

[Metal](#) / MTLComputePipelineState

## Protocol

# MTLComputePipelineState

An interface that represents a GPU pipeline configuration for running kernels in a compute pass.

iOS 8.0+ | iPadOS 8.0+ | Mac Catalyst 13.1+ | macOS 10.11+ | tvOS | visionOS 1.0+

```
protocol MTLComputePipelineState : MTLAllocation, Sendable
```

## Mentioned in

- 📄 Calculating threadgroup and grid sizes
- 📄 Improving CPU performance by using argument buffers

## Overview

The [MTLComputePipelineState](#) protocol is an interface that represents a specific configuration for the GPU pipeline for a compute pass. Use a pipeline state to configure a compute pass by calling the [setComputePipelineState\(\\_:\)](#) method of an [MTLComputeCommandEncoder](#) instance.

To create a pipeline state, call the appropriate [MTLDevice](#) method (see [Pipeline state creation](#)). You typically make pipeline states at a noncritical time, like when your app first launches. This is because graphics drivers may need time to evaluate and build each pipeline state. However, you can quickly use and reuse each pipeline state throughout your app's lifetime.

## Topics

## Identifying a pipeline state

`var device: any MTLDevice`

The device instance that created the pipeline state.

Required

`var gpuResourceID: MTLResourceID`

An unique identifier that represents the pipeline state, which you can add to an argument buffer.

Required

`var label: String?`

A string that helps you identify the compute pipeline state during debugging.

Required

## Checking threadgroup attributes

`var maxTotalThreadsPerThreadgroup: Int`

The maximum number of threads in a threadgroup that you can dispatch to the pipeline.

Required

`var threadExecutionWidth: Int`

The number of threads that the GPU executes simultaneously.

Required

`var staticThreadgroupMemoryLength: Int`

The length, in bytes, of statically allocated threadgroup memory.

Required

## Checking imageblock attributes

`func imageblockMemoryLength(forDimensions: MTLSize) -> Int`

Returns the length of reserved memory for an imageblock of a given size.

Required

## Checking indirect command buffer support

`var supportIndirectCommandBuffers: Bool`

A Boolean value that indicates whether the compute pipeline supports indirect command buffers.

Required

## Checking shader validation

```
var shaderValidation: MTLShaderValidation
```

The current state of shader validation for the pipeline.

Required

## Creating function handles

```
func functionHandle(function: any MTLFunction) -> (any MTLFunction  
Handle)?
```

Creates a function handle for a visible function.

Required

## Adding visible functions

```
func makeComputePipelineStateWithAdditionalBinaryFunctions(functions:  
[any MTLFunction]) throws -> any MTLComputePipelineState
```

Creates a new pipeline state object with additional callable functions.

Required

## Creating function tables

```
func makeVisibleFunctionTable(descriptor: MTLVisibleFunctionTable  
Descriptor) -> (any MTLVisibleFunctionTable)?
```

Creates a new visible function table.

Required

```
func makeIntersectionFunctionTable(descriptor: MTLIntersectionFunction  
TableDescriptor) -> (any MTLIntersectionFunctionTable)?
```

Creates a new intersection function table.

Required

## Instance Properties

```
var reflection: MTLComputePipelineReflection?
```

Provides access to this compute pipeline's reflection.

Required

```
var requiredThreadsPerThreadgroup: MTLSize
```

Required

## Instance Methods

```
func functionHandle(function: any MTL4BinaryFunction) -> (any MTLFunctionHandle)?
```

Gets the function handle for a function this pipeline links at the binary level.

Required

```
func functionHandle(withName: String) -> (any MTLFunctionHandle)?
```

Gets the function handle for a function this pipeline links at the Metal IR level by name.

Required

```
func makeComputePipelineState(additionalBinaryFunctions: [any MTL4BinaryFunction]) throws -> any MTLComputePipelineState
```

Allocates a new compute pipeline state by adding binary functions to this pipeline state.

Required

---

## Relationships

### Inherits From

MTLAllocation  
NSObjectProtocol  
Sendable  
SendableMetatype

---

## See Also

### Configuring a compute pipeline state

```
class MTL4ComputePipelineDescriptor
```

Describes a compute pipeline state.

`class MTLComputePipelineDescriptor`

An instance describing the desired GPU state for a kernel call in a compute pass.

`class MTLStageInputOutputDescriptor`

A description of the input and output data of a function.

`class MTLPipelineBufferDescriptor`

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

`class MTLPipelineBufferDescriptorArray`

An array of pipeline buffer descriptors.

`struct MTLPipelineOption`

Options that determine how Metal prepares the pipeline.