Documentation   Language: **Swift**

AVFoundation / AVCaptureVideoDataOutput / videoSettings

Instance Property

videoSettings

A dictionary that contains the compression settings for the output.

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

```
var videoSettings: [String : Any]! { get set }
```

Discussion

To receive samples in their device-native format, set this value to an empty dictionary:

Swift   Objective-C

```
let myVideoOutput = AVCaptureVideoDataOutput()
myVideoOutput.videoSettings = [:] // Receive samples in device format.
```

To receive samples in a default uncompressed format, set this value to `nil`. Then you can query this value to receive a dictionary of the settings the session uses.

In iOS versions prior to iOS 16, the only key supported is `kCVPixelBufferPixelFormatTypeKey`. In iOS 16 and later, the supported keys include the following:

- For compressed video output, only use `AVVideoPixelAspectRatioKey`, `AVVideoCleanApertureKey`, `AVVideoScalingModeKey`, `AVVideoColorPropertiesKey`, and `AVVideoAllowWideColorKey`.
- For uncompressed video output, you can also use `kCVPixelBufferPixelFormatTypeKey`, `kCVPixelBufferWidthKey`, and `kCVPixelBufferHeightKey`, in addition to the compressed video output keys.

You can use `availableVideoPixelFormatTypes` and `availableVideoCodecTypes` to get a list of the supported pixel formats and video codecs, respectively. The width and height need to match the `videoOrientation` specified in the output's `AVCaptureConnection`, otherwise the system throws an `invalidArgumentException`. The aspect ratio of the width and height also need to match the aspect ratio of the source's `activeFormat`, corrected for the connection's `videoOrientation`, otherwise the system throws an `invalidArgumentException`. If the width or height exceeds the source's `activeFormat`'s width or height, the system throws an `invalidArgumentException`. Don't change the width and height if `deliversPreviewSizedOutputBuffers` is `true`, otherwise the system throws an `invalidArgumentException`.

See Also

Configuring video capture

Video settings

Configure video processing settings using standard key and value constants.

`var alwaysDiscardsLateVideoFrames: Bool`

Indicates whether to drop video frames if they arrive late.

`var automaticallyConfiguresOutputBufferDimensions: Bool`

A Boolean value that indicates whether the output automatically configures the size of output buffers.

`var deliversPreviewSizedOutputBuffers: Bool`

A Boolean value that indicates whether the output is configured to deliver preview-sized buffers.

`var preparesCellularRadioForNetworkConnection: Bool`

Indicates whether the receiver should prepare the cellular radio for imminent network activity.

`var preservesDynamicHDRMetadata: Bool`

Indicates whether the receiver should preserve dynamic HDR metadata as an attachment on the output sample buffer's underlying pixel buffer.

`var recommendedMediaTimeScaleForAssetWriter: CMTimeScale`

Indicates the recommended media timescale for the video track.

```
func recommendedMovieMetadata(forVideoCodecType: AVVideoCodecType,
assetWriterOutputFileType: AVFileType) -> [AVMetadataItem]?
```

Recommends movie-level metadata for a particular video codec type and output file type, to be used with an asset writer input.

```
func recommendedVideoSettings(forVideoCodecType: AVVideoCodecType,
assetWriterOutputFileType: AVFileType) -> [String : Any]?
```

Returns a video settings dictionary appropriate for capturing video to a file with the specified codec and type.

```
func recommendedVideoSettings(forVideoCodecType: AVVideoCodecType,
assetWriterOutputFileType: AVFileType, outputFileURL: URL?) -> [String
: Any]?
```

Returns a dictionary of recommended output settings for writing the specified code, file type, and output URL.

```
func recommendedVideoSettingsForAssetWriter(writingTo: AVFileType) -> [
String : Any]?
```

Specifies the recommended settings for use with an AVAssetWriterInput.