

# AV Foundation Constants Reference

# Contents

## **AV Foundation Constants Reference 3**

Overview 3

Constants 3

Media Types 3

Video Gravity 5

Media Characteristics 5

Video Settings 6

File Format UTIs 11

Core Animation 12

Video Color Properties 13

Video Scaling Mode 15

Metadata Keys 16

AVMediaSelectionOption Constants 19

## **Document Revision History 23**

# AV Foundation Constants Reference

---

Framework	AVFoundation/AVFoundation.h
Declared in	AVAnimation.h AVMediaFormat.h AVMetadataFormat.h AVVideoSettings.h

---

## Overview

This document describes constants defined in the AV Foundation framework not described in individual classes or in domain-specific constants references. See also:

- *AV Foundation Audio Settings Constants*
- *AV Foundation Error Constants*
- *AV Foundation ID3 Constants*
- *AV Foundation iTunes Metadata Constants*
- *AV Foundation QuickTime Constants*

## Constants

### Media Types

---

*Constants to identify various media types.*

```
NSString *const AVMediaTypeVideo;  
NSString *const AVMediaTypeAudio;  
NSString *const AVMediaTypeText;  
NSString *const AVMediaTypeClosedCaption;  
NSString *const AVMediaTypeSubtitle;  
NSString *const AVMediaTypeTimecode;
```

```
NSString *const AVMediaTypeTimedMetadata;  
NSString *const AVMediaTypeMuxed;
```

## Constants

### AVMediaTypeVideo

Specifies video.

Available in OS X v10.7 and later.

Declared in `AVMediaFormat.h`.

### AVMediaTypeAudio

Specifies audio.

Available in OS X v10.7 and later.

Declared in `AVMediaFormat.h`.

### AVMediaTypeText

Specifies text.

Available in OS X v10.7 and later.

Declared in `AVMediaFormat.h`.

### AVMediaTypeClosedCaption

Specifies closed-caption content.

Available in OS X v10.7 and later.

Declared in `AVMediaFormat.h`.

### AVMediaTypeSubtitle

Specifies subtitles.

Available in OS X v10.7 and later.

Declared in `AVMediaFormat.h`.

### AVMediaTypeTimecode

Specifies a time code.

Available in OS X v10.7 and later.

Declared in `AVMediaFormat.h`.

### AVMediaTypeTimedMetadata

Specifies timed metadata.

Available in OS X v10.7 through OS X v10.7.

Declared in `AVMediaFormat.h`.

### AVMediaTypeMuxed

Specifies muxed media.

Available in OS X v10.7 and later.

Declared in `AVMediaFormat.h`.

## Video Gravity

---

*These string constants define how the video is displayed within a layer's bounds rectangle.*

```
NSString * const AVLayerVideoGravityResize;  
NSString * const AVLayerVideoGravityResizeAspect;  
NSString * const AVLayerVideoGravityResizeAspectFill;
```

### Constants

#### AVLayerVideoGravityResize

Specifies that the video should be stretched to fill the layer's bounds.

Available in OS X v10.7 and later.

Declared in `AVAnimation.h`.

#### AVLayerVideoGravityResizeAspect

Specifies that the player should preserve the video's aspect ratio and fit the video within the layer's bounds.

Available in OS X v10.7 and later.

Declared in `AVAnimation.h`.

#### AVLayerVideoGravityResizeAspectFill

Specifies that the player should preserve the video's aspect ratio and fill the layer's bounds.

Available in OS X v10.7 and later.

Declared in `AVAnimation.h`.

### Discussion

You use these constants when setting the `videoGravity` property of an `AVPlayerLayer` or `AVCaptureVideoPreviewLayer` instance.

## Media Characteristics

---

*Constants to specify the characteristics of media types.*

```
NSString *const AVMediaCharacteristicVisual;  
NSString *const AVMediaCharacteristicAudible;  
NSString *const AVMediaCharacteristicLegible;  
NSString *const AVMediaCharacteristicFrameBased;
```

## Constants

### `AVMediaCharacteristicVisual`

Indicates that the media is visual.

Available in OS X v10.7 and later.

Declared in `AVMediaFormat.h`.

### `AVMediaCharacteristicAudible`

Indicates that the media is audible.

Available in OS X v10.7 and later.

Declared in `AVMediaFormat.h`.

### `AVMediaCharacteristicLegible`

Indicates that the media is legible.

Available in OS X v10.7 and later.

Declared in `AVMediaFormat.h`.

### `AVMediaCharacteristicFrameBased`

Indicates that the media is frame-based.

Available in OS X v10.7 and later.

Declared in `AVMediaFormat.h`.

## Video Settings

---

*These constants define dictionary keys for configuring video compression and compression settings for video assets.*

```
NSString *const AVVideoCodecKey;  
NSString *const AVVideoCodecH264;  
NSString *const AVVideoCodecJPEG;  
NSString *const AVVideoCodecAppleProRes4444;  
NSString *const AVVideoCodecAppleProRes422;  
NSString *const AVVideoWidthKey;  
NSString *const AVVideoHeightKey;  
NSString *const AVVideoCompressionPropertiesKey;  
NSString *const AVVideoAverageBitRateKey;  
NSString *const AVVideoQualityKey;  
NSString *const AVVideoMaxKeyFrameIntervalKey;  
NSString *const AVVideoProfileLevelKey;  
NSString *const AVVideoProfileLevelH264Baseline30;  
NSString *const AVVideoProfileLevelH264Baseline31;  
NSString *const AVVideoProfileLevelH264Baseline41;  
NSString *const AVVideoProfileLevelH264Main30;  
NSString *const AVVideoProfileLevelH264Main31;
```

```
NSString *const AVVideoProfileLevelH264Main32;  
NSString *const AVVideoProfileLevelH264Main41;  
NSString *const AVVideoPixelAspectRatioKey;  
NSString *const AVVideoPixelAspectRatioHorizontalSpacingKey;  
NSString *const AVVideoPixelAspectRatioVerticalSpacingKey;  
NSString *const AVVideoCleanApertureKey;  
NSString *const AVVideoCleanApertureWidthKey;  
NSString *const AVVideoCleanApertureHeightKey;  
NSString *const AVVideoCleanApertureHorizontalOffsetKey;  
NSString *const AVVideoCleanApertureVerticalOffsetKey;
```

## Constants

### AVVideoCodecKey

Specifies a key to access the name of the codec used to encode the video.

The corresponding value is an instance of `NSString`; equivalent to `CMVideoCodecType`.

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

### AVVideoCodecH264

Specifies that the video was encoded using H264.

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

### AVVideoCodecJPEG

Specifies that the video was encoded using the JPEG encoder.

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

### AVVideoCodecAppleProRes4444

Specifies that the video was encoded using the Apple ProRes 4444 encoder.

Files are identified with the `.ap4h` extension.

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

### AVVideoCodecAppleProRes422

Specifies that the video was encoded using the ProRes 422 Standard Definition encoder.

Files are identified with the `.apcn` extension.

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### `AVVideoWidthKey`

Specifies a key to access the width of the video in pixels.

The corresponding value is an instance of `NSNumber`.

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### `AVVideoHeightKey`

Specifies a key to access the height of the video in pixels.

The corresponding value is an instance of `NSNumber`.

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### `AVVideoCompressionPropertiesKey`

Specifies a key to access the compression properties.

The corresponding value is an instance of `NSDictionary`.

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### `AVVideoAverageBitRateKey`

Specifies a key to access the average bit rate (as bits per second) used in encoding.

The corresponding value is an instance of `NSNumber`.

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### `AVVideoQualityKey`

Specifies a key to access the JPEG coded quality.

The corresponding value is an instance of `NSNumber` 0.0-1.0.

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### `AVVideoMaxKeyFrameIntervalKey`

Specifies a key to access the maximum interval between key frames.

The corresponding value is an instance of `NSNumber`. 1 means key frames only.

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### `AVVideoProfileLevelKey`

Specifies a key to access the video profile.

Available in OS X v10.8 and later.

Declared in `AVVideoSettings.h`.



`AVVideoProfileLevelH264Baseline30`

Specifies a baseline level 3.0 profile.

Available in OS X v10.8 and later.

Declared in `AVVideoSettings.h`.

`AVVideoProfileLevelH264Baseline31`

Specifies a baseline level 3.1 profile.

Available in OS X v10.8 and later.

Declared in `AVVideoSettings.h`.

`AVVideoProfileLevelH264Baseline41`

Specifies a baseline level 4.1 profile.

Available in OS X v10.8 and later.

Declared in `AVVideoSettings.h`.

`AVVideoProfileLevelH264Main30`

Specifies a main level 3.0 profile.

Available in OS X v10.8 and later.

Declared in `AVVideoSettings.h`.

`AVVideoProfileLevelH264Main31`

Specifies a main level 3.1 profile.

Available in OS X v10.8 and later.

Declared in `AVVideoSettings.h`.

`AVVideoProfileLevelH264Main32`

Specifies a main level 3.2 profile.

Available in OS X v10.8 and later.

Declared in `AVVideoSettings.h`.

`AVVideoProfileLevelH264Main41`

Specifies a main level 4.2 profile.

Available in OS X v10.8 and later.

Declared in `AVVideoSettings.h`.

`AVVideoPixelAspectRatioKey`

Specifies a key to access the pixel aspect ratio.

The corresponding value is an instance of `NSDictionary`.

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### `AVVideoPixelFormatAspectHorizontalSpacingKey`

Specifies a key to access the pixel aspect ratio horizontal spacing.

The corresponding value is an instance of `NSNumber`.

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### `AVVideoPixelFormatAspectVerticalSpacingKey`

Specifies a key to access the pixel aspect ratio vertical spacing.

The corresponding value is an instance of `NSNumber`.

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### `AVVideoCleanApertureKey`

Specifies a key to access the clean aperture.

The corresponding value is an instance of `NSDictionary`.

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### `AVVideoCleanApertureWidthKey`

Specifies a key to access the clean aperture width.

The corresponding value is an instance of `NSNumber`.

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### `AVVideoCleanApertureHeightKey`

Specifies a key to access the clean aperture height.

The corresponding value is an instance of `NSNumber`.

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### `AVVideoCleanApertureHorizontalOffsetKey`

Specifies a key to access the clean aperture horizontal offset.

The corresponding value is an instance of `NSNumber`.

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### `AVVideoCleanApertureVerticalOffsetKey`

Specifies a key to access the clean aperture vertical offset.

The corresponding value is an instance of `NSNumber`.

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

## File Format UTIs

---

*These constants specify UTIs for various file formats.*

```
NSString *const AVFileTypeAIFC;  
NSString *const AVFileTypeAIFF;  
NSString *const AVFileTypeAMR;  
NSString *const AVFileTypeCoreAudioFormat;  
NSString *const AVFileTypeAppleM4V;  
NSString *const AVFileTypeMPEG4;  
NSString *const AVFileTypeAppleM4A;  
NSString *const AVFileTypeQuickTimeMovie;  
NSString *const AVFileTypeWAVE;
```

### Constants

#### AVFileTypeAIFC

UTI for the AIFC audio file format.

The value of this UTI is `public.aifc-audio`. Files are identified with the `.aifc` and `.cdda` extensions.

Available in OS X v10.7 and later.

Declared in `AVMediaFormat.h`.

#### AVFileTypeAIFF

UTI for the AIFF audio file format.

The value of this UTI is `public.aiff-audio`. Files are identified with the `.aif` and `.aiff` extensions.

Available in OS X v10.7 and later.

Declared in `AVMediaFormat.h`.

#### AVFileTypeCoreAudioFormat

UTI for the CoreAudio file format.

The value of this UTI is `com.apple.coreaudio-format`. Files are identified with the `.caf` extension.

Available in OS X v10.7 and later.

Declared in `AVMediaFormat.h`.

#### AVFileTypeAppleM4V

UTI for the iTunes video file format.

The value of this UTI is `com.apple.mpeg-4-video`. Files are identified with the `.m4v` extension.

Available in OS X v10.7 and later.

Declared in `AVMediaFormat.h`.

#### AVFileTypeMPEG4

UTI for the MPEG-4 file format.

The value of this UTI is `public.mpeg-4`. Files are identified with the `.mp4` extension.

Available in OS X v10.7 and later.

Declared in `AVMediaFormat.h`.

#### AVFileTypeAppleM4A

UTI for the Apple m4a audio file format.

The value of this UTI is `com.apple.m4a-audio`. Files are identified with the `.m4a` extension.

Available in OS X v10.7 and later.

Declared in `AVMediaFormat.h`.

#### AVFileTypeQuickTimeMovie

UTI for the QuickTime movie file format.

The value of this UTI is `com.apple.quicktime-movie`. Files are identified with the `.mov` and `.qt` extensions.

Available in OS X v10.7 and later.

Declared in `AVMediaFormat.h`.

#### AVFileTypeWAVE

A UTI for the WAVE audio file format.

The value of this UTI is `com.microsoft.waveform-audio`. Files are identified with the `.wav`, `.wave`, and `.bwf` extensions.

Available in OS X v10.7 and later.

Declared in `AVMediaFormat.h`.

#### AVFileTypeAMR

UTI for the adaptive multi-rate audio file format.

The value of this UTI is `org.3gpp.adaptive-multi-rate-audio`. Files are identified with the `.amr` extension.

Available in OS X v10.7 and later.

Declared in `AVMediaFormat.h`.

---

## Core Animation

*Support for integration with Core Animation.*

```
const CTimeInterval AVCoreAnimationBeginTimeAtZero
```

## Constants

### AVCoreAnimationBeginTimeAtZero

Use this constant to set the CoreAnimation's animation `beginTime` property to be time 0.

The constant is a small, non-zero, positive value which prevents CoreAnimation from replacing 0.0 with `CACurrentMediaTime`.

Available in OS X v10.7 and later.

Declared in `AVAnimation.h`.

## Video Color Properties

---

*Key to specify video properties, and corresponding keys and values for the color primary, transfer function, and YCbCr matrix.*

```
NSString *const AVVideoColorPropertiesKey;  
NSString *const AVVideoColorPrimariesKey;  
NSString *const AVVideoColorPrimaries_ITU_R_709_2;  
NSString *const AVVideoColorPrimaries_EBU_3213;  
NSString *const AVVideoColorPrimaries_SMPTE_C;  
NSString *const AVVideoTransferFunctionKey;  
NSString *const AVVideoTransferFunction_ITU_R_709_2;  
NSString *const AVVideoTransferFunction_SMPTE_240M_1995;  
NSString *const AVVideoYCbCrMatrixKey;  
NSString *const AVVideoYCbCrMatrix_ITU_R_709_2;  
NSString *const AVVideoYCbCrMatrix_ITU_R_601_4;  
NSString *const AVVideoYCbCrMatrix_SMPTE_240M_1995;
```

## Constants

### AVVideoColorPropertiesKey

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

The dictionary must contain the keys [AVVideoColorPrimariesKey](#) (page 13), [AVVideoTransferFunctionKey](#) (page 14), and [AVVideoYCbCrMatrixKey](#) (page 14).

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

### AVVideoColorPrimariesKey

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

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

### AVVideoColorPrimaries\_ITU\_R\_709\_2

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### `AVVideoColorPrimaries_EBU_3213`

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### `AVVideoColorPrimaries_SMPTE_C`

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### `AVVideoTransferFunctionKey`

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

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### `AVVideoTransferFunction_ITU_R_709_2`

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### `AVVideoTransferFunction_SMPTE_240M_1995`

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### `AVVideoYCbCrMatrixKey`

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

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### `AVVideoYCbCrMatrix_ITU_R_709_2`

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### `AVVideoYCbCrMatrix_ITU_R_601_4`

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### `AVVideoYCbCrMatrix_SMPTE_240M_1995`

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

### Discussion

If you specify the `AVVideoColorPropertiesKey`, you must specify a color primary, transfer function, and Y'CbCr matrix. Typically you specify HD, which consists of [AVVideoColorPrimaries\\_ITU\\_R\\_709\\_2](#) (page 13) [AVVideoTransferFunction\\_ITU\\_R\\_709\\_2](#) (page 14), and [AVVideoYCbCrMatrix\\_ITU\\_R\\_709\\_2](#) (page 14).

If you require SD colorimetry use [AVVideoColorPrimaries\\_SMPTE\\_C](#) (page 14), [AVVideoTransferFunction\\_ITU\\_R\\_709\\_2](#) (page 14), and [AVVideoYCbCrMatrix\\_ITU\\_R\\_601\\_4](#) (page 14).

AVFoundation will color match if the source and destination color properties differ.

It is important that the source be tagged.

## Video Scaling Mode

---

*Constants to specify how video should be scaled to fit a given area.*

```
NSString *const AVVideoScalingModeKey;  
NSString *const AVVideoScalingModeFit;  
NSString *const AVVideoScalingModeResize;  
NSString *const AVVideoScalingModeResizeAspect;  
NSString *const AVVideoScalingModeResizeAspectFill;
```

### Constants

#### AVVideoScalingModeKey

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

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### AVVideoScalingModeFit

Crop to remove edge processing region; preserve aspect ratio of cropped source by reducing specified width or height if necessary.

This mode does not scale a small source up to larger dimensions.

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### AVVideoScalingModeResize

Crop to remove edge processing region; scale remainder to destination area.

This mode does not preserve the aspect ratio.

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

#### AVVideoScalingModeResizeAspect

Preserve aspect ratio of the source, and fill remaining areas with black to fit destination dimensions.

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

## AVVideoScalingModeResizeAspectFill

Preserve aspect ratio of the source, and crop picture to fit destination dimensions.

Available in OS X v10.7 and later.

Declared in `AVVideoSettings.h`.

## Metadata Keys

---

*Common metadata and common keys for metadata.*

```
NSString *const AVMetadataKeySpaceCommon;

NSString *const AVMetadataCommonKeyTitle;
NSString *const AVMetadataCommonKeyCreator;
NSString *const AVMetadataCommonKeySubject;
NSString *const AVMetadataCommonKeyDescription;
NSString *const AVMetadataCommonKeyPublisher;
NSString *const AVMetadataCommonKeyContributor;
NSString *const AVMetadataCommonKeyCreationDate;
NSString *const AVMetadataCommonKeyLastModifiedDate;
NSString *const AVMetadataCommonKeyType;
NSString *const AVMetadataCommonKeyFormat;
NSString *const AVMetadataCommonKeyIdentifier;
NSString *const AVMetadataCommonKeySource;
NSString *const AVMetadataCommonKeyLanguage;
NSString *const AVMetadataCommonKeyRelation;
NSString *const AVMetadataCommonKeyLocation;
NSString *const AVMetadataCommonKeyCopyrights;
NSString *const AVMetadataCommonKeyAlbumName;
NSString *const AVMetadataCommonKeyAuthor;
NSString *const AVMetadataCommonKeyArtist;
NSString *const AVMetadataCommonKeyArtwork;
NSString *const AVMetadataCommonKeyMake;
NSString *const AVMetadataCommonKeyModel;
NSString *const AVMetadataCommonKeySoftware;
```

### Constants

#### AVMetadataKeySpaceCommon

Available in OS X v10.7 and later.

Declared in `AVMetadataFormat.h`.

#### AVMetadataCommonKeyTitle

Available in OS X v10.7 and later.

Declared in `AVMetadataFormat.h`.



AVMetadataCommonKeyCreator

Available in OS X v10.7 and later.

Declared in AVMetadataFormat.h.

AVMetadataCommonKeySubject

Available in OS X v10.7 and later.

Declared in AVMetadataFormat.h.

AVMetadataCommonKeyDescription

Available in OS X v10.7 and later.

Declared in AVMetadataFormat.h.

AVMetadataCommonKeyPublisher

Available in OS X v10.7 and later.

Declared in AVMetadataFormat.h.

AVMetadataCommonKeyContributor

Available in OS X v10.7 and later.

Declared in AVMetadataFormat.h.

AVMetadataCommonKeyCreationDate

Available in OS X v10.7 and later.

Declared in AVMetadataFormat.h.

AVMetadataCommonKeyLastModifiedDate

Available in OS X v10.7 and later.

Declared in AVMetadataFormat.h.

AVMetadataCommonKeyType

Available in OS X v10.7 and later.

Declared in AVMetadataFormat.h.

AVMetadataCommonKeyFormat

Available in OS X v10.7 and later.

Declared in AVMetadataFormat.h.

AVMetadataCommonKeyIdentifier

Available in OS X v10.7 and later.

Declared in AVMetadataFormat.h.

AVMetadataCommonKeySource

Available in OS X v10.7 and later.

Declared in AVMetadataFormat.h.

`AVMetadataCommonKeyLanguage`

Available in OS X v10.7 and later.

Declared in `AVMetadataFormat.h`.

`AVMetadataCommonKeyRelation`

Available in OS X v10.7 and later.

Declared in `AVMetadataFormat.h`.

`AVMetadataCommonKeyLocation`

Available in OS X v10.7 and later.

Declared in `AVMetadataFormat.h`.

`AVMetadataCommonKeyCopyrights`

Available in OS X v10.7 and later.

Declared in `AVMetadataFormat.h`.

`AVMetadataCommonKeyAlbumName`

Available in OS X v10.7 and later.

Declared in `AVMetadataFormat.h`.

`AVMetadataCommonKeyAuthor`

Available in OS X v10.7 and later.

Declared in `AVMetadataFormat.h`.

`AVMetadataCommonKeyArtist`

Available in OS X v10.7 and later.

Declared in `AVMetadataFormat.h`.

`AVMetadataCommonKeyArtwork`

Available in OS X v10.7 and later.

Declared in `AVMetadataFormat.h`.

`AVMetadataCommonKeyMake`

Available in OS X v10.7 and later.

Declared in `AVMetadataFormat.h`.

`AVMetadataCommonKeyModel`

Available in OS X v10.7 and later.

Declared in `AVMetadataFormat.h`.

`AVMetadataCommonKeySoftware`

Available in OS X v10.7 and later.

Declared in `AVMetadataFormat.h`.

## AVMediaSelectionOption Constants

---

*Media characteristic that may be present in an `AVMediaSelectionOption` object.*

```
NSString *const AVMediaCharacteristicIsMainProgramContent;  
NSString *const AVMediaCharacteristicIsAuxiliaryContent;  
NSString *const AVMediaCharacteristicContainsOnlyForcedSubtitles;  
NSString *const AVMediaCharacteristicTranscribesSpokenDialogForAccessibility;  
NSString *const AVMediaCharacteristicDescribesMusicAndSoundForAccessibility;  
NSString *const AVMediaCharacteristicDescribesVideoForAccessibility;
```

### Constants

#### `AVMediaCharacteristicIsMainProgramContent`

Indicates that the option includes content that's marked by the content author as intrinsic to the presentation of the asset.

Example: an option that presents the main program audio for the presentation, regardless of locale, would typically have this characteristic.

The value of this characteristic is `"public.main-program-content"`.

The presence of this characteristic for a media option is inferred; any option that does not have the characteristic [AVMediaCharacteristicIsAuxiliaryContent](#) (page 19) is considered to have the characteristic.

Available in OS X v10.8 and later.

Declared in `AVMediaFormat.h`.

#### `AVMediaCharacteristicIsAuxiliaryContent`

Indicates that the option includes content that's marked by the content author as auxiliary to the presentation of the asset.

Example: an option that presents audio media containing commentary on the presentation would typically have this characteristic.

The value of this characteristic is `"public.auxiliary-content"`.

For QuickTime movie and .m4v files, a media option is considered to have the characteristic `AVMediaCharacteristicIsAuxiliaryContent` if it's explicitly tagged with that characteristic or if, as a member of an alternate track group, its associated track is excluded from autoselection.

Available in OS X v10.8 and later.

Declared in `AVMediaFormat.h`.

#### `AVMediaCharacteristicContainsOnlyForcedSubtitles`

Indicates that the options presents only forced subtitles.

Media options with forced-only subtitles are typically selected when 1) the user has not selected a legible option with an accessibility characteristic or an auxiliary purpose and 2) its locale matches the locale of the selected audible media selection option.

The value of this characteristic is `"public.subtitles.forced-only"`.

The presence of this characteristic for a legible media option is inferred from the format description of the associated track that presents the subtitle media.

Available in OS X v10.8 and later.

Declared in `AVMediaFormat.h`.

#### `AVMediaCharacteristicTranscribesSpokenDialogForAccessibility`

Indicates that the option includes legible content in the language of its specified locale that transcribes spoken dialog.

It is possible for a legible media option to include both transcriptions of spoken dialog and descriptions of music and sound effects.

The value of this characteristic is `"public.accessibility.transcribes-spoken-dialog"`.

For QuickTime movie and .m4v files, a media option is considered to have the characteristic `AVMediaCharacteristicTranscribesSpokenDialogForAccessibility` only if it's explicitly tagged with that characteristic.

Available in OS X v10.8 and later.

Declared in `AVMediaFormat.h`.

#### `AVMediaCharacteristicDescribesMusicAndSoundForAccessibility`

Indicates that the option includes legible content in the language of its specified locale that describes music and sound effects occurring in program audio.

It is possible for a legible media option to include both transcriptions of spoken dialog and descriptions of music and sound effects.

The value of this characteristic is `"public.accessibility.describes-music-and-sound"`.

For QuickTime movie and .m4v files, a media option is considered to have the characteristic `AVMediaCharacteristicDescribesMusicAndSoundForAccessibility` only if it's explicitly tagged with that characteristic.

Available in OS X v10.8 and later.

Declared in `AVMediaFormat.h`.

### `AVMediaCharacteristicDescribesVideoForAccessibility`

Indicates that the option includes audible content that describes the visual portion of the presentation. It is possible for a legible media option to include both transcriptions of spoken dialog and descriptions of music and sound effects.

The value of this characteristic is `"public.accessibility.describes-video"`.

For QuickTime movie and .m4v files a media option is considered to have the characteristic `AVMediaCharacteristicDescribesVideoForAccessibility` only if it's explicitly tagged with that characteristic.

Available in OS X v10.8 and later.

Declared in `AVMediaFormat.h`.

### Discussion

Each track of .mov files and .m4v files (that is, files of type [AVFileTypeQuickTimeMovie](#) (page 12) and [AVFileTypeAppleM4V](#) (page 11)) can optionally carry one or more tagged media characteristics, each of which declares a purpose, a trait, or some other distinguishing property of the track's media.

For example, a track containing audio that mixes original program content with additional narrative descriptions of visual action may be tagged with the media characteristic `"public.accessibility.describes-video"` in order to distinguish it from other audio tracks stored in the same file that do not contain additional narrative.

Each tagged media characteristic in .mov and .m4v files is stored in track userdata as a userdata item of type 'tagc' (represented as a FourCharCode) that consists of a standard atom header (size and type) followed by an array of US-ASCII characters (8-bit, high bit clear) comprising the value of the tag. The character array is not a C string; there is no terminating zero. The userdata item atom size is sum of the standard atom header size (8) and the size of the US-ASCII character array.

You can inspect the tagged media characteristics of a track as follows:

```
NSArray *trackUserDataItems = [myAVAssetTrack
metadataForFormat:AVMetadataFormatQuickTimeUserData];

NSArray *trackTaggedMediaCharacteristics = [AVMetadataItem
metadataItemsFromArray:trackUserDataItems
    withKey:AVMetadataQuickTimeUserDataKeyTaggedCharacteristic
    keySpace:AVMetadataKeySpaceQuickTimeUserData];

for (AVMetadataItem *metadataItem in trackTaggedMediaCharacteristics) {
    NSString *thisTrackMediaCharacteristic = [metadataItem stringValue];
}
```

You can use `hasMediaCharacteristic:` to determine whether a track has a particular media characteristic, whether the characteristic is inferred from its media type or format descriptions (such as [AVMediaCharacteristicAudible](#) (page 6) or [AVMediaCharacteristicContainsOnlyForcedSubtitles](#) (page 20)) or requires explicit tagging (such as [AVMediaCharacteristicTranscribesSpokenDialogForAccessibility](#) (page 20) or [AVMediaCharacteristicDescribesVideoForAccessibility](#) (page 21)). Note that explicit tagging can't be used to override inferences from tracks' media types or format descriptions; for example:

```
[anAVAssetTrack hasMediaCharacteristic:AVMediaCharacteristicVisual]
```

will return `NO` for any audio track, even if the track has been perversely tagged with the visual characteristic.

Tagged media characteristics can be written to the QuickTime userdata of an output track associated with an `AVAssetWriterInput` object as follows, provided that the output file type of the asset writer is either [AVFileTypeQuickTimeMovie](#) (page 12) or [AVFileTypeAppleM4V](#) (page 11):

```
AVMutableMetadataItem *myTaggedMediaCharacteristic = [[AVMutableMetadataItem alloc]
    init];
[myTaggedMediaCharacteristic
    setKey:AVMetadataQuickTimeUserDataKeyTaggedCharacteristic];
[myTaggedMediaCharacteristic setKeySpace:AVMetadataKeySpaceQuickTimeUserData];
[myTaggedMediaCharacteristic setValue:aMeaningfulCharacteristicAsString];
[myMutableArrayOfMetadata addObject:myTaggedMediaCharacteristic];
[myAssetWriterInput setMetadata:myMutableArrayOfMetadata];
```

# Document Revision History

This table describes the changes to *AV Foundation Constants Reference*.

Date	Notes
2011-06-06	First release for OS X v10.7.
2010-07-14	Corrected UTI definition for AVFileType3GPP.
2010-05-15	New document that describes the constants defined in the AV Foundation framework not described in individual classes.



Apple Inc.  
Copyright © 2011 Apple Inc.  
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Apple Inc.  
1 Infinite Loop  
Cupertino, CA 95014  
408-996-1010

Apple, the Apple logo, Aperture, iTunes, Mac, OS X, and QuickTime are trademarks of Apple Inc., registered in the U.S. and other countries.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.