

[AVFoundation](#) / [AVOutputSettingsAssistant](#)

Class

AVOutputSettingsAssistant

An object that builds audio and video output settings dictionaries.

iOS 7.0+ | iPadOS 7.0+ | Mac Catalyst 13.1+ | macOS 10.9+ | tvOS 9.0+ | visionOS 1.0+

```
class AVOutputSettingsAssistant
```

Overview

Use an output settings assistant to create the audio and video settings that you use to configure instances of [AVAssetWriter](#) and [AVAssetWriterInput](#). You create an assistant with a specific preset configuration, such as [hevc3840x2160WithAlpha](#) or [preset1920x1080](#). You can accept the settings dictionaries as is to generate a file that conforms to the criteria that the preset implies. You may also use the dictionaries it generates as a base configuration that you can customize as you require.

Providing the assistant additional details about your source media helps it generate more complete results. For example, setting a value for its [sourceVideoFormat](#) property ensures that the assistant generates settings that don't scale up video frames from a smaller size.

Topics

Creating an assistant

```
convenience init?(preset: AVOutputSettingsPreset)
```

Creates an output setting assistant with a preset configuration.

```
struct AVOutputSettingsPreset
```

A structure that defines preset configurations for an output settings assistant.

```
class func availableOutputSettingsPresets() -> [AVOutputSettingsPreset]
```

Returns an array of preset values to use to initialize an output settings assistant.

Configuring output settings

```
var fileType: AVFileType
```

A uniform type identifier (UTI) that indicates the type of file to write.

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

An audio settings dictionary.

```
var sourceAudioFormat: CMAudioFormatDescription?
```

The format of the source audio data.

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

A video settings dictionary.

```
var sourceVideoFormat: CMVideoFormatDescription?
```

The format of the source video data.

```
var sourceVideoMinFrameDuration: CMTime
```

A time value that describes the minimum frame duration of the video data.

```
var sourceVideoAverageFrameDuration: CMTime
```

A time value that describes the average frame duration of the video data.

Relationships

Inherits From

NSObject

Conforms To

CVarArg

CustomDebugStringConvertible
CustomStringConvertible
Equatable
Hashable
NSObjectProtocol

See Also

Media writing

- { } Converting projected video to Apple Projected Media Profile
 - Convert content with equirectangular or half-equirectangular projection to APMP.
- { } Converting side-by-side 3D video to multiview HEVC and spatial video
 - Create video content for visionOS by converting an existing 3D HEVC file to a multiview HEVC format, optionally adding spatial metadata to create a spatial video.
- { } Writing fragmented MPEG-4 files for HTTP Live Streaming
 - Create an HTTP Live Streaming presentation by turning a movie file into a sequence of fragmented MPEG-4 files.
- 📄 Creating spatial photos and videos with spatial metadata
 - Add spatial metadata to stereo photos and videos to create spatial media for viewing on Apple Vision Pro.
- 📄 Tagging media with video color information
 - Inspect and set video color space information when writing and transcoding media.
- ☰ Evaluating an app's video color
 - Check color reproduction for a video in your app by using test patterns, video test equipment, and light-measurement instruments.

`class AVAssetWriter`
An object that writes media data to a container file.

`class AVAssetWriterInput`
An object that appends media samples to a track in an asset writer's output file.

`class AVAssetWriterInputPixelBufferAdaptor`

An object that appends video samples to an asset writer input.

`class AVAssetWriterInputTaggedPixelBufferGroupAdaptor`

An object that appends tagged buffer groups to an asset writer input.

`class AVAssetWriterInputMetadataAdaptor`

An object that appends timed metadata groups to an asset writer input.

`class AVAssetWriterInputGroup`

A group of inputs with tracks that are mutually exclusive to each other for playback or processing.