Documentation Language: Swift

Wi-Fi Aware / WAError

Enumeration

WAError

An error in Wi-Fi Aware.

iOS 26.0+ | iPadOS 26.0+

```
enum WAError
```

Overview

Wi-Fi Aware may throw a `WAError` to indicate when an underlying issue occurs, with the Network framework throwing an `NWError`. If the `NWError` occurs due to Wi-Fi Aware, then the `Network/NWError/wifiAware` property on `Network/NWError` can provide access to additional information about the underlying Wi-Fi Aware error.

Topics

Checking for general errors

`case error(WAError.ErrorDetails)`

A general error.

`struct ErrorDetails`

The optional details describing the error.

Checking for missing entitlement

`struct EntitlementMissingDetails`

The optional details describing the missing entitlement.

`case entitlementMissing(WAError.EntitlementMissingDetails)`

An error that occurs if your app is missing the entitlement needed for the requested operation.

Checking for unsupported host hardware

`case wifiAwareUnsupported(WAError.WiFiAwareUnsupportedDetails)`

An error that occurs if Wi-Fi Aware isn’t supported on the device.

`struct WiFiAwareUnsupportedDetails`

The optional details describing the unavailability of Wi-Fi Aware on the device.

Checking for insufficient radio resources

`case noRadioResources(WAError.NoRadioResourcesDetails)`

An error that occurs if the radio lacks resources.

`struct NoRadioResourcesDetails`

The optional details describing what resources are lacking.

Checking for undeclared services

`case serviceNotDeclared(WAError.ServiceNotDeclaredDetails)`

An error that occurs if your app didn’t declare the necessary services.

`struct ServiceNotDeclaredDetails`

The optional details that describe the app service wasn’t declared.

Checking for service already in use

`case serviceAlreadySubscribing(WAError.ServiceAlreadySubscribingDetails)`

An error that occurs if a new subscriber or `Network/NetworkBrowser` can’t be created.

`struct ServiceAlreadySubscribingDetails`

The optional details describing the service that’s subscribing.

```
case serviceAlreadyPublishing(WAError.ServiceAlreadyPublishingDetails)
```

An error that occurs if a new publisher or `Network/NetworkListener` can’t be created.

```
struct ServiceAlreadyPublishingDetails
```

The optional details describing the service that’s publishing.

Checking if paired devices are present or specified

```
case noPairedDevices(WAError.NoPairedDevicesDetails)
```

An error that occurs if your app doesn’t have access to any paired devices.

```
struct NoPairedDevicesDetails
```

The optional details describing the lack of paired devices.

Checking for invalid device

```
case deviceInvalid(WAError.DeviceInvalidDetails)
```

An error that occurs if your app specifies a paired device it doesn’t have access to.

```
struct DeviceInvalidDetails
```

The optional details describing the device that’s invalid.

Checking for unavailable device

```
case deviceNoLongerAvailable(WAError.DeviceNoLongerAvailableDetails)
```

An error that occurs if a device is no longer available.

```
struct DeviceNoLongerAvailableDetails
```

The optional details describing the unavailable device.

Checking for connection

```
case connectionFailed(WAError.ConnectionFailedDetails)
```

An error that occurs if the service is unable to connect.

```
struct ConnectionFailedDetails
```

The optional details describing the failed connection.

Checking for timeouts

`case connectionIdleTimeout(WAError.ConnectionIdleTimeoutDetails)`

An error that occurs due to an idle or unused connection.

`case publisherTimeout(WAError.PublisherTimeoutDetails)`

An error that occurs due to publisher timeout.

`case subscriberTimeout(WAError.SubscriberTimeoutDetails)`

An error that occurs due to subscriber timeout.

`struct ConnectionIdleTimeoutDetails`

The optional details describing the missing resources.

`struct PublisherTimeoutDetails`

The optional details describing the timed out publisher.

`struct SubscriberTimeoutDetails`

The optional details describing the timed out subscriber.

Checking for terminated connection

`case connectionTerminated(WAError.ConnectionTerminatedDetails)`

An error that occurs if the connection was terminated.

`struct ConnectionTerminatedDetails`

The optional details describing the terminated connection.

Relationships

Conforms To

```
Decodable
Encodable
Error
LocalizedError
Sendable
SendableMetatype
```

See Also

Errors

`enum NWError`

The errors returned by objects in the Network framework.