Class

MTL4RenderPipelineDescriptor

Groups together properties to create a render pipeline state object.

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

```
class MTL4RenderPipelineDescriptor
```

Overview

Compared to `MTLRenderPipelineDescriptor`, this interface doesn't offer a mechanism to hint to Metal mutability of vertex and fragment buffers. Additionally, using this descriptor, you don't specify binary archives.

Topics

Instance Properties

`var alphaToCoverageState: MTL4AlphaToCoverageState`

Indicates whether to read and use the alpha channel fragment output of color attachments to compute a sample coverage mask.

`var alphaToOneState: MTL4AlphaToOneState`

Indicates whether the pipeline forces alpha channel values of color attachments to the largest representable value.

`var colorAttachmentMappingState: MTL4LogicalToPhysicalColorAttachmentMappingState`

Configures a logical-to-physical rendering remap state.

`var colorAttachments: MTL4RenderPipelineColorAttachmentDescriptorArray`

Accesses an array containing descriptions of the color attachments this pipeline writes to.

`var fragmentFunctionDescriptor: MTL4FunctionDescriptor?`

Assigns the shader function that this pipeline executes for each fragment.

`var fragmentStaticLinkingDescriptor: MTL4StaticLinkingDescriptor!`

Provides static linking information for the fragment stage of the render pipeline.

`var inputPrimitiveTopology: MTLPrimitiveTopologyClass`

Assigns type of primitive topology this pipeline renders.

`var isRasterizationEnabled: Bool`

Determines whether the pipeline rasterizes primitives.

`var maxVertexAmplificationCount: Int`

Determines the maximum value that can you can pass as the pipeline's amplification count.

`var rasterSampleCount: Int`

Controls the number of samples this pipeline applies for each fragment.

`var supportFragmentBinaryLinking: Bool`

Indicates whether you can use the pipeline to create new pipelines by adding binary functions to the fragment shader function's callable functions list.

`var supportIndirectCommandBuffers: MTL4IndirectCommandBufferSupportState`

Indicates whether the pipeline supports indirect command buffers.

`var supportVertexBinaryLinking: Bool`

Indicates whether you can use the render pipeline to create new pipelines by adding binary functions to the vertex shader function's callable functions list.

`var vertexDescriptor: MTLVertexDescriptor?`

Configures an optional vertex descriptor for the vertex input.

`var vertexFunctionDescriptor: MTL4FunctionDescriptor?`

Assigns the shader function that this pipeline executes for each vertex.

`var vertexStaticLinkingDescriptor: MTL4StaticLinkingDescriptor!`

Provides static linking information for the vertex stage of the render pipeline.

Instance Methods

`func reset()`

Resets this descriptor to its default state.

Relationships

Inherits From

`MTL4PipelineDescriptor`

Conforms To

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

See Also

Render pipeline states

`protocol MTLRenderPipelineState`

An interface that represents a graphics pipeline configuration for a render pass, which the pass applies to the draw commands you encode.

`class MTLRenderPipelineDescriptor`

An argument of options you pass to a GPU device to get a render pipeline state.

`class MTLRenderPipelineFunctionsDescriptor`

A collection of functions for updating a render pipeline.

`class` `MTL4MeshRenderPipelineDescriptor`

Groups together properties you use to create a mesh render pipeline state object.

`class` `MTLMeshRenderPipelineDescriptor`

An object that configures new render pipeline state objects for mesh shading.

`class` `MTLPipelineBufferDescriptor`

The mutability options for a buffer that a render or compute pipeline uses.

`class` `MTLPipelineBufferDescriptorArray`

An array of pipeline buffer descriptors.

`class` `MTL4RenderPipelineColorAttachmentDescriptor`

`class` `MTLRenderPipelineColorAttachmentDescriptor`

A color render target that specifies the color configuration and color operations for a render pipeline.

`class` `MTLRenderPipelineColorAttachmentDescriptorArray`

An array of render pipeline color attachment descriptor objects.

`class` `MTL4TileRenderPipelineDescriptor`

Groups together properties you use to create a tile render pipeline state object.

`class` `MTLTileRenderPipelineDescriptor`

An object that configures new render pipeline state objects for tile shading.

`class` `MTLTileRenderPipelineColorAttachmentDescriptor`

A description of a tile-shading render pipeline's color render target.

`struct` `MTLPipelineOption`

Options that determine how Metal prepares the pipeline.

`class` `MTL4RenderPipelineBinaryFunctionsDescriptor`

Allows you to specify additional binary functions to link to each stage of a render pipeline.