Protocol

Actor

Common protocol to which all actors conform.

iOS 13.0+ | iPadOS 13.0+ | Mac Catalyst 13.0+ | macOS 10.15+ | tvOS 13.0+ | visionOS 1.0+ | watchOS 6.0+

```
protocol Actor : AnyObject, Sendable
```

Overview

The `Actor` protocol generalizes over all `actor` types. Actor types implicitly conform to this protocol.

Actors and SerialExecutors

By default, actors execute tasks on a shared global concurrency thread pool. This pool is shared by all default actors and tasks, unless an actor or task specified a more specific executor requirement.

It is possible to configure an actor to use a specific `SerialExecutor`, as well as impact the scheduling of default tasks and actors by using a `TaskExecutor`.

See Also

`SerialExecutor`

See Also

`TaskExecutor`

Topics

Instance Properties

`var unownedExecutor: UnownedSerialExecutor`

Retrieve the executor for this actor as an optimized, unowned reference.

Required

Instance Methods

`func assertIsolated(@autoclosure () -> String, file: StaticString, line: UInt)`

Stops program execution if the current task is not executing on this actor’s serial executor.

`func assumeIsolated<T>((isolated Self) throws -> T, file: StaticString, line: UInt) rethrows -> T`

Assume that the current task is executing on this actor’s serial executor, or stop program execution otherwise.

`func preconditionIsolated(@autoclosure () -> String, file: StaticString, line: UInt)`

Stops program execution if the current task is not executing on this actor’s serial executor.

Relationships

Inherits From

`Sendable`, `SendableMetatype`

Conforming Types

`MainActor`

See Also

Actors

`protocol Sendable`

A thread-safe type whose values can be shared across arbitrary concurrent contexts without introducing a risk of data races.

~~`typealias AnyActor`~~

Common marker protocol providing a shared “base” for both (local) `Actor` and (potentially remote) `DistributedActor` types.

Deprecated

`actor MainActor`

A singleton actor whose executor is equivalent to the main dispatch queue.

`protocol GlobalActor`

A type that represents a globally-unique actor that can be used to isolate various declarations anywhere in the program.

`protocol SendableMetatype`

A type whose metatype can be shared across arbitrary concurrent contexts without introducing a risk of data races. When a generic type T conforms to `SendableMetatype`, its metatype `T.Type` conforms to `Sendable`. All concrete types implicitly conform to the `SendableMetatype` protocol, so its primary purpose is in generic code to prohibit the use of isolated conformances along with the generic type.

~~`typealias ConcurrentValue`~~ Deprecated

~~`protocol UnsafeSendable`~~

A type whose values can safely be passed across concurrency domains by copying, but which disables some safety checking at the conformance site.

Deprecated

~~`typealias UnsafeConcurrentValue`~~ Deprecated

`macro isolation<T>() -> T`

Produce a reference to the actor to which the enclosing code is isolated, or `nil` if the code is nonisolated.

```
~~func extractIsolation<each Arg, Result>((repeat each Arg) async throws
-> Result) -> (any Actor)?~~
```

Deprecated