

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

Article

Debugging a custom kernel extension

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

Overview

The kernel supports every other process on the system, including the debugger itself, so you can't debug it using normal techniques. If you develop a kernel extension (kext), you have several options to debug it:

- Write debug information to logs and analyze the logs later.
- Use a second computer to debug the kernel on another Mac.
- Debug or inspect a kernel core file from a kernel panic.

Use the Kernel Debug Kit to set up two-machine debugging and an optional core-dump server. Use two-machine debugging to dynamically examine the state of your kext at runtime. If you don't save core-dump files directly to the Mac you use for debugging, a core-dump server captures them automatically over the network.

Install the Kernel Debug Kit

The Kernel Debug Kit (KDK) provides the tools and support you need to debug the kernel. This separate download includes custom versions of the kernel that include extra assertions and error checking to help you catch problems.

To use the KDK to perform two-machine debugging:

1. Connect both Macs to the same network.
2. Log in as an administrator on both Macs.
3. Download the KDK for the version of macOS you're debugging.

4. Install the KDK on both Mac computers.

5. Install your kext (and the appropriate kernel) on the computer you want to debug.

The KDK includes instructions on how to install the custom kernels you use during debugging. It also provides detailed instructions for setting up the connection between your computers. For more information, sign in and download the appropriate KDK from <https://developer.apple.com/download/more/>.

Debug the Kext

To begin a debugging session, designate one machine as the debugging machine and the other as the target. Run `lldb` from Terminal on your debugging machine and use the `kdp-remote` command to connect to the remote computer. The `kdp-remote` command accepts the hostname or IP address and creates a debugger connection to the corresponding Mac. Specify the name or IP address of your target Mac as the parameter to the command.

```
(lldb) kdp-remote MyComputer.local
```

After you establish a connection, set breakpoints and step through your custom code as you would for any other process. The debugger automatically searches any spotlight-indexed directories and the `/Library/Developer/KDKs/` directory on the local system for symbol information to use during debugging. You can specify the location of the kernel symbol files by launching `lldb` with the path to the kernel file that contains those symbols. For example:

```
% lldb /Library/Developer/KDKs/<KDK Version>/System/Library/Kernels/kernel
```

For more information about two-machine debugging, see the KDK Readme file located in the `/Library/Developer/KDKs/<KDK Version>/` directory of the KDK.

.

See Also

Kernel and drivers

 Installing a custom kernel extension

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

