Protocol

Sendable

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

iOS 8.0+ | iPadOS 8.0+ | Mac Catalyst 13.0+ | macOS 10.10+ | tvOS 9.0+ | visionOS 1.0+ | watchOS 2.0+

```
protocol Sendable : SendableMetatype
```

Overview

Values of the type may have no shared mutable state, or they may protect that state with a lock or by forcing it to only be accessed from a specific actor.

You can safely pass values of a sendable type from one concurrency domain to another — for example, you can pass a sendable value as the argument when calling an actor's methods. All of the following can be marked as sendable:

- Value types
- Reference types with no mutable storage
- Reference types that internally manage access to their state
- Functions and closures (by marking them with `@Sendable`)

Although this protocol doesn't have any required methods or properties, it does have semantic requirements that are enforced at compile time. These requirements are listed in the sections below. Conformance to `Sendable` must be declared in the same file as the type's declaration.

To declare conformance to `Sendable` without any compiler enforcement, write `@unchecked Sendable`. You are responsible for the correctness of unchecked sendable types, for example, by protecting all access to its state with a lock or a queue. Unchecked conformance to `Sendable` also disables enforcement of the rule that conformance must be in the same file.

For information about the language-level concurrency model that `Task` is part of, see Concurrency in The Swift Programming Language.

Sendable Structures and Enumerations

To satisfy the requirements of the `Sendable` protocol, an enumeration or structure must have only sendable members and associated values. In some cases, structures and enumerations that satisfy the requirements implicitly conform to `Sendable`:

- Frozen structures and enumerations
- Structures and enumerations that aren't public and aren't marked `@usableFromInline`.

Otherwise, you need to declare conformance to `Sendable` explicitly.

Structures that have nonsendable stored properties and enumerations that have nonsendable associated values can be marked as `@unchecked Sendable`, disabling compile-time correctness checks, after you manually verify that they satisfy the `Sendable` protocol's semantic requirements.

Sendable Actors

All actor types implicitly conform to `Sendable` because actors ensure that all access to their mutable state is performed sequentially.

Sendable Classes

To satisfy the requirements of the `Sendable` protocol, a class must:

- Be marked `final`
- Contain only stored properties that are immutable and sendable
- Have no superclass or have `NSObject` as the superclass

Classes marked with `@MainActor` are implicitly sendable, because the main actor coordinates all access to its state. These classes can have stored properties that are mutable and nonsendable.

Classes that don't meet the requirements above can be marked as `@unchecked Sendable`, disabling compile-time correctness checks, after you manually verify that they satisfy the `Sendable` protocol's semantic requirements.

Sendable Functions and Closures

Instead of conforming to the `Sendable` protocol, you mark sendable functions and closures with the `@Sendable` attribute. Any values that the function or closure captures must be sendable. In addition, sendable closures must use only by-value captures, and the captured values must be of a sendable type.

In a context that expects a sendable closure, a closure that satisfies the requirements implicitly conforms to `Sendable` — for example, in a call to `Task.detached(priority:operation:)`.

You can explicitly mark a closure as sendable by writing `@Sendable` as part of a type annotation, or by writing `@Sendable` before the closure's parameters — for example:

```
let sendableClosure = { @Sendable (number: Int) -> String in
    if number > 12 {
        return "More than a dozen."
    } else {
        return "Less than a dozen"
    }
}
```

Sendable Tuples

To satisfy the requirements of the `Sendable` protocol, all of the elements of the tuple must be sendable. Tuples that satisfy the requirements implicitly conform to `Sendable`.

Sendable Metatypes

Metatypes such as `Int.Type` implicitly conform to the `Sendable` protocol.

Relationships

Inherits From

`SendableMetatype`

Inherited By

`Actor`
`Clock`
`CodingKey`
`DistributedActor`

DistributedActorSystem
DistributedActorSystemError
DurationProtocol
Error
Executor
InstantProtocol
SerialExecutor
TaskExecutor
UnsafeSendable

Conforming Types

Array

Conforms when `Element` conforms to `Copyable`, `Escapable`, and `Sendable`.

ArraySlice

Conforms when `Element` conforms to `Copyable`, `Escapable`, and `Sendable`.

AsyncCompactMapSequence

Conforms when `Base` conforms to `Sendable`, `Base` conforms to `AsyncSequence`, `ElementOfResult` conforms to `Copyable`, `ElementOfResult` conforms to `Escapable`, `ElementOfResult` conforms to `Sendable`, and `Base.Element` conforms to `Sendable`.

AsyncCompactMapSequence.Iterator

Conforms when `Base` conforms to `AsyncSequence`, `ElementOfResult` conforms to `Copyable`, `ElementOfResult` conforms to `Escapable`, `ElementOfResult` conforms to `Sendable`, `Base.AsyncIterator` conforms to `Sendable`, and `Base.Element` conforms to `Sendable`.

AsyncDropFirstSequence

Conforms when `Base` conforms to `Sendable`, `Base` conforms to `AsyncSequence`, and `Base.Element` conforms to `Sendable`.

AsyncDropFirstSequence.Iterator

Conforms when `Base` conforms to `AsyncSequence`, `Base.AsyncIterator` conforms to `Sendable`, and `Base.Element` conforms to `Sendable`.

AsyncDropWhileSequence

Conforms when `Base` conforms to `Sendable`, `Base` conforms to `AsyncSequence`, and `Base.Element` conforms to `Sendable`.

AsyncDropWhileSequence.Iterator

Conforms when `Base` conforms to `AsyncSequence`, `Base.AsyncIterator` conforms to `Sendable`, and `Base.Element` conforms to `Sendable`.

AsyncFilterSequence

Conforms when `Base` conforms to `Sendable`, `Base` conforms to `AsyncSequence`, and `Base.Element` conforms to `Sendable`.

AsyncFilterSequence.Iterator

Conforms when `Base` conforms to `AsyncSequence`, `Base.AsyncIterator` conforms to `Sendable`, and `Base.Element` conforms to `Sendable`.

AsyncFlatMapSequence

Conforms when `Base` conforms to `Sendable`, `Base` conforms to `AsyncSequence`, `SegmentOfResult` conforms to `Sendable`, `SegmentOfResult` conforms to `AsyncSequence`, `Base.Element` conforms to `Sendable`, and `SegmentOfResult.Element` conforms to `Sendable`.

AsyncFlatMapSequence.Iterator

Conforms when `Base` conforms to `AsyncSequence`, `SegmentOfResult` conforms to `Sendable`, `SegmentOfResult` conforms to `AsyncSequence`, `Base.AsyncIterator` conforms to `Sendable`, `Base.Element` conforms to `Sendable`, `SegmentOfResult.AsyncIterator` conforms to `Sendable`, and `SegmentOfResult.Element` conforms to `Sendable`.

AsyncMapSequence

Conforms when `Base` conforms to `Sendable`, `Base` conforms to `AsyncSequence`, `Transformed` conforms to `Copyable`, `Transformed` conforms to `Escapable`, `Transformed` conforms to `Sendable`, and `Base.Element` conforms to `Sendable`.

AsyncMapSequence.Iterator

Conforms when `Base` conforms to `AsyncSequence`, `Transformed` conforms to `Copyable`, `Transformed` conforms to `Escapable`, `Transformed` conforms to `Sendable`, `Base.AsyncIterator` conforms to `Sendable`, and `Base.Element` conforms to `Sendable`.

AsyncPrefixSequence

Conforms when `Base` conforms to `Sendable`, `Base` conforms to `AsyncSequence`, and `Base.Element` conforms to `Sendable`.

AsyncPrefixSequence.Iterator

Conforms when `Base` conforms to `AsyncSequence`, `Base.AsyncIterator` conforms to `Sendable`, and `Base.Element` conforms to `Sendable`.

AsyncPrefixWhileSequence

Conforms when `Base` conforms to `Sendable`, `Base` conforms to `AsyncSequence`, and `Base.Element` conforms to `Sendable`.

AsyncPrefixWhileSequence.Iterator

Conforms when `Base` conforms to `AsyncSequence`, `Base.AsyncIterator` conforms to `Sendable`, and `Base.Element` conforms to `Sendable`.

AsyncStream

Conforms when `Element` conforms to `Copyable`, `Escapable`, and `Sendable`.

AsyncStream.Continuation

AsyncStream.Continuation.BufferingPolicy

AsyncStream.Continuation.Termination

AsyncStream.Continuation.YieldResult

Conforms when `Element` conforms to `Copyable`, `Escapable`, and `Sendable`.

AsyncThrowingCompactMapSequence

Conforms when `Base` conforms to `Sendable`, `Base` conforms to `AsyncSequence`, `ElementOfResult` conforms to `Copyable`, `ElementOfResult` conforms to `Escapable`, and `Base.Element` conforms to `Sendable`.

AsyncThrowingCompactMapSequence.Iterator

Conforms when `Base` conforms to `AsyncSequence`, `ElementOfResult` conforms to `Copyable`, `ElementOfResult` conforms to `Escapable`, `Base.AsyncIterator` conforms to `Sendable`, and `Base.Element` conforms to `Sendable`.

AsyncThrowingDropWhileSequence

Conforms when `Base` conforms to `Sendable`, `Base` conforms to `AsyncSequence`, and `Base.Element` conforms to `Sendable`.

AsyncThrowingDropWhileSequence.Iterator

Conforms when `Base` conforms to `AsyncSequence`, `Base.AsyncIterator` conforms to `Sendable`, and `Base.Element` conforms to `Sendable`.

AsyncThrowingFilterSequence

Conforms when `Base` conforms to `Sendable`, `Base` conforms to `AsyncSequence`, and `Base.Element` conforms to `Sendable`.

AsyncThrowingFilterSequence.Iterator

Conforms when `Base` conforms to `AsyncSequence`, `Base.AsyncIterator` conforms to `Sendable`, and `Base.Element` conforms to `Sendable`.

AsyncThrowingFlatMapSequence

Conforms when `Base` conforms to `Sendable`, `Base` conforms to `AsyncSequence`, `SegmentOfResult` conforms to `Sendable`, `SegmentOfResult` conforms to `AsyncSequence`, `Base.Element` conforms to `Sendable`, and `SegmentOfResult.Element` conforms to `Sendable`.

AsyncThrowingFlatMapSequence.Iterator

Conforms when `Base` conforms to `AsyncSequence`, `SegmentOfResult` conforms to `Sendable`, `SegmentOfResult` conforms to `AsyncSequence`, `Base.AsyncIterator` conforms to `Sendable`, `Base.Element` conforms to `Sendable`, `SegmentOfResult.AsyncIterator` conforms to `Sendable`, and `SegmentOfResult.Element` conforms to `Sendable`.

AsyncThrowingMapSequence

Conforms when `Base` conforms to `Sendable`, `Base` conforms to `AsyncSequence`, `Transformed` conforms to `Copyable`, `Transformed` conforms to `Escapable`, `Transformed` conforms to `Sendable`, and `Base.Element` conforms to `Sendable`.

AsyncThrowingMapSequence.Iterator

Conforms when `Base` conforms to `AsyncSequence`, `Transformed` conforms to `Copyable`, `Transformed` conforms to `Escapable`, `Transformed` conforms to `Sendable`, `Base.AsyncIterator` conforms to `Sendable`, and `Base.Element` conforms to `Sendable`.

AsyncThrowingPrefixWhileSequence

Conforms when `Base` conforms to `Sendable`, `Base` conforms to `AsyncSequence`, and `Base.Element` conforms to `Sendable`.

AsyncThrowingPrefixWhileSequence.Iterator

Conforms when `Base` conforms to `AsyncSequence`, `Base.AsyncIterator` conforms to `Sendable`, and `Base.Element` conforms to `Sendable`.

AsyncThrowingStream

Conforms when `Element` conforms to `Copyable`, `Element` conforms to `Escapable`, `Element` conforms to `Sendable`, and `Failure` conforms to `Error`.

AsyncThrowingStream.Continuation

AsyncThrowingStream.Continuation.BufferingPolicy
AsyncThrowingStream.Continuation.Termination
AsyncThrowingStream.Continuation.YieldResult

Conforms when `Element` conforms to `Copyable`, `Element` conforms to `Escapable`, `Element` conforms to `Sendable`, and `Failure` conforms to `Error`.

Atomic

Conforms when `Value` conforms to `Sendable` and `AtomicRepresentable`.

AtomicLazyReference

Conforms when `Instance` conforms to `Copyable`, `Escapable`, and `Sendable`.

AtomicLoadOrdering
AtomicStoreOrdering
AtomicUpdateOrdering
Bool
CancellationError
Character
CheckedContinuation
ClosedRange

Conforms when `Bound` conforms to `Comparable` and `Sendable`.

ClosedRange.Index

Conforms when `Bound` conforms to `Sendable`, `Bound` conforms to `Strideable`, and `Bound.Stride` conforms to `SignedInteger`.

CodingUserInfoKey
CollectionDifference

Conforms when `ChangeElement` conforms to `Copyable`, `Escapable`, and `Sendable`.

CollectionDifference.Change

Conforms when `ChangeElement` conforms to `Copyable`, `Escapable`, and `Sendable`.

CollectionDifference.Index

Conforms when `ChangeElement` conforms to `Copyable`, `Escapable`, and `Sendable`.

CollectionOfOne

Conforms when `Element` conforms to `Copyable`, `Escapable`, and `Sendable`.

CollectionOfOne.Iterator

Conforms when `Element` conforms to `Copyable`, `Escapable`, and `Sendable`.

CommandLine
ContiguousArray

Conforms when `Element` conforms to `Copyable`, `Escapable`, and `Sendable`.

ContinuousClock
ContinuousClock.Instant
DecodingError
DecodingError.Context

DefaultIndices

Conforms when Elements conforms to Collection, Elements conforms to Sendable, and Elements.Index conforms to Sendable.

DefaultStringInterpolation

Dictionary

Conforms when Key conforms to Hashable, Key conforms to Sendable, Value conforms to Copyable, Value conforms to Escapable, and Value conforms to Sendable.

Dictionary.Index

Conforms when Key conforms to Hashable, Key conforms to Sendable, Value conforms to Copyable, Value conforms to Escapable, and Value conforms to Sendable.

Dictionary.Iterator

Conforms when Key conforms to Hashable, Key conforms to Sendable, Value conforms to Copyable, Value conforms to Escapable, and Value conforms to Sendable.

Dictionary.Keys

Conforms when Key conforms to Hashable, Key conforms to Sendable, Value conforms to Copyable, Value conforms to Escapable, and Value conforms to Sendable.

Dictionary.Keys.Iterator

Conforms when Key conforms to Hashable, Key conforms to Sendable, Value conforms to Copyable, Value conforms to Escapable, and Value conforms to Sendable.

Dictionary.Values

Conforms when Key conforms to Hashable, Key conforms to Sendable, Value conforms to Copyable, Value conforms to Escapable, and Value conforms to Sendable.

Dictionary.Values.Iterator

Conforms when Key conforms to Hashable, Key conforms to Sendable, Value conforms to Copyable, Value conforms to Escapable, and Value conforms to Sendable.

DiscontiguousSlice

Conforms when Base conforms to Collection, Base conforms to Sendable, and Base.Index conforms to Sendable.

DiscontiguousSlice.Index

Conforms when Base conforms to Collection and Base.Index conforms to Sendable.

DistributedActorCodingError

Double

Double.SIMD16Storage

Double.SIMD2Storage

Double.SIMD32Storage

Double.SIMD4Storage

Double.SIMD64Storage

Double.SIMD8Storage

DropFirstSequence

Conforms when Base conforms to Sendable and Sequence.

DropWhileSequence

Conforms when `Base` conforms to `Sequence`, `Base.Element` conforms to `Sendable`, and `Base.Iterator` conforms to `Sendable`.

DropWhileSequence.Iterator

Conforms when `Base` conforms to `Sequence`, `Base.Element` conforms to `Sendable`, and `Base.Iterator` conforms to `Sendable`.

Duration

Duration.TimeFormatStyle

Duration.TimeFormatStyle.Attributed

Duration.TimeFormatStyle.Pattern

Duration.UnitsFormatStyle

Duration.UnitsFormatStyle.Attributed

Duration.UnitsFormatStyle.FractionalPartDisplayStrategy

Duration.UnitsFormatStyle.Unit

Duration.UnitsFormatStyle.UnitWidth

Duration.UnitsFormatStyle.ZeroValueUnitsDisplayStrategy

EmptyCollection

Conforms when `Element` conforms to `Copyable` and `Escapable`.

EmptyCollection.Iterator

Conforms when `Element` conforms to `Copyable` and `Escapable`.

EncodingError

EncodingError.Context

EnumeratedSequence

Conforms when `Base` conforms to `Sendable` and `Sequence`.

EnumeratedSequence.Iterator

Conforms when `Base` conforms to `Sequence` and `Base.Iterator` conforms to `Sendable`.

ExecuteDistributedTargetError

ExecutorJob

ExecutorJob.Kind

FlattenSequence

Conforms when `Base` conforms to `Sendable`, `Base` conforms to `Sequence`, and `Base.Element` conforms to `Sequence`.

FlattenSequence.Index

Conforms when `Base` conforms to `Collection`, `Base.Element` conforms to `Collection`, `Base.Index` conforms to `Sendable`, and `Base.Element.Index` conforms to `Sendable`.

FlattenSequence.Iterator

Conforms when `Base` conforms to `Sequence`, `Base.Element` conforms to `Sequence`, `Base.Iterator` conforms to `Sendable`, and `Base.Element.Iterator` conforms to `Sendable`.

Float

Float.SIMD16Storage

Float.SIMD2Storage

Float.SIMD32Storage

Float.SIMD4Storage
Float.SIMD64Storage
Float.SIMD8Storage
Float16
Float16.SIMD16Storage
Float16.SIMD2Storage
Float16.SIMD32Storage
Float16.SIMD4Storage
Float16.SIMD64Storage
Float16.SIMD8Storage
Float80
FloatingPointClassification
FloatingPointRoundingRule
FloatingPointSign
Hasher
IndexingIterator

Conforms when `Elements` conforms to `Collection`, `Elements` conforms to `Sendable`, and `Elements.Index` conforms to `Sendable`.

InlineArray

Conforms when `Element` conforms to `Escapable` and `Sendable`.

Int
Int.SIMD16Storage
Int.SIMD2Storage
Int.SIMD32Storage
Int.SIMD4Storage
Int.SIMD64Storage
Int.SIMD8Storage
Int.Words
Int128
Int16
Int16.SIMD16Storage
Int16.SIMD2Storage
Int16.SIMD32Storage
Int16.SIMD4Storage
Int16.SIMD64Storage
Int16.SIMD8Storage
Int16.Words
Int32
Int32.SIMD16Storage
Int32.SIMD2Storage
Int32.SIMD32Storage
Int32.SIMD4Storage

Int32.SIMD64Storage
Int32.SIMD8Storage
Int32.Words
Int64
Int64.SIMD16Storage
Int64.SIMD2Storage
Int64.SIMD32Storage
Int64.SIMD4Storage
Int64.SIMD64Storage
Int64.SIMD8Storage
Int64.Words
Int8
Int8.SIMD16Storage
Int8.SIMD2Storage
Int8.SIMD32Storage
Int8.SIMD4Storage
Int8.SIMD64Storage
Int8.SIMD8Storage
Int8.Words
IteratorSequence

Conforms when `Base` conforms to `IteratorProtocol` and `Sendable`.

Job
JobPriority
JoinedSequence

Conforms when `Base` conforms to `Sendable`, `Base` conforms to `Sequence`, `Base.Element` conforms to `Sequence`, and `Base.Element.Element` conforms to `Sendable`.

JoinedSequence.Iterator

Conforms when `Base` conforms to `Sequence`, `Base.Element` conforms to `Sequence`, `Base.Iterator` conforms to `Sendable`, `Base.Element.Element` conforms to `Sendable`, and `Base.Element.Iterator` conforms to `Sendable`.

KeyValuePairs

Conforms when `Key` conforms to `Copyable`, `Key` conforms to `Escapable`, `Key` conforms to `Sendable`, `Value` conforms to `Copyable`, `Value` conforms to `Escapable`, and `Value` conforms to `Sendable`.

LazyPrefixWhileSequence.Index

Conforms when `Base` conforms to `Collection` and `Base.Index` conforms to `Sendable`.

LazySequence

Conforms when `Base` conforms to `Sendable` and `Sequence`.

LocalTestingActorID
LocalTestingDistributedActorSystem
LocalTestingDistributedActorSystemError
MainActor
MemoryLayout

Conforms when `T` conforms to `Copyable` and `Escapable`.

`Mirror.DisplayStyle`

`MutableRawSpan`

`MutableSpan`

Conforms when `Element` conforms to `Copyable`, `Escapable`, and `Sendable`.

`Mutex`

Conforms when `Value` conforms to `Escapable`.

`Never`

`ObjectIdentifier`

`ObservationRegistrar`

`Observations`

`Observations.Iteration`

`Optional`

Conforms when `Wrapped` conforms to `Sendable`.

`OutputRawSpan`

`OutputSpan`

Conforms when `Element` conforms to `Escapable` and `Sendable`.

`PartialRangeFrom`

Conforms when `Bound` conforms to `Comparable` and `Sendable`.

`PartialRangeFrom.Iterator`

Conforms when `Bound` conforms to `Sendable`, `Bound` conforms to `Strideable`, and `Bound.Stride` conforms to `SignedInteger`.

`PartialRangeThrough`

Conforms when `Bound` conforms to `Comparable` and `Sendable`.

`PartialRangeUpTo`

Conforms when `Bound` conforms to `Comparable` and `Sendable`.

`PrefixSequence`

Conforms when `Base` conforms to `Sendable` and `Sequence`.

`PrefixSequence.Iterator`

Conforms when `Base` conforms to `Sequence` and `Base.Iterator` conforms to `Sendable`.

`Range`

Conforms when `Bound` conforms to `Comparable` and `Sendable`.

`RangeSet`

Conforms when `Bound` conforms to `Comparable` and `Sendable`.

`RangeSet.Ranges`

Conforms when `Bound` conforms to `Comparable` and `Sendable`.

`RawSpan`

`Repeated`

Conforms when `Element` conforms to `Copyable`, `Escapable`, and `Sendable`.

Result

Conforms when `Success` conforms to `Sendable` and `Failure` conforms to `Error`.

ReversedCollection

Conforms when `Base` conforms to `BidirectionalCollection` and `Sendable`.

ReversedCollection.Index

Conforms when `Base` conforms to `BidirectionalCollection` and `Base.Index` conforms to `Sendable`.

ReversedCollection.Iterator

Conforms when `Base` conforms to `BidirectionalCollection`, `Base` conforms to `Sendable`, and `Base.Index` conforms to `Sendable`.

SIMD16

Conforms when `Scalar` conforms to `SIMDScalar`, `Scalar` conforms to `Sendable`, and `Scalar.SIMD16Storage` conforms to `Sendable`.

SIMD2

Conforms when `Scalar` conforms to `SIMDScalar`, `Scalar` conforms to `Sendable`, and `Scalar.SIMD2Storage` conforms to `Sendable`.

SIMD3

Conforms when `Scalar` conforms to `SIMDScalar`, `Scalar` conforms to `Sendable`, and `Scalar.SIMD4Storage` conforms to `Sendable`.

SIMD32

Conforms when `Scalar` conforms to `SIMDScalar`, `Scalar` conforms to `Sendable`, and `Scalar.SIMD32Storage` conforms to `Sendable`.

SIMD4

Conforms when `Scalar` conforms to `SIMDScalar`, `Scalar` conforms to `Sendable`, and `Scalar.SIMD4Storage` conforms to `Sendable`.

SIMD64

Conforms when `Scalar` conforms to `SIMDScalar`, `Scalar` conforms to `Sendable`, and `Scalar.SIMD64Storage` conforms to `Sendable`.

SIMD8

Conforms when `Scalar` conforms to `SIMDScalar`, `Scalar` conforms to `Sendable`, and `Scalar.SIMD8Storage` conforms to `Sendable`.

SIMDMask

Conforms when `Storage` conforms to `SIMD`, `Storage` conforms to `Sendable`, `Storage.Scalar` conforms to `FixedWidthInteger`, and `Storage.Scalar` conforms to `SignedInteger`.

Set

Conforms when `Element` conforms to `Hashable` and `Sendable`.

Set.Index

Conforms when `Element` conforms to `Hashable` and `Sendable`.

Set.Iterator

Conforms when `Element` conforms to `Hashable` and `Sendable`.

Slice

Conforms when `Base` conforms to `Collection`, `Base` conforms to `Sendable`, and `Base.Index` conforms to `Sendable`.

Span

Conforms when `Element` conforms to `Escapable` and `Sendable`.

StaticBigInt

StaticString

StrideThrough

Conforms when `Element` conforms to `Sendable`, `Element` conforms to `Strideable`, and `Element.Stride` conforms to `Sendable`.

StrideThroughIterator

Conforms when `Element` conforms to `Sendable`, `Element` conforms to `Strideable`, and `Element.Stride` conforms to `Sendable`.

StrideTo

Conforms when `Element` conforms to `Sendable`, `Element` conforms to `Strideable`, and `Element.Stride` conforms to `Sendable`.

StrideToIterator

Conforms when `Element` conforms to `Sendable`, `Element` conforms to `Strideable`, and `Element.Stride` conforms to `Sendable`.

String

String.Comparator

String.Encoding

String.Index

String.Iterator

String.LocalizationValue

String.LocalizationValue.Placeholder

String.StandardComparator

String.UTF16View

String.UTF16View.Iterator

String.UTF8View

String.UnicodeScalarView

String.UnicodeScalarView.Iterator

Substring

Substring.UTF16View

Substring.UTF8View

Substring.UnicodeScalarView

SuspendingClock

SuspendingClock.Instant

SystemRandomNumberGenerator

Task

TaskLocal

TaskPriority

UInt
UInt.SIMD16Storage
UInt.SIMD2Storage
UInt.SIMD32Storage
UInt.SIMD4Storage
UInt.SIMD64Storage
UInt.SIMD8Storage
UInt.Words
UInt128
UInt128.Words
UInt16
UInt16.SIMD16Storage
UInt16.SIMD2Storage
UInt16.SIMD32Storage
UInt16.SIMD4Storage
UInt16.SIMD64Storage
UInt16.SIMD8Storage
UInt16.Words
UInt32
UInt32.SIMD16Storage
UInt32.SIMD2Storage
UInt32.SIMD32Storage
UInt32.SIMD4Storage
UInt32.SIMD64Storage
UInt32.SIMD8Storage
UInt32.Words
UInt64
UInt64.SIMD16Storage
UInt64.SIMD2Storage
UInt64.SIMD32Storage
UInt64.SIMD4Storage
UInt64.SIMD64Storage
UInt64.SIMD8Storage
UInt64.Words
UInt8
UInt8.SIMD16Storage
UInt8.SIMD2Storage
UInt8.SIMD32Storage
UInt8.SIMD4Storage
UInt8.SIMD64Storage
UInt8.SIMD8Storage
UInt8.Words
UnboundedRange_

UnfoldSequence

Conforms when `Element` conforms to `Copyable`, `Element` conforms to `Escapable`, `Element` conforms to `Sendable`, `State` conforms to `Copyable`, `State` conforms to `Escapable`, and `State` conforms to `Sendable`.

Unicode

Unicode.ASCII

Unicode.ASCII.Parser

Unicode.CanonicalCombiningClass

Unicode.GeneralCategory

Unicode.NumericType

Unicode.ParseResult

Conforms when `T` conforms to `Copyable`, `Escapable`, and `Sendable`.

Unicode.Scalar

Unicode.Scalar.Properties

Unicode.Scalar.UTF16View

Unicode.Scalar.UTF8View

Unicode.UTF16

Unicode.UTF16.ForwardParser

Unicode.UTF16.ReverseParser

Unicode.UTF32

Unicode.UTF32.Parser

Unicode.UTF8

Unicode.UTF8.ForwardParser

Unicode.UTF8.ReverseParser

Unicode.UTF8.ValidationError

Unicode.UTF8.ValidationError.Kind

UnicodeDecodingResult

Unmanaged

Conforms when `Instance` conforms to `Copyable`, `Escapable`, and `Sendable`.

UnownedJob

UnownedSerialExecutor

UnownedTaskExecutor

UnsafeContinuation

WordPair

Zip2Sequence

Conforms when `Sequence1` conforms to `Sendable`, `Sequence1` conforms to `Sequence`, `Sequence2` conforms to `Sendable`, and `Sequence2` conforms to `Sequence`.

Zip2Sequence.Iterator

Conforms when `Sequence1` conforms to `Sequence`, `Sequence2` conforms to `Sequence`, `Sequence1.Iterator` conforms to `Sendable`, and `Sequence2.Iterator` conforms to `Sendable`.

See Also

Actors

`protocol Actor`

Common protocol to to which all actors conform.

~~`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