Class

vImageCVImageFormat

A mutable description of image encoding in a Core Video pixel buffer.

iOS | iPadOS | Mac Catalyst | macOS | tvOS | visionOS | watchOS

```
class vImageCVImageFormat
```

Mentioned in

Converting chroma-subsampled images

Overview

The vImage library uses the information in an image format to construct `vImageConverter` instances that convert to and from images encoded with the format. The format stores a description of the pixels in the image, such as color representation, bit depth, and number of channels.

A `vImageCVImageFormat` instance is capable of holding an incomplete encoding representation. In this case, the `vImageConverter_CreateForCGToCVImageFormat(_:_:_:_:_:)` and `vImageConverter_CreateForCVToCGImageFormat(_:_:_:_:_:)` functions return an error code that indicates what information is missing.

`kvImageCVImageFormat_ConversionMatrix`
: Use `vImageCVImageFormat_CopyConversionMatrix(_:_:_:)` to add the missing conversion matrix.

`kvImageCVImageFormat_ChromaSiting`
: Use `vImageCVImageFormat_SetChromaSiting(_:_:)` to add the missing chrominance siting information.

`kvImageCVImageFormat_ColorSpace`

Use `vImageCVImageFormat_SetColorSpace(_:_:)` to add the missing color space that contains primaries and transfer function.

Reuse a `vImageCVImageFormat` instance with other Core Video pixel buffers of the same format, such as other frames from the same movie.

Topics

Creating a Core Video image format

```
static func make(buffer: CVPixelBuffer) -> vImageCVImageFormat?
```

Creates the description of the image encoding in an existing Core Video pixel buffer.

```
static func make(format: vImageCVImageFormat.Format, matrix: vImage_ARGBToYpCbCrMatrix, chromaSiting: vImageCVImageFormat.ChromaSiting, colorSpace: CGColorSpace, alphaIsOpaqueHint: Bool) -> vImageCVImageFormat?
```

Creates the description of image encoding in a Core Video pixel buffer from the specified properties.

```
static func make(format: vImageCVImageFormat.Format, colorSpace: CGColorSpace, alphaIsOpaqueHint: Bool) -> vImageCVImageFormat?
```

Creates the description of an RGB image encoding in a Core Video pixel buffer from the specified properties.

Inspecting a Core Video image format's properties

```
var channelCount: UInt32
```

The number of channels, including alpha, for the Core Video image format.

```
var channels: [vImage.BufferType]
```

The channels of the Core Video image format.

```
func channelDescription(bufferType: vImage.BufferType) -> vImageChannelDescription?
```

Returns the range and clamp limits for a specified channel in a Core Video image format.

```
var formatCode: UInt32
```

The four-character code that encodes the pixel format of the Core Video image format.

`var chromaSiting: vImageCVImageFormat.ChromaSiting?`

The chrominance siting of the Core Video image format.

`var colorSpace: CGColorSpace?`

The color space of the Core Video image format.

`var alphaIsOpaqueHint: Bool`

The alpha hint of the Core Video image format.

Supporting types

`enum ChromaSiting`

Constants that specify the chrominance siting of a Core Video image format.

`enum Format`

Constants that specify the format of a Core Video image format.

Relationships

Conforms To

`Equatable`, `Hashable`

See Also

Creating Core Video image formats

`class vImageConstCVImageFormat`

An immutable description of image encoding in a Core Video pixel buffer.

`func vImageCVImageFormat_CreateWithCVPixelBuffer(CVPixelBuffer!) -> Unmanaged<vImageCVImageFormat>!`

Creates the description of the image encoding in an existing Core Video pixel buffer.

```
func vImageCVImageFormat_Create(UInt32, UnsafePointer<vImage_ARGBToYpCbCrMatrix>!, CFString!, CGColorSpace!, Int32) -> Unmanaged<vImageCVImageFormat>!
```

Creates the description of image encoding in a Core Video pixel buffer from the specified properties.