AVFoundation / AVAssetExportSession

Class

# AVAssetExportSession

An object that exports assets in a format that you specify using an export preset.

iOS 4.0+ | iPadOS 4.0+ | Mac Catalyst 13.1+ | macOS 10.7+ | tvOS 9.0+ | visionOS 1.0+

```
class AVAssetExportSession
```

## Mentioned in

Exporting video to alternative formats

## Overview

You configure this object to export an instance of `AVAsset` by setting an export preset, an output file type, and an output URL.

## Topics

### Creating an export session

`init?(asset: AVAsset, presetName: String)`

Creates an export session with a preset configuration.

Export presets

Configure an export session to output media in standard sizes and formats.

## Accessing export presets

`var presetName: String`

The name of the preset that the asset export session uses.

`func determineCompatibleFileTypes(completionHandler: ([AVFileType]) -> Void)`

Determines the output file types an asset export session supports writing in its current configuration.

`class func allExportPresets() -> [String]`

Returns all available export preset names.

`class func determineCompatibility(ofExportPreset: String, with: AVAsset, outputFileType: AVFileType?, completionHandler: (Bool) -> Void)`

Determines an export preset’s compatibility to export the asset in a container of the output file type.

~~`class func exportPresets(compatibleWith: AVAsset) -> [String]`~~

Returns compatible export presets for the asset.

Deprecated

## Configuring output

~~`var outputURL: URL?`~~

A URL where an asset export session writes its output.

Deprecated

~~`var outputFileType: AVFileType?`~~

The file type of the output an asset export session writes.

Deprecated

`var supportedFileTypes: [AVFileType]`

An array containing the types of files the session can write.

`var allowsParallelizedExport: Bool`

A Boolean value that indicates whether the session can parallelize its export operation.

`var shouldOptimizeForNetworkUse: Bool`

A Boolean value that indicates whether to optimize the movie for network use.

`var canPerformMultiplePassesOverSourceMediaData: Bool`

A Boolean value that indicates whether the export session can perform multiple passes over the source media to achieve better results.

`var timeRange: CMTimeRange`

The time range of the source asset to export.

`var fileLengthLimit: Int64`

The file length that the output of the session must not exceed.

`var directoryForTemporaryFiles: URL?`

A directory suitable to store temporary files that the export process generates.

## Configuring metadata

`var metadata: [AVMetadataItem]?`

The metadata an export session writes to the output container file.

`var metadataItemFilter: AVMetadataItemFilter?`

An object the export session uses to filter the metadata items it transfers to the output asset.

## Configuring video output

`var videoComposition: AVVideoComposition?`

An optional object that provides instructions for how to composite frames of video.

`var customVideoCompositor: (any AVVideoCompositing)?`

An optional custom object to use when compositing video frames.

## Configuring track groups

`var audioTrackGroupHandling: AVAssetTrackGroupOutputHandling`

A policy that defines how the session exports alternate audio tracks.

`struct AVAssetTrackGroupOutputHandling`

A type that specifies policies for how an export session processes alternate tracks in a track group.

## Configuring audio output

`var audioMix: AVAudioMix?`

The parameters for audio mixing and an indication of whether to enable nondefault audio mixing for export.

`var audioTimePitchAlgorithm: AVAudioTimePitchAlgorithm`

A processing algorithm for managing audio pitch for scaled audio edits.

## Exporting media

`func export(to: URL, as: AVFileType, isolation: isolated (any Actor)?) async throws`

Exports the asset to the output location in the specified file type.

`func cancelExport()`

Cancels the execution of an export session.

~~`func exportAsynchronously(completionHandler: () -> Void)`~~

Starts the asynchronous execution of an export session.

Deprecated

## Monitoring export progress

`func states(updateInterval: TimeInterval) -> some Sendable & AsyncSequence<AVAssetExportSession.State, Never>`

Monitors the progress state of an export operation.

`enum State`

Constants that indicate the state of an export operation.

~~`var status: AVAssetExportSession.Status`~~

The status of the export session.

Deprecated

`enum Status`

Values that indicate the state of an export session.

~~`var progress: Float`~~

A value that indicates the progress of the export.

Deprecated

~~var error: (any Error)?~~

An optional error object.

Deprecated

## Estimating file length and duration

```
func estimateOutputFileLength(completionHandler: (Int64, (any Error)?)
-> Void)
```

Starts estimating the output file length of the export while considering the asset, preset, and time range configuration of the export session.

~~var estimatedOutputFileLength: Int64~~

The estimated length of the exported file, in bytes.

Deprecated

## Estimating duration

```
func estimateMaximumDuration(completionHandler: (CMTime, (any Error)?)
-> Void)
```

Starts estimating the maximum duration of the export while considering the asset, preset, and time range configuration of the export session.

~~var maxDuration: CMTime~~

Provides an estimate of the maximum duration of the exported media.

Deprecated

## Accessing the asset

```
var asset: AVAsset
```

An asset that a session exports.

# Relationships

## Inherits From

```
NSObject
```

### Conforms To

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

## See Also

### Media export

Exporting video to alternative formats

Convert an existing movie file to a different format.