Documentation    Language: **Swift**

AVFoundation / AVCaptureDevice

Class

# AVCaptureDevice

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

iOS 4.0+ | iPadOS 4.0+ | Mac Catalyst 14.0+ | macOS 10.7+ | tvOS 17.0+ | visionOS 1.0+

```
class AVCaptureDevice
```

## Mentioned in

- Choosing a capture device
- Enhancing your app experience with the Camera Control
- Requesting authorization to capture and save media
- Setting up a capture session

## Overview

Capture devices provide media data to capture session inputs that you connect to an `AVCaptureSession`. An individual device can provide one or more streams of media of a particular type.

You don’t create capture device instances directly. Instead, retrieve them using an instance of `AVCaptureDevice.DiscoverySession`, or by calling the `default(_:for:position:)` method.

A capture device provides several configuration options. Before attempting to configure device properties, such as its focus mode, exposure mode, and so on, you must first acquire a lock on the device by calling the `lockForConfiguration()` method. You should also query the device’s capabilities to ensure that the new modes you intend to set are valid for the device. You can then set the properties and release the lock using the `unlockForConfiguration()` method. You may hold the lock if you want all settable device properties to remain unchanged. However, holding the device lock unnecessarily may degrade capture quality in other apps sharing the device and isn’t recommended.

# Topics

## Finding and monitoring devices

```
class DiscoverySession
```

An object that finds capture devices that match specific search criteria.

```
class func `default`(AVCaptureDevice.DeviceType, for: AVMediaType?,
position: AVCaptureDevice.Position) -> AVCaptureDevice?
```

Returns the default device for the specified device type, media type, and position.

```
class func `default`(for: AVMediaType) -> AVCaptureDevice?
```

Returns the default device that captures the specified media type.

```
init?(uniqueID: String)
```

Creates an object that represents a device with the specified identifier.

```
class let wasConnectedNotification: NSNotification.Name
```

A notification the system posts when a new capture device becomes available.

```
class let wasDisconnectedNotification: NSNotification.Name
```

A notification the system posts when an existing device becomes unavailable.

~~class func devices(for: AVMediaType) -> [AVCaptureDevice]~~

Returns devices capable of capturing media of the specified type.

Deprecated

~~class func devices() -> [AVCaptureDevice]~~

Returns all available capture devices on the system.

Deprecated

## Authorizing device access

```
class func requestAccess(for: AVMediaType, completionHandler: (Bool) ->
Void)
```

Requests the user’s permission to allow the app to capture media of a particular type.

```
class func authorizationStatus(for: AVMediaType) -> AVAuthorizationStatus
```

Returns an authorization status that indicates whether the user grants the app permission to capture media of a particular type.

```
enum AVAuthorizationStatus
```

Constants that indicate the status of an app's authorization to capture media.

## Identifying a device

```
var uniqueID: String
```

An identifier that uniquely identifies the device.

```
var modelID: String
```

A model identifier for the device.

```
var localizedName: String
```

A localized device name for display in the user interface.

```
var manufacturer: String
```

A human-readable string for the manufacturer of the device.

```
var deviceType: AVCaptureDevice.DeviceType
```

The type of device, such as a built-in microphone or wide-angle camera.

```
struct DeviceType
```

A structure that defines the device types the framework supports.

```
var position: AVCaptureDevice.Position
```

The physical position of the capture device hardware.

```
enum Position
```

Constants that indicate the physical position of a capture device.

## Accessing device state

```
var isConnected: Bool
```

A Boolean value that indicates whether a device is currently connected to the system and available for use.

```
var isSuspended: Bool
```

A Boolean value that indicates whether the device is in a suspended state.

```
var isInUseByAnotherApplication: Bool
```

A Boolean value that indicates whether another app is using the device.

## Inspecting device characteristics

```
var isVirtualDevice: Bool
```

A Boolean value that indicates whether the device consists of two or more physical devices.

```
var constituentDevices: [AVCaptureDevice]
```

An array of physical devices that make up a virtual device.

```
func hasMediaType(AVMediaType) -> Bool
```

Returns a Boolean value that indicates whether the device captures media of a particular type.

```
var transportType: Int32
```

The transport type of the device.

```
func supportsSessionPreset(AVCaptureSession.Preset) -> Bool
```

Returns a Boolean value that indicates whether you can use the device with capture session configured with the specified preset.

## Monitoring device rotation

```
class RotationCoordinator
```

A class that monitors the physical orientation of a capture device and provides adjustment angles to keep images level, relative to gravity.

## Configuring camera hardware

```
func lockForConfiguration() throws
```

Requests exclusive access to configure device hardware properties.

```
func unlockForConfiguration()
```

Releases exclusive control over device hardware properties.

```
var isSubjectAreaChangeMonitoringEnabled: Bool
```

A Boolean value that indicates whether the device monitors the subject area for changes.

```
class let subjectAreaDidChangeNotification: NSNotification.Name
```

A notification the system posts when a capture device detects a substantial change to the video subject area.

Formats

Configure capture formats and camera frame rates.

Focus

Configure the automatic focus behavior of a camera, or manually set its lens position.

Exposure

Configure the automatic exposure behavior of a camera, or manually control its exposure settings.

White balance

Configure the automatic white balance behavior of a camera, or manually control white balance settings.

Lighting

Configure the device flash, torch, and low light settings.

Color

Manage HDR and color space settings for a device.

Zoom

Configure device zooming behavior and inspect hardware capabilities.

## Configuring Cinematic video

```
func setCinematicVideoFixedFocus(at: CGPoint, focusMode: AVCaptureDevice.CinematicVideoFocusMode)
```

Fix focus at a distance.

```
func setCinematicVideoTrackingFocus(at: CGPoint, focusMode: AVCaptureDevice.CinematicVideoFocusMode)
```

Focus on and start tracking an object if it can be detected at the region specified by the point.

```
func setCinematicVideoTrackingFocus(detectedObjectID: Int, focusMode: AVCaptureDevice.CinematicVideoFocusMode)
```

Focus on and start tracking a detected object.

```
enum CinematicVideoFocusMode
```

Constants indicating the focus behavior when recording a Cinematic Video.

```
struct AVCaptureSceneMonitoringStatus
```

An informative status about the scene observed by the device.

```
static let notEnoughLight: AVCaptureSceneMonitoringStatus
```

The light level of the current scene is insufficient for the current set of features to function optimally.

```
var cinematicVideoCaptureSceneMonitoringStatuses: Set<AVCaptureSceneMonitoringStatus>
```

The current scene monitoring statuses related to Cinematic Video capture.

## Configuring smart framing

```
var smartFramingMonitor: AVCaptureSmartFramingMonitor?
```

A monitor owned by the device that recommends an optimal framing based on the content in the scene.

```
class AVCaptureSmartFramingMonitor
```

An object associated with a capture device that monitors the scene and suggests an optimal framing.

```
class AVCaptureFraming
```

A framing, consisting of an aspect ratio and a zoom factor.

## Configuring dynamic aspect ratio

```
func setDynamicAspectRatio(AVCaptureDevice.AspectRatio, completionHandler: ((CMTime, (any Error)?) -> Void)?)
```

Updates the dynamic aspect ratio of the device.

```
struct AspectRatio
```

String constants describing the different video aspect ratios you can configure for a particular device.

```
var dynamicAspectRatio: AVCaptureDevice.AspectRatio?
```

A key-value observable property indicating the current aspect ratio for a device.

```
var dynamicDimensions: CMVideoDimensions
```

A key-value observable property describing the output dimensions of the video buffer based on the device's dynamic aspect ratio.

## Enabling automatic frame rate

```
var isAutoVideoFrameRateEnabled: Bool
```

A Boolean value that indicates whether the capture device performs automatic video frame rate adjustments.

## Supporting spatial capture

```
var spatialCaptureDiscomfortReasons: Set<AVSpatialCaptureDiscomfortReason>
```

Reasons why current environmental conditions aren't suitable to capturing spatial videos that are comfortable to view.

```
struct AVSpatialCaptureDiscomfortReason
```

Constants that indicate the suitability of the current scene to create a comfortable viewing experience.

## Supporting Continuity Camera

```
class var systemPreferredCamera: AVCaptureDevice?
```

A camera the system prefers to use for video and photo capture.

```
class var userPreferredCamera: AVCaptureDevice?
```

A camera the user prefers to use for video and photo capture.

```
var isContinuityCamera: Bool
```

A Boolean value that indicates whether the device is a Continuity Camera.

```
var companionDeskViewCamera: AVCaptureDevice?
```

A Desk View camera associated with a device.

## Supporting system features

System video effects and microphone modes

Configure the state of system video effects like Center Stage, and inspect enhancements the system applies to microphone audio.

## Monitoring system pressure

```
var systemPressureState: AVCaptureDevice.SystemPressureState
```

A value that indicates the capture device’s current system pressure state.

```
class SystemPressureState
```

An object that provides information about OS and hardware status affecting capture system performance and availability.

```
let AVCaptureSessionInterruptionSystemPressureStateKey: String
```

A key to retrieve a state value that indicates the system pressure level and contributing factors that caused the interruption.

## Restricting camera switching

```
func setPrimaryConstituentDeviceSwitchingBehavior(AVCaptureDevice.PrimaryConstituentDeviceSwitchingBehavior, restrictedSwitchingBehaviorConditions: AVCaptureDevice.PrimaryConstituentDeviceRestrictedSwitchingBehaviorConditions)
```

Sets the switching behavior of the primary constituent device.

```
var primaryConstituentDeviceSwitchingBehavior: AVCaptureDevice.PrimaryConstituentDeviceSwitchingBehavior
```

The switching behavior for the primary constituent device.

```
var primaryConstituentDeviceRestrictedSwitchingBehaviorConditions: AVCaptureDevice.PrimaryConstituentDeviceRestrictedSwitchingBehaviorConditions
```

The conditions that restrict the primary constituent device’s switching behavior.

```
var activePrimaryConstituentDeviceSwitchingBehavior: AVCaptureDevice.PrimaryConstituentDeviceSwitchingBehavior
```

The switching behavior of the active constituent device.

```
var activePrimaryConstituentDeviceRestrictedSwitchingBehaviorConditions: AVCaptureDevice.PrimaryConstituentDeviceRestrictedSwitchingBehaviorConditions
```

The conditions that restrict camera switching behavior for the active primary constituent device.

```
var activePrimaryConstituent: AVCaptureDevice?
```

A virtual device's active primary constituent device.

```
enum PrimaryConstituentDeviceSwitchingBehavior
```

Constants that control when to allow a virtual device to switch its active primary constituent device.

```
struct PrimaryConstituentDeviceRestrictedSwitchingBehaviorConditions
```

A structure that defines the conditions in which to restrict camera switching.

```
var supportedFallbackPrimaryConstituentDevices: [AVCaptureDevice]
```

The constituent devices available to select as a fallback for a longer focal length primary constituent device.

```
var fallbackPrimaryConstituentDevices: [AVCaptureDevice]
```

The fallback devices to use when a constituent device with a longer focal length becomes limited by its light sensitivity or minimum focus distance.

## Configuring macOS features

macOS capture features

Control the transport behavior and input sources of capture hardware in macOS.

## Accessing camera extrinsics

```
class func extrinsicMatrix(from: AVCaptureDevice, to: AVCaptureDevice)
-> Data?
```

Returns the relative extrinsic matrix from one capture device to another.

## Accessing the focal length

```
var nominalFocalLengthIn35mmFilm: Float
```

The nominal 35mm equivalent focal length of the capture device's lens.

## Determining lens stabilization

```
enum LensStabilizationStatus
```

Constants that indicate the status of optical image stabilization hardware during a bracketed photo capture.

## Configuring lens smudge detection

`var isCameraLensSmudgeDetectionEnabled: Bool`

Whether camera lens smudge detection is enabled.

`func setCameraLensSmudgeDetectionEnabled(Bool, detectionInterval: CMTime)`

Specify whether to enable camera lens smudge detection, and the interval time between each run of detections.

`var cameraLensSmudgeDetectionInterval: CMTime`

The camera lens smudge detection interval.

`var cameraLensSmudgeDetectionStatus: AVCaptureCameraLensSmudgeDetectionStatus`

A value specifying the status of camera lens smudge detection.

`enum AVCaptureCameraLensSmudgeDetectionStatus`

Constants indicating the current camera lens smudge detection status.

## Synchronizing with external devices

`var isFollowingExternalSyncDevice: Bool`

Whether the device is following an external sync device.

`var minSupportedExternalSyncFrameDuration: CMTime`

The minimum frame duration that can be passed as the `videoFrameDuration` when directing your device input to follow an external sync device.

`var isVideoFrameDurationLocked: Bool`

Whether the device's video frame rate (expressed as a duration) is currently locked.

`var minSupportedLockedVideoFrameDuration: CMTime`

The maximum frame rate (expressed as a minimum duration) that can be set on an input associated with this device.

# Relationships

### Inherits From

`NSObject`

### Conforms To

`CVarArg`
`CustomDebugStringConvertible`
`CustomStringConvertible`
`Equatable`
`Hashable`
`NSObjectProtocol`

## See Also

### 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 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.