

[AVFoundation](#) / [AVCapturePhoto](#)

Class

# AVCapturePhoto

A container for image data from a photo capture output.

iOS 11.0+ | iPadOS 11.0+ | Mac Catalyst 14.0+ | macOS 10.15+ | tvOS 17.0+

```
class AVCapturePhoto
```

## Mentioned in

- 📄 Configuring camera capture to collect a Portrait Effects matte
- 📄 Saving captured photos
- 📄 Capturing and saving Live Photos
- 📄 Capturing thumbnail and preview images
- 📄 Capturing a bracketed photo sequence

## Overview

When you capture photos with the [AVCapturePhotoOutput](#) class, your delegate object receives each resulting image and related data in the form of an [AVCapturePhoto](#) object. This object is an immutable wrapper from which you can retrieve various results of the photo capture.

In addition to the photo image pixel buffer, an [AVCapturePhoto](#) object can also contain a preview-sized pixel buffer, capture metadata, and, on supported devices, depth data and camera calibration data. From an [AVCapturePhoto](#) object, you can generate data appropriate for writing to a file, such as HEVC encoded image data containerized in the HEIC file format and including a preview image, depth data and other attachments.

An [AVCapturePhoto](#) instance wraps a single image result. For example, if you request a bracketed capture of three images, your callback is called three times, each time delivering a

single `AVCapturePhoto` object.

---

## Topics

### Resolving photo capture requests

```
var resolvedSettings: AVCaptureResolvedPhotoSettings
```

The settings object that was used to request this photo capture.

```
var photoCount: Int
```

The 1-based index of this photo capture relative to other results from the same capture request.

```
var timestamp: CMTime
```

The time at which the image was captured.

### Accessing photo pixel data

```
var isRawPhoto: Bool
```

A Boolean value indicating whether this photo object contains RAW format data.

```
var pixelBuffer: CVPixelBuffer?
```

The uncompressed or RAW image sample buffer for the photo, if requested.

### Accessing preview photo data

```
var embeddedThumbnailPhotoFormat: [String : Any]?
```

A dictionary describing the data format for a preview-sized image accompanying the captured photo.

```
var previewPixelBuffer: CVPixelBuffer?
```

The pixel data for a preview-sized version of the photo, if requested.

### Accessing photo metadata

```
var depthData: AVDepthData?
```

Depth or disparity map data captured with the photo.

```
var cameraCalibrationData: AVCameraCalibrationData?
```

Calibration information for the camera device that captured the photo.

```
var sourceDeviceType: AVCaptureDevice.DeviceType?
```

The type of device that captured the photo.

```
var metadata: [String : Any]
```

A dictionary of metadata describing the captured image.

```
var portraitEffectsMatte: AVPortraitEffectsMatte?
```

The portrait effects matte captured with the photo.

## Packaging data for file output

```
func fileDataRepresentation(with: any AVCapturePhotoFileDataRepresentationCustomizer) -> Data?
```

Gets a customized representation of the photo data.

```
protocol AVCapturePhotoFileDataRepresentationCustomizer
```

A protocol that defines the methods to implement to customize the packaging of photo data.

```
func fileDataRepresentation() -> Data?
```

Generates and returns a flat data representation of the photo and its attachments.

```
func cgImageRepresentation() -> CGImage?
```

Extracts and returns the captured photo's primary image as a Core Graphics image object.

```
func previewCGImageRepresentation() -> CGImage?
```

Extracts and returns the captured photo's preview image as a Core Graphics image object.

```
func fileDataRepresentation(withReplacementMetadata: [String : Any]?,  
replacementEmbeddedThumbnailPhotoFormat: [String : Any]?, replacement  
EmbeddedThumbnailPixelBuffer: CVPixelBuffer?, replacementDepthData:  
AVDepthData?) -> Data?
```

Generates and returns a flat data representation of the photo using the specified replacements for some or all of its attachments.

Deprecated

## Enabling constant color

```
var constantColorCenterWeightedMeanConfidenceLevel: Float
```

A score that summarizes the overall confidence level of a constant color photo.

```
var constantColorConfidenceMap: CVPixelBuffer?
```

A pixel buffer where each pixel value indicates how fully the system achieves the constant color effect in the corresponding region of the photo.

```
var isConstantColorFallbackPhoto: Bool
```

A Boolean value that indicates whether this photo is a fallback photo for a constant color capture.

## Examining bracketed capture information

```
var bracketSettings: AVCaptureBracketedStillImageSettings?
```

The variations available for bracketed capture settings for this photo.

```
var sequenceCount: Int
```

The 1-based index of this photo in a bracketed capture sequence.

```
var lensStabilizationStatus: AVCaptureDevice.LensStabilizationStatus
```

Information about the use of lens stabilization during bracketed photo capture.

```
enum LensStabilizationStatus
```

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

## Accessing segmentation mattes

```
func semanticSegmentationMatte(for: AVSemanticSegmentationMatte.MatteType) -> AVSemanticSegmentationMatte?
```

Retrieves the semantic segmentation matte associated with this photo.

---

## Relationships

### Inherits From

NSObject

### Inherited By

AVCaptureDeferredPhotoProxy

## Conforms To

CVarArg  
CustomDebugStringConvertible  
CustomStringConvertible  
Equatable  
Hashable  
NSObjectProtocol

---

## See Also

### Photo capture

- { } Capturing consistent color images

Add the power of a photography studio and lighting rig to your app with the new Constant Color API.
- :≡ Capturing still and Live Photos

Configure and capture single or multiple still images, Live Photos, and other forms of photography.
- 📄 Capturing photos in RAW and Apple ProRAW formats

Support professional photography workflows by enabling minimally processed image capture in your camera app.
- 📄 Supporting Continuity Camera in Your Mac App

Incorporate scanned documents and pictures from a user's iPhone, iPad, or iPod touch into your Mac app using Continuity Camera.

#### class AVCaptureDeferredPhotoProxy

A lightly-processed photo with data that the system may use to process and fetch a higher-resolution asset at a later time.

#### class AVCapturePhotoOutput

A capture output for still image, Live Photos, and other photography workflows.

#### protocol AVCapturePhotoCaptureDelegate

Methods for monitoring progress and receiving results from a photo capture output.

```
class AVCapturePhotoOutputReadinessCoordinator
```

An object that monitors changes to a photo output's capture readiness.

```
protocol AVCapturePhotoOutputReadinessCoordinatorDelegate
```

A delegate protocol to receive updates about a photo output's capture readiness.

```
class AVCaptureStillImageOutput
```

A capture output for capturing still photos.

Deprecated