Sample Code

Building an Audio Server Plug-in and Driver Extension

Create a plug-in and driver extension to support an audio device in macOS.

Download

DriverKit 20.0+ | macOS 11.0+ | Xcode 12.2+

Overview

This sample shows how to create an Audio Server plug-in built on top of a DriverKit Driver Extension. The sample provides C++ implementations of the plug-in and the Driver Extension, and shows how to implement the communication between the two, including discovering, connecting to, and calling into the Driver Extension from the plug-in.

The sample implements a dynamic environment that supports multiple audio devices. The plug-in's audio devices provide the following features:

- Configurable device master volume.
- Support for 44.1 kHz and 48 kHz sample rates
- Support for two channels of audio I/O in 32-bit, floating point, linear PCM format

Configure the Sample Code Project

To deploy the sample driver, you need to create an explicit App ID and provisioning profile with the following entitlements:

- com.apple.developer.driverkit
- com.apple.developer.driverkit.transport.usb

To learn how to perform this configuration, see Requesting Entitlements for DriverKit Development.

To bypass this configuration and use ad hoc signing to test the driver in your local development environment, perform the following steps:

1. Disable System Integrity Protection (SIP) on your system so it recognizes ad hoc-signed DriverKit extensions. For more information, see Disabling and Enabling System Integrity Protection.

2. Configure the `SimpleAudioDriver` target to use local signing. Select the target, and then select its Build Settings tab. Find the Code Signing Identity build setting and select Sign to Run Locally.

Use the included installer script to install the driver on your system. Navigate to the root of the sample project and run the following command:

```
./Scripts/install.sh
```

Reboot your computer so that the system recognizes the driver. After your computer restarts, open Audio MIDI Setup to inspect the newly installed device.

To uninstall the driver, run the `uninstall.sh` script and reboot your computer.

See Also

Drivers

{} Creating an Audio Server Driver Plug-in

Build a virtual audio device by creating a custom driver plug-in.

{} Capturing system audio with Core Audio taps

Use a Core Audio tap to capture outgoing audio from a process or group of processes.