

[Metal](#) / MTL4CommandEncoder

Protocol


MTL4CommandEncoder

An encoder that writes GPU commands into a command buffer.

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

```
protocol MTL4CommandEncoder : NSObjectProtocol
```

Mentioned in

 Understanding the Metal 4 core API

Topics

Instance Properties

```
var commandBuffer: (any MTL4CommandBuffer)?
```

Returns the command buffer that is currently encoding commands.

Required

```
var label: String?
```

Provides an optional label to assign to the command encoder for debug purposes.

Required

Instance Methods

```
func barrier(afterEncoderStages: MTLStages, beforeEncoderStages: MTLStages, visibilityOptions: MTL4VisibilityOptions)
```

Encodes an intra-pass barrier.

```
func barrier(afterQueueStages: MTLStages, beforeStages: MTLStages, visibilityOptions: MTL4VisibilityOptions)
```

Encodes a consumer barrier on work you commit to the same command queue.

```
func barrier(afterStages: MTLStages, beforeQueueStages: MTLStages, visibilityOptions: MTL4VisibilityOptions)
```

Encodes a producer barrier on work committed to the same command queue.

```
func endEncoding()
```

Declares that all command generation from this encoder is complete.

Required

```
func insertDebugSignpost(String)
```

Inserts a debug string into the frame data to aid debugging.

Required

```
func popDebugGroup()
```

Pops the latest debug group string from this encoder's stack of debug groups.

Required

```
func pushDebugGroup(String)
```

Pushes a string onto this encoder's stack of debug groups.

Required

```
func updateFence(any MTLFence, afterEncoderStages: MTLStages)
```

Encodes a command to update a GPU fence.

Required

```
func waitForFence(any MTLFence, beforeEncoderStages: MTLStages)
```

Encodes a command to wait on a GPU fence.

Required

Relationships

Inherits From

NSObjectProtocol

Inherited By

MTL4ComputeCommandEncoder, MTL4MachineLearningCommandEncoder, MTL4RenderCommandEncoder

See Also

Submitting work to a GPU with Metal 4

`protocol MTL4CommandQueue`

An abstraction representing a command queue that you use commit and synchronize command buffers and to perform other GPU operations.

`class MTL4CommandQueueDescriptor`

Groups together parameters for the creation of a new command queue.

`struct MTL4CommandQueueError`

`enum Code`

Enumeration of kinds of errors that committing an array of command buffers instances can produce.

`let MTL4CommandQueueErrorDomain: String`

`protocol MTL4CommandBuffer`

Records a sequence of GPU commands.

`class MTL4CommandBufferOptions`

Options to configure a command buffer before encoding work into it.

`struct MTL4RenderEncoderOptions`

Custom render pass options you specify at encoder creation time.

`protocol MTL4ArgumentTable`

Provides a mechanism to manage and provide resource bindings for buffers, textures, sampler states and other Metal resources.

`class MTL4ArgumentTableDescriptor`

Groups parameters for the creation of a Metal argument table.

`protocol MTL4CommandAllocator`

Manages the memory backing the encoding of GPU commands into command buffers.

`class MTL4CommandAllocatorDescriptor`

Groups together parameters for creating a command allocator.

`class MTL4CommitOptions`

Represents options to configure a commit operation on a command queue.

`protocol MTL4CommitFeedback`

Describes an object containing debug information from Metal to your app after completing a workload.

`typealias MTL4CommitFeedbackHandler`

Defines the block signature for a callback Metal invokes to provide your app feedback after completing a workload.