Class

MTL4RenderPassDescriptor

Describes a render pass.

iOS 26.0+ | iPadOS 26.0+ | Mac Catalyst 26.0+ | macOS 26.0+ | tvOS 26.0+ | visionOS 26.0+

```
class MTL4RenderPassDescriptor
```

Overview

You use render pass descriptors to create instances of `MTL4RenderCommandEncoder` and encode draw commands into instances of `MTL4CommandBuffer`.

To create render command encoders, you typically call `renderCommandEncoderWithDescriptor:`. The `makeRenderCommandEncoder(descriptor:options:)` variant of this method allows you to specify additional options to encode a render pass in parallel from multiple CPU cores by creating *suspending* and *resuming* render passes.

Topics

Instance Properties

`var colorAttachments: MTLRenderPassColorAttachmentDescriptorArray`

Accesses the array of state information for render attachments that store color data.

`var defaultRasterSampleCount: Int`

Sets the default raster sample count for the render pass when it references no attachments.

`var depthAttachment: MTLRenderPassDepthAttachmentDescriptor!`

Accesses state information for a render attachment that stores depth data.

`var imageblockSampleLength: Int`

Assigns the per-sample size, in bytes, of the largest explicit imageblock layout in the render pass.

`var rasterizationRateMap: (any MTLRasterizationRateMap)?`

Assigns an optional variable rasterization rate map that Metal uses in the render pass.

`var renderTargetArrayLength: Int`

Assigns the number of layers that all attachments this descriptor references have.

`var renderTargetHeight: Int`

Sets the height, in pixels, to which Metal constrains the render target.

`var renderTargetWidth: Int`

Sets the width, in pixels, to which Metal constrains the render target.

`var samplePositions: [MTLSamplePosition]`

Configures the custom sample positions to use in MSAA rendering.

`var stencilAttachment: MTLRenderPassStencilAttachmentDescriptor!`

Accesses state information for a render attachment that stores stencil data.

`var supportColorAttachmentMapping: Bool`

Controls if the render pass supports color attachment mapping.

`var threadgroupMemoryLength: Int`

Assigns the per-tile size, in bytes, of the persistent threadgroup memory allocation of this render pass.

`var tileHeight: Int`

The height of the tiles, in pixels, a render pass you create with this descriptor applies to its attachments.

`var tileWidth: Int`

The width of the tiles, in pixels, a render pass you create with this descriptor applies to its attachments.

`var visibilityResultBuffer: (any MTLBuffer)?`

Configures a buffer into which Metal writes counts of fragments (pixels) passing the depth and stencil tests.

`var visibilityResultType: MTLVisibilityResultType`

Determines if Metal accumulates visibility results between render encoders or resets them.

Relationships

Inherits From

`NSObject`

Conforms To

`CVarArg`
`CustomDebugStringConvertible`
`CustomStringConvertible`
`Equatable`
`Hashable`
`NSCopying`
`NSObjectProtocol`

See Also

Configuring a render command encoder

`class MTLRenderPassDescriptor`

A group of render targets that hold the results of a render pass.

`class MTLRenderPassAttachmentDescriptor`

A render target that serves as the output destination for pixels generated by a render pass.

`class MTLRenderPassColorAttachmentDescriptorArray`

An array of render pass color attachment descriptor objects.

`class MTLRenderPassColorAttachmentDescriptor`

A color render target that serves as the output destination for color pixels generated by a render pass.

`struct MTLClearColor`

An RGBA value used for a color pixel.

`class MTLRenderPassDepthAttachmentDescriptor`

A depth render target that serves as the output destination for depth pixels generated by a render pass.

`enum MTLMultisampleDepthResolveFilter`

Filtering options for controlling an MSAA depth resolve operation.

`class MTL4RenderPipelineColorAttachmentDescriptorArray`

An array of color attachment descriptions for a render pipeline.

`class MTLTileRenderPipelineColorAttachmentDescriptorArray`

An array of color attachment descriptors for the tile render pipeline.

`class MTLRenderPassStencilAttachmentDescriptor`

A stencil render target that serves as the output destination for stencil pixels generated by a render pass.

`enum MTLMultisampleStencilResolveFilter`

Constants used to control the multisample stencil resolve operation.

`class MTLRenderPassSampleBufferAttachmentDescriptorArray`

An array of sample buffer attachments for a render pass.

`class MTLRenderPassSampleBufferAttachmentDescriptor`

A description of where to store GPU counter information at the start and end of a render pass.

`class MTLLogicalToPhysicalColorAttachmentMap`

Allows you to easily specify color attachment remapping from logical to physical indices.

`struct MTLDispatchThreadsIndirectArguments`