Class

MDLTexture

A source of texel data to be used in rendering material surface appearances.

iOS 9.0+ | iPadOS 9.0+ | Mac Catalyst 13.1+ | macOS 10.11+ | tvOS 9.0+ | visionOS 1.0+

```
class MDLTexture
```

Overview

You use the `MDLTexture` class or one of its subclasses to identify, load, or create texture data, and then associate textures with materials using the `MDLTextureSampler` and `MDLMaterialProperty` classes. When you load 3D objects from an asset file (in a format that supports texturing) with the `MDLAsset` class, Model I/O automatically creates texture objects and material objects and associates them with the `MDLSubmesh` objects in the asset.

Topics

Loading Textures from a Bundle

`convenience init?(named: String)`

Loads the texture with the specified filename from the app's main bundle.

`convenience init?(named: String, bundle: Bundle?)`

Loads the texture with the specified filename from the specified bundle.

`convenience init?(cubeWithImagesNamed: [String])`

Loads a cube texture from the specified image files in the app's main bundle.

`convenience init?(cubeWithImagesNamed: [String], bundle: Bundle?)`

Loads a cube texture from the specified image files in the specified bundle.

Creating Textures

`init(data: Data?, topLeftOrigin: Bool, name: String?, dimensions: vector_int2, rowStride: Int, channelCount: Int, channelEncoding: MDLTextureChannelEncoding, isCube: Bool)`

Initializes a texture object with the specified image data and properties.

Exporting Textures

`func write(to: URL) -> Bool`

Exports the texture data to an image file at the specified URL.

`func write(to: URL, type: CFString) -> Bool`

Exports the texture data to an image file at the specified URL, of the specified type.

`func imageFromTexture() -> Unmanaged<CGImage>?`

Exports the texture data as a CoreGraphics image.

Accessing Texture Data

`func texelDataWithTopLeftOrigin() -> Data?`

Returns the texture's image data, organized such that its first pixel represents the top-left corner of the image.

`func texelDataWithBottomLeftOrigin() -> Data?`

Returns the texture's image data, organized such that its first pixel represents the bottom-left corner of the image.

`func texelDataWithTopLeftOrigin(atMipLevel: Int, create: Bool) -> Data?`

Returns the texture's image data for the specified mipmap level, organized such that its first pixel represents the top-left corner of the image.

`func texelDataWithBottomLeftOrigin(atMipLevel: Int, create: Bool) -> Data?`

Returns the texture's image data for the specified mipmap level, organized such that its first pixel represents the bottom-left corner of the image.

Examining Texture Attributes

`var dimensions: vector_int2`

The width and height, in texels, of the texture image.

`var rowStride: Int`

The number of bytes between the first texel in a row of image data and the first texel in the next row.

`var channelCount: Int`

The number of channels per texel.

`var channelEncoding: MDLTextureChannelEncoding`

The data format for each channel value per texel.

`var isCube: Bool`

A Boolean value that indicates whether the texture is a cube textures.

`var mipLevelCount: Int`

The number of mipmap levels contained in the texture image data.

Creating Irradiance Textures

`class func irradianceTextureCube(with: MDLTexture, name: String?, dimensions: vector_int2) -> Self`

Generates an irradiance texture from the specified reflectance cube texture.

`class func irradianceTextureCube(with: MDLTexture, name: String?, dimensions: vector_int2, roughness: Float) -> Self`

Generates an irradiance texture from the specified reflectance cube texture, assuming a surface of the specified roughness.

Constants

`enum MDLTextureChannelEncoding`

Options for the data size and type of texel channel values, used by the `channelEncoding` property.

Initializers

```
init()
```

```
convenience init?(named: String, assetResolver: any MDLAssetResolver)
```

Instance Properties

```
var hasAlphaValues: Bool
```

Instance Methods

```
func imageFromTexture(atLevel: Int) -> Unmanaged<CGImage>?
```

```
func write(to: URL, level: Int) -> Bool
```

```
func write(to: URL, type: CFString, level: Int) -> Bool
```

Relationships

Inherits From

NSObject

Inherited By

MDLCheckerboardTexture
MDLColorSwatchTexture
MDLNoiseTexture
MDLNormalMapTexture
MDLSkyCubeTexture
MDLURLTexture

Conforms To

CVarArg
CustomDebugStringConvertible
CustomStringConvertible
Equatable
Hashable
MDLNamed

NSObjectProtocol

See Also

Textures

`class` MDLCheckerboardTexture

A generator of texel data that creates a checkerboard pattern with two specified colors.

`class` MDLColorSwatchTexture

A generator of texel data that creates a gradient between two specified colors.

`class` MDLNoiseTexture

A generator of texel data that creates a field of random noise.

`class` MDLNormalMapTexture

A generator of texel data that computes a normal map from a supplied texture.

`class` MDLSkyCubeTexture

A generator of texel data that creates cube textures using a physically realistic simulation of the sunlit sky.

`class` MDLURLTexture

A lightweight reference to a URL from which to load texture data.

`class` MDLTextureFilter

A description of filtering modes for a renderer to use when sampling from a texture.

`class` MDLTextureSampler

An object that pairs a source of texture data with sampling parameters to be used in rendering the texture.