AVFoundation / Capture setup / Choosing a capture device

Article

Choosing a capture device

Select the front or back camera, or use advanced features like the TrueDepth camera or dual camera.

Overview

Devices offer many options for capturing photos and video, including front- and back-facing cameras, dual cameras, and the TrueDepth camera. Choosing the appropriate camera automatically or offering a user interface for camera selection is an important part of developing any app with camera features.

AVFoundation offers two main paths to selecting a camera device: the `default(_:for:position:)` method and the `AVCaptureDevice.DiscoverySession` class.

Quickly choose a default device

If you know exactly what kind of capture devices you're looking for, use one of the `AVCaptureDevice` convenience methods to select a default device. For example, the code below selects the best available back-facing camera: either the dual camera on supported devices, or the single (wide-angle) camera on single-camera devices.

```swift
if let device = AVCaptureDevice.default(.builtInDualCamera,
                                        for: .video, position: .back) {
    return device
} else if let device = AVCaptureDevice.default(.builtInWideAngleCamera,
                                               for: .video, position: .back) {
    return device
} else {
    fatalError("Missing expected back camera device.")
}
```

Sort and filter devices with a discovery session

To see the entire set of devices matching certain criteria so that you can use your own logic to choose one, use the `AVCaptureDevice.DiscoverySession` class. First, create a discovery session for the kinds of devices you need:

```
let discoverySession = AVCaptureDevice.DiscoverySession(deviceTypes:
    [.builtInTrueDepthCamera, .builtInDualCamera, .builtInWideAngleCamera],
    mediaType: .video, position: .unspecified)
```

Then, read the discovery session's `devices` list to find matching devices and choose one that suits your needs. The discovery session automatically sorts its `devices` list based on the device types you asked for, so you can use the array order to find the best device with certain features. For example, because the discovery session shown above searches for depth-capable devices before the wide-angle camera and allows any device position, the first item in its devices list matching a specified position is the best depth capture device for that position (or a fallback device):

```
func bestDevice(in position: AVCaptureDevice.Position) -> AVCaptureDevice {
    let devices = self.discoverySession.devices
    guard !devices.isEmpty else { fatalError("Missing capture devices.")}

    return devices.first(where: { device in device.position == position })!
}
```

See Also

Capture devices

Adopting smart framing in your camera app

Capture the optimal shot by providing automatic framing recommendations.

`class AVCaptureDevice`

An object that represents a hardware or virtual capture device like a camera or microphone.

`class AVCaptureDeviceInput`

An object that provides media input from a capture device to a capture session.

`class AVContinuityDevice`

A class that represents a physical iOS device that’s nearby and can provide access to its cameras and microphones.

```
class AVExternalStorageDevice
```

Represents a physical external storage device that stores media assets.

```
class AVExternalStorageDeviceDiscoverySession
```

Informs your app when the external storage devices connect to and disconnect from the system.