background services, which use it the most. You don't need any tools beyond a Property List or text editor to craft its configuration files, although Lingon from [Peter Borg](#) makes them much more accessible. It's also widely abused by malware, which often installs its own LaunchAgents and LaunchDaemons to ensure its persistence across reboots, and to perform services which it requires.

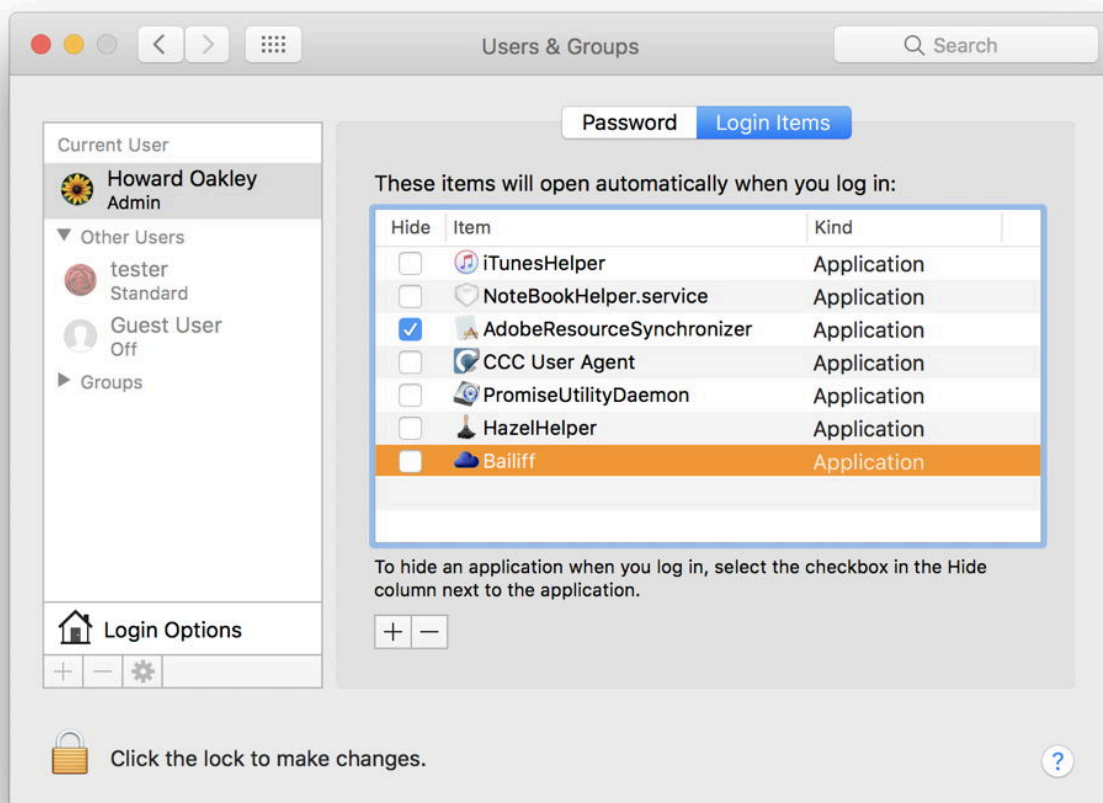
Apple defines a **LaunchDaemon** as a general background service, without any form of GUI. Specifically, LaunchDaemons are not allowed to connect to the macOS window server. **LaunchAgents** do have access to the GUI, and to other system features such as locating the current user's Home folder, but generally have much more limited interfaces than do regular apps.

Uninstalling a LaunchAgent or LaunchDaemon is simply a matter of trashing its Property List from the appropriate folder – a task sometimes necessary when you have uninstalled an app which doesn't clean up properly after itself. If software does need to install one or more Property Lists in any of these folders, then it should also provide an effective uninstaller to remove them when required.

[Apple's documentation](#) on `launchd` is quite old now, but is still a valuable reference, as is [TN2083](#).

Login Items

These are controlled quite differently, through LaunchServices rather than `launchd`, which runs later following startup. You can set any app, service, or other executable code to run at startup by adding the item to the list of **Login Items** in the **Users & Groups** pane in System Preferences.



Once added, LaunchServices puts them into its list of apps to launch at startup, which is also kept in a Property List file. In Sierra and earlier, that is located in `~/Library/Preferences/com.apple.loginitems.plist`, but High Sierra

has moved that, changed its extension, and the file structure, to ~/Library/Application Support/com.apple.backgroundtaskmanagementagent/backgrounditems.btm

The older Property List isn't intended to be maintained directly by the user. For each Login Item, it gives an 'alias' which isn't a normal macOS Bookmark, and some CustomItemProperties, which are also Base-64 encoded opaque data. The new format in High Sierra is even more opaque: although a Property List, it is now a keyed archive containing objects which are referenced by UUID, and unsuitable for manual editing.

There's an easy way to add apps which are already in the Dock to the Login Items list: click and hold on their icon in the Dock until its menu pops up. In that, select **Options**, then the **Open at Login** command.

You can add and remove Login Items using a scripting language; for example, in AppleScript

```
tell application "System Events" to make login item at end with properties  
{path:"/Applications/MyApp.app", hidden:false}
```

adds MyApp.app as a Login Item,

```
tell application "System Events" to get the name of every login item  
lists all Login Items, and
```

```
tell application "System Events" to delete login item "name"
```

removes the named Login Item from the current list.

Login Items are better suited to apps of any size which have significant user interfaces, and anything which a user wants to control easily. Many apps, such as menubar or 'Status Bar' apps, offer the user the option of installing them as a Login Item. Unfortunately, particularly when this is done for an app provided through the App Store, this turns out to be a complex development task which requires a helper app.

If you're interested in understanding this in detail, [this item on Stack Overflow](#) considers some of the problems, [this tutorial](#) proposes one solution, and [this is another tutorial](#) account. Tomorrow I'll be showing a much simpler way which works with apps which are not sandboxed (thus not intended for the App Store).

Sorting out LaunchServices problems is not easy. If you're seeing weird things going on with your Login Items list, the best solution is usually to start from scratch by trashing the current file (located according to whether you're running Sierra and earlier, or High Sierra), restarting, and setting up your Login Items again. Note that in some versions of macOS, until there is at least one Login Item, the settings file may not exist.

Just leave it running

The third and last way of opening an app when you next start up is simply to leave the app running when you shut down or restart. It does, of course, rely on you remembering to check that the app is running beforehand.