API Collection

VTDecompressionSession

An object that decompresses video data.

Overview

A decompression session supports the decompression of a sequence of video frames. Here’s the basic workflow:

1. Create a decompression session by calling `VTDecompressionSessionCreate(allocator:formatDescription:decoderSpecification:imageBufferAttributes:outputCallback:decompressionSessionOut:)`.
2. Optionally, configure the session with your desired Decompression Properties by calling `VTSessionSetProperty(_:key:value:)` or `VTSessionSetProperties(_:propertyDictionary:)`.
3. Decode video frames using `VTDecompressionSessionDecodeFrame(_:sampleBuffer:flags:frameRefcon:infoFlagsOut:)`.
4. When you finish with the decompression session, call `VTDecompressionSessionInvalidate(_:)` to tear it down, and call `CFRelease` to free its memory.

Topics

Creating a Session

```
func VTDecompressionSessionCreate(allocator: CFAllocator?, format
Description: CMVideoFormatDescription, decoderSpecification:
```

```
CFDictionary?, imageBufferAttributes: CFDictionary?, decompression
SessionOut: UnsafeMutablePointer<VTDecompressionSession?>) -> OSStatus
```

```
func VTDecompressionSessionCreate(allocator: CFAllocator?, format
Description: CMVideoFormatDescription, decoderSpecification:
CFDictionary?, imageBufferAttributes: CFDictionary?, outputCallback:
UnsafePointer<VTDecompressionOutputCallbackRecord>?, decompression
SessionOut: UnsafeMutablePointer<VTDecompressionSession?>) -> OSStatus
```

Creates a session for decompressing video frames.

Configuring a Session

Decompression Properties

Properties used to configure a VideoToolbox decompression session.

```
func VTVideoDecoderExtensionProperties(CMFormatDescription) throws -> [
VTExtensionPropertiesKey : Any]
```

Decoding Frames

```
func VTDecompressionSessionCanAcceptFormatDescription(VTDecompression
Session, formatDescription: CMFormatDescription) -> Bool
```

Indicates whether the session can decode frames with the given format description.

```
func VTDecompressionSessionDecodeFrame(VTDecompressionSession, sample
Buffer: CMSampleBuffer, flags: VTDecodeFrameFlags, frameRefcon: Unsafe
MutableRawPointer?, infoFlagsOut: UnsafeMutablePointer<VTDecodeInfo
Flags>?) -> OSStatus
```

Decompresses a video frame.

```
func VTDecompressionSessionDecodeFrame(VTDecompressionSession, sample
Buffer: CMSampleBuffer, flags: VTDecodeFrameFlags, infoFlagsOut: Unsafe
MutablePointer<VTDecodeInfoFlags>?, outputHandler: VTDecompression
OutputHandler) -> OSStatus
```

Decompresses a video frame and invokes the output callback when the decompression completes.

```
func VTDecompressionSessionDecodeFrame(VTDecompressionSession, sample
Buffer: CMSampleBuffer, flags: VTDecodeFrameFlags, infoFlagsOut: Unsafe
MutablePointer<VTDecodeInfoFlags>?, completionHandler: (OSStatus,
VTDecodeInfoFlags, CVImageBuffer?, [CMTaggedBuffer]?, CMTime, CMTime) -
> Void) -> OSStatus
```

Decompresses a video frame and calls the provided output closure when decompression completes.

`func VTDecompressionSessionFinishDelayedFrames(VTDecompressionSession) -> OSStatus`

Directs the decompression session to emit all delayed frames.

`func VTDecompressionSessionWaitForAsynchronousFrames(VTDecompressionSession) -> OSStatus`

Waits for any and all outstanding asynchronous and delayed frames to complete, then returns.

`func VTDecompressionSessionCopyBlackPixelBuffer(VTDecompressionSession, pixelBufferOut: UnsafeMutablePointer<CVPixelBuffer?>) -> OSStatus`

Copies a black pixel buffer from the decompression session.

Decoding Multi-Image Frames

`func VTIsStereoMVHEVCDecodeSupported() -> Bool`

Returns a Boolean value that indicates whether the system supports MV-HEVC decoding.

`typealias VTDecompressionMultiImageCapableOutputHandler`

A type alias for callback that the system invokes when it finishes decompressing a frame.

Invalidating a Session

`func VTDecompressionSessionInvalidate(VTDecompressionSession)`

Tears down a decompression session.

Accessing the Type Identifier

`func VTDecompressionSessionGetTypeID() -> CFTypeID`

Returns the Core Foundation type identifier for the decompression session.

Data Types

`class VTDecompressionSession`

A reference to a decompression session.

`struct VTDecodeFrameFlags`

Flags to pass to a decompression session and the video decoder.

`struct VTDecodeInfoFlags`

Flags that provide information about the status of a decode operation.

`typealias VTDecompressionOutputCallback`

The prototype for the callback invoked when frame decompression is complete.

`struct VTDecompressionOutputCallbackRecord`

`typealias VTDecompressionOutputHandler`

The prototype for the block invoked when frame decompression is complete.

See Also

Compression

{} Encoding video for low-latency conferencing

Configure a compression session to optimize encoding for video-conferencing apps.

{} Encoding video for live streaming

Configure a compression session to encode video for live streaming.

{} Encoding video for offline transcoding

Configure a compression session to transcode video in offline workflows.

VTCompressionSession

An object that compresses video data.

VTFrameSilo

An object that stores sample buffers from a multipass encoding session.

VTMultiPassStorage

An object that stores video encoding metadata from a multipass encoding session.