

[Metal](#) / [MTLCommandQueue](#)

Protocol

# MTLCommandQueue

An instance you use to create, submit, and schedule command buffers to a specific GPU device to run the commands within those buffers.

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

```
protocol MTLCommandQueue : NSObjectProtocol, Sendable
```

## Mentioned in

-  [Understanding the Metal 4 core API](#)
-  [Simplifying GPU resource management with residency sets](#)

## Overview

A command queue maintains an ordered list of command buffers. You use a command queue to:

- Create command buffers, which you fill with commands for the GPU device that creates the queue
- Submit command buffers to run on that GPU

Create a command queue from an [MTLDevice](#) instance by calling its [makeCommandQueue\(\)](#) or [makeCommandQueue\(maxCommandBufferCount:\)](#) method. Typically, you create one or more command queues when your app launches and then keep them throughout your app's lifetime.

With each [MTLCommandQueue](#) instance you create, you can create [MTLCommandBuffer](#) instances for that queue by calling its [makeCommandBuffer\(\)](#) or [makeCommandBufferWithUnretainedReferences\(\)](#) method.

## Note

Each command queue is thread-safe and allows you to encode commands in multiple command buffers simultaneously.

For more information about command buffers and encoding GPU commands to them — such as rendering images and computing data in parallel — see [Setting up a command structure](#).

# Topics

## Creating command buffers

```
func makeCommandBuffer(descriptor: MTLCommandBufferDescriptor) -> (any MTLCommandBuffer)?
```

Returns a command buffer from the command queue that you configure with a descriptor.  
**Required**

```
func makeCommandBuffer() -> (any MTLCommandBuffer)?
```

Returns a command buffer from the command queue that maintains strong references to resources.  
**Required**

```
func makeCommandBufferWithUnretainedReferences() -> (any MTLCommandBuffer)?
```

Returns a command buffer from the command queue that doesn't maintain strong references to resources.  
**Required**

## Attaching residency sets

```
func addResidencySet(any MTLResidencySet)
```

Applies a residency set to a queue, which Metal applies to the queue's command buffers as you commit them.  
**Required**

```
func addResidencySets([any MTLResidencySet])
```

Applies multiple residency sets to a queue, which Metal applies to the queue's command buffers as you commit them.

## Detaching residency sets

```
func removeResidencySet(any MTLResidencySet)
```

Removes a residency set from a command queue's list, which means Metal doesn't apply it to the queue's command buffers as you commit them.

Required

```
func removeResidencySets([any MTLResidencySet])
```

Removes multiple residency sets from a command queue's list, which means Metal doesn't apply them to the queue's command buffers as you commit them.

## Identifying the command queue

```
var device: any MTLDevice
```

The GPU device that creates the command queue.

Required

```
var label: String?
```

An optional name that can help you identify the command queue.

Required

## Deprecated

```
func insertDebugCaptureBoundary()
```

Informs Xcode about when GPU Frame Capture starts and stops.

Required

Deprecated

## Relationships

### Inherits From

NSObjectProtocol, Sendable, SendableMetatype

# See Also

## Submitting work to a GPU with Metal

 Setting up a command structure

Discover how Metal executes commands on a GPU.

`class MTLCommandQueueDescriptor`

A configuration that customizes the behavior for a new command queue.

`protocol MTLCommandBuffer`

A container that stores a sequence of GPU commands that you encode into it.

`class MTLCommandBufferDescriptor`

A configuration that customizes the behavior for a new command buffer.

`struct MTLCommandBufferError`

The command buffer error codes that indicate why the GPU doesn't finish executing a command buffer.

`protocol MTLCommandEncoder`

An encoder that writes GPU commands into a command buffer.