Documentation

Language: Swift

AVFoundation / Video settings

API Collection

Video settings

Configure video processing settings using standard key and value constants.

Topics

Clean aperture

```
let AVVideoCleanApertureKey: String
```

A key that defines the region within the video dimension displayed during playback.

```
let AVVideoCleanApertureWidthKey: String
```

A key to access the width of video that’s free from transition artifacts caused by signal encoding.

```
let AVVideoCleanApertureHeightKey: String
```

A key to access the height of video that’s free from transition artifacts caused by signal encoding.

```
let AVVideoCleanApertureVerticalOffsetKey: String
```

A key to access the vertical offset of video that’s free from transition artifacts caused by signal encoding.

```
let AVVideoCleanApertureHorizontalOffsetKey: String
```

A key to access the horizontal offset of video that’s free from transition artifacts caused by signal encoding.

Video codecs

```
let AVVideoCodecKey: String
```

A key to access the name of the codec for compressing video.

```
struct AVVideoCodecType
```

A set of constants that describe the codecs the system supports for video capture.

Color properties

Keys specify video properties, and corresponding keys and values specify the color primary, transfer function, and Y′CbCr matrix.

Setting color properties for a specific resolution

Choose the proper color property keys for the desired color range.

```
let AVVideoAllowWideColorKey: String
```

The key for a dictionary that indicates whether the client can process wide color.

```
let AVVideoColorPrimariesKey: String
```

The key to identify color primaries in a color properties dictionary.

```
let AVVideoColorPrimaries_EBU_3213: String
```

The color primary is in the EBU Tech. 3213 color space.

```
let AVVideoColorPrimaries_ITU_R_2020: String
```

The color primary is in the ITU_R BT.2020 color space for ultra high definition television.

```
let AVVideoColorPrimaries_ITU_R_709_2: String
```

The color primary is in the ITU_R BT.709 color space.

```
let AVVideoColorPrimaries_P3_D65: String
```

The color primary uses the DCI-P3 D65 color space.

```
let AVVideoColorPrimaries_SMPTE_C: String
```

The color primary uses the SMPTE C color space.

```
let AVVideoColorPropertiesKey: String
```

The key for a dictionary that contains properties specifying video color.

```
let AVVideoTransferFunctionKey: String
```

The key to identify the transfer function in a color properties dictionary.

```
let AVVideoTransferFunction_IEC_sRGB: String
```

The transfer function for the IEC sRGB color space.

`let AVVideoTransferFunction_ITU_R_2100_HLG: String`

The transfer function for the ITU_R BT.2100 color space.

`let AVVideoTransferFunction_ITU_R_709_2: String`

The transfer function for the ITU_R BT.709 color space.

`let AVVideoTransferFunction_Linear: String`

The transfer function for the linear color space.

`let AVVideoTransferFunction_SMPTE_240M_1995: String`

The transfer function for the SMPTE 240M color space.

`let AVVideoTransferFunction_SMPTE_ST_2084_PQ: String`

The transfer function for the SMPTE 2084 color space.

`let AVVideoYCbCrMatrixKey: String`

The key to identify the Y′CbCr matrix in a color properties dictionary.

`let AVVideoYCbCrMatrix_ITU_R_2020: String`

The Y′CbCr color matrix for ITU-R BT.2020 conversion.

`let AVVideoYCbCrMatrix_ITU_R_601_4: String`

The Y′CbCr color matrix for ITU-R BT.601 conversion.

`let AVVideoYCbCrMatrix_ITU_R_709_2: String`

The Y′CbCr color matrix for ITU-R BT.709 conversion.

`let AVVideoYCbCrMatrix_SMPTE_240M_1995: String`

The Y′CbCr color matrix for SMPTE 240M conversion.

Compression

`let AVVideoCompressionPropertiesKey: String`

A key to access the dictionary of compression properties for a video asset.

`let AVVideoDecompressionPropertiesKey: String`

The key that indicates the video decompression properties to pass to the video decoder.

`let AVVideoAverageBitRateKey: String`

A key to access the average bit rate—as bits per second—used in compressing video.

`let AVVideoQualityKey: String`

A key to set the JPEG compression quality of the video.

```
let AVVideoMaxKeyFrameIntervalKey: String
```

A key to access the maximum interval between keyframes.

```
let AVVideoMaxKeyFrameIntervalDurationKey: String
```

A key to access the maximum interval duration between keyframes.

```
let AVVideoAllowFrameReorderingKey: String
```

A key to access permission to reorder frames.

```
let AVVideoAppleProRAWBitDepthKey: String
```

A key to access the Apple ProRAW bit depth.

Entropy mode

```
let AVVideoH264EntropyModeKey: String
```

The entropy encoding mode for H.264 compression.

```
let AVVideoH264EntropyModeCABAC: String
```

The encoder uses Context-based Adaptive Binary Arithmetic Coding.

```
let AVVideoH264EntropyModeCAVLC: String
```

The encoder uses Context-based Adaptive Variable Length Coding.

FairPlay

```
let AVStreamingKeyDeliveryContentKeyType: String
```

A URL for a content key.

```
let AVStreamingKeyDeliveryPersistentContentKeyType: String
```

A URL for a persistent content key.

Frame rate

```
let AVVideoExpectedSourceFrameRateKey: String
```

The expected source frame rate.

```
let AVVideoAverageNonDroppableFrameRateKey: String
```

The desired average number of non-droppable frames to be encoded for each second of video.

Geometry

```
let AVVideoWidthKey: String
```

A key to access the width of the video in pixels.

```
let AVVideoHeightKey: String
```

A key to access the height of the video in pixels.

```
let AVVideoPixelAspectRatioKey: String
```

A key to access the video’s pixel aspect ratio.

```
let AVVideoPixelAspectRatioVerticalSpacingKey: String
```

A key to access the pixel aspect ratio vertical spacing.

```
let AVVideoPixelAspectRatioHorizontalSpacingKey: String
```

A key to access the pixel aspect ratio horizontal spacing.

Profile level

```
let AVVideoProfileLevelKey: String
```

A key to access the video profile.

```
let AVVideoProfileLevelH264High40: String
```

A high-level 4.0 profile.

```
let AVVideoProfileLevelH264High41: String
```

A high-level 4.1 profile.

```
let AVVideoProfileLevelH264Main30: String
```

A main-level 3.0 profile.

```
let AVVideoProfileLevelH264Main31: String
```

A main-level 3.1 profile.

```
let AVVideoProfileLevelH264Main32: String
```

A main-level 3.2 profile.

```
let AVVideoProfileLevelH264Main41: String
```

A main-level 4.1 profile.

```
let AVVideoProfileLevelH264Baseline30: String
```

A baseline-level 3.0 profile.

```
let AVVideoProfileLevelH264Baseline31: String
```

A baseline-level 3.1 profile.

```
let AVVideoProfileLevelH264Baseline41: String
```

A baseline-level 4.1 profile.

```
let AVVideoProfileLevelH264HighAutoLevel: String
```

A high profile auto level profile.

```
let AVVideoProfileLevelH264MainAutoLevel: String
```

A main profile auto level profile.

```
let AVVideoProfileLevelH264BaselineAutoLevel: String
```

A baseline auto level profile.

Scaling mode

```
let AVVideoScalingModeFit: String
```

The string identifier for scaling a video to fit the surrounding view’s dimensions.

```
let AVVideoScalingModeKey: String
```

A key to retrieve the video scaling mode from a dictionary.

```
let AVVideoScalingModeResize: String
```

The string identifier for resizing a video to fit the surrounding view’s dimensions.

```
let AVVideoScalingModeResizeAspect: String
```

The string identifier for resizing a video to its surrounding view’s shorter dimension while preserving its aspect ratio.

```
let AVVideoScalingModeResizeAspectFill: String
```

The string identifier for resizing a video to fit the surrounding view’s longer dimension while preserving aspect ratio.

VideoToolbox options

```
let AVVideoEncoderSpecificationKey: String
```

The video encoder specification includes options for choosing a specific video encoder.

See Also

Common

Media assets

Load media assets from files and streams to inspect their attributes, tracks, and embedded metadata.

Media reading and writing

Read images from video, export to alternative formats, and perform sample-level reading and writing of media data.

Media types and utilities

Identify the types of content and file formats that AVFoundation supports.

Audio settings

Configure audio processing settings using standard key and value constants.