

[Core Location](#) / CLLocation

Class

CLLocation

The latitude, longitude, and course information reported by the system.

iOS 2.0+ iPadOS 2.0+ Mac Catalyst 13.1+ macOS 10.6+ tvOS 9.0+ visionOS 1.0+ watchOS 2.0+

```
class CLLocation
```

Mentioned in

-  [Getting heading and course information](#)
-  [Converting between coordinates and user-friendly place names](#)

Overview

A [CLLocation](#) object contains the geographical location and altitude of a device, along with values indicating the accuracy of those measurements and when they were collected. In iOS, a location object also contains course information — that is, the speed and heading in which the device was moving.

Typically, you don't create location objects yourself. After you request location updates from your [CLLocationManager](#) object, the system uses onboard sensors to gather location data and report that data to your app. Some services also return previously collected location data, which you can use as context to improve your services. You can always retrieve the most recently collected location from the [location](#) property of your [CLLocationManager](#) object.

You may create location objects yourself when you want to cache custom location data or calculate the distance between two geographical coordinates.

Use CLLocation objects as-is, and don't subclass them.

Topics

Creating a location object

```
init(latitude: CLLocationDegrees, longitude: CLLocationDegrees)
```

Creates a location object with the specified latitude and longitude.

```
init(coordinate: CLLocationCoordinate2D, altitude: CLLocationDistance,  
horizontalAccuracy: CLLocationAccuracy, verticalAccuracy: CLLocation  
Accuracy, timestamp: Date)
```

Creates a location object with the specified coordinate and altitude information.

```
init(coordinate: CLLocationCoordinate2D, altitude: CLLocationDistance,  
horizontalAccuracy: CLLocationAccuracy, verticalAccuracy: CLLocation  
Accuracy, course: CLLocationDirection, speed: CLLocationSpeed,  
timestamp: Date)
```

Creates a location object with the specified coordinate, altitude, and course information.

```
init(coordinate: CLLocationCoordinate2D, altitude: CLLocationDistance,  
horizontalAccuracy: CLLocationAccuracy, verticalAccuracy: CLLocation  
Accuracy, course: CLLocationDirection, courseAccuracy: CLLocation  
DirectionAccuracy, speed: CLLocationSpeed, speedAccuracy: CLLocation  
SpeedAccuracy, timestamp: Date)
```

Creates a location object with the specified coordinate, altitude, course, and accuracy information.

```
init(coordinate: CLLocationCoordinate2D, altitude: CLLocationDistance,  
horizontalAccuracy: CLLocationAccuracy, verticalAccuracy: CLLocation  
Accuracy, course: CLLocationDirection, courseAccuracy: CLLocation  
DirectionAccuracy, speed: CLLocationSpeed, speedAccuracy: CLLocation  
SpeedAccuracy, timestamp: Date, sourceInfo: CLLocationSource  
Information)
```

Getting the location attributes

`var coordinate: CLLocationCoordinate2D`

The geographical coordinate information.

`var altitude: CLLocationDistance`

The altitude above mean sea level associated with a location, measured in meters.

`var ellipsoidalAltitude: CLLocationDistance`

The altitude as a height above the World Geodetic System 1984 (WGS84) ellipsoid, measured in meters.

`typealias CLLocationDistance`

A distance in meters from an existing location.

`var floor: CLFloor?`

The logical floor of the building in which the user is located.

`var timestamp: Date`

The time at which this location was determined.

`var sourceInformation: CLLocationSourceInformation?`

Information about the source that provides the location.

Getting the location accuracy

`var horizontalAccuracy: CLLocationAccuracy`

The radius of uncertainty for the location, measured in meters.

`var verticalAccuracy: CLLocationAccuracy`

The validity of the altitude values, and their estimated uncertainty, measured in meters.

`typealias CLLocationAccuracy`

The accuracy of a geographical coordinate.

Measuring the distance between coordinates

```
func distance(from: CLLocation) -> CLLocationDistance
```

Returns the distance (measured in meters) from the current object's location to the specified location.

```
func getDistanceFrom(CLLocation) -> CLLocationDistance
```

Returns the distance (measured in meters) from the current object's location to the specified location.

Deprecated

Getting speed and course information

```
var speed: CLLocationSpeed
```

The instantaneous speed of the device, measured in meters per second.

```
var speedAccuracy: CLLocationSpeedAccuracy
```

The accuracy of the speed value, measured in meters per second.

```
var course: CLLocationDirection
```

The direction in which the device is traveling, measured in degrees and relative to due north.

```
var courseAccuracy: CLLocationDirectionAccuracy
```

The accuracy of the course value, measured in degrees.

```
 typealias CLLocationSpeed
```

The velocity (measured in meters per second) at which the device is moving.

```
 typealias CLLocationDirection
```

An azimuth that is measured in degrees relative to true north.

```
 typealias CLLocationSpeedAccuracy
```

The accuracy of a speed.

```
 typealias CLLocationDirectionAccuracy
```

The accuracy of a compass heading.

Relationships

Inherits From

NSObject

Conforms To

CKRecordValue
CKRecordValueProtocol
CVarArg
Copyable
CustomDebugStringConvertible
CustomStringConvertible
Equatable
Hashable
NSCoding
NSCopying
NSObjectProtocol
NSSecureCoding
Sendable
SendableMetatype

See Also

Location updates

-  **Getting the current location of a device**
Start location services and provide information the system needs to optimize power usage for those services.
-  **Handling location updates in the background**
Configure your app to receive location updates when it isn't running in the foreground.

 Creating a location push service extension

Add and configure an extension to enable your location-sharing app to access a user's location in response to a request from another user.

`struct CLLocationCoordinate2D`

The latitude and longitude associated with a location, specified using the WGS 84 reference frame.

`class CLFloor`

The floor of a building on which the user's device is located.

`class CLVisit`

Information about the user's location during a specific period of time.

`class CLLocationSourceInformation`

Information about the source that provides a location.

`{}` Monitoring location changes with Core Location

Define boundaries and act on user location updates.

`class CLServiceSession`