AVFoundation / AVAssetWriterInput

Class

AVAssetWriterInput

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

iOS 4.1+ | iPadOS 4.1+ | Mac Catalyst 13.1+ | macOS 10.7+ | tvOS 9.0+ | visionOS 1.0+

```
class AVAssetWriterInput
```

Mentioned in

Tagging media with video color information

Overview

Create an asset writer input to write a single track of media, and optional track-level metadata, to the output file. To write multiple concurrent tracks with ideal interleaving of media data, observe the value of the `isReadyForMoreMediaData` property of each input.

You can use an asset writer input to create tracks in a QuickTime movie file that aren't self-contained, and instead reference sample data that exists in another file.

Topics

Creating an input

```
convenience init(mediaType: AVMediaType, outputSettings: [String : Any
]?)
```

Creates an input to append sample buffers of the specified type to the output file.

`init(mediaType: AVMediaType, outputSettings: [String : Any]?, sourceFormatHint: CMFormatDescription?)`

Creates an input that appends sample buffers of the specified type and format hint to the output file.

Configuring presentation

`var naturalSize: CGSize`

The natural display dimensions of the output's visual media.

`var transform: CGAffineTransform`

The transform to use for display of the output's visual media.

`var preferredVolume: Float`

The volume to prefer for playback of the output's audio data.

`var mediaTimeScale: CMTimeScale`

The time scale of the track in the output file.

`var marksOutputTrackAsEnabled: Bool`

A Boolean value that indicates whether to enable a track in the output for playback and processing.

Configuring language support

`var languageCode: String?`

The language code of the input's track.

`var extendedLanguageTag: String?`

The extended language for the input's track.

Configuring metadata

`var metadata: [AVMetadataItem]`

The track-level metadata to write to the output.

Configuring media data layout

`var preferredMediaChunkAlignment: Int`

The boundary, in bytes, for aligning media chunks.

`var preferredMediaChunkDuration: CMTime`

The duration to use for each chunk of sample data in the output file.

`var sampleReferenceBaseURL: URL?`

The base URL sample references are relative to.

`var mediaDataLocation: AVAssetWriterInput.MediaDataLocation`

Specifies how the input lays out and interleaves media data.

`struct MediaDataLocation`

A structure that indicates how to lay out and interleave media data.

Configuring track associations

`func canAddTrackAssociation(withTrackOf: AVAssetWriterInput, type: String) -> Bool`

Determines whether it's valid to associate another input's track with this input's track.

`func addTrackAssociation(withTrackOf: AVAssetWriterInput, type: String)`

Adds an association between input tracks.

Appending media samples

`var expectsMediaDataInRealTime: Bool`

A Boolean value that indicates whether the input tailors its processing for real-time sources.

`var isReadyForMoreMediaData: Bool`

A Boolean value that indicates whether the input is ready to accept media data.

`func requestMediaDataWhenReady(on: dispatch_queue_t, using: () -> Void)`

Tells the input to request media data, at its convenience, to write to the output file.

`func append(CMSampleBuffer) -> Bool`

Appends a sample buffer to an input to write to the output file.

`func markAsFinished()`

Marks the input as finished to indicate that you're done appending samples to it.

`class SampleBufferReceiver`

Provides an interface for writing sample buffers to an input.

`class PixelBufferReceiver`

Provides an interface for writing pixel buffers to an input.

`class TaggedPixelBufferGroupReceiver`

Provides an interface for writing tagged pixel buffers to an input.

`class MetadataReceiver`

Provides an interface for writing timed metadata groups to an input.

`class CaptionReceiver`

Provides an interface for writing caption data to an input.

Performing multiple-pass encoding

`var canPerformMultiplePasses: Bool`

A Boolean value that indicates whether the input may perform multiple passes over appended media data.

`var currentPassDescription: AVAssetWriterInputPassDescription?`

An object that describes the requirements for the current pass.

`class AVAssetWriterInputPassDescription`

An object that defines the interface to query for the requirements of the current pass.

`func markCurrentPassAsFinished()`

Tells the input to analyze the appended media to determine whether it can improve the results by reencoding certain segments.

`var performsMultiPassEncodingIfSupported: Bool`

A Boolean value that indicates whether the input attempts to encode the source media data using multiple passes.

```
func respondToEachPassDescription(on: dispatch_queue_t, using: () ->
Void)
```

Tells the input to invoke a callback whenever it begins a new pass.

`class MultiPassController`

Provides an interface to receive an async sequence of pass descriptions for the writer input receiver, if multi-pass is supported.

Inspecting an input

`var mediaType: AVMediaType`

The media type of the samples that the input accepts.

`var outputSettings: [String : Any]?`

The settings to use for encoding media data you append to the output.

`var sourceFormatHint: CMFormatDescription?`

A hint about the format of the sample buffers to append to the input.

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 AVOutputSettingsAssistant`

An object that builds audio and video output settings dictionaries.

`class AVAssetWriter`

An object that writes media data to a container 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.