

[Foundation](#) / [NotificationCenter](#)

Class

NotificationCenter

A notification dispatch mechanism that enables the broadcast of information to registered observers.

iOS 2.0+ | iPadOS 2.0+ | Mac Catalyst 13.0+ | macOS 10.0+ | tvOS 9.0+ | visionOS 1.0+ | watchOS 2.0+

```
class NotificationCenter
```

Overview

Callers register with a notification center to receive one or both of the following:

- [NSNotification](#) objects, when working in Objective-C or with frameworks that only support [NSNotification](#). Objects register with a notification center to receive notifications ([NSNotification](#) objects) using the [addObserver\(_ :selector:name:object:\)](#) or [addObserver\(forName:object:queue:using:\)](#) methods, specifying a notification name and optionally a source object. When a caller adds itself as an observer, it specifies which notifications it should receive.
- [NotificationCenter.MainActorMessage](#) and [NotificationCenter.AsyncMessage](#) instances for use with Swift code, providing strong typing, appropriate actor isolation, and a more idiomatic Swift experience. Callers register with the notification center using the various flavors of the [addObserver\(of:for:using:\)](#) method, specifying either a message type or a convenience [NotificationCenter.MessageIdentifier](#) to identify the notification messages to receive. See [Notification center messages](#) for more information about this API.

Callers may add observers for many different notifications, or even the same notification name or message type as produced by different source objects.

Each running app has a [default](#) notification center, and you can create new notification centers to organize communications in particular contexts.

A notification center can deliver notifications only within a single program. On macOS, if you want to post a notification to other processes or receive notifications from other processes, use [DistributedNotificationCenter](#) instead.

Topics

Getting the default notification center

```
class var `default`: NotificationCenter
```

The app's default notification center.

Adding and removing notification observers

```
func addObserver(forName: NSNotification.Name?, object: Any?, queue: OperationQueue?, using: (Notification) -> Void) -> any NSObjectProtocol
```

Adds an entry to the notification center to receive notifications that passed to the provided block.

```
func addObserver(Any, selector: Selector, name: NSNotification.Name?, object: Any?)
```

Adds an entry to the notification center to call the provided selector with the notification.

```
func removeObserver(Any, name: NSNotification.Name?, object: Any?)
```

Removes matching entries from the notification center's dispatch table.

```
func removeObserver(Any)
```

Removes all entries specifying an observer from the notification center's dispatch table.

Posting notifications

```
func post(Notification)
```

Posts a given notification to the notification center.

```
func post(name: NSNotification.Name, object: Any?, userInfo: [Any Hashable : Any]?)
```

Creates a notification with a given name, sender, and information and posts it to the notification center.

```
func post(name: NSNotification.Name, object: Any?)
```

Creates a notification with a given name and sender and posts it to the notification center.

Receiving notifications as an asynchronous sequence

```
func notifications(named: Notification.Name, object: (any AnyObject & Sendable)?) -> NotificationCenter.Notifications
```

Returns an asynchronous sequence of notifications produced by this center for a given notification name and optional source object.

```
class Notifications
```

An asynchronous sequence of notifications generated by a notification center.

Receiving notifications as a Combine publisher

```
func publisher(for: Notification.Name, object: AnyObject?) -> NotificationCenter.Publisher
```

Returns a publisher that emits events when broadcasting notifications.

```
struct Publisher
```

A publisher that emits elements when broadcasting notifications.

Using notification center with Swift actor isolation

≡ Notification center messages

Use Foundation's notification center with Swift concurrency.

Relationships

Inherits From

NSObject

Inherited By

DistributedNotificationCenter

Conforms To

CVarArg
CustomDebugStringConvertible
CustomStringConvertible
Equatable
Hashable
NSObjectProtocol
Sendable
SendableMetatype

See Also

Notifications

`struct Notification`

A container for information broadcast through a notification center to all registered observers.

`class NotificationQueue`

A notification center buffer.