

[AVFoundation](#) / [AVCaptureConnection](#)

Class

AVCaptureConnection

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

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

```
class AVCaptureConnection
```

Overview

Capture inputs have one or more input ports (instances of [AVCaptureInput.Port](#)). Capture outputs can accept data from one or more sources (for example, an [AVCaptureMovieFileOutput](#) object accepts both video and audio data).

You can add an `AVCaptureConnection` instance to a session using the [`addConnection\(_:_\)`](#) method only if the [`canAddConnection\(_:_\)`](#) method returns `true`. When using the [`addInput\(_:_\)`](#) or [`addOutput\(_:_\)`](#) method, the session forms connections automatically between all compatible inputs and outputs. You only need to add connections manually when adding an input or output with no connections. You can also use connections to enable or disable the flow of data from a given input or to a given output.

Topics

Creating a connection

`init(inputPorts: [AVCaptureInput.Port], output: AVCaptureOutput)`

Creates a capture connection that represents a connection between multiple input ports and an output.

```
init(inputPort: AVCaptureInput.Port, videoPreviewLayer: AVCaptureVideoPreviewLayer)
```

Creates a capture connection that represents a connection between an input port and a video preview layer.

Enabling a connection

```
var isEnabled: Bool
```

Turns the connection on and off.

```
var isActive: Bool
```

Indicates whether the connection is active.

Inspecting a connection

```
var inputPorts: [AVCaptureInput.Port]
```

An array of the connection's input ports.

```
var output: AVCaptureOutput?
```

The connection's output port, if applicable.

```
var videoPreviewLayer: AVCaptureVideoPreviewLayer?
```

The video preview layer associated with the connection.

```
var audioChannels: [AVCaptureAudioChannel]
```

An array of audio channels that the connection provides.

Rotating a video

```
func isVideoRotationAngleSupported(CGFloat) -> Bool
```

Returns a Boolean value that indicates whether the connection supports a rotation angle.

```
var videoRotationAngle: CGFloat
```

A rotation angle the connection applies to a video flowing through it.

Mirroring a video

```
var isVideoMirroringSupported: Bool
```

A Boolean value that indicates whether the connection supports video mirroring.

```
var isVideoMirrored: Bool
```

A Boolean value that indicates whether the connection horizontally flips the video flowing through it.

```
var automaticallyAdjustsVideoMirroring: Bool
```

A Boolean value that indicates whether you can enable mirroring based on a session's configuration.

Stabilizing video

```
var isVideoStabilizationSupported: Bool
```

A Boolean value that indicates whether this connection supports video stabilization.

```
var activeVideoStabilizationMode: AVCaptureVideoStabilizationMode
```

The connection's current stabilization mode.

```
var preferredVideoStabilizationMode: AVCaptureVideoStabilizationMode
```

The stabilization mode that's the most appropriate for a video connection.

Delivering camera calibration settings

```
var isCameraIntrinsicMatrixDeliverySupported: Bool
```

A Boolean value that indicates whether the capture connection currently supports delivering camera intrinsics information.

```
var isCameraIntrinsicMatrixDeliveryEnabled: Bool
```

A Boolean value that indicates whether the connection can configure the capture pipeline to deliver camera intrinsics information.

Configuring a video's frame rate

```
var isVideoMinFrameDurationSupported: Bool
```

A Boolean value that indicates whether the connection supports a minimum frame duration.

Deprecated

```
var videoMinFrameDuration: CMTime
```

The smallest time interval the connection can apply between consecutive video frames.

Deprecated

```
var isVideoMaxFrameDurationSupported: Bool
```

A Boolean value that indicates whether the connection supports a maximum frame duration.

Deprecated

~~var videoMaxFrameDuration: CMTime~~

The largest time interval the connection can apply between consecutive video frames.

Deprecated

Scaling a video

~~var videoMaxScaleAndCropFactor: CGFloat~~

The connection's maximum video scale and crop factor.

~~var videoScaleAndCropFactor: CGFloat~~

The current scale and crop factor the video output uses.

Interlacing video

~~var isVideoFieldModeSupported: Bool~~

A Boolean value that indicates whether the connection supports setting a video field mode.

~~var videoFieldMode: AVVideoFieldMode~~

A setting that tells the connection how to interlace video flowing through it.

~~enum AVVideoFieldMode~~

Constants that indicate which interlacing modes the connection applies to video flowing through it.

Deprecated

~~var isVideoStabilizationEnabled: Bool~~

A Boolean value that indicates whether video stabilization is active for the connection.

Deprecated

~~var enablesVideoStabilizationWhenAvailable: Bool~~

A Boolean value that indicates whether the system enables video stabilization when it's available.

Deprecated

~~var isVideoOrientationSupported: Bool~~

A Boolean value that indicates whether the connection supports changing the orientation of the video.

Deprecated

~~var videoOrientation: AVCaptureVideoOrientation~~

An orientation that tells the connection how to rotate a video flowing through it.

Deprecated

~~enum AVCaptureVideoOrientation~~

Constants indicating video orientation.

Deprecated

Relationships

Inherits From

NSObject

Conforms To

CVarArg

CustomDebugStringConvertible

CustomStringConvertible

Equatable

Hashable

NSObjectProtocol

See Also

Capture sessions

 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.