Cinematic / CNScript

Class

CNScript

A collection of focus decisions, focus transitions, detections, and detection tracks associated with a movie captured in Cinematic mode and methods to change them.

iOS 17.0+ | iPadOS 17.0+ | Mac Catalyst | macOS 14.0+ | tvOS 17.0+

```
final class CNScript
```

Overview

The Cinematic script provides thread-safe access to information about the focus decisions made in the original recorded Cinematic movie. The script supports changing those decisions and obtaining updated information about where to focus each frame. You can snapshot changes to a script and later reload them.

> **Tip**
>
> Look up what you need up front, outside your critical code, and pass the immutable results to where it's needed. That way, you're not blocked when you access the information inside the rendering portion of your code.

Topics

Structures

```
struct Changes
```

An object that represents a snapshot of the changes made to a movie script, including the added user decisions and detection tracks.

```
struct Frame
```

An object that represents what to focus on, and where to focus, in a given movie frame.

Initializers

```
init(asset: AVAsset, changes: CNScript.Changes?, progress: Progress?)
async throws
```

Creates a Cinematic script based on a movie and applying changes to the movie.

Instance Properties

```
var addedDetectionTracks: [CNDetectionTrack]
```

An array of the detection tracks added since recording the original Cinematic movie.

```
var fNumber: Float
```

The f-stop value which inversely affects the aperture used to render the Cinematic image.

```
var timeRange: CMTimeRange
```

The time range of the selected track.

Instance Methods

```
func addDetectionTrack(CNDetectionTrack) -> CNDetectionID
```

Adds a user-created detection track.

```
func addUserDecision(CNDecision) -> Bool
```

Adds a new user decision, and replaces an existing user decision if the times are identical.

```
func baseDecisions(in: CMTimeRange) -> [CNDecision]
```

All base decisions made automatically during recording in the given time range.

```
func changes() -> CNScript.Changes
```

Changes made since recording the Cinematic asset.

```
func changes(trimmedBy: CMTimeRange) -> CNScript.Changes
```

Changes trimmed and time range shifted to start at zero.

```
func decision(after: CMTime) -> CNDecision?
```

The decision that occurs after the given time.

```
func decision(at: CMTime, tolerance: CMTime) -> CNDecision?
```

The closest decision to the given time within the given tolerance.

```
func decision(before: CMTime) -> CNDecision?
```

The decision that occurs before the given time.

```
func decisions(in: CMTimeRange) -> [CNDecision]
```

All decisions within the given time range.

```
func detectionTrack(for: CNDecision) -> CNDetectionTrack?
```

A detection track representing all detections selected by a given decision.

```
func detectionTrack(for: CNDetectionID) -> CNDetectionTrack?
```

A detection track representing all detections with the given detection ID, over the entire Cinematic script.

```
func frame(at: CMTime, tolerance: CMTime) -> CNScript.Frame?
```

The closest frame to the given time within the given tolerance.

```
func frames(in: CMTimeRange) -> [CNScript.Frame]
```

All frames within the given time range.

```
func primaryDecision(at: CMTime) -> CNDecision?
```

The primary decision that’s in effect at the specified time, unless it’s outside the time range of the Cinematic script.

```
func reload(changes: CNScript.Changes?)
```

Reloads the Cinematic script with optional changes applied, removing any previous changes made.

```
func removeAllUserDecisions()
```

Removes all user decisions and reverts to base decisions only.

```
func removeDetectionTrack(CNDetectionTrack) -> Bool
```

Removes the user-created detection track.

```
func removeUserDecision(CNDecision) -> Bool
```

Removes an existing user decision.

```
func secondaryDecision(at: CMTime) -> CNDecision?
```

If a given time is during a focus transition, the system transitions toward a secondary decision.

```
func timeRangeOfTransition(after: CNDecision) -> CMTimeRange
```

The time range during which the focus transitioned away from the given decision.

```
func timeRangeOfTransition(before: CNDecision) -> CMTimeRange
```

The time range during which the focus transitioned toward the given decision.

```
func userDecisions(in: CMTimeRange) -> [CNDecision]
```

All user decisions in the given time range.

Relationships

Conforms To

`Sendable`, `SendableMetatype`

See Also

Essentials

{} Playing and editing Cinematic mode video

Play and edit Cinematic mode video with an adjustable depth of field and focus points.