

[MapKit](#) / MKMapItem

Class

MKMapItem

A point of interest on the map.

iOS 6.0+ | iPadOS 6.0+ | Mac Catalyst 13.1+ | macOS 10.9+ | tvOS 9.2+ | visionOS 1.0+ | watchOS 2.0+

```
class MKMapItem
```

Mentioned in

📄 Identifying unique locations with Place IDs

Overview

A map item includes a geographic location and any interesting data that might apply to that location, such as the address at that location and the name of a business at that address. You can also create a special `MKMapItem` object representing the user's location.

Use this class to do the following:

- Share map-related data with the Maps app.
- Handle requests for directions that originate from the Maps app.

To display information in the Maps app, create an `MKMapItem` object with the information you want to display and call the `openMaps(with:launchOptions:)` method. The Maps app displays that location on the map and shows the information you provide.

If you implement a routing app, the Maps app provides two `MKMapItem` objects representing the start and end points. Use the information in those two objects to plot the route and generate directions.

Topics

Creating map items

~~`init(placemark: MKPlacemark)`~~

Creates and returns a map item object using the specified placemark object.

Deprecated

`class func forCurrentLocation() -> MKMapItem`

Creates and returns a singleton map item object representing the user's location.

Accessing the map item attributes

`class Identifier`

A unique identifier for a place.

`var alternateIdentifiers: Set<MKMapItem.Identifier>`

A set of alternative identifiers for a place.

`var identifier: MKMapItem.Identifier?`

A unique identifier for a place.

`var isCurrentLocation: Bool`

A Boolean value that indicates whether the map item represents the user's location.

`var name: String?`

The descriptive name associated with the map item.

~~`var placemark: MKPlacemark`~~

The placemark object containing the location information.

Deprecated

`var pointOfInterestCategory: MKPointOfInterestCategory?`

The point-of-interest category for the map item.

`var phoneNumber: String?`

The phone number associated with a business at the specified location.

`var timeZone: TimeZone?`

The time zone of the specified location.

```
var url: URL?
```

The URL associated with the specified location.

Launching the Maps app

```
class func openMaps(with: [MKMapItem], launchOptions: [String : Any]?)  
-> Bool
```

Opens the Maps app and displays the specified map items.

```
class func openMaps(with: [MKMapItem], launchOptions: [String : Any]?,  
completionHandler: ((Bool) -> Void)?)
```

Opens the Maps app using the specified map items and options.

```
class func openMaps(with: [MKMapItem], launchOptions: [String : Any]?,  
from: UIScene?, completionHandler: ((Bool) -> Void)?)
```

Opens the Maps app from a particular scene using the specified map items and options.

```
func openInMaps(launchOptions: [String : Any]?) -> Bool
```

Opens the Maps app and displays the map item.

```
func openInMaps(launchOptions: [String : Any]?, completionHandler: ((  
Bool) -> Void)?)
```

Opens the Maps app and displays the map item.

```
func openInMaps(launchOptions: [String : Any]?, from: UIScene?,  
completionHandler: ((Bool) -> Void)?)
```

Opens the Maps app from a particular scene using the specified options.

Serializing a map item

```
let MKMapItemTypeIDentifier: String
```

A constant that indicates the type of a serialized map item.

Opening items at launch time

☰ Launch options dictionary keys

Launch options to specify when opening map items in the Maps app.

☰ Directions mode values

Strings that represent the possible values of the launch options direction mode key.

Initializers

```
init?(coder: NSCoder)
```

```
init(location: CLLocation, address: MKAddress?)
```

Creates and returns a map item object using the specified location and address objects.

Instance Properties

```
var address: MKAddress?
```

The address object.

```
var addressRepresentations: MKAddressRepresentations?
```

The address representations object that contains various address representations useful for display purposes.

```
var location: CLLocation
```

The location object.

Relationships

Inherits From

NSObject

Conforms To

CVarArg

Copyable

CustomDebugStringConvertible

CustomStringConvertible

Equatable

Hashable

NSCoding

NSItemProviderReading

NSItemProviderWriting

NSObjectProtocol

See Also

Essentials

 [Enabling Maps capability in Xcode](#)

Configure your routing app to support providing directions.

 [Identifying unique locations with Place IDs](#)

Obtain information about a point of interest that persists over its lifetime.

`class` [MKMapView](#)

An embeddable map interface, similar to the one that the Maps app provides.