AVFoundation / AVSynchronizedLayer

Class

AVSynchronizedLayer

A Core Animation layer that derives its timing from a player item so that you can synchronize layer animations with media playback.

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

```
class AVSynchronizedLayer
```

Overview

You can create an arbitrary number of synchronized layers from the same `AVPlayerItem` object.

A synchronized layer is similar to a `CATransformLayer` object in that it doesn’t display anything itself, it just confers state upon its layer subtree. `AVSynchronizedLayer` confers its timing state, synchronizing the timing of layers in its subtree with that of a player item.

Any `CoreAnimation` layer with animation property set that is added as a sublayer of `AVSynchronizedLayer` should set the animation’s `beginTime` property to a non-zero positive value so animations will be interpreted on the player item’s timeline. `CoreAnimation` replaces the default `beginTime` of 0.0 with `CACurrentMediaTime()`. To start the animation from time 0, use a small positive value like `AVCoreAnimationBeginTimeAtZero`.

You might use a layer as shown in the following example:

```
AVPlayerItem *playerItem = <#Get a player item#>;
CALayer *superLayer =  <#Get a layer#>;
// Set up a synchronized layer to sync the layer timing of its subtree
// with the playback of the playerItem/
AVSynchronizedLayer *syncLayer = [AVSynchronizedLayer synchronizedLayerWithPlayerIte
[syncLayer addSublayer:<#Another layer#>];    // These sublayers will be synchronize
[superLayer addSublayer:syncLayer];
```

Topics

Creating a synchronized layer

`init(playerItem: AVPlayerItem)`

Creates a new synchronized layer with timing synchronized with a given player item.

Managing the player item

`var playerItem: AVPlayerItem?`

The player item to which the timing of the layer is synchronized.

Supporting types

`let AVCoreAnimationBeginTimeAtZero: CFTimeInterval`

A value that sets an animation begin time to `0`.

Relationships

Inherits From

`CALayer`

Conforms To

`CAMediaTiming`
`CVarArg`
`CustomDebugStringConvertible`

```
CustomStringConvertible
Equatable
Hashable
NSCoding
NSObjectProtocol
NSSecureCoding
Sendable
SendableMetatype
```

See Also

Presentation

Monitoring playback progress in your app

Observe the playback of a media asset to update your app’s user-interface state.

{} Using HEVC video with alpha

Play, write, and export HEVC video with an alpha channel to add overlay effects to your video processing.

`class AVPlayerLayer`

An object that presents the visual contents of a player object.