

[AVFoundation](#) / Capture setup

API Collection

Capture setup

Configure built-in cameras and microphones, and external capture devices, for media capture.

Overview

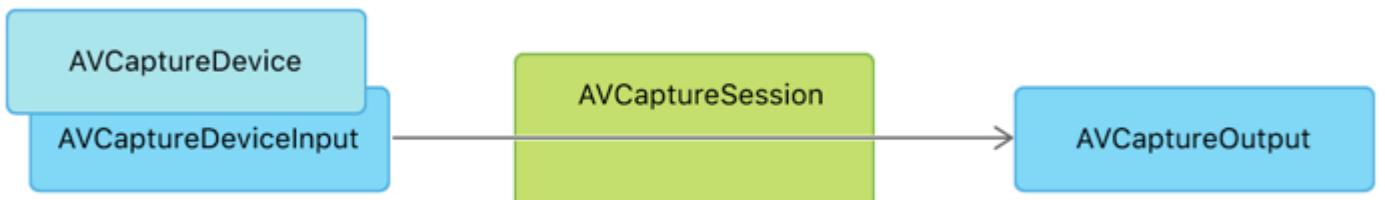
The AVFoundation Capture subsystem provides a common high-level architecture for video, photo, and audio capture services in iOS and macOS. Use this system if you want to:

- Build a custom camera UI to integrate shooting photos or videos into your app's user experience.
- Give users more direct control over photo and video capture, such as focus, exposure, and stabilization options.
- Produce different results than the system camera UI, such as RAW format photos, depth maps, or videos with custom timed metadata.
- Get live access to pixel or audio data streaming directly from a capture device.

Note

To instead let the user capture media with the system camera UI within your app, see [UIImagePickerController](#).

The main parts of the capture architecture are sessions, inputs, and outputs: Capture sessions connect one or more inputs to one or more outputs. Inputs are sources of media, including capture devices like the cameras and microphones built into an iOS device or Mac. Outputs acquire media from inputs to produce useful data, such as movie files written to disk or raw pixel buffers available for live processing.



Topics

Essentials

- 📄 Requesting authorization to capture and save media
 - Prompt the user to authorize access to the camera, microphone, and photo library.

- 📄 Setting up a capture session
 - Configure input devices, output media, preview views, and basic settings before capturing photos or video.

- 📄 Accessing the camera while multitasking on iPad
 - Operate the camera in Split View, Slide Over, Picture in Picture, and Stage Manager modes.

- { } AVCam: Building a camera app
 - Capture photos and record video using the front and rear iPhone and iPad cameras.

- { } Capturing Cinematic video
 - Capture video with an adjustable depth of field and focus points.

- { } AVMultiCamPiP: Capturing from Multiple Cameras
 - Simultaneously record the output from the front and back cameras into a single movie file by using a multi-camera capture session.

- { } AVCamBarcode: detecting barcodes and faces
 - Identify machine readable codes or faces by using the camera.

class AVCaptureSession

An object that configures capture behavior and coordinates the flow of data from input devices to capture outputs.

class AVCaptureMultiCamSession

A capture session that supports simultaneous capture from multiple inputs of the same media type.

class `AVCaptureInput`

An abstract superclass for objects that provide input data to a capture session.

class `AVCaptureOutput`

An abstract superclass for objects that provide media output destinations for a capture session.

class `AVCaptureConnection`

An object that represents a connection from a capture input to a capture output.

Capture devices

Choosing a capture device

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

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.

Capture preview

```
class AVCaptureVideoPreviewLayer
```

A Core Animation layer that displays video from a camera device.

```
class AVCaptureAudioPreviewOutput
```

A capture output that provides a preview of the captured audio.

Continuity Camera

{ } Supporting Continuity Camera in your tvOS app

Capture high-quality photos, video, and audio in your Apple TV app by connecting an iPhone or iPad as a continuity device.

{ } Supporting Continuity Camera in your macOS app

Enable high-quality photo and video capture by using an iPhone camera as an external capture device.

```
class AVCaptureDeskViewApplication
```

An object that programmatically presents Desk View.

Capture controls

📄 Enhancing your app experience with the Camera Control

Provide direct access to your camera app's features to help people quickly capture the perfect shot.

```
class AVCaptureControl
```

An abstract base class for controls that interact with the camera system.

```
class AVCaptureSystemZoomSlider
```

A control that adjusts the video zoom factor of a capture device within the system-recommended range.

```
class AVCaptureSystemExposureBiasSlider
```

A control that adjusts the exposure bias of a capture device within the system-recommended range.

```
class AVCaptureSlider
```

A slider control that selects a value from a bounded range.

```
class AVCaptureIndexPicker
```

A control for selecting from a set of mutually exclusive values by index.

External display output

```
class AVCaptureExternalDisplayConfiguration
```

A class you use to specify a configuration to your external display configurator.

```
class AVCaptureExternalDisplayConfigurator
```

A configurator class allowing you to configure properties of an external display to match the camera's active video format.

Timecode generation

```
class AVCaptureTimecodeGenerator
```

Generates and synchronizes timecode data from various sources for precise video and audio synchronization.

```
protocol AVCaptureTimecodeGeneratorDelegate
```

A protocol for receiving real-time timecode updates and error notifications from a timecode generator.

```
enum SynchronizationStatus
```

Constants defining the synchronization status of a timecode generator .

```
class Source
```

Describes a timecode source that a timecode generator can synchronize to.

```
enum SourceType
```

Defines possible sources for generating timecode in using a timecode generator.

```
struct AVCaptureTimecode
```

This structure represents a timecode, adhering to SMPTE standards, which define precise time information and associated timestamps for video or audio synchronization.

```
static func advanced(AVCaptureTimecode, by: Int64) -> AVCaptureTimecode
```

Generates a new timecode by adding a specified number of frames to the given timecode, handling overflow for seconds, minutes, and hours.

```
static func createMetadataSampleBuffer(from: AVCaptureTimecode,  
associatedWithPresentationTimeStamp: CMTime) -> Unmanaged<CMSSample  
Buffer>?
```

Creates a sample buffer containing Timecode Media Description metadata for integration with a video track.

```
static func createMetadataSampleBuffer(from: AVCaptureTimecode, for Duration: CMTime) -> Unmanaged<CMSampleBuffer>?
```

Creates a sample buffer containing Timecode Media Description metadata for a specified duration.

External synchronization

```
class AVExternalSyncDevice
```

An external sync device connected to a host device that can be used to drive the timing of an internal component, such as a camera sensor.

```
protocol AVExternalSyncDeviceDelegate
```

Defines an interface for delegates of [AVCaptureDeviceInput](#) to respond to events that occur when connecting, calibrating, and disconnecting external sync devices.

```
enum AVExternalSyncDeviceStatus
```

Connection state of an external sync device

```
class DiscoverySession
```

A means of discovering and monitoring connection / disconnection of external sync devices to the host.

See Also

Capture

☰ Photo capture

Capture high-quality still images, Live Photos, and supporting photo data.

☰ Audio and video capture

Capture audio and video directly to media files, or capture streams of media for direct access to media sample buffers.

☰ Additional data capture

Capture additional data including depth and metadata, and synchronize capture from multiple outputs.