

[Apple silicon](#) / Installing a custom kernel extension

Article

Installing a custom kernel extension

Install kernel extensions using a custom installer package, and help users understand the installation process.

Overview

A kernel extension (or kext) is a bundle that performs low-level tasks. Kexts run in kernel space, which gives them elevated privileges and the ability to perform tasks that user-space apps can't.

Always consider alternatives before creating a kext. The system provides numerous APIs to minimize the need for kexts. For example, you can use the DriverKit SDK to implement drivers for most types of hardware. In addition, the kernel runtime environment has more stringent code-level requirements, which makes the creation of kexts more challenging, and the consequences more severe if you do it incorrectly.

For information about alternatives to kexts, see [Implementing drivers, system extensions, and kexts](#).

Load Kexts as the Final Step in Your Installer Package

The most common way to install kexts is with a custom installer package. (Unlike apps and system extensions, you cannot distribute kexts through the Mac App Store.) Typically, you use a single package to copy each kext and any supporting software to the user's system, and then you use scripts to load and configure the kexts for execution.

When creating your installer package, load your kexts as the final step in the installation process using the `kmutil` or `kextload` tool. These tools initiate the kext-loading process, which on macOS 11 requires getting the user's permission and rebooting the user's system. Loading them as the final step prevents the reboot from disrupting the rest of the installation process. To avoid multiple user prompts when installing multiple kexts, make a single call to `kmutil` or `kextload` and include all of your kexts in the parameter list.

Note

In macOS 11 and later, `kmutil` replaces `kextload`, `kextunload`, and other earlier tools for loading and managing kexts. The older tools still work, but their implementations now call `kmutil`.

For more information about how to load kexts, run `kmutil load -h` in Terminal.

Explain the Kext-Loading Process to Users

In macOS 11 and later, the loading of a kext requires the user to take actions outside of your installation package. When loading a kext, tell the user to perform the following one-time setup on Apple silicon:

1. Reboot your Mac with Apple silicon into Recovery mode.
2. Set the security level to Reduced security.
3. Allow the loading of third-party kexts.
4. Reboot back to macOS.

Tell the user to perform the following steps on all Macs whenever you load or install a previously unused kext:

1. Open the Security & Privacy System Preferences.
2. Authenticate to make changes.
3. Allow the system to load your kext.
4. Wait for the system to load the kext and rebuild the auxiliary kext collection.
5. Reboot to load the new auxiliary kext collection.

The system prompts the user through most of this process, but make sure your installer package provides clear instructions about what to do. Rebooting is a disruptive process for the user, and lowering the security protections may cause concern. Providing a clear explanation of what your kext does, and why its installation is necessary may alleviate some of those concerns. Providing explicit instructions for the user to follow also helps them navigate the reboot process. If you don't provide these instructions, the user may accidentally or purposefully abort the installation process for your kext.

Note

For codeless kexts, the system asks the user for permission to install the kext, but doesn't reboot the user's system.

Update an Existing Kernel Extension

To update an existing kext, install that kext and use the `kmutil` or `kextload` tool to initiate the loading process. For a previously approved and loaded kext, the user must perform the following steps to update that kext:

1. Open the Security & Privacy System Preferences.
2. Authenticate to make changes.
3. Allow the system to update the kext.
4. Reboot to install the updated kext.

When the user consents to update the kext, the system rebuilds the kext collection with the updated version. However, the new collection doesn't take effect until after the user reboots the system. If the user defers the reboot process, the system continues to run with the previous version of the kext.

Unload a Kernel Extension

Unloading a kext on macOS 11 and later requires a call to `kmutil` or `kextunload`, followed by a system reboot. The `kmutil` tool builds a new kext collection without the specified kext, but it doesn't install that collection immediately. The system installs the new kext collection only after the computer reboots. As a result, the unloaded kext actually remains active and running until the user reboots the system.

For more details, see the `kmutil(8)` man page.

See Also

Kernel and drivers

 [Debugging a custom kernel extension](#)

Configure your system to enable the debugging of custom kernel extensions from a second Mac.