



HERE iOS SDK

Developer's Guide

Starter Edition Version 3.6

Important Information

Notices

Topics:

This section contains document notices.

- [*Legal Notices*](#)
- [*Document Information*](#)
- [*Service Support*](#)

Legal Notices

© 2017 HERE Global B.V. and its Affiliate(s). All rights reserved.

This material, including documentation and any related computer programs, is protected by copyright controlled by HERE. All rights are reserved. Copying, including reproducing, storing, adapting or translating, any or all of this material requires the prior written consent of HERE. This material also contains confidential information, which may not be disclosed to others without the prior written consent of HERE.

Trademark Acknowledgements

HERE is trademark or registered trademark of HERE Global B.V.

Other product and company names mentioned herein may be trademarks or trade names of their respective owners.

Disclaimer

This content is provided "as-is" and without warranties of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, satisfactory quality and non-infringement. HERE does not warrant that the content is error free and HERE does not warrant or make any representations regarding the quality, correctness, accuracy, or reliability of the content. You should therefore verify any information contained in the content before acting on it.

To the furthest extent permitted by law, under no circumstances, including without limitation the negligence of HERE, shall HERE be liable for any damages, including, without limitation, direct, special, indirect, punitive, consequential, exemplary and/ or incidental damages that result from the use or application of this content, even if HERE or an authorized representative has been advised of the possibility of such damages.

Document Information

Product

Name:	HERE iOS SDK
Version:	Starter Edition Version 3.6

Document

Name:	HERE iOS SDK Developer's Guide
ID:	ec87f01-1513029594-f5a1ee71
Status:	FINAL
Date:	2017-Dec-11, 22:04 (GMT)

Service Support

If you need assistance with this or any other HERE product, select one of the following options.

- If you have a HERE representative, contact them when you have questions/issues.
- If you manage your applications and accounts through developer.here.com, log into your account and check the pages on the SLA report or API Health. If this does not clarify the issue, then check stackoverflow.com/questions/tagged/here-api.
- If you have an evaluation plan, check stackoverflow.com/questions/tagged/here-api.
- If you have questions about billing or your account, [Contact Us](#).
- If you have purchased your plan/product from a HERE reseller, contact your reseller.

Contents

Chapter 1: Overview.....	7
What is the HERE iOS SDK?.....	8
Feature List.....	8
Legal Requirements.....	8
Chapter 2: Quick Start.....	10
Run the Sample Application.....	11
Create a Simple App Using the HERE SDK.....	11
Chapter 3: User Guide.....	19
System Requirements.....	20
Authenticating Applications.....	20
Examples on GitHub.....	21
App Submission Requirements.....	21
Mapping.....	22
Maps.....	22
Gestures.....	24
Map Schemes.....	26
Objects and Interaction.....	28
Custom Raster Tiles.....	31
Positioning.....	35
Basic positioning.....	35
Directions.....	38
Route Calculation.....	39
Search.....	43
Geocoding and Reverse Geocoding.....	43
Search and Discovery.....	46
Offline Search.....	53
Chapter 4: Supplemental Information.....	55
Create a Simple HERE SDK App Using Swift.....	56

HERE iOS SDK Developer's Guide

► Contents



Swift Support in HERE SDK.....	61
Supported Thread Usage.....	61
Chapter 5: Coverage Information.....	62
Chapter 6: API reference.....	63

Chapter 1

Overview

Topics:

- [*What is the HERE iOS SDK?*](#)
- [*Feature List*](#)
- [*Legal Requirements*](#)

The articles that follow introduce the HERE iOS SDK, explain essential concepts, and describe the common use cases it supports.

What is the HERE iOS SDK?

The HERE iOS SDK provides a set of programming interfaces that enable developers to build immersive, geographically-aware iOS applications by leveraging a powerful and flexible mapping platform. Through this SDK, developers can add rich location features such as routing, interactive maps, and global place search to their applications. The powerful client-side HERE iOS SDK also includes a sophisticated engine for rendering map data and calculated routes.

Feature List

The main features offered by the HERE iOS SDK are listed below.

Mapping:

- Raster map tiles with high resolution support
- Map styles: Normal, Satellite, Terrain, and more
- Touch gestures such as tap, pan, and pinch
- Overlay objects on the map such as polylines, polygons, icons, routes

Search:

- Search through a broad set of geographical content across the globe (including streets, address points, and categorized places)
- Search Places for somewhere specific or explore by categories
- Access rich details for a Point of Interest from third-party content sources (including images, ratings, reviews, and editorials)
- Perform geocoding and reverse geocoding lookups

Directions:

- Online Car and Pedestrian Route Directions
- Routing options (Highways, Tolls, Fastest etc.)
- Specify preferred route type (fastest or shortest) and route options (such as avoiding toll roads, motorways, and parks)
- Alternate routes

Legal Requirements

In addition to the applicable *terms and conditions* under which you have licensed the SDK, the following shall apply.

Components of the HERE SDK collect certain information from your application. Such information includes access credentials (App_Id and App_Code – see also *Authenticating Applications* on page 20) and the types of features utilized by your application when used by end users. The information does not identify an

individual end user. However, your application's privacy policy must disclose to the end users that you have licensed products and services from HERE and that such information is collected from your application as it is being used by end users and that HERE collects and processes such information from the application.

Chapter 2

Quick Start

Topics:

- [*Run the Sample Application*](#)
- [*Create a Simple App Using ...*](#)

The articles in this section provide information to help you start using the HERE iOS SDK.

Run the Sample Application

This tutorial provides instructions on how to run the Objective-C and Swift sample applications to render a map on an iOS device. The tutorial assumes you are using Xcode 9 and the iOS 11 SDK. For more details, see [System Requirements](#) on page 20.

Tasks for this basic application include:

- Check HERE Credentials.
- Launch the sample project using CocoaPods or manually from the package.

Acquire HERE SDK Credentials

Before working with the HERE SDK, you need to acquire a set of credentials by registering your application on <http://developer.here.com>. Each application requires a unique set of credentials.

Launch the Project Using Cocoapods

You can start trying a sample HERE iOS SDK project by using Cocoapods. Cocoapods is an open source dependency manager for Objective-C and Swift Xcode projects. To get started with using Cocoapods on macOS, follow the instructions in the [Getting Started](#) guide on CocoaPods.org.

Once you have configured Cocoapods in your development environment, try the tutorial project by using the following command.

```
pod try HEREMapsStarter
```

 **Note:** This command is only intended for product demonstration purposes. It only puts the HERE SDK in a temporary location. For more information on using Cocoapods in a Xcode project, see [Create a Simple App Using the HERE SDK](#) on page 11.

After invoking this command, you are prompted to choose between an Objective-C or a Swift sample project. Make a selection, and the script opens a project in Xcode, with HERE SDK dependencies already configured. You can then follow the instructions in the project README.txt to add app credentials and run the app.

Launch the Project from the Downloaded HERE iOS SDK Package

You can also open a sample project from the HERE iOS SDK package, which you can download from <http://developer.here.com>. The Xcode projects are available in the sample-apps folder. To run the project, double-click on HelloMap.xcodeproj or SwiftHelloMap.xcodeproj, then follow the instructions in the README.txt file.

Create a Simple App Using the HERE SDK

This guide provides instructions on creating a simple HERE iOS SDK application to render a map on an iOS device. Users are able to navigate the map by way of touch gestures such as panning, rotating, tilting, and pinching. The contents of this guide apply to Xcode 9 and the iOS 11 SDK.

This tutorial applies to development using Objective-C. For information on how to perform the same tasks using Swift, see [Create a Simple HERE SDK App Using Swift](#) on page 56.

- **Note:** The HERE iOS SDK is now distributed as a dynamic framework instead of a static library. Please review the following steps and update your Xcode project configuration if you are upgrading from an older versions of the HERE SDK. Also, ensure you first remove the old `NMAKit.framework`, `NMABundle.bundle`, and linked libraries from your Xcode project, before you add the new dynamic framework.

Acquire HERE SDK Credentials

Before working with the HERE SDK, you need to acquire a set of credentials by registering your application on <http://developer.here.com>. Each application requires a unique set of credentials.

Create a New Single View Application

1. From XCode menu, select **File > New > Project** to open the New project dialog (or press Shift + Command + N).
2. Select **iOS > Application > Single View Application** as the application type you want to create. Press Next.
3. In the next dialog, enter your **Product Name** (such as HelloMap) and **Organization Identifier** (such as `edu.self`).
4. Next, choose "Objective-C" under **Language**, then click **Next**. Navigate to the directory where you want your project to be stored and then select **Create**.
5. The next step is to configure this project to use the HERE SDK.

Add HERE iOS SDK

There are two ways to add the HERE iOS SDK to your Xcode project:

- CocoaPods
- Manually import the dynamic framework

To use CocoaPods to import the HERE SDK:

1. Create a Podfile in your project and add the following. Replace `YourApp` with your project name.

```
platform :ios, '9.0'
target 'YourApp' do
  pod 'HEREMapsStarter'
end
```

2. Next, open a command-line terminal on your workstation. From your project's root folder, run this command:

```
pod install
```

Alternatively, to manually import the HERE SDK dynamic framework:

1. Download the HERE iOS SDK package from <http://developer.here.com> and extract it to somewhere in your local file system.
2. Add the `NMAKit` dynamic framework to your Xcode project. Click on your app target and choose the "General" tab. Find the section called "Embedded Binaries", click the plus (+) sign, and then click the

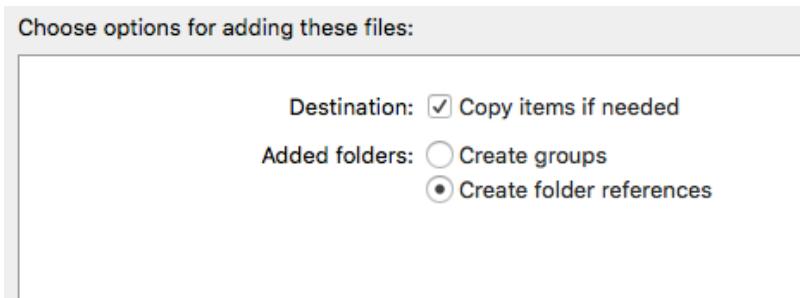
HERE iOS SDK Developer's Guide

► Quick Start



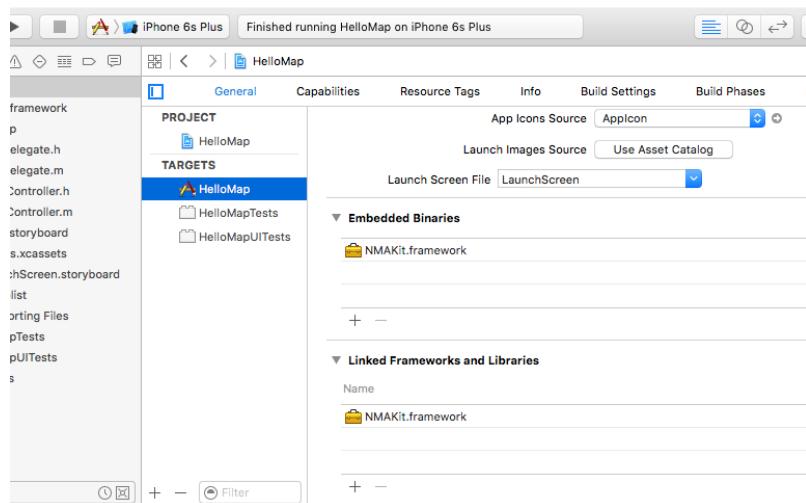
"Add Other" button. From the file dialog box, select the "NMAKit.framework" folder. Ensure that the "Copy items if needed" and "Create folder reference" options are selected, then click **Finish**.

Figure 1: Adding NMAKit.framework



3. Ensure that `NMAKit.framework` appears in the "Embedded Binaries" and the "Linked Frameworks and Libraries" sections.

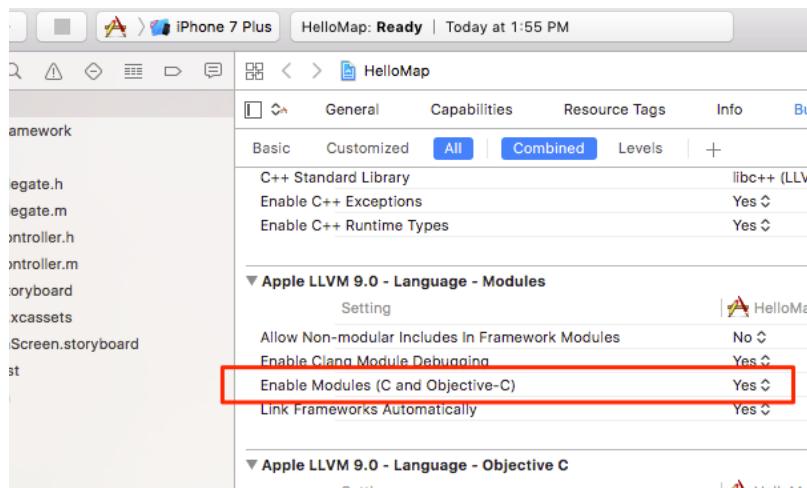
Figure 2: Added Embedded Binaries



Verify and Run the App

1. Next, ensure that modules are enabled. Click on the "Build Settings" tab and navigate to the "Apple LLVM 7.1 - Language" - Modules section. Ensure that the "Enable Modules (C and Objective-C)" has the value "YES".

Figure 3: Enable Modules



2. Run the application. From the Xcode menu bar, select **Product > Run**. Ensure that the project runs in the iOS Simulator without errors.
3. The HERE iOS SDK is now ready for use in your Xcode project. Now that you have your project configured to work with the HERE SDK, try extending the sample application to render a map.

Create the Map View

In this section, we utilize the `NMAMapView` and `NMAGeoCoordinates` classes to render a Map.

1. Create an `NMAMapView`.
 - a. Select `Main.storyboard` in the navigator, then open the Utilities view by pressing the key combination Command + Option + Control + 3. Drag and drop a View object from the Object Library onto the View Controller. If necessary, resize the View so it takes up the entire viewable area.
 - b. In the Interface Builder, click on the created View and then open the Identity Inspector in the Utilities view by pressing the key combination Command + Option + 3. Change the class value from

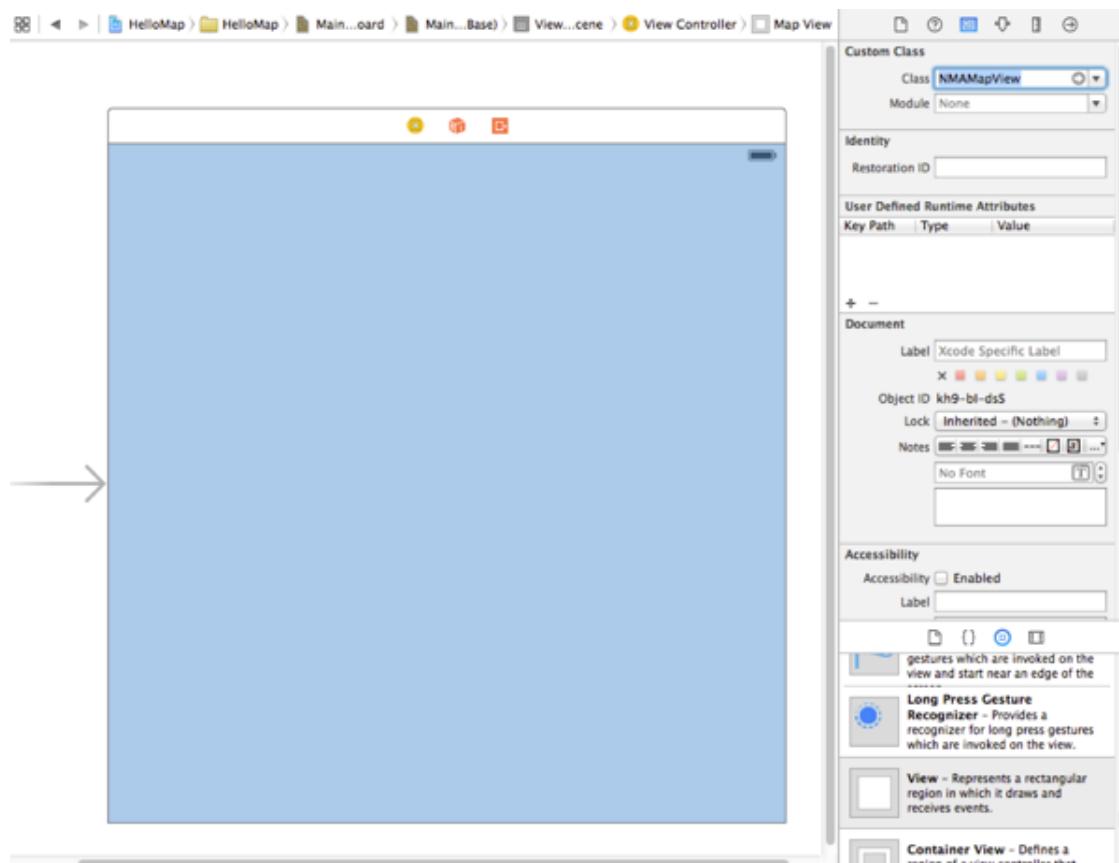
HERE iOS SDK Developer's Guide

► Quick Start



UIView to NMAMapView and press return. In the Document Outline, you should see that the name of the View has changed from to View to Map View.

Figure 4: MapView



2. Create an outlet to NMAMapView in ViewController.
 - a. Select Main.storyboard in the navigator.
 - b. Press Command + Option + Return to open the Assistant Editor. It should show ViewController.m.
 - c. Add the following import statement to the top of this file:

```
@import NMAKit;
```

- d. Hold the Control key on the keyboard and click to drag from the Map View to the interface block in ViewController.m. You should see a blue line and tooltip which says "Insert Outlet or Outlet Connection". Release the mouse button and a dialog appears, allowing you to create an outlet.

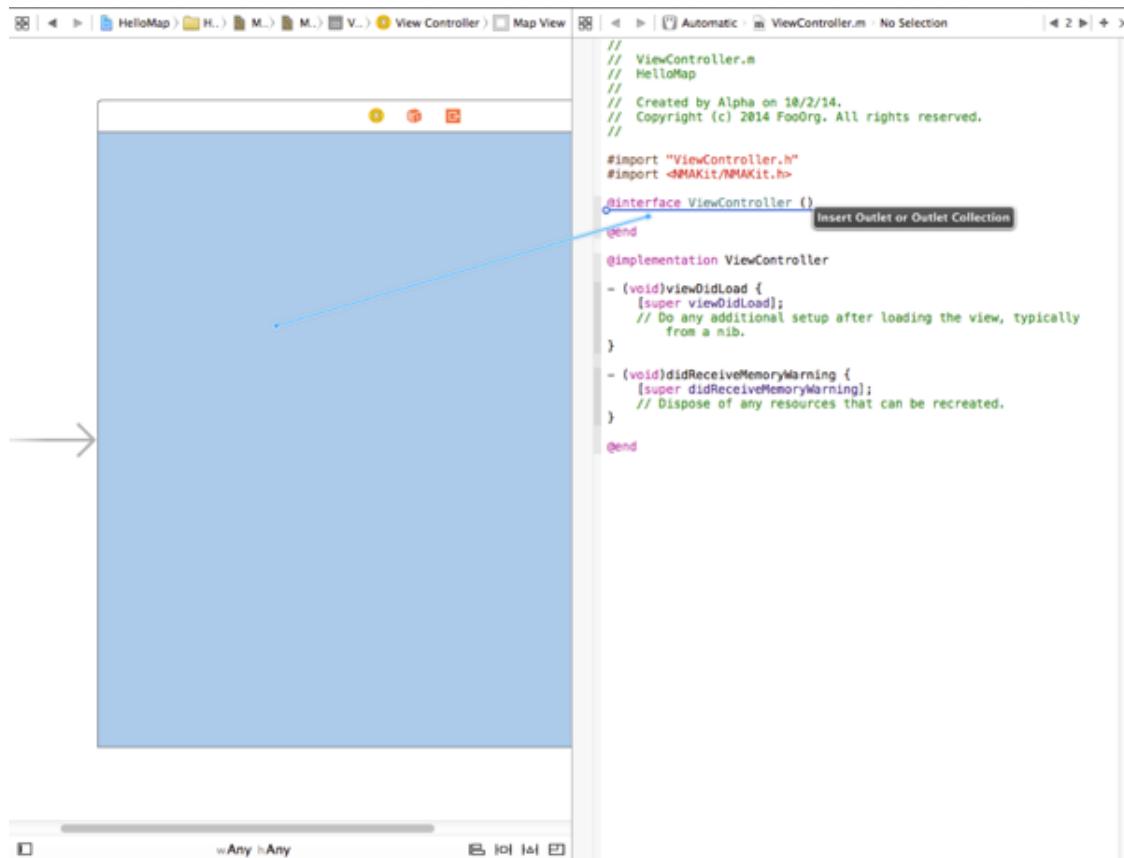
HERE iOS SDK Developer's Guide

► Quick Start



- e. Name the outlet *mapView*, keep the other default options and then select Connect.

Figure 5: Create an Outlet



3. Now an outlet to NMAMapView is set. The modified file should be as follows:

```
#import "HelloMapViewController.h"

#import NMAKit;

@interface HelloMapViewController ()
@property (weak, nonatomic) IBOutlet NMAMapView *mapView;
@end

@implementation HelloMapViewController

- (void)viewDidLoad
{
    [super viewDidLoad];
}

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
}

@end
```

4. Implement NMAMapView setup and lifecycle code by modifying the `viewDidLoad` method as shown:

```
- (void)viewDidLoad
{
    [super viewDidLoad];
```

```
//set geo center
NMAGeoCoordinates *geoCoordCenter =
[[NMAGeoCoordinates alloc] initWithLatitude:49.260327 longitude:-123.115025];
[self.mapView setGeoCenter:geoCoordCenter withAnimation:NMAMapAnimationNone];
self.mapView.copyrightLogoPosition = NMALayoutPositionBottomCenter;

//set zoom level
self.mapView.zoomLevel = 13.2;
}
```

5. Add your HERE application credentials.

- Open *AppDelegate.m* and import *NMAKit* by adding the following import statement to the top of the file.

```
@import NMAKit;
```

- Add the following in the *didFinishLaunchingWithOptions* method, replacing *YOUR_APP_ID* and *YOUR_APP_CODE* with the credentials that you received from <http://developer.here.com>.

```
[NMAApplicationContext setappId:@"{YOUR_APP_ID}"
appCode:@"{YOUR_APP_CODE}"];
```

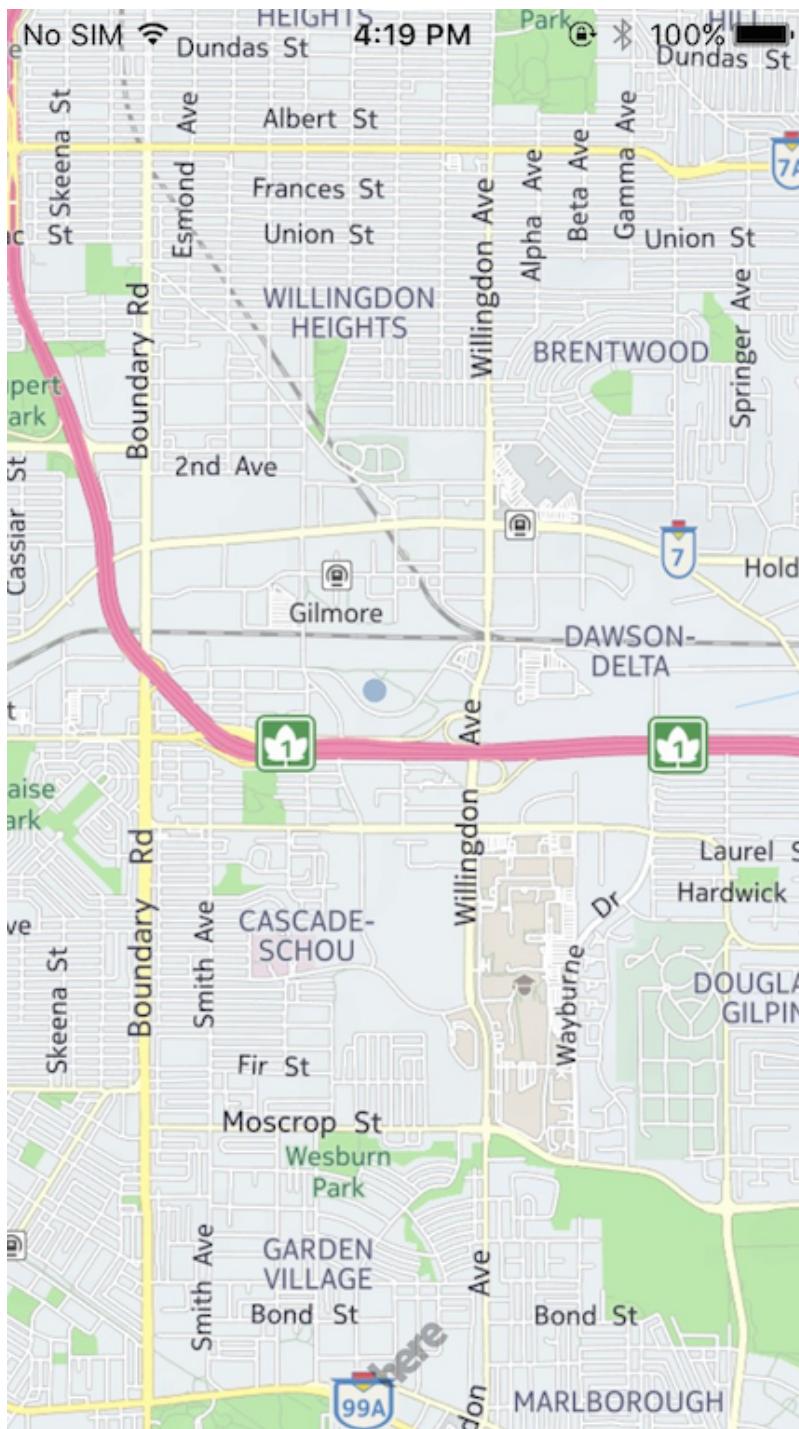
HERE iOS SDK Developer's Guide

► Quick Start



6. Build and run the application. If the build is successful, you now have an application that displays a map similar to the following screenshot and allows you to manipulate it using gestures.

Figure 6: Running the App



Chapter 3

User Guide

Topics:

- [*System Requirements*](#)
- [*Authenticating Application...*](#)
- [*Examples on GitHub*](#)
- [*App Submission Requirement...*](#)
- [*Mapping*](#)
- [*Positioning*](#)
- [*Directions*](#)
- [*Search*](#)

The articles in this section provide a guide to using the HERE iOS SDK.

System Requirements

- HERE SDK supports iOS 10 or above. iOS 11 is recommended for optimal operation.
- HERE iOS SDK is designed and tested for use with iOS phones and tablets. Supported devices are:
 - iPhone 5 or newer
 - iPad 3 or newer
 - iPad Mini 2 or newer
- **Note:** While the HERE iOS SDK supports both 32-bit and 64-bit devices, iOS 11 is 64-bit only, and only these devices are supported:
 - iPhone 5s or newer
 - iPad Air or newer, including iPad (2017) and iPad Pro
 - iPad Mini 2 or newer
- Apps should be developed using XCode 9 or above, running on macOS 10.12.6 or above
- A minimum of 3.5MB per application should be made available for the storage of the HERE SDK libraries
- A minimum of 32MB should be made available for the storage of map data.
- Data connectivity (WiFi or Cellular) is required to download map data.

Authenticating Applications

Developers using the HERE SDK with their app are required to register for a set of HERE credentials, and to specify these credentials (`App_Id` and `App_Code`) in their application. Failure to do so results in blocked access to certain features and degradation in the quality of other services.

To obtain these credentials, visit the developer portal at <https://developer.here.com/?create=Evaluation> and register for a free Evaluation license. Once your project is created, you can generate these credentials on your Project Details page. If you already have a commercial (public or business) plan, you can also retrieve these credentials from your Project Details page.

- **Note:** Credentials are unique to your application. Do not reuse credentials across multiple applications.

Adding Credentials

Ensure that you have provided the `app_id` and `app_code` before using the HERE SDK. For example, set them in your app delegate:

```
- (BOOL)application:(UIApplication *)application  
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions  
{  
    [NMAApplicationContext setAppId:@"{YOUR_APP_ID}" appCode:@"{YOUR_APP_CODE}"];  
  
    return YES;  
}
```

Examples on GitHub

You can find more HERE SDK sample projects on GitHub: <https://www.github.com/heremaps>

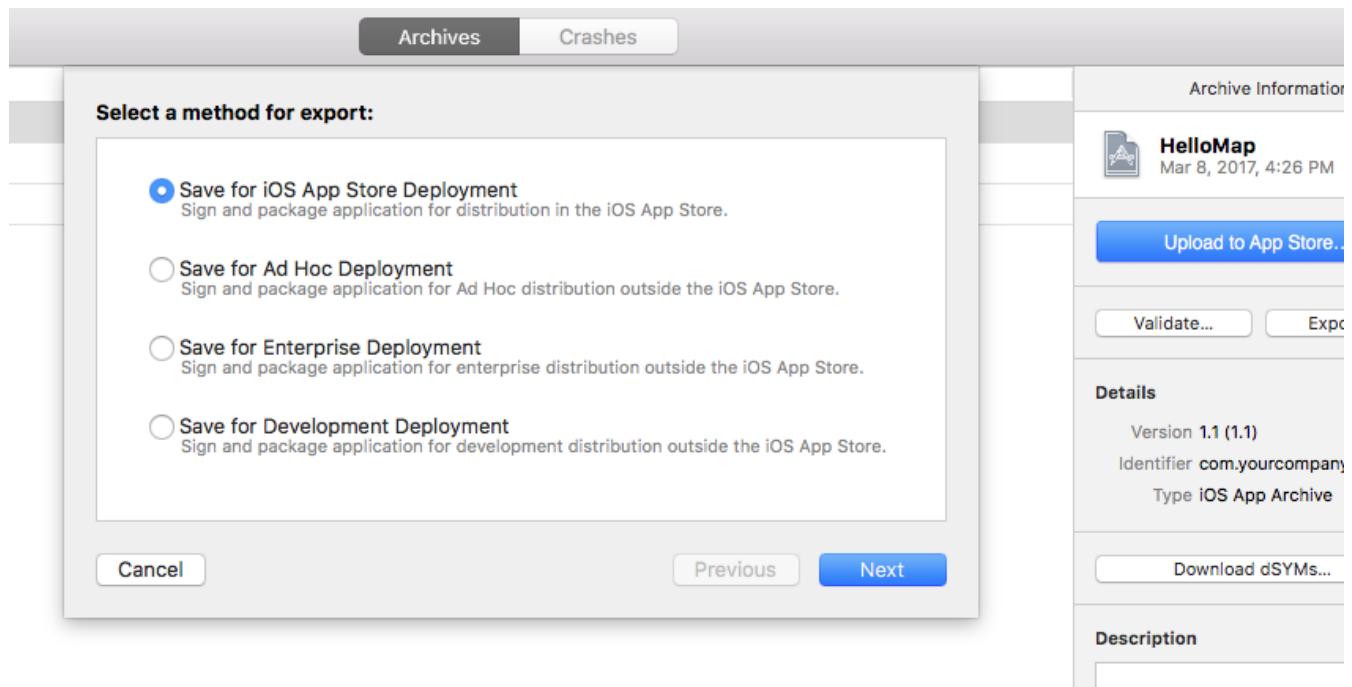
App Submission Requirements

This section contains important information on how to prepare your HERE iOS SDK-enabled app for App Store submission. Be sure to follow the advice here to avoid store rejections caused by the HERE SDK.

Remove Simulator Architectures

Apple rejects apps that are submitted to the App Store with simulator-specific architectures (x86_64, i386). Similarly, Xcode also gives an error if you try to export these apps for other types of deployment. Prior to exporting your app, you must strip simulator-specific architectures from the NMAKit framework binary and rebuild your app.

Figure 7: Exporting According to Deployment Type



A script is provided with the HERE SDK to strip the simulator architectures from the NMAKit library. Before you build your app to upload to the store, make a copy of `NMAKit.framework` and run the script to create a framework that only contains ARM device architectures (arm64, armv7). This modified version of the framework should be included in any builds destined for the app store.

This script is located at the following location:

```
{SDK Root}/framework/stripe_sim.sh
```



To run the script, open a terminal and go to the directory containing the script and execute it from there.

You can also add a script as part of your Xcode build process to automatically strip unnecessary architectures from frameworks. Depending on your build environment, this may work better than managing separate versions of `NMAKit.framework`. For more details about this solution, see [this article on Stack Overflow](#).

Declare Private Data Access

Starting with iOS 10, you must statically declare access to private user data, such as address book and device location, before submitting your app to the App Store. Each declaration requires adding a key and a purpose string in your app's `Info.plist` file.

The following shows the required HERE iOS SDK data access declarations. You can use the recommended `<string>` entries in your app's `Info.plist`, or you can choose strings that are appropriate to your app.

■ **Note:** All HERE iOS SDK apps must have these declaration entries for the App Store submission process. App users are only prompted for the appropriate data access if your app uses the relevant HERE SDK feature.

```
<key>NSLocationWhenInUseUsageDescription</key>
<string>This is needed to determine your current location</string>
```

You can find more information about these keys in the following article: <https://developer.apple.com/library/content/documentation/General/Reference/InfoPlistKeyReference/Articles/CocoaKeys.html>.

Mapping

Maps

The core feature of the HERE iOS SDK is Mapping. The key concepts covered in this section include adding a map to an iOS application, changing the location displayed by the map, and modifying its properties. The primary component of the mapping API is the `NMAMapView`, which is integrated with the Cocoa Touch framework as a `UIView` subclass. `NMAMapView` represents the view to display map and various properties. The `NMAMapView` is derived from `UIView` and part of iOS Cocoa Touch framework.

■ **Note:** To create a simple map application, refer to the [Quick Start](#) section.

The first step to integrate a map into an application is to insert a `NMAMapView` to your view controller's `.xib` file or the storyboard of the application using the Interface Builder. Alternatively, you can also add `NMAMapView` to your view controller programmatically as follows:

```
- (void)viewDidLoad
{
    mapView = [[NMAMapView alloc] initWithFrame:self.view.frame];
    [self.view addSubview:mapView];
}
```

The `NMAMapView` class handles all user interactions in the form of touch gestures. More details about the supported gesture types can be found in the [Map Gestures](#) section.

Working with NMAMapView

Once the NMAMapView is initialized, it can be manipulated and interacted in a variety of ways. Some key attributes of the NMAMapView are its geographical center (`geoCenter`) and zoom level (`zoomLevel`). These properties may be used to customize the NMAMapView display. For example, the following code demonstrates how to show a view of Vancouver, Canada.

```
NMAGeoCoordinates *geoCoordCenter = [[NMAGeoCoordinates alloc]
    initWithLatitude:49.260327 longitude:-123.115025];
[self.mapView setGeoCenter:geoCoordCenter withAnimation:NMAMapAnimationNone];
```

In the preceding code:

- The geographical location [`NMAGeoCoordinates`] for the new map center is created by a call to the `-(id)initWithLatitude:(double)aLatitude longitude:(double)aLongitude` method.
- When setting the center of a map, the transition can be animated by passing the `NMAMapAnimationLinear` enum value to the `animation` parameter. Animation can also be suppressed by using the `NMAMapAnimationNone` value.

The beginning and ending of these events may be observed by assigning an object to the NMAMapView's `delegate` property. The object should implement the methods of the `NMAMapViewDelegate` protocol corresponding to the events you wish it to receive. This delegate can also be used to detect *map object selection*.

- **Note:** For optimum performance, avoid resizing a map after it has been created. If resizing is necessary, create the map at the largest size to be used and reduce it later.

Resolution and Text Size

By default, the HERE SDK uses high-resolution (512 x 512 pixels) map tiles. You can set the `useHighResolutionMap` property in `NMAMapView` to `NO` to use lower-resolution (256 x 256 pixels) map tiles instead.

You can also use the `mapPPI` property to change the map's pixel-per-inch setting. The default setting is `NMAMapPPILow`. Setting the property to `NMAMapPPIHight` makes street labels and other text to be bigger.

Properties of NMAMapView

The following examples show how to work with some of the properties in `NMAMapView`:

Map Center

The center of the map determines the geographical area to be displayed. It can be read using the `NMAMapView geoCenter` property, and set using one of the `setGeoCenter:` methods. Its type is `NMAGeoCoordinates`.

```
// Move the map to London.
NMAGeoCoordinates *geoCoordCenter = [[NMAGeoCoordinates alloc]
    initWithLatitude:51.51 longitude:-0.11];
[self.mapView setGeoCenter:geoCoordCenter withAnimation:NMAMapAnimationNone];
```

Zoom Level

The size of the geographical area displayed by the map can be controlled by changing the zoom level. The zoom level ranges from `NMAMapViewMinimumZoomLevel` to `NMAMapViewMaximumZoomLevel`, with a higher zoom value being closer to the ground. The following code sets the zoom level to the median zoom level:

```
// Set the zoom level to the median.  
mapView.zoomLevel =  
(NMAMapViewMinimumZoomLevel + NMAMapViewMaximumZoomLevel)/2.0f;
```

Animations

The `NMAMapView` supports the following animation settings to be used while changing properties, defined by the `NMAMapAnimation` enum:

- `NMAMapAnimationNone`
- `NMAMapAnimationLinear`

```
// Move to London using bow animation  
NMAGeoCoordinates *geoCoordCenter = [[NMAGeoCoordinates alloc]  
initWithLatitude:51.51 longitude:-0.11];  
[mapView setGeoCenter:geoCoordCenter withAnimation:NMAMapAnimationLinear];
```

Setting Multiple Attributes

An extended API is provided to change one or more attributes at the same time.

```
-(void) setGeoCenter:(NMAGeoCoordinates*) coordinates  
zoomLevel:(float) level  
withAnimation:(NMAMapAnimation) animation
```

To leave a map attribute unchanged, pass the `NMAMapViewPreserveValue` constant to the relevant method parameter.

```
// Move to Vancouver, preserving zoom level  
NMAGeoCoordinates* coord = [[NMAGeoCoordinates alloc]  
initWithLatitude:49.0  
longitude:123.0];  
[mapView setGeoCenter:coord  
zoomLevel:NMAMapViewPreserveValue  
withAnimation:NMAMapAnimationNone];
```

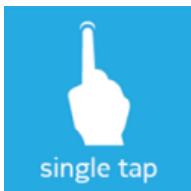
For more information about the APIs introduced and demonstrated in this section, refer to the API Reference documentation.

Map Gestures

The `NMAMapView` class responds to a number of predefined touch gestures. The default behavior of the map for each gesture type may be used as-is, supplemented, or replaced entirely. The following table is a summary of the available gestures and their default behavior.

HERE iOS SDK Developer's Guide

► User Guide



To select a visible map object, tap the screen with one finger.



To zoom the map in a fixed amount, tap the screen twice with one finger. Tap continuously to make a continuous zoom.



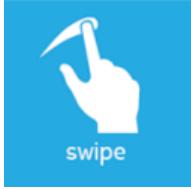
To zoom out a fixed amount, tap the screen with two fingers. Tap continuously to make a continuous zoom.



To move the map, press and hold one finger to the screen, and move it in any direction.



Press and hold two fingers to the screen and move them in the same direction. This gesture does not have a predefined map action.



To pan the map with momentum, press and swipe one finger on the screen. The map continues to move in the same direction, and gradually slow to a stop.



To continuously zoom in or out, press and hold two fingers to the screen, and increase or decrease the distance between them.



Pressing and holding one finger to the screen activates the long press gesture. This gesture does not have a predefined map action.

The time required to trigger a long press gesture can be customized using the `NMAMapView` `longPressDuration` property. The default value for this property is one second.

Controlling the NMAMapView Gesture Response

Any of the gestures listed above may be selectively enabled or disabled on an `NMAMapView` instance using the `enableMapGestures:` and `disableMapGestures:` methods. These methods take a single input parameter that is an "or" combination of `NMAMapGestureType` values, which are defined in `NMAMapGesture.h`. The state of a specific gesture may be checked with `isMapGestureEnabled:`.

The following code shows how to disable all panning gestures:

```
// mapView is a valid NMAMapView instance  
[mapView disableMapGestures:NMAMapGestureTypePan];
```

Gesture Delegation

To receive notifications of gestures, you can implement the `NMAMapGestureDelegate` and use the corresponding handler methods. For example:

```
@implementation MyGestureDelegate  
// ...  
-(void)mapView:(NMAMapView*)mapView didReceiveTapAtLocation:(CGPoint)location  
{  
    // a gesture was received  
    // location is available through the "location" parameter  
}  
@end
```

■ **Note:** Implementing the handler method does not have any impact on how the map responds to the gesture. For example, the logic within `mapView:didReceiveDoubleTapAtLocation:` is called when the map has zoomed in.

Map Schemes

Specific map schemes are available to offer your application users a choice among different kinds of map appearance.

Setting the Scheme

The `NMAMapScheme.h` file defines schemes that the HERE map service supports. You can set a desired scheme by changing the `mapScheme` property of `NMAMapView`. For example:

```
mapView.mapScheme = NMAMapSchemeNormalDay;
```

HERE iOS SDK Developer's Guide

► User Guide



Examples of Map Schemes

All available schemes are defined as constant strings in the `NMAMapScheme.h` file. The string values that your application can use to set a map scheme include:

Figure 8: `NMAMapSchemeNormalDay`

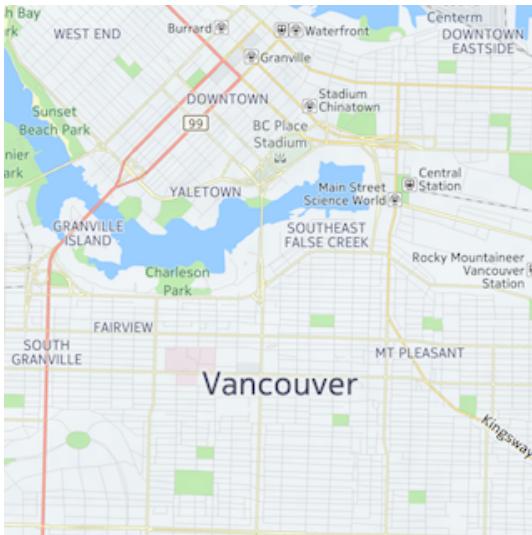


Figure 9: `NMAMapSchemeSatelliteDay`

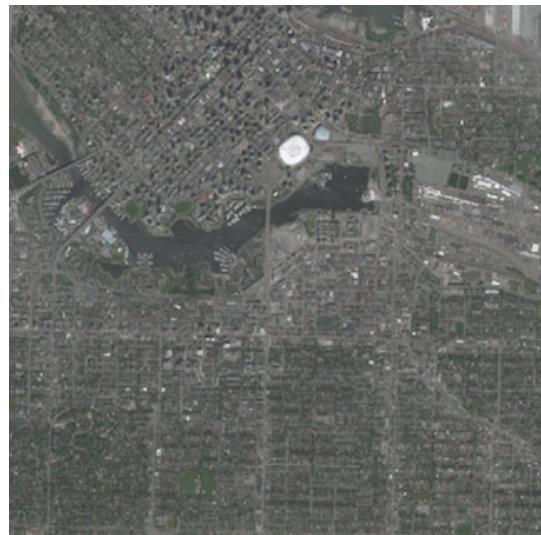
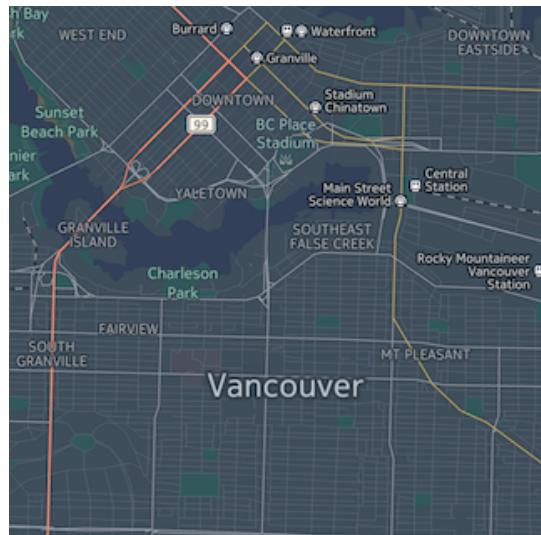


Figure 10: `NMAMapSchemeHybridDay`



Figure 11: `NMAMapSchemeNormalNight`



HERE iOS SDK Developer's Guide

► User Guide



Figure 12: NMAMapSchemeTerrainDay

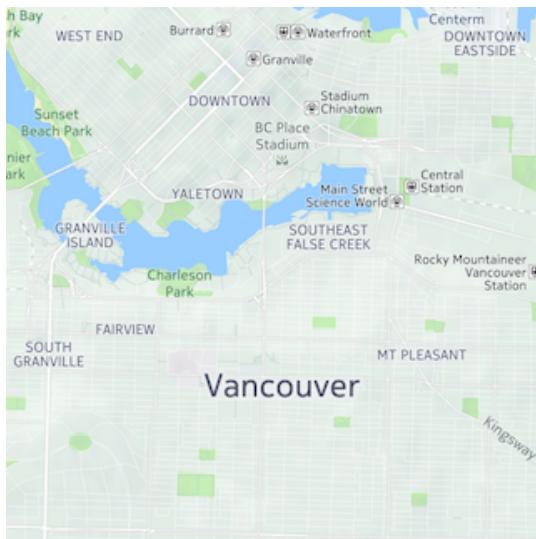


Figure 13: NMAMapSchemeReducedDay

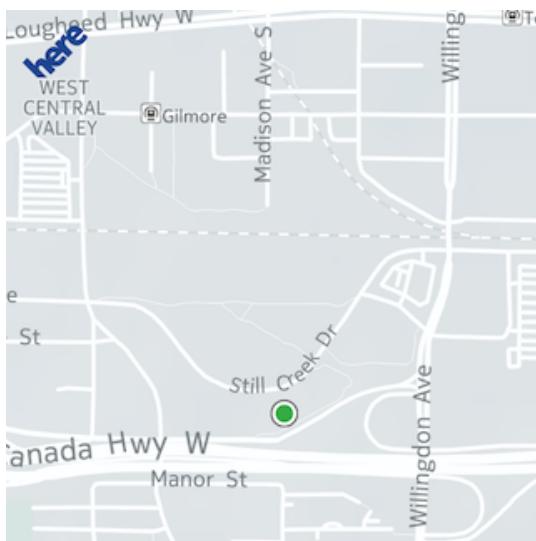
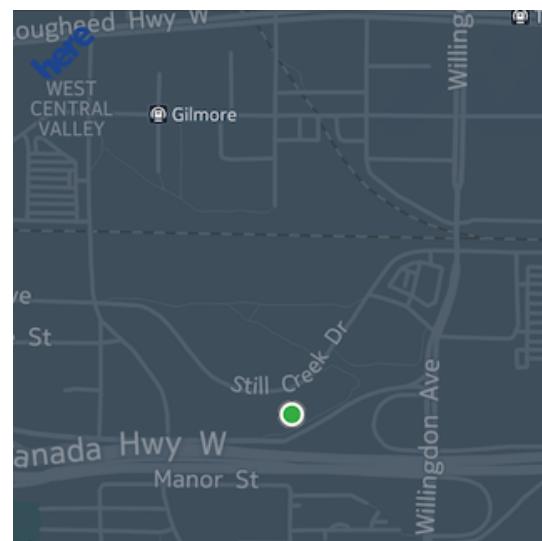


Figure 14: NMAMapSchemeReducedNight



Objects and Interaction

The HERE SDK allows the addition of a variety of objects, each with a specific purpose, to a map view. The types of available object include map markers, routes, polylines, and overlays. These objects are described in more detail below.

The NMAMapObject Class

The `NMAMapObject` class provides a generic base class from which most types of specialized map object inherit. Functionality that is common to all these object types is encapsulated in `NMAMapObject`. The following is a list of the important properties and methods in `NMAMapObject`.

- `zIndex` - determines the objects stacking order, which controls how the object is displayed on the map relative to other objects that may overlap it
- `visible` - determines whether or not the object is drawn when the map is rendered
- `type` - contains the type of the map object, such as marker, polyline, and route. For the full list, see the `NMAMapObject` API reference.
- `parent` - the `NMAMapContainer` instance holding this object, if any
- `uniqueId` - uniquely identifies the object for the duration of the application launch

■ **Note:**

- `NMAMapObject` serves as a base class to other map object types and should not be instantiated directly.
- Any change in a map object's visual appearance causes the entire map view to be redrawn, since map objects are drawn as part of the map itself. For optimal performance, map objects should not be frequently updated unless it is necessary.

The `NMAMapContainer` class

Map containers are a special type of map object that can be used to group together other map objects of certain types. The types of object allowed are `NMAMapMarker`, `NMAMapCircle`, `NMAMapPolygon`, and `NMAMapPolyline`. Containers provide a convenient way to control the stacking order and visibility of a large group of objects.

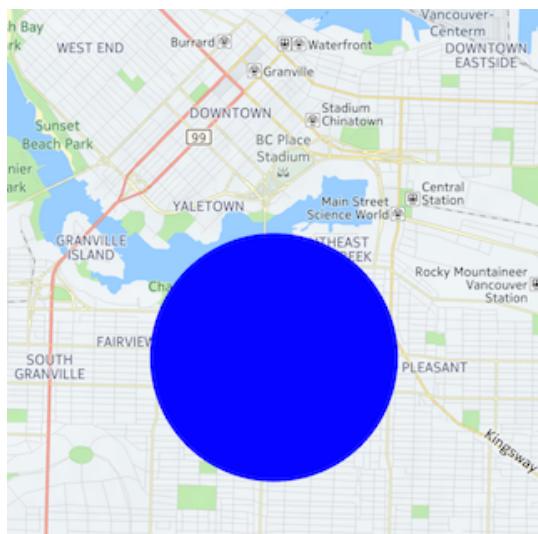
To use a map container, create one or more map objects and add them to the container using the `addMapObject` method. To show the objects on a map, add the map container to the map with the `addMapObject` method of `NMAMapView`.

■ **Note:** A container may also hold other instances of `NMAMapContainer`.

The `NMAMapCircle` class

An `NMAMapCircle` class is used to draw a circle on the map at a fixed geographical location; custom border and fill colors may be defined.

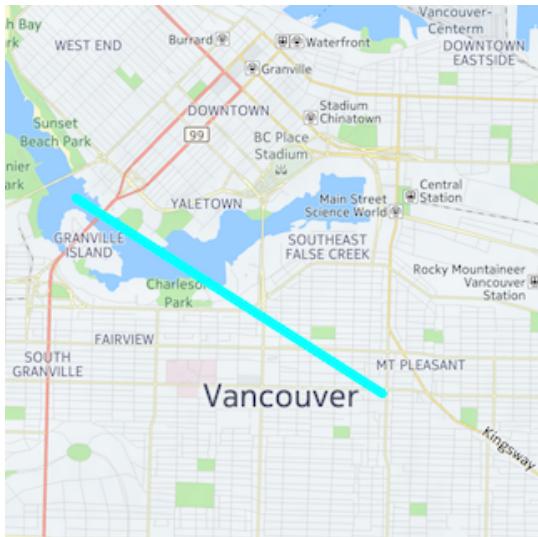
Figure 15: A MapCircle object



The NMAMapPolyline class

The NMAMapPolyline class is used to draw one or more connected line segments on the map. The segment vertices are specified by a series of NMAGeoCoordinates. The visual appearance of the polyline can be customized.

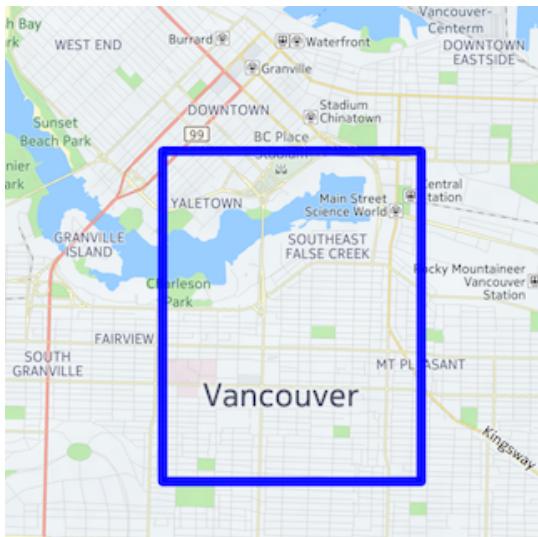
Figure 16: A MapPolyline object



The NMAMapPolygon interface

The NMAMapPolygon class is similar to NMAMapPolyline, but the first and last points of the line are automatically joined to create a closed shape. Polygon objects can have different border and fill colors.

Figure 17: A MapPolygon object with transparent fill and a border



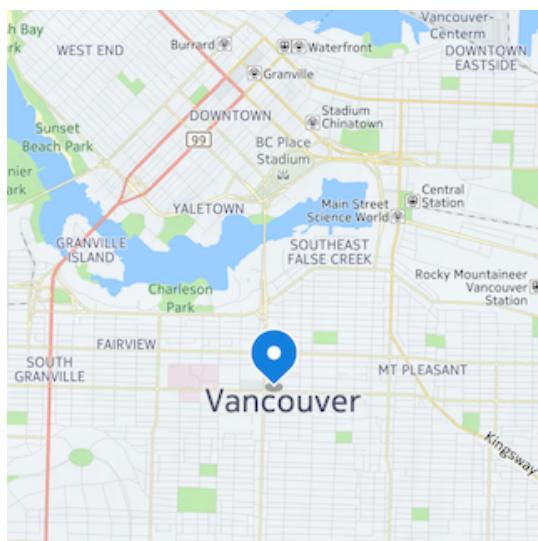
- Note:** Polygons that have a path that crosses over itself are not supported. For example, it is not possible to create a "bowtie" shape using four line segments, where one line segment crosses over another. However, you can create the shape using two triangles.

The NMAMapMarker class

The NMAMapMarker class is used to display a custom icon at a fixed geographical position on the map.

Custom icons can be provided as an `UIImage`.

Figure 18: A NMAMapMarker object



You can set `NMAMapMarker` to be draggable by setting the `draggable` property to YES. To listen for drag events, such as marker position changes, use the `respondToEvents:withBlock:` method in `NMAMapView`.

Map Object Selection

All user-defined objects with a visual representation can be selected. Selection occurs when a visible object on the map is tapped. By default, the map does not take any action when objects are selected. To implement selection handling, a custom class must implement the `NMAMapViewDelegate` protocol and its `onMapObjectsSelected` method. The `onMapObjectsSelected` callback returns an array that contains instances of `NMAViewObject`, which is a superclass of `NMAMapObject`.

Object selection can also be programmatically invoked by using the `objectsAtPoint:` or `visibleObjectsAtPoint:` method. Each of these methods takes a `CGPoint` screen coordinate and returns an `NSArray` of `NMAMapObject` at that location. The `visibleObjectsAtPoint` method does not return any object that has the `visible` property set as NO.

For more information, see the `NMAMapView` API documentation.

Custom Raster Tiles

You can use the HERE iOS SDK to enhance maps with the custom raster tiles API — `NMAMapTileLayer`.

Custom raster tiles are tile images that you can add to a map for enhancing the map with extra information over a large geographical area. If the application is set to display custom raster tiles, then users see them whenever they view a designated geographical area at a specified zoom level or range of zoom levels.

You can provide tile images in two ways:

1. Store custom raster tile images on a remote server and return URLs via the `NMAMapTileLayerDataSource mapTileLayer:urlForTileAtX:y:zoomLevel:` protocol method.
2. Provide raw bitmap data using the `NMAMapTileLayerDataSource mapTileLayer:requestDataForTileAtX:y:zoomLevel:` protocol method.

Dividing a Map and Using Tile Coordinates

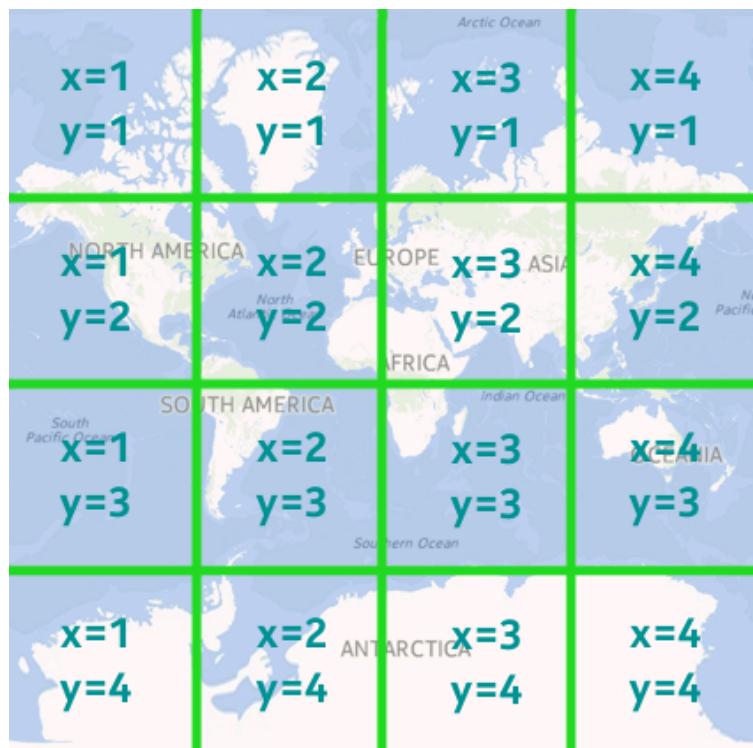
`NMAMapTileLayer` uses a scheme that divides the world map into tiles, specified by x, y and zoom level coordinates. This coordinate system is used by the `NMAMapTileLayerDataSource` protocol when it requests tiles.

At each zoom level, it is expected that the world map is rendered on $(2^{\text{zoomlevel}})^2$ tiles:

- at level 3: $8 \times 8 = 64$ tiles
- at level 4: $16 \times 16 = 256$ tiles
- continuing on until zoom level 20

For example, at zoom level 2, the world map would be divided up as follows:

Figure 19: World Map at Zoom Level 2



The x and y parameters indicate which tile is being requested for the given zoom level.

You need to provide enough tile images to cover all the zoom levels you are supporting within a geographical area. You can restrict custom tile rendering to a specific `NMAGeoBoundingBox` using the `boundingBox` property of `NMAMapTileLayer`. You can restrict the zoom level using `showAtZoomLevel:` and related methods.

Supplying Tiles from a Web Server

The steps for providing custom tiles from a web server to a map view are as follows:

1. Host the appropriate number of tiles on a server according to the zoom levels and NMAGeoBoundingBox you are supporting. The tiles must be in either PNG or JPG format and should be sized at 256 x 256 or 512 x 512 pixels.
 - **Note:** As long as your tiles conform to 256 by 256 or 512 by 512 pixels, the NMAMapView can scale the image to fit the current map resolution. For example, if the useHighResolutionMap property is set to YES (512 by 512 pixels) but your tiles are 256 by 256 pixels, the NMAMapView enlarges your image tiles.
2. Create an object that derives from NMAMapTileLayerDataSource and implement the mapTileLayer:urlForTileAtX:y:zoomLevel:method to return a URL pointing to the specified tile on your server.
3. Create an NMAMapTileLayer object and set its properties to correspond with the tile data source server. At a minimum, set the boundingBox and zoomLevel properties to reflect the tiles hosted on your server. Set the dataSource property.
4. Add the NMAMapTileLayer object to the NMAMapView by calling the addMapTileLayer: method.

The following code snippet shows a class that renders the Queen Elizabeth Olympic Park in London. The park is displayed as a tile layer that is added to an NMAMapView, and the raster tiles are served from the HERE server. To use this class, call [OlympicParkTileLayer addOlympicParkTileLayerToMapView:myMapView].

```
@interface OlympicParkTileLayer : NMAMapTileLayer <NMAMapTileLayerDataSource>
@end

@implementation OlympicParkTileLayer

+(void)addOlympicParkTileLayerToMapView:(NMAMapView*)mapView
{
    OlympicParkTileLayer *tileLayer = [OlympicParkTileLayer new];
    [mapView addMapTileLayer:tileLayer];
    [mapView setGeoCenter:titleLayer.boundingBox.center
        zoomLevel:14.0
        withAnimation:NMAMapAnimationNone ];
}

-(id)init
{
    if (self = [super init]) {
        // Set the data source
        self.dataSource = self;

        // Limit the tiles to the bounding box supported by the server
        NMAGeoBoundingBox *olympicParkBoundingBox =
        [NMAGeoBoundingBox geoBoundingBoxWithTopLeft:[NMAGeoCoordinates
            geoCoordinatesWithLatitude:51.557000 longitude:-0.042772]
        bottomRight:[NMAGeoCoordinates geoCoordinatesWithLatitude:51.525941
            longitude: 0.028296]];
        self.boundingBox = olympicParkBoundingBox;

        // Enable caching
        self.cacheTimeToLive = 60 * 60 * 24;      // 24 hours
        self.cacheSizeLimit = 1024 * 1024 * 64; // 64MB
    }
}
```

```
[self setCacheEnabled:YES withIdentifier:@"OlympicParkTileLayer"];
}
return self;
}

-(NSString *)mapTileLayer:(NMAMapTileLayer *)mapTileLayer
    urlForTileAtX:(NSUInteger)x
    y:(NSUInteger)y
    zoomLevel:(NSUInteger)zoomLevel
{
// Return a URL for the specified tile
// This tile source is hosted by HERE Global B.V. and may be removed at any time
return [NSString stringWithFormat:
    @"http://api.maps.example.org/maptiles/olympic_park/normal.day/%d/%d/%d.png",
    zoomLevel,
    y,
    x ];
}

@end
```

Supplying Tiles as Bitmaps

You can choose to supply tiles as bitmaps if your app uses bundled static tiles, dynamically-generated tiles, or if the server that you are using for tile images requires authentication. In the third case, since `NMAMapTileLayer` only uses simple HTTP GET requests to retrieve tile images, it is up to you to provide code that handles authentication and downloads the tiles. Once the tiles have been downloaded, you can use them as local bitmaps with the `NMAMapTileLayer` class.

The steps for providing custom tiles as local bitmaps to a map view are as follows:

1. Create an object that derives from `NMAMapTileLayerDataSource` and implements the `mapTileLayer:requestDataForTileAtX:y:zoomLevel:` method.
2. Create an `NMAMapTileLayer` object and set its properties to correspond with the tile data source. Set the `dataSource` property.
3. Add the `NMAMapTileLayer` object to the `NMAMapView` by calling the `addMapTileLayer:` method.

Caching Tiles

Tiles can be cached to the disk to improve performance and reduce data traffic in the URL fetching case.

When you enable caching, you must provide a cache identifier. This identifier must be unique for each `NMAMapTileLayer` used within your application. Since the cache persists across app sessions, it is important to use the same identifier across sessions (by defining a constant, for example).

You can optionally limit the cache size and time-to-live for each cached tile. The cache can be cleared at any time by calling `[NMAMapTileLayer clearCache]`. To be sure the cache is completely cleared first remove the `NMAMapTileLayer` from the map view before calling `[NMAMapTileLayer clearCache]`.

The following code enables disk caching with a 128MB maximum size and a tile time-to-live of 7 days:

```
NMAMapTileLayer *tileLayer = [[NMAMapTileLayer alloc] init];
tileLayer.dataSource = self; // Assuming self is a valid data source
[tileLayer setCacheEnabled:YES withIdentifier:@"MyUniqueTileCacheIdentifier"];
tileLayer.cacheTimeToLive = 60 * 60 * 24 * 7; // 7 days
tileLayer.cacheSizeLimit = 1024 * 1024 * 128; // 128 MB
[mapView addMapTileLayer:tileLayer]; // NOTE: add to map view after setting tile properties
```

Performance Tips

1. IMPORTANT: Set `NMAMapTileLayer` properties before adding the tile layer to the map view. Most properties ignore attempts to set them after being added to the view.
2. Ensure the properties you set on the tile layer match the data you are supplying via the `NMAMapTileLayerDataSource` protocol.
3. Do not block `NMAMapTileLayerDataSource` methods for extended periods of time. For example, if it takes a while to generate tiles on the fly move the processing to a separate GCD queue.
4. If requesting a specific tile is constantly failing consider implementing `mapTileLayer:hasTileAtX:y:zoomLevel:`, returning NO.
5. Use the provided disk caching mechanism.

Positioning

Basic Positioning

An application created using the HERE iOS SDK can use information from the positioning capabilities of a user's device to display its position, and, optionally, provide real-time updates. Getting the current position requires an application to make use of the `NMAPositioningManager` interface from the HERE SDK. In order to receive position updates or position-lost notifications, an application should use `NSNotificationCenter addObserver` with the notification names found in `NMAPositioningManager.h`:

- `NMAPositioningManagerDidUpdatePositionNotification`
- `NMAPositioningManagerDidLosePositionNotification`

The user's current position can be easily displayed on the map using the `NMAPositionIndicator` class. Each instance of `NMAMapView` owns an instance of this class, accessed via the `positionIndicator` property.

NMAPositioningManager

The `NMAPositioningManager` class provides information related to the device's geographical location, such as the current position and the average speed. `NMAPositioningManager` is a singleton class and thus should only be accessed through the `sharedPositioningManager` class method.

- **Note:** Add `NSLocationWhenInUseUsageDescription` to your project's `Info.plist` so that `CLLocationManager` can properly access the user's location. The value of this key is displayed to the user when the system requests for permission to use location services. See [App Submission Requirements](#) on page 21 for recommended strings.

To start receiving real time positioning updates, the application needs to call `NMAPositioningManager startPositioning`, which uses the internal GPS as the update mechanism. This method returns a `BOOL` value, indicating whether or not positioning was successfully started.



While position updates are being received, the application can retrieve the current position of the client device through the `NMAPositioningManager currentPosition` property. This current position is equal to the `rawPosition` property. `rawPosition` is a position value from the current data source that has not been modified by the HERE SDK engine. If the positioning manager is not active, or it has an invalid position, then the `currentPosition` method returns `nil`.

- **Note:** Map matching is disabled by default. It can be enabled automatically through the use of any HERE SDK feature which requires map matching, such as navigation, or it can be manually enabled by setting `mapMatchingEnabled` to `YES`. When map matching is disabled, `mapMatchedPosition` returns `nil`, and `currentPosition` returns the raw position.

When the application no longer requires position updates, it should notify the `NMAPositioningManager` by calling `stopPositioning`. Position updates are then stopped, provided that no other SDK services (such as `NMAPositionIndicator`) that require position updates are in use.

NMAPositioningManager Notifications

The `NMAPositioningManager` notifications can be used to track position updates of a client device as determined by its positioning mechanism (for example, its GPS). To register or unregister for these notifications, use the following methods:

```
[[NSNotificationCenter defaultCenter] addObserver:self
    selector:@selector(methodName)
    name:NMAPositioningManagerDidUpdatePositionNotification
    object:[NMAPositioningManager sharedPositioningManager]];
```

```
[[NSNotificationCenter defaultCenter] removeObserver:self
    name:NMAPositioningManagerDidUpdatePositionNotification
    object:[NMAPositioningManager sharedPositioningManager]];
```

Applications can register for two types of notifications:

- `NMAPositioningManagerDidUpdatePositionNotification`
- `NMAPositioningManagerDidLosePositionNotification`

- **Note:** `NSNotificationCenter` does not limit how many times an object can register to the same notification. You should be careful not to register the same object more than once to a notification. Otherwise, the object receives duplicate notifications.

The following is an example of registering and handling these notifications in a `UIViewController`:

```
// Start positioning and register for position update notifications
- (void)viewDidLoad
{
    ...
    if ([[NMAPositioningManager sharedPositioningManager] startPositioning]) {
        // Register to positioning manager notifications
        [[NSNotificationCenter defaultCenter] addObserver:self
            selector:@selector(positionDidUpdate) name:NMAPositioningManagerDidUpdatePositionNotification
            object:[NMAPositioningManager sharedPositioningManager]];

        [[NSNotificationCenter defaultCenter] addObserver:self
            selector:@selector(didLosePosition) name: NMAPositioningManagerDidLosePositionNotification
            object:[NMAPositioningManager sharedPositioningManager]];
    }
}
```

```
...
}
// Handle NMAPositioningManagerDidUpdatePositionNotification
- (void)positionDidUpdate
{
    NMAGeoPosition *position = [[NMAPositioningManager sharedPositioningManager] currentPosition];
    [_mapView setGeoCenter:position.coordinates
        withAnimation:NMAMapAnimationLinear];
}
// Handle NMAPositioningManagerDidLosePositionNotification
- (void)didLosePosition
{
    ...
}
```

In order to avoid unnecessary position updates while the application is in the background, you can stop positioning and restart it when the application returns to the foreground using `UIApplicationDelegate` protocol callbacks.

The following code snippet demonstrates how to stop positioning and unregister from the notifications:

```
- (void)viewWillDisappear:(BOOL)animated
{
    [[NMAPositioningManager sharedPositioningManager] stopPositioning];
    [[NSNotificationCenter defaultCenter] removeObserver:self
        name:NMAPositioningManagerDidUpdatePositionNotification
        object:[NMAPositioningManager sharedPositioningManager]];
    [[NSNotificationCenter defaultCenter] removeObserver:self
        name:NMAPositioningManagerDidLosePositionNotification
        object:[NMAPositioningManager sharedPositioningManager]];
}
```

Note: If you enable background location updates in the Xcode project, then `NMAPositioningManager` provides position updates even when your application is no longer in the foreground. To avoid this behavior, you can either stop the positioning manager before the app goes into the background, or you can disable this setting in Xcode.

Creating Position Logs

You can also use the HERE SDK to create GPX logs that can be replayed by `NMALoggedPositionSource`. To do this, set the `logType` property in `NMAPositionManager` to the value `NMAPositionLogTypeDataSource`, indicating that the position received from the current data source should be logged. GPX logs are created in the `Documents` folder of your application. To disable position logging, set `NMAPositionLogType` to `NMAPositionLogTypeNone`.

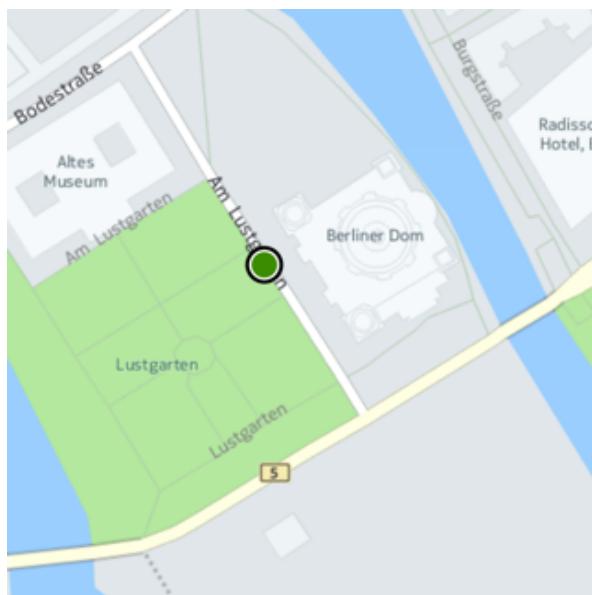
Note: This feature is only intended for debugging purposes. Do not use Position Logging in a production application.

NMAPositionIndicator

The `NMAPositionIndicator` class provides a convenient way to add a map object that marks the user's current location as reported by the `NMAPositioningManager`. The position indicator is rendered as a circular object within a translucent circle, the diameter of which illustrates the accuracy of the indicated

position. The types of map objects can be used to customize `NMAPositionIndicator` are `NMAMapMarker` and `NMAMapCircle`.

Figure 20: An `NMAPositionIndicator`



Each `NMAMapView` instance has an `NMAPositionIndicator` instance which can be accessed from the `NMAMapView positionIndicator` property.

You can customize the accuracy circle's color and whether it is visible by using the `accuracyIndicatorColor` and `accuracyIndicatorVisible` properties.

```
// Display position indicator  
mapView.positionIndicator.visible = YES;
```

Note: Setting `NMAPositionIndicator` to visible automatically enables `NMAPositioningManager` updates.

For the position indicator to stay in the center of the map and illustrate real-time updates of the device position, it is necessary to update the map's center whenever a new location update is received. Please note that frequently redrawing the map in this manner consumes device battery life. You should be aware of battery power implications while performing real-time updates. The following code can be used to update the map location when a position update is received:

```
- (void)positionDidUpdate  
{  
    NMAGeoPosition *position = [[NMAPositioningManager sharedPositioningManager] currentPosition];  
    [_MapView setGeoCenter:position.coordinates  
              withAnimation:NMAMapAnimationLinear];  
}
```

For more information about the classes introduced and demonstrated in this section, refer to the [API reference documentation](#).

Directions

Car and Pedestrian Routing

The HERE iOS SDK supports route calculation with multiple waypoints, optimized for walking or driving.

A route describes a path between at least two waypoints, the starting point and the destination, with optional intermediate waypoints in between. Applications can provide route information to users in two ways:

- A line rendered on a map that displays a connecting path between all waypoints
- Turn-by-turn directions in text format

NMARouteManager

The `NMARouteManager` class is responsible for calculating an `NMARoute` using a list of stops and an `NMARoutingMode`. It also provides an `NMARouteManagerDelegate` protocol for monitoring calculation progress and triggering appropriate callback methods upon completion. To calculate a route, the application needs to call the `calculateRouteWithStops:routingMode:` method. `NMARouteManager` only supports one routing request at a time. Making another request before completion of the current request is not supported.

■ **Note:** `NMARouteManager` supports routes with a maximum of 32 waypoints.

NMARoutingMode

The `NMARoutingMode` class is a model of the parameters required to calculate an `NMARoute`, such as:

- `routingType` - the routing type, such as Fastest or Shortest
- `transportMode` - the mode of transportation
- `routingOptions` - the routing options (represented by the `NMARoutingOption` enums) that are applicable for this route
- `departureTime` - the departure time for the route
- `resultLimit` - the maximum number of alternate routes to calculate (the actual number of results may be less than this limit)

■ **Note:** The HERE SDK allows for more than one route to be returned from a route calculation between two waypoints. You can use the `NMARoutingMode` class to set the desired number of routes, and the HERE SDK then returns different routes according to this limit. Note that the first element of the returned array is the best result based on the routing options, and the rest of the returned routes are not listed in any specific order.

NMARoute

The `NMARoute` class represents a distinct calculated path connecting two or more waypoints, and consists of a list of maneuvers and route links. A call to the `calculateRouteWithStops:routingMode:` method of `NMARouteManager` triggers a route calculation, while the `NMARouteManagerDelegate` implements callback methods to monitor the operation and process the resulting `NMARoute` objects.

An `NMARoute` object contains route information that can be accessed by calling one or more of the following methods:

- `routingMode` - the NMARoutingMode for the route
- `waypoints` - the array of all waypoints for the route
- `start` - the starting waypoint for the route
- `destination` - the destination waypoint for the route
- `maneuvers` - the array of maneuvers for the route
- `length` - the length of the route, in meters
- `tta` - the NMARouteTta indicating the estimated time to arrival
- `boundingBox` - gets the smallest NMAGeoBoundingBox that contains the entire route
- `routeGeometry` - gets the array of all NMAGeoCoordinates along the route
- `mapPolyline` - gets the NMAMapPolyline representation of the route

NMARouteTta

The NMARouteTta ("time-to-arrival") class provides useful information about the route, such as duration and route details that impact travel duration, including car pool restrictions, turn restrictions, and blocked roads. For example, to retrieve a duration for a calculated route, use the `tta` property. For example,

```
NSTimeInterval duration = route.tta.duration;
```

You can also retrieve the duration for a subleg from a route by using the `ttaForSubleg:` method. For example,

```
if (myRoute.sublegCount > 0)
{
    NSTimeInterval duration = [myRoute ttaForSubleg:0].duration;
}
```

NMAManeuver

The NMAManeuver class represents the action required to go from one segment to the next within a calculated NMARoute. Each NMAManeuver object provides information such as:

- location of the maneuver
- action required to complete the maneuver
- distance between maneuvers
- current road
- next road
- estimated times of the maneuver
- highway signpost (if any) indicating entrance, exit, or merge information
- a list of route elements representing portions of this maneuver

For more information, please consult the API Reference.

NMARouteElement and NMARoadElement

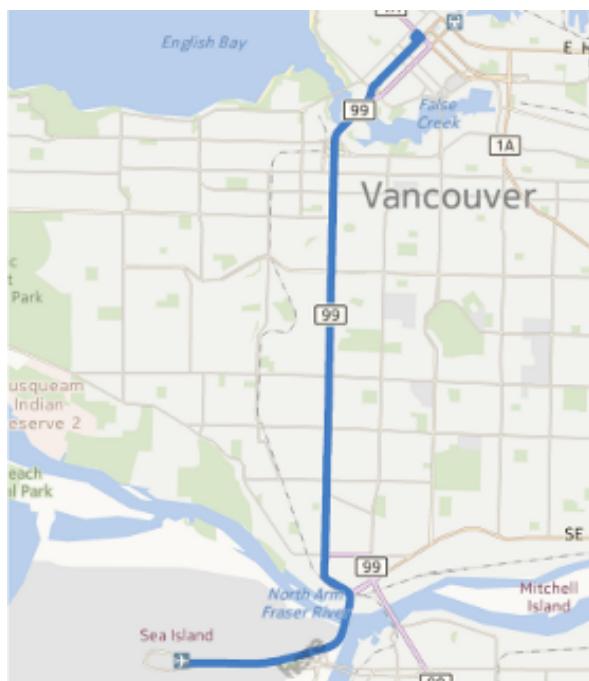
NMARouteElement and NMARoadElement represent portions within a maneuver. For example, a maneuver may ask the driver to turn left and then remain on a street, but this street may be comprised of multiple

sections, including a tunnel, a dirt road, and a toll road. In this situation, the maneuver contains multiple `NMARouteElement` objects, with each element containing a `NMARoadElement` property that can provide your application with information about the individual section of the road.

NMAMapRoute

The `NMAMapRoute` class is a type of `NMAMapObject` that displays a calculated route on a map. Typically, an application creates a `NMAMapRoute` after a route calculation, and add the `NMAMapRoute` to the map by calling `NMAMapView addMapObject:`.

Figure 21: An NMAMapRoute added to an NMAMapView



For example, if you want to render a route that connects two waypoints (start and destination), you can add the following application logic:

1. Adopt `NMARouteManagerDelegate` protocol and create a `NMARouteManager`

```
@interface ClassName : NSObject <NMARouteManagerDelegate>
{
    // Setup your class
}

(void)setup
{
    // Create a NMARouteManager.
    NMARouteManager* routeManager = [NMARouteManager sharedRouteManager];

    // Setup delegate
    [routeManager setDelegate:self];
}
```

2. Create an `NSMutableArray` and add two `NMAGeoCoordinates` stops

```
NSMutableArray* stops = [[NSMutableArray alloc] initWithCapacity:4];
NMAGeoCoordinates* geoCoord1 = [[NMAGeoCoordinates alloc]
initWithLatitude:49.1966286 longitude:-123.0053635];
NMAGeoCoordinates* geoCoord2 = [[NMAGeoCoordinates alloc]
```

```
initWithLatitude:49.1947289 longitude:-123.1762924];
[stops addObject:geoCoord1];
[stops addObject:geoCoord2];
```

3. Create an `NMARoutingMode` and set its `NMATransportMode`, `NMARoutingType` and `NMARoutingOption` values

```
NMARoutingMode* routingMode = [[NMARoutingMode alloc]
    initWithRoutingType:NMARoutingTypeFastest
    transportMode:NMATransportModeCar
    routingOptions:0];
```

4. Calculate the route

```
[routeManager calculateRouteWithStops:stops routingMode:routingMode];
```

5. To receive the results of the route calculation, implement the `NMARouteManagerDelegate` protocol method `routeManager:didCalculateRoutes:withError:violatedOptions:` in your delegate class.

 **Note:** Routes are returned even if you receive the `NMARouteManagerErrorViolatesOptions` error. It is up to you to handle these route results that violate routing options.

```
-(void) routeManager: (NMARouteManager*)routeManager
    didCalculateRoutes:(NSArray*)routes
    withError:(NMARouteManagerError)error
    violatedOptions:(NSArray*)violatedOptions
{
    // If the route was calculated successfully
    if (!error && routes && routes.count > 0)
    {
        NMARoute* route = [routes objectAtIndex:0];
        // Render the route on the map
        NMAMapRoute *mapRoute = [NMAMapRoute mapRouteWithRoute:route];
        [mapView addMapObject:mapRoute];
    }
    else if (error)
    {
        // Display a message indicating route calculation failure
    }
}
```

Routing-related enumerations

Route calculations make use of HERE SDK enumerations that include:

- The `NMARoutingType` enum - represents values describing different routing types, such as `NMARoutingTypeFastest` or `NMARoutingTypeShortest`
- The `NMATransportMode` enum - represents values describing different transport modes, such as `NMATransportModeCar` or `NMATransportModePedestrian`
- The `NMARoutingOption` enum - represents values describing special conditions for route calculation, such as `NMARoutingOptionAvoidBoatFerry` or `NMARoutingOptionAvoidTollRoad`. Values from this enum are also returned after a route calculation to indicate the options that a route result violates.
- The `NMARouteViolatedOption` enum - represents values describing special conditions that are violated in a route calculation in addition to `NMARoutingOption`. This enum contains values for blocked roads and turn restrictions. For example, after specifying a route calculation that avoids tolls and ferries, you may get a `NMARoute` with the `NMARouteViolatedOptionBlockedRoad` violated

option. This indicates that although a route was found, this route goes through at least one blocked road – violating a condition of your route request.

- The `NMARouteManagerError` enum - represents values describing possible route calculation errors, such as `NMARouteManagerErrorNone` or `NMARouteManagerErrorViolatesOptions`

Search

Geocoding and Reverse Geocoding

Geocoding and reverse geocoding APIs from the HERE iOS SDK allow application developers to offer search functionality for requesting `NMAPlaceLocation` information. Geocoding APIs resolve a free-formatted text query to an `NMAGeoCoordinates`, while reverse geocoding APIs resolve from an `NMAGeoCoordinates` to geographic data, such as `NMAAddress`.

`NMAAddress` provides textual address information including house number, street name, city, country, district and more. It encompasses everything about an address or a point on the map. The `NMAPlaceLocation` class represents an area on the map where additional attributes can be retrieved. These additional attributes include `NMAAddress`, unique identifier, label, location, access locations, and `NMAGeoBoundingBox` for the location.

The NMAGEocoder Interface

The `NMAGEocoder` interface represents a factory class used to instantiate location search requests. Two types of requests are available: `NMAGEocodeRequest` and `NMAResponseGeocodeRequest`.

The NMAGEocodeRequest Interface

The `NMAGEocodeRequest` interface represents an extended `NMARequest`. The `NMAGEocodeRequest` can be created using a combination of a search area and a free text query string. This is known as a "one-box" request. It returns `NMAPlaceLocation` results according to the specified search area and text query. You can specify a search area by providing a `NMAGeoBoundingBox` or a location with a search radius.

The following shows the methods used to create one-box requests:

```
NMAGEocodeRequest* request = [[NMAGEocoder sharedGeocoder] createGeocodeRequestWithQuery:string  
    searchArea:geoBoundingBox  
    locationContext:geoCoordinates];
```

```
NMAGEocodeRequest* request = [[NMAGEocoder sharedGeocoder] createGeocodeRequestWithQuery:string  
    searchRadius:radius  
    locationContext:geoCoordinates];
```

The preceding methods return an `NMAGEocodeRequest` object. To perform the request, call its `startWithListener:` method. The parameter of this method is an object which receives the request results; the object must implement the `NMAResultListener` protocol. Once a request is invoked, it can be canceled using the `cancel` method of `NMARequest`, which returns a `BOOL` value indicating whether the

result was canceled successfully. If the NMAGeocodeRequest is successful, a list of NMAGeocodeResult objects is returned to the listener.

The following code example demonstrates how to use a NMAGeocodeRequest:

```
// Implementation of NMAResultListener
@interface NMAGeocodeTest : NSObject<NMAResultListener> {
}
@end
@implementation NMAGeocodeTest

// NMAResultListener protocol callback implementation
- (void)request:(NMAResponse*)request
didCompleteWithData:(id)data
error:(NSError*)error
{
    if ( ( [request isKindOfClass:[NMAGeocodeRequest class]]) &&
        ( error.code == NMAResponseErrorNone ) )
    {
        // Process result NSArray of NMAGeocodeResult objects
        [self processResult:(NSMutableArray *)data];
    }
    else
    {
        // Handle error
        ...
    }
}

- (void) startSearch
{
    NMAGeoCoordinates *topLeft =
    [[NMAGeoCoordinates alloc]
     initWithLatitude:52.537413 longitude:13.365641];
    NMAGeoCoordinates *bottomRight =
    [[NMAGeoCoordinates alloc]
     initWithLatitude:52.522428 longitude:13.39345];
    NMAGeoBoundingBox *boundingBox =
    [NMAGeoBoundingBox
     geoBoundingBoxWithTopLeft:topLeft bottomRight:bottomRight];

    NMAGeocodeRequest* request = [[NMAGeocoder sharedGeocoder]
        createGeocodeRequestWithQuery:@"/100 INVALIDENSTRASSE"
        searchArea:boundingBox
        locationContext:nil];

    // limit the number of results to 10
    request.collectionSize = 10;

    NSError* error = [request startWithListener:self];
    if (error.code != NMAResponseErrorNone)
    {
        // Handle request error
        ...
    }
}
@end
```

The NMAResponseGeocodeRequest interface

The NMAResponseGeocodeRequest interface represents an extended NMAResponse used to retrieve NMAPlaceLocation data. The request is created using an NMAGeoCoordinates as shown below:

```
NMAGeocodeRequest* request = [[NMAGeocoder sharedGeocoder]
    createReverseGeocodeRequestWithGeoCoordinates:geoCoordinates];
```

The above method returns an NMAResponseGeocodeRequest object. Reverse geocode requests are used in the same way as regular geocode requests (described in the previous section), but the results are returned as an array of NMAResponseGeocodeResult objects.

The following example shows how to create and use an NMAResponseGeocodeRequest:

```
// Implementation of NMAResultListener
@interface NMAResponseGeocodeTest : NSObject<NMAResultListener> {
}
@end
@implementation NMAResponseGeocodeTest

// NMAResultListener protocol callback implementation
- (void)request:(NMAResponse*)request
didCompleteWithData:(id)data
error:(NSError*)error
{
    if ( ( [request isKindOfClass:[NMAResponseGeocodeRequest class]] ) &&
        ( error.code == NMAResponseErrorNone ) )
    {
        // Process result NSArray of NMAResponseGeocodeResult objects
        [self processResult:(NSMutableArray *)data];
    }
    else
    {
        // Handle error
        ...
    }
}

- (void) startSearch
{
    // Instantiate an Address object
    NMAGeoCoordinates* vancouver = [[NMAGeoCoordinates alloc] initWithLatitude:49.2849
        longitude:-123.1252];

    NMAResponseGeocodeRequest* request = [[NMAGeocoder sharedGeocoder]
        createReverseGeocodeRequestWithGeoCoordinates:vancouver];

    NSError* error = [request startWithListener:self];
    if (error.code != NMAResponseErrorNone)
    {
        // Handle request error
        ...
    }
}

@end
```

For more information about the APIs introduced and demonstrated in this section, refer to the [API Reference documentation](#).

Search and Discovery

The HERE iOS SDK includes a Places API which provides functionality to search, discover, and obtain more information about places in the real world.

HERE Places helps to determine whether a business meets your needs through reviews and photos from real people. In addition to basic information such as opening hours and contact details, HERE Places can also include editorials from popular guides to help identify the best Places for you to visit.

Steps for Performing a Search

1. Implement the `NMAResultListener` protocol to handle the completion of the search
 2. Create a request using the `NMAPlaces` factory
 3. Invoke the request by calling `NMARequest startWithListener:`
 4. The `NMAResultListener request:didCompleteWithData:error:` callback is triggered when the request is finished
- **Note:** Applications that use the Places API must honor the following prescribed workflow:
1. Search
 2. Request for Details
 3. Perform Actions

Do not preload results that are linked from a response in order to improve performance, as doing so violates HERE's guidelines. For more information about usage restrictions, consult the [API Implementation Check List](#) section in the Places RESTful API documentation.

Discovery Requests

The HERE Places API supports the following discovery requests. Requests are created through factory methods in `NMAPlaces`.

Request	<code>NMAPlaces</code> method	Purpose
Search	<code>createSearchRequestWithLocation: query:</code>	Finds places that match user-provided search terms.
Explore	<code>createExploreRequestWithLocation:searchArea:filters:</code>	Finds interesting places nearby, or in the map viewport, sorted by popularity. Use this type of request if you are trying to answer the question "What are the interesting places near here?" The results may be optionally restricted to a given set of categories, which acts as a filter in terms of what places get returned.

Request	NMAPPlaces method	Purpose
Here	createHereRequestWithLocation:filters:	<p>Helps users identify places at the given location by finding places of interest near a given point, sorted by distance. Use this type of request if you are trying to answer the question "What is near this location?" or "Where am I?" You can use this endpoint to implement features like "check-in" (by identifying places at the user's current position) or "tap to get more information about this place of interest".</p> <p>Note: Normally, the closest known places are returned with the Here Discovery request, but if the uncertainty in the given position is high, then some nearer places are excluded from the result in favor of more popular places in the area of uncertainty.</p>

The following code example demonstrates how to perform a search discovery request.

You need to implement the `NMAResultListener` protocol by implementing the `request:didCompleteWithData:error` callback method, and also initialize the request by calling `request startWithListener::`:

```
// Sample Search request listener
@interface NMASearchTest : NSObject<NMAResultListener> {
    NMADiscoveryPage* _result;
}
@end
@implementation NMASearchTest

// NMAResultListener protocol callback implementation
- (void)request:(NMAResponse*)request
    didCompleteWithData:(id)data
    error:(NSError*)error
{
    if ( ( [request isKindOfClass:[NMADiscoveryRequest class]]) &&
        ( error.code == NMAResponseErrorNone ) )
    {
        // Process result NMADiscoveryPage objects
        _result = (NMADiscoveryPage*) data;
    }
    else
    {
        // Handle error
        ...
    }
}
- (void) startSearch
{
    // Create a request to search for restaurants in Vancouver
    NMAGeoCoordinates* vancouver =
    [[NMAGeoCoordinates alloc] initWithLatitude:48.263392
                                    longitude:-123.12203];
```

```
NMADiscoveryRequest* request =
[[NMAPlaces sharedPlaces] createSearchRequestWithLocation:vancouver
query:@"restaurant"];

// optionally, you can set a bounding box to limit the results within it.
NMAGeoCoordinates *boundingTopLeftCoords = [[NMAGeoCoordinates alloc] initWithLatitude:49.277484
longitude:-123.133693];
NMAGeoCoordinates *boundingBottomRightCoords = [[NMAGeoCoordinates alloc]
initWithLatitude:49.257209 longitude:-123.11275];
NMAGeoBoundingBox *bounding = [[NMAGeoBoundingBox alloc] initWithTopLeft:boundingTopLeftCoords
bottomRight:boundingBottomRightCoords];

request.viewport = bounding;

// limit number of items in each result page to 10
request.collectionSize = 10;

NSError* error = [request startWithListener:self];
if (error.code != NMAResponseErrorNone)
{
    // Handle request error
    ...
}
@end
```

To ensure that your application get the best search results, you can set a location context to your search request by setting a bounding box to the `viewport` property. In the previous example, you can also replace the `NMAGeoBoundingBox` with the `viewport` from `NMAMapView`.

The result of a search or explore discovery request is an `NMADiscoveryPage`. The `NMADiscoveryPage` represents a paginated collection of items from which the following can be retrieved:

- Next page request - an `NMADiscoveryRequest` used to retrieve additional pages of search items
- Items for the current page - an `NSArray` of `NMALink`, either `NMAPlaceLink` or `NMADiscoveryLink`

If `NMADiscoveryPage.nextPageRequest` is `nil`, no additional results are available.

The following is an example:

```
...
@interface NMANextPageTest : NSObject<NMAResultListener>
    NMADiscoveryPage* _page; // valid NMADiscoveryPage instance
}
@implementation NMANextPageTest
- (void)onNextPageAction
{
    NSError* error = [_page.nextPageRequest startWithListener:self];
    if (error.code == NMAResponseErrorNone)
    {
        // More data is available
    }
}

// NMAResultListener protocol callback implementation
- (void)request:(NMADiscoveryRequest*)request
didCompleteWithData:(id)data
error:(NSError*)error
{
    if ( ( [request isKindOfClass:[NMADiscoveryRequest class]] ) &&
        ( error.code == NMAResponseErrorNone ) )
```

```
{  
    // Process NMADiscoveryPage objects  
}  
else  
{  
    // Handle error  
    ...  
}  
...  
}@end
```

The `NMADiscoveryPage discoveryResults` property contains an array of `NMALink` objects. The items are actually a collection of `NMALink` subclasses:

- `NMAPlaceLink` - Represents discovery information about a `NMAPlace`. The `NMAPlaceLink` contains a brief summary about a place. Details about a place are available from the `NMAPlace` that the `NMAPlaceLink` references.
- `NMADiscoveryLink` - Represents a discovery-related API link used to retrieve additional `NMADiscoveryPage` instances. This type of `NMALink` can be a result item in an Explore or Here type of search. The `NMADiscoveryLink` references refine discovery requests resulting in more specific results. For example, the `NMADiscoveryLink` may link to a discovery request to search for 'Eat & Drink', 'Going Out', 'Accommodation', and so on.

It is recommended that each type of `NMADiscoveryPage` be checked before it is used. In the following example, it is shown how an `NMAPlace` is retrieved through a `NMAPlaceLink`:

```
@interface NMASearchTest : NSObject<NMAResultListener> {  
    NMADiscoveryPage* _result;  
}  
@end  
@implementation NMASearchTest  
// Retrieve the place details when the user selects a displayed PlaceLink.  
- (void)onPlaceLinkSelected:(NMAPlaceLink*)placeLink  
{  
    NSError* error = [[placeLink detailsRequest] startWithListener:self];  
    if ( error.code == NMAResultErrorNone )  
    {  
        // More data will available.  
        ...  
    }  
  
// NMAResultListener protocol callback implementation  
- (void)request:(NMAResult*)request  
didCompleteWithData:(id)data  
error:(NSError*)error  
{  
    if ( ( [request isKindOfClass:[NMADiscoveryRequest class]] ) &&  
        ( error.code == NMAResultErrorNone ) )  
    {  
        _result = (NMADiscoveryPage*) data;  
        NSArray* discoveryResult = _result.discoveryResults;  
  
        for ( NMALink* link in discoveryResult )  
        {  
            if ( link isKindOfClass:[NMADiscoveryLink class] )  
            {  
                NMADiscoveryLink* discoveryLink = (NMADiscoveryLink*) link;  
  
                // NMADiscoveryLink can also be presented to the user.  
            }  
        }  
    }  
}
```

```
// When a NMADiscoveryLink is selected, another search request should be
// performed to retrieve results for a specific category.
...
}
else if ( link isKindOfClass:[NMAPlaceLink class] )
{
    NMAPlaceLink* placeLink = (NMAPlaceLink*) link;

    // NMAPlaceLink should be presented to the user, so the link can be
    // selected in order to retrieve additional details about a place
    // of interest.
    ...
}
}
else if ( ([request isKindOfClass:[NMAPlaceRequest class]] ) &&
          ( error.code == NMAResponseErrorNone ) )
{
    NMAPlace* place = (NMAPlace*)data;
    // Access to additional details about a place of interest.
}
else
{
    // Handle error
    ...
}
}
@end
```

The NMAPlace Class

The **NMAPlace** class represents a detailed set of data about a physical place, acting as a container for various attributes, collections of media about a place, and key-value pairs of related places. An **NMAPlace** object can belong to a specific **NMACategory**, and has attributes such as:

- A unique identifier (ID)
- A name
- An **NMAPlaceLocation** object representing the physical location of the place. **NMAPlaceLocation** also contains a street address and a list of the geocoordinate positions to access this place.
- An array of **NMACategory** objects that link to the categories assigned to the place
- An **NMALink** object containing a link to the origin of supplied information, typically a website of the supplier
- An **NSString** representing a URL to an icon (optional)
- Optional information, such as related places, user ratings, reviews, and other editorial media

Category Filters

A place of interest can be associated with categories such as museum, restaurant, and coffee shop. While creating an Explore or Here discovery request, you can choose to provide category filters in order to get a more specific set of results. For example, you may want to search for sushi restaurants near the Vancouver city hall.

To get a list of categories, call the **topLevelCategories** method in **NMAPlaces**. From this list of categories, you can then retrieve one or more level of sub-categories. For example, "Bars/Pubs" under the "Restaurant" category. Once you have the categories, you can then create an **NMACategoryFilter** object

and call the `addCategoryFilterFromUniqueId` method. Note that each `NMACategoryFilter` object can represent multiple categories.

```
NSArray* categories = [[NMAPlaces sharedPlaces] topLevelCategories];

for (id category in categories)
{
    if (category.uniqueId == "restaurant")
    {
        NMACategory* restCategory = category;
        NMAGeoCoordinates* vancouver = [[NMAGeoCoordinates alloc]
            initWithLatitude:47.592229
            longitude:-122.315147];

        NMACategoryFilter *categoryFilter = [NMACategoryFilter new];
        [categoryFilter addCategoryFilterFromUniqueId:restCategory.uniqueId];

        NMADiscoveryRequest* request = [ [NMAPlaces sharedPlaces]
            createHereRequestWithLocation:vancouver
            filters:categoryFilter];
        //...
    }
}
```

Text AutoSuggestion Requests

The HERE Places Search API also supports text autosuggestion requests. This type of request is used for retrieving a list of suggested search terms (`NMAAutoSuggestTypeQuery`), instant results (`NMAAutoSuggestTypePlace`), and refined search links (`NMAAutoSuggestTypeSearch`) that are related to a specified location context and a partial search term. For example, if you make a request with the String "rest" in Berlin, the results then contain search terms such as "Restaurant", "Rest area", and "Restorf, Höhbeck, Germany".

To use text autosuggestions, implement a listener to handle a list of `NMAAutoSuggest` and call `createAutoSuggestionRequestWithLocation:partialTerm:` as follows:

```
// Sample Search request listener
@interface NMATextAutoSuggestionSearchTest : NSObject<NMAResultListener> {

}

@end

@implementation NMATextAutoSuggestionSearchTest

// NMAResultListener protocol callback implementation
- (void)request:(NMAResponse*)request
didCompleteWithData:(id)data
error:(NSError*)error
{
    if ( ( [request isKindOfClass:[NMAAutoSuggestionRequest class]] ) &&
        ( error.code == NMAResponseErrorNone ) )
    {
        // Results are held in an array of NMAAutoSuggest objects
        // You can then check the subclass type using the NMAAutoSuggest.type property
        NSArray* textAutoSuggestionResult = (NSArray*) data;
    }
    else
    {
        // Handle error
        ...
    }
}
```

```
}

- (void) startSearch
{
    NMAGeoCoordinates* vancouver =
        [[NMAGeoCoordinates alloc] initWithLatitude:47.592229
                                         longitude:-122.315147];

    NMAAutoSuggestionRequest* request =
        [[NMAPlaces sharedPlaces] createAutoSuggestionRequestWithLocation:vancouver
                                         partialTerm:@"rest"];

    // limit number of items in each result page to 10
    request.collectionSize = 10;

    NSError* error = [request startWithListener:self];
    if (error.code != NMAResponseErrorNone)
    {
        // Handle request error
        ...
    }
}
@end
```

You can retrieve the results of `NMAAutoSuggestionRequest` by first checking the `NMAAutoSuggest` object type, as shown in the following example. If the object is `NMAAutoSuggestTypeSearch`, it contains additional paginated results through its `NMADiscoveryRequest` object. If the object is `NMAAutoSuggestTypePlace`, you can request for more details through its `NMAPlaceRequest`. `NMAAutoSuggestTypeQuery` contains additional results through the `NMAAutoSuggestionRequest` object.

```
+ (BOOL)checkAutoSuggestResults:(NSArray*)array
{
    for (id item in array) {
        NMAAutoSuggestType type = ((NMAAutoSuggest*)item).type;
        NSString *typeString;
        switch (type){
            caseNMAAutoSuggestTypePlace:
            {
                typeString = @Place";
            }
            break;
            caseNMAAutoSuggestTypeSearch:
            {
                typeString = @Search";
            }
            break;
            caseNMAAutoSuggestTypeQuery:
            {
                typeString = @Query";
            }
            break;
        }

        NSLog(@"Type = %@", typeString);

        NMAAutoSuggest* suggestItem = (NMAAutoSuggest*)item;
        //Retrieve information such as suggestItem.title
        if (type == NMAAutoSuggestTypePlace) {

            NMAAutoSuggestPlace* suggestPlace = (NMAAutoSuggestPlace*)item;
            //Retrieve information such as suggestPlace.vicinityDescription
```

```
NMAPlaceRequest* detailsRequest = suggestPlace.placeDetailsRequest;  
// Get NMAPlaceResult by calling detailsRequest startWithListener:  
// ...  
  
{ else if (type == NMAAutoSuggestTypeSearch) {  
  
    NMAAutoSuggestSearch* suggestSearch = (NMAAutoSuggestPlace*)item;  
  
    //Retrieve information such as suggestSearch.position  
  
    NMADiscoveryPage* discoveryPage;  
    NMADiscoveryRequest* discoveryRequest = suggestSearch.suggestedSearchRequest;  
  
    // Get discoveryPage by calling [discoveryRequest startWithListener:]  
    // ...  
}  
else if (type == NMAAutoSuggestTypeQuery) {  
  
    NMAAutoSuggestionRequest* autoSuggestionRequest =  
    ((NMAAutoSuggestQuery*)item).autoSuggestionRequest;  
  
    // Call autoSuggestionRequest startWithBlock:^(NMAResponse, id, NSError) {}  
    // ...  
}  
}  
return YES;  
}
```

Offline Search

Force Online or Offline

You can launch online or offline searches without changing the device or HERE SDK connectivity by using the `connectivity` property on an `NMAResult` instance. This property is applicable to all `NMAResult` subclasses, except `NMAAutoSuggestionRequest`, which can only be used online.

- **Note:** In HERE SDK v3.4, we have updated the behavior of the routing connectivity modes (`NMAResultConnectivity`) to be more consistent with other parts of the SDK.

The `connectivity` property can be set to three possible values:

- If you launch a request using the `NMAResultConnectivityDefault` connectivity mode, the request is performed according to the `NMAApplicationContext` connectivity setting. If the device is offline while `NMAApplicationContext` is set to online mode, the request fails.
- If you launch a request using the `NMAResultConnectivityOnline` connectivity mode, an online request is performed, regardless of the `NMAApplicationContext` connectivity setting.
- If you launch a request using the `NMAResultConnectivityOffline` connectivity mode, an offline request is performed, regardless of the `NMAApplicationContext` connectivity setting.

In all cases, if the request fails, no fallback action is automatically performed.

To ensure that the connectivity mode is applied, set the `connectivity` property before executing an `NMAResult`.

Depending on your search request and the connectivity mode, you can get a few different errors:

- If an `NMAResponseConnectivityOnline` search request fails due to connection issues, HERE SDK returns the `NMAResponseErrorUnknown` error code.
- If an `NMAResponseConnectivityOnline` Geocoding or Reverse Geocoding request fails due to connection issues, HERE SDK returns the `NMAResponseErrorNetworkCommunication` error code.
- If an `NMAResponseConnectivityOffline` search request fails due to not enough cached data, HERE SDK returns with zero results.
- If you attempt to start an `NMAAutoSuggestionRequest` with the `NMAResponseConnectivityOffline` connectivity mode, HERE SDK returns the `NMAResponseErrorNotSupported` error code, since auto suggestions are only supported online.

Chapter 4

Supplemental Information

Topics:

- [*Create a Simple HERE SDK A...*](#)
- [*Swift Support in HERE SDK*](#)
- [*Supported Thread Usage*](#)

This section provides supplemental information for using the HERE iOS SDK.

Create a Simple HERE SDK App Using Swift

This tutorial provides instructions on how to create a simple application using the *Swift programming language*. It is equivalent to the Objective-C tutorial, which is located at *Run the Sample Application* on page 11.

Development tasks for this basic application include:

- Create a new Xcode project
- Add necessary resources and a map view to the project
- Acquire credentials from HERE for accessing map services
- Initialize the map view such that a map instance is created for rendering on the client device

The contents of this guide apply to Xcode 9 and the iOS 11 SDK.

Sample Project in the HERE iOS SDK

A copy of the Xcode project described in this tutorial is available in the `sample-apps` folder in your HERE iOS SDK package. To run the project, double-click on `SwiftHelloMap.xcodeproj` and follow the instructions in the `README.txt` file.

Create a New Single View Application

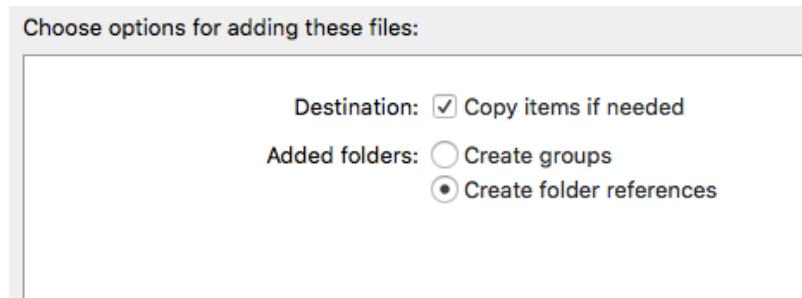
1. From XCode menu, select **File > New > Project** to open the New project dialog (or press Shift + Command + N).
2. Select **iOS > Application > Single View Application** as the application type you want to create. Press Next.
3. In the next dialog, enter your **Product Name** (such as `HelloMap`) and **Organization Identifier** (such as `edu.self`).
4. Choose "Swift" under **Language**, then click **Next**.
5. Navigate to the directory where you want your project to be stored and then select **Create**.
6. The next step is to configure this project to use the HERE SDK.

Configure the Application

1. Extract the HERE iOS SDK archive to somewhere in your local file system.
2. Add the `NMAKit` framework to your Xcode project. To add the `NMAKit` framework to your Xcode project, click on your app target and choose the "General" tab. Find the section called "Embedded Binaries", click the plus (+) sign, and then click the "Add Other" button. From the file dialog box, select the

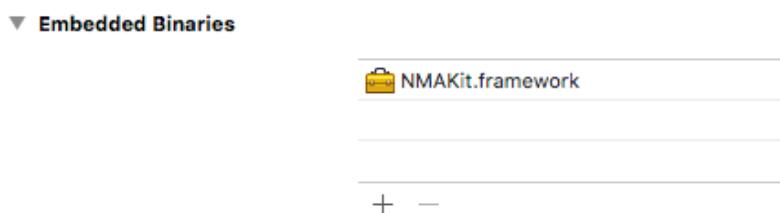
"NMAKit.framework" folder. Ensure that the "Copy items if needed" and "Create folder reference" options are selected, then click **Finish**.

Figure 22: Add File to Target



3. Ensure that `NMAKit.framework` appears in the "Embedded Binaries" and the "Linked Frameworks and Libraries" sections.

Figure 23: Embedded Binaries



4. Run the application. From the Xcode menu bar, select **Product > Run**. Ensure that the project runs in the iOS Simulator without errors.
5. The HERE iOS SDK is now ready for use in your Xcode project. Now that you have your project configured to work with the HERE SDK, try extending the sample application to render a map.

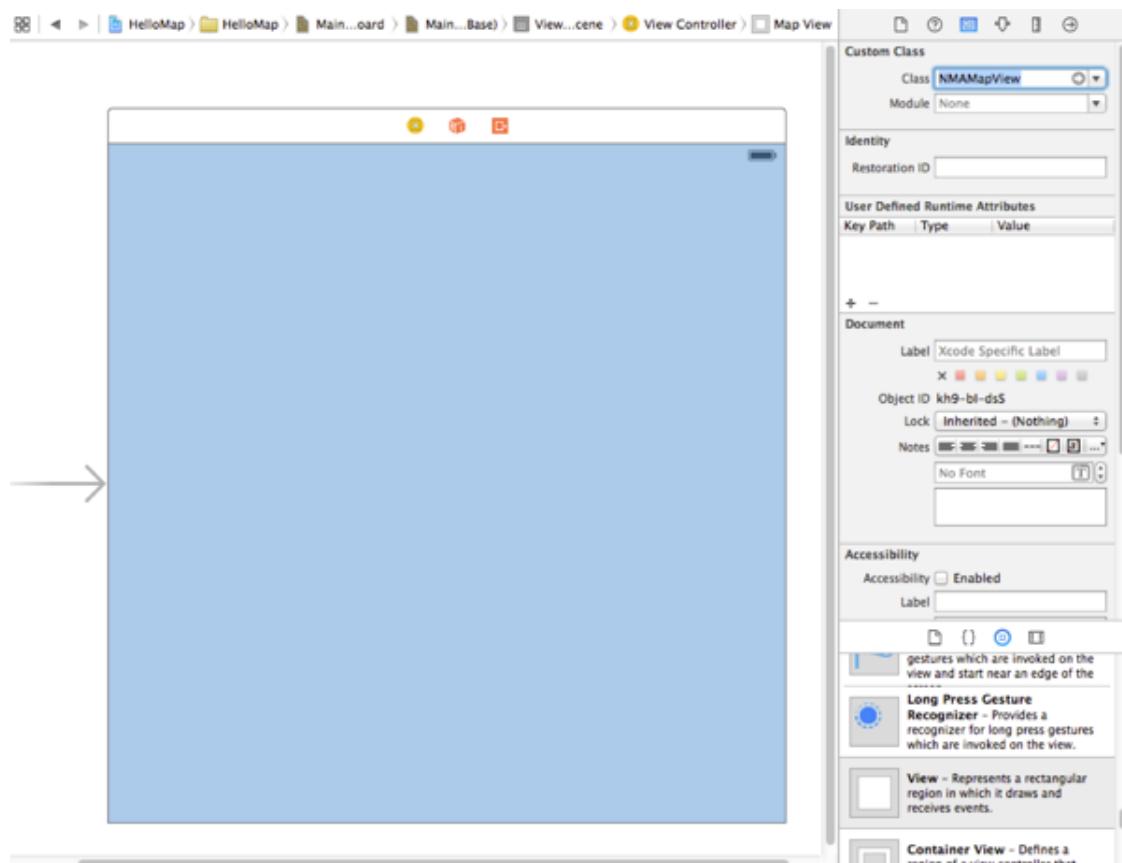
Create the Map View

In this section, we utilize the `NMAMapView` and `NMAGeoCoordinates` classes to render a Map.

1. Create an `NMAMapView`.
 - a. Select `Main.storyboard` in the navigator, then open the Utilities view by pressing the key combination Command + Option + Control + 3. Drag and drop a View object from the Object Library onto the View Controller. If necessary, resize the View so it takes up the entire viewable area.
 - b. In the Interface Builder, click on the created View and then open the Identity Inspector in the Utilities view by pressing the key combination Command + Option + 3. Change the class value from

UIView to NMAMapView and press return. In the Document Outline, you should see that the name of the View has changed from to View to Map View.

Figure 24: MapView



2. Create an outlet to NMAMapView in ViewController.
 - a. Select Main.storyboard in the navigator.
 - b. Press Command + Option + Return to open the Assistant Editor. It should show ViewController.swift.
 - c. Add the following import statement to the top of this file:

```
import NMAKit
```

- d. Hold the Control key on the keyboard and click to drag from the Map View to the interface block in ViewController.swift. You should see a blue line and tooltip which says "Insert Outlet or Outlet Connection". Release the mouse button and a dialog appears, allowing you to create an outlet.

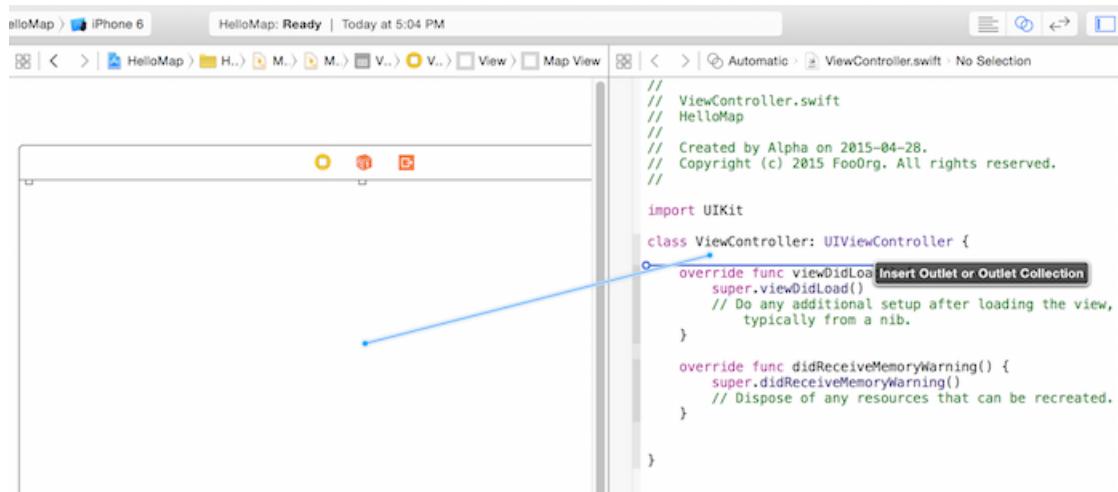
HERE iOS SDK Developer's Guide

► Supplemental Information



- e. Name the outlet *mapView*, keep the other default options and then select Connect.

Figure 25: Create an Outlet



3. Now an outlet to `NMAMapView` is set. The modified file should be as follows:

```
import UIKit
import NMKit

class ViewController: UIViewController {

    @IBOutlet weak var mapView: NMAMapView!

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }
}
```

4. Implement `NMAMapView` setup and lifecycle code by replacing the `viewDidLoad()` function with `viewWillAppear(animated)` and `addMapCircle()`:

```
override func viewWillAppear(_ animated: Bool) {
    super.viewWillAppear(animated)
    mapView.useHighResolutionMap = true
    mapView.zoomLevel = 13.2
    mapView.set(geoCenter: NMAGeoCoordinates(latitude: 49.258867, longitude: -123.008046),
               animation: .linear)
    mapView.copyrightLogoPosition = NMALayoutPosition.bottomCenter
    addMapCircle()
}

func addMapCircle() {
    if mapCircle == nil {
        let coordinates: NMAGeoCoordinates =
            NMAGeoCoordinates(latitude: 49.258867, longitude: -123.008046)
        mapCircle = NMAMapCircle(coordinates: coordinates, radius: 50)
        mapView.add(mapCircle!)
    }
}
```

HERE iOS SDK Developer's Guide

► Supplemental Information



```
}
```

5. Add your HERE application credentials.

- Open the `AppDelegate.swift` file and import `NMAKit` by adding the following import statement to the top of the file.

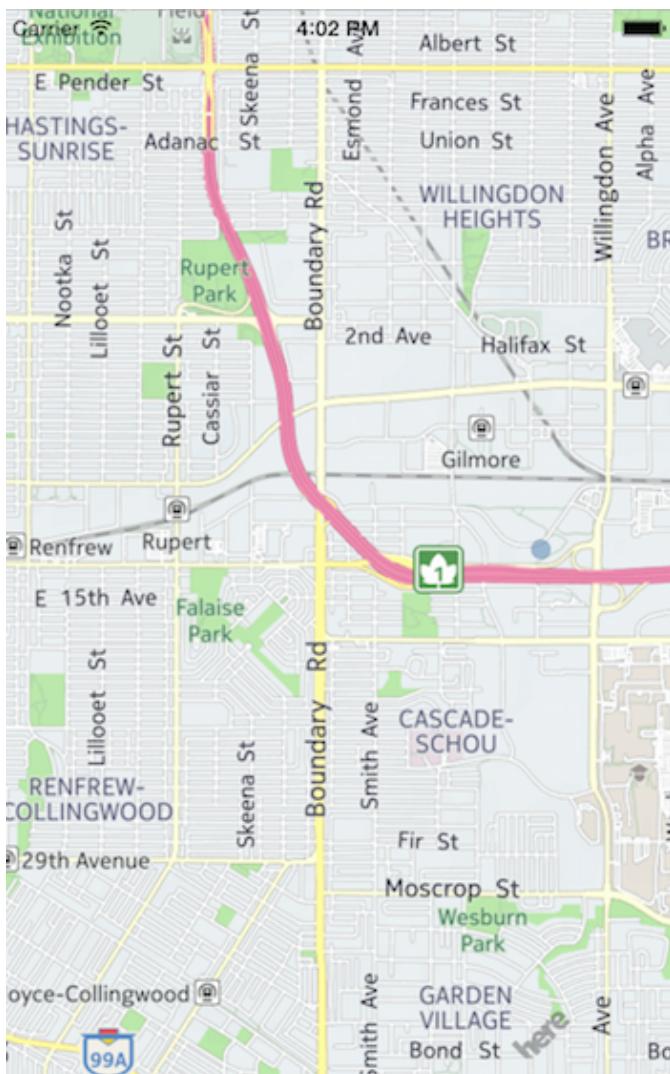
```
import NMAKit
```

- Add the following in the `application(_:didFinishLaunchingWithOptions)` function, replacing `YOUR_APP_ID` and `YOUR_APP_CODE` with the credentials that you received from your <http://developer.here.com>.

```
NMAApplicationContext.set(appId: "YOUR_APP_ID",
    appCode: "YOUR_APP_CODE")
```

6. Build and run the application. If the build is successful, you now have an application that displays a map similar to the following screenshot and allows you to manipulate it using gestures.

Figure 26: Running the App



Swift Support in HERE SDK

This section provides information on the ways that HERE iOS SDK supports development using Swift 3 or above.

Nullability Annotations

Where appropriate, properties, method parameters, and return types are marked with nullability annotations. This provides more clarity on the generated Swift types and avoids implicitly unwrapped optionals.

For more information on this, read the blog post on "["Nullability and Objective-C"](#)".

Designated Initializers

Where appropriate, class header files have annotations to indicate the designated initializer method.

For more information, see the documentation on [initialization in Swift](#) and [initialization in Objective-C](#).

Suggested Names for Swift

The HERE SDK contains suggested Swift method names to better align with standard Swift naming conventions. For example, `createGeocodeRequestWithQuery:searchArea:locationContext:` is `create(query:searchArea:locationContext:)` in Swift.

 **Attention:** An API Reference containing Swift method names is not currently available. You can find Swift method names in the `.h` header files.

For more information, see the blog post on [Swift 3 API Design Guidelines](#).

Stronger Typing Using Lightweight Generics

Where appropriate, the HERE SDK contains lightweight generics information for collection types. This provides stronger typing in the generated Swift code. For example, the `referenceIdentifiers(forSource:)` method in `NMAPlace` returns `[String]` rather than a `[AnyObject]`.

For more information, see [Importing Objective-C Lightweight Generics](#).

Supported Thread Usage

Developers should be aware of the following threading guidelines while using the HERE iOS SDK:

- All NMA interface methods are designed to be called from the main thread.
- All NMA protocol callbacks and blocks are dispatched to the main thread.
- Any exceptions to this rule are clearly documented in the applicable protocol or block definition.

Chapter 5

Coverage Information

The following list provides coverage information for HERE iOS SDK features. Feature support in the HERE SDK may differ depending on the language and locale.

- [*Public Transit*](#)
- [*Routing*](#)
- [*Point Addresses* \(such as house numbers\)](#)
- [*Online Geocoding / Reverse Geocoding*](#)
- [*Online Places and Search*](#)
- Satellite Imagery: Worldwide

Chapter 6

API reference

Topics:

- [*NMA Common*](#)
- [*NMA Mapping*](#)
- [*NMA Routing*](#)
- [*NMA Search*](#)

The HERE iOS SDK allows you to add HERE Maps, Routing and Search functionality to your iOS applications. The following pages provide a detailed reference to the packages and classes that make up the SDK.

Open Source Software Notices

If you use the HERE iOS SDK in your application, you must embed or link to the HERE copyright and various open source software licenses. You can find a copy of these licenses in the SDK Release Notes, or in the SDK package.

NMA Common

Group Summary

group NMA Common

The Common group includes classes, protocols, and enumerations that are generally used by other packages in the iOS SDK.

[For complete information, see the section [Group Details](#)]

Nested Classes

Table 1: Public Nested Classes

Public Nested Classes
NMAApplicationContext
NMAGeoBoundingBox
NMAGeoCoordinates
NMAGeoPosition
NMAPositioningManager
NMARoadElement

Public Enumeration Summary

Table 2: Public Enumerations

Public Enumerations
NMALayoutPosition Represents fixed locations or regions within a bounding rectangle.
NMAResponseError Error codes that may be returned from places search requests.
NMARoadElementAttribute Attributes of a road element.
NMARoadElementPluralType Plural type identifies when a Junction is made up of multiple Road Elements. It also implies that a maneuver may require different explication than implied by the geometry.
NMARoadElementType Road element classifications.

Public Variable Summary

Table 3: Public Variables

Public Variables
<pre>const double NMAGeoPositionUnknownValue</pre> <p>A constant used to indicate an unknown measurement value</p>
<pre>FOUNDATION_EXPORT const NSUInteger NMAMapObjectMaximumZIndex</pre>
<pre>FOUNDATION_EXPORT const NSUInteger NMAMapObjectMinimumZIndex</pre>
<pre>FOUNDATION_EXPORT const float NMAMapViewMaximumZoomLevel</pre> <p>Upper limit of the NMAMapView zoomLevel property</p>
<pre>FOUNDATION_EXPORT const float NMAMapViewMinimumZoomLevel</pre> <p>Lower limit of the NMAMapView zoomLevel property</p>
<pre>FOUNDATION_EXPORT NSString *const _Nonnull NMAPositioningManagerDidLosePositionNotification</pre> <p>Indicates that the <i>NMAPositioningManager</i> currentPosition has been lost.</p>
<pre>FOUNDATION_EXPORT NSString *const _Nonnull NMAPositioningManagerDidUpdatePositionNotification</pre> <p>Indicates that the <i>NMAPositioningManager</i> currentPosition has changed.</p>

Group Details

The Common group includes classes, protocols, and enumerations that are generally used by other packages in the iOS SDK. Some important classes in this group are *NMAGeoCoordinates* and *NMAPositioningManager*.

Public Enumeration Details

NMALayoutPosition

Include: NMAKit.framework/headers/NMATypes.h

Represents fixed locations or regions within a bounding rectangle.

Enumeration Members:

- **NMALayoutPositionTopLeft**
- **NMALayoutPositionTopCenter**
- **NMALayoutPositionTopRight**
- **NMALayoutPositionCenterLeft**
- **NMALayoutPositionCenter**
- **NMALayoutPositionCenterRight**
- **NMALayoutPositionBottomLeft**
- **NMALayoutPositionBottomCenter**
- **NMALayoutPositionBottomRight**

NMAResponseError

Include: NMAKit.framework/headers/NMATypes.h

Error codes that may be returned from places search requests.

Enumeration Members:

- **NMARequestErrorNone**

No error.

- **NMARequestErrorUnknown**

Unknown error.

- **NMARequestErrorAddressMissing**

Request is missing an address parameter.

- **NMARequestErrorLocationContextInvalid**

Location context is invalid.

- **NMARequestErrorLocationContextMissing**

Request is missing location context parameter.

- **NMARequestErrorCountryCodeMissing**

Request is missing a country code in the address parameter.

- **NMARequestErrorSearchTextMissing**

Request missing search text parameter.

- **NMARequestErrorInvalidParameter**

Invalid query parameter.

- **NMARequestErrorNetworkCommunication**

Network communications error.

- **NMARequestErrorNetworkServer**

Backend server unreachable.

- **NMARequestErrorNetworkEmptyInput**

Request input missing.

- **NMARequestErrorInvalidCredentials**

Application ID and/or token are missing or invalid.

- **NMARequestErrorHttp**

Request failed due to an HTTP error.

- **NMARequestErrorNetworkUnknown**

Unknown network error.

- **NMARequestErrorNotSupported**

Functionality not supported.

- **NMARequestErrorNotReady**

An entity is not ready. All poll functions return this when one needs to poll again. Also if an operation is in progress.

- **NMARequestErrorNotFound**

When something was not found.

- **NMARequestErrorAlreadyExists**

When resource already exists.

- **NMARequestErrorOutOfMemory**

Out of memory.

- **NMARequestErrorOperationFailed**

When operation fails (e.g. extracting of file failed)

- **NMARequestErrorAborted**

Operation aborted by user.

- **NMARequestErrorFinderIndexFailure**

Search index failure.

- **NMARequestErrorMovedPermanently**

Resource moved (e.g. resource URI changed)

- **NMARequestErrorNotModified**

Resource contend remains the same.

- **NMARequestErrorBadRequest**

Invalid request (places component might need to be updated)

- **NMARequestErrorResourceGone**

Resource no longer exists.

- **NMARequestErrorParse**

An error occurred during document parsing (json, etc.)

- **NMARequestErrorTimeout**

Request timeout.

NMARoadElementAttribute

Include: NMKit.framework/headers/NMARoadElement.h

Attributes of a road element.

Enumeration Members:

- **NMARoadElementAttributeUndefined**

- **NMARoadElementAttributeDirtRoad**

Road type.

- **NMARoadElementAttributeUsageFeeRequired**

- **NMARoadElementAttributeCarpool**

- **NMARoadElementAttributeUrban**

- **NMARoadElementAttributeTollroad**

- **NMARoadElementAttributeNoThroughTraffic**

- **NMARoadElementAttributeTunnel**

- **NMARoadElementAttributeSliproad**

- **NMARoadElementAttributeHighway**

- **NMARoadElementAttributeUnderConstruction**

- **NMARoadElementAttributeHasLaneDir**

- **NMARoadElementAttributeHasLaneExit**
- **NMARoadElementAttributeFerry**
- **NMARoadElementAttributeRailFerry**

NMARoadElementPluralType

Include: `NMAKit.framework/headers/NMARoadElement.h`

Plural type identifies when a Junction is made up of multiple Road Elements. It also implies that a maneuver may require different explication than implied by the geometry.

Enumeration Members:

- **NMARoadElementPluralTypeNone**
Junction is not plural.
- **NMARoadElementPluralTypeManeuver**
Indicates that only one command should be given despite the fact that two Junctions occur – one at each end of the turn lane. It is only necessary to state "turn left/right" near the beginning of the maneuver because at the end the driver does not have a choice in direction.
- **NMARoadElementPluralTypeConnector**
Indicates that a road segment should not be viewed as an individual piece of road but as part of the intersection. A separate guidance manoeuvre should not exist for this segment.
- **NMARoadElementPluralTypeIndeterminate**
Indicates a maneuver that cannot be explained in one command or at all. A graphic may be needed to illustrate the turn. For example, a driver may need to go right to make a left turn.

NMARoadElementType

Include: `NMAKit.framework/headers/NMARoadElement.h`

Road element classifications.

The NAMRoadElementType describes the nature of the road that a [NMARoadElement](#) is part of. A road element may only have one type, or none if the road is an unclassified type or the information is not available.

Enumeration Members:

- **NMARoadElementTypeUndefined**
- **NMARoadElementTypeMotorway**
- **NMARoadElementTypeMultiCarriageway**
- **NMARoadElementTypeSingleCarriageway**
- **NMARoadElementTypeRoundabout**
- **NMARoadElementTypeSliproad**
- **NMARoadElementTypePedestrianZone**
- **NMARoadElementTypePedestrianWalkway**
- **NMARoadElementTypeServiceAccessParking**
- **NMARoadElementTypeServiceAccessOther**

- `NMARoadElementTypeServiceRoad`

Public Variable Details

`const double NMAGeoPositionUnknownValue`

Include: NMAKit.framework/headers/NMAGeoPosition.h

A constant used to indicate an unknown measurement value.

`FOUNDATION_EXPORT const NSUInteger NMAMapObjectMaximumZIndex`

Include: NMAKit.framework/headers/NMATypes.h

The maximum Z Index for an `NMAMapObject`

`FOUNDATION_EXPORT const NSUInteger NMAMapObjectMinimumZIndex`

Include: NMAKit.framework/headers/NMATypes.h

The minimum Z Index for an `NMAMapObject`

`FOUNDATION_EXPORT const float NMAMapViewMaximumZoomLevel`

Include: NMAKit.framework/headers/NMATypes.h

Upper limit of the `NMAMapView` zoomLevel property.

`FOUNDATION_EXPORT const float NMAMapViewMinimumZoomLevel`

Include: NMAKit.framework/headers/NMATypes.h

Lower limit of the `NMAMapView` zoomLevel property.

`FOUNDATION_EXPORT NSString *const _Nonnull NMAPositioningManagerDidLosePositionNotification`

Include: NMAKit.framework/headers/NMAPositioningManager.h

Indicates that the `NMAPositioningManager` currentPosition has been lost.

In order to receive this notification, register with NSNotificationCenter using this notification name and the [`NMAPositioningManager` sharedPositioningManager] instance as the object (notification sender).

`FOUNDATION_EXPORT NSString *const _Nonnull NMAPositioningManagerDidUpdatePositionNotification`

Include: NMAKit.framework/headers/NMAPositioningManager.h

Indicates that the *NMAPositioningManager* currentPosition has changed.

NMAPositioningManager Notifications In order to receive this notification, register with NSNotificationCenter using this notification name and the [*NMAPositioningManager* sharedPositioningManager] instance as the object (notification sender).

NMAApplicationContext

Class Summary

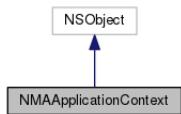
class **NMAApplicationContext**

Derived from: NSObject

Include: NMAKit.framework/headers/NMAApplicationContext.h

Inheritance Diagrams

Figure 27: Public inheritance diagram for NMAApplicationContext



[For complete information, see the section *Class Details*]

Class Method Summary

Table 4: Class Methods

Class Methods
<code>+ (NSString *) sdkVersion</code>
Return the SDK version.
<code>+ (void) setAppId:(NSString *) appId appCode:(appId:appCode:set)</code>
Sets the mandatory HERE Authentication Credentials

Class Details

Use this interface to set mandatory authentication credentials

Note:

YOU MUST SET AUTHENTICATION CREDENTIALS in order to use the APIs contained within this SDK. Some APIs may appear to work without valid credentials but they may stop functioning in the near future when server side configurations change, so please obtain and use valid credentials.

Class Method Details

+ (NSString *) sdkVersion

Return the SDK version.

Returns:

The SDK verion

+ (void) setAppId:(NSString *) appId appCode:(appId:appCode:set)

Sets the mandatory HERE Authentication Credentials.

These credentials MUST be set in order to use the APIs in this SDK. It is recommended to set the credentials in your application delegate, such as in [UIApplicationDelegate didFinishLaunchingWithOptions]

NMAGeoBoundingBox

Class Summary

class **NMAGeoBoundingBox**

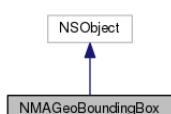
Derived from: NSObject

Represents a bounding box that defines a rectangular area in a geographic coordinate system.

Include: `NMAKit.framework/headers/NMAGeoBoundingBox.h`

Inheritance Diagrams

Figure 28: Public inheritance diagram for NMAGeoBoundingBox



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 5: Public Properties

Public Properties
<code>[readable, writable, strong] NMAGeoCoordinates * bottomLeft</code> The NMAGeoCoordinates representing the bottom-left corner of the NMAGeoBoundingBox
<code>[readable, writable, strong] NMAGeoCoordinates * bottomRight</code> The NMAGeoCoordinates representing the bottom-right corner of the NMAGeoBoundingBox

Public Properties

[readable, weak] `NMAGeoCoordinates * center`

The NMAGeoCoordinates representing the center of the NMAGeoBoundingBox

[readable, assign] BOOL `crossesInternationalDateLine`

[readable, assign] double `height`

The NMAGeoBoundingBox height, in degrees

[readable, writable, strong] `NMAGeoCoordinates * topLeft`

The NMAGeoCoordinates representing the top-left corner of the NMAGeoBoundingBox

[readable, writable, strong] `NMAGeoCoordinates * topRight`

The NMAGeoCoordinates representing the top-right corner of the NMAGeoBoundingBox

[readable, assign] double `width`

The NMAGeoBoundingBox width, in degrees

Instance Method Summary

Table 6: Instance Methods

Instance Method Summary

- (BOOL) `containsGeoBoundingBox:(_:contains)`

Determines whether the specified NMAGeoBoundingBox is covered entirely by this NMAGeoBoundingBox (boundary included)

- (BOOL) `containsGeoCoordinates:(_:contains)`

Determines whether the specified NMAGeoCoordinates object is contained within this NMAGeoBoundingBox (boundary included)

- (instancetype) `initWithTopLeft:(NMAGeoCoordinates *) topLeft bottomRight:(NMAGeoCoordinates *) bottomRight`

Initializes a NMAGeoBoundingBox instance with specified top-left and bottom-right NMAGeoCoordinates

- (nullable NMAGeoBoundingBox *) `intersectionWithGeoBoundingBox:(_:intersection)`

Computes the intersection of this and another NMAGeoBoundingBox

- (BOOL) `intersectsGeoBoundingBox:(_:intersects)`

Determines whether the intesection of this NMAGeoBoundingBox and the specified NMAGeoBoundingBox is non-empty

- (BOOL) `isEmpty`

Determines whether the size of the enclosed NMAGeoBoundingBox is 0

- (BOOL) `isEqualGeoBoundingBox:(_:equals)`

Determines whether this NMAGeoBoundingBox is equal to another

Class Method Summary

Table 7: Class Methods

Class Methods
<pre>+ (instancetype) geoBoundingBoxContainingGeoBoundingBoxes:(NSArray<NMAGeoBoundingBox> *) containingBoundingBoxes:init</pre> <p>Constructs a NMAGeoBoundingBox which contains all bounding boxes in an array.</p>
<pre>+ (instancetype) geoBoundingBoxContainingGeoCoordinates:(NSArray<NMAGeoCoordinate> *) containingCoordinates:init</pre> <p>Constructs a NMAGeoBoundingBox which contains all coordinates in an array.</p>
<pre>+ (instancetype) geoBoundingBoxWithCenter:(NMAGeoCoordinate *) center width:(double) width height:(double) height</pre> <p>Constructs a NMAGeoBoundingBox from a NMAGeoCoordinate center point and a height and width in degrees</p>
<pre>+ (instancetype) geoBoundingBoxWithTopLeft:(NMAGeoCoordinate *) topLeft bottomRight:(NMAGeoCoordinate *) bottomRight</pre> <p>Constructs a NMAGeoBoundingBox from NMAGeoCoordinates defining its top left and bottom right corners</p>
<pre>+ (instancetype) geoBoundingBoxWithTopRight:(NMAGeoCoordinate *) topRight bottomLeft:(NMAGeoCoordinate *) bottomLeft</pre> <p>Constructs a NMAGeoBoundingBox from NMAGeoCoordinates defining its top right and bottom left corners</p>

Class Details

Represents a bounding box that defines a rectangular area in a geographic coordinate system.

Although the bounding box is specified by its top-left and bottom-right corner, the box is not necessarily the smallest rectangle spanned by these two points. It is possible to define bounding boxes that are wider than 180 degrees or higher than 90 degrees (e.g. by setting the longitude of top-left corner to a bigger value than the longitude of the bottom-right corner).

For example, a bounding box with a longitude of -180 degrees for the top-left corner and a longitude of 180 degrees for the bottom-right corner will construct an area that encircles the globe, whereas a bounding box with the same longitude values for both corners will construct a bounding box with a width of 0 degrees.

Public Property Details

[readable, writable, strong] `NMAGeoCoordinate * bottomLeft`

The NMAGeoCoordinate representing the bottom-left corner of the NMAGeoBoundingBox.

[readable, writable, strong] `NMAGeoCoordinate * bottomRight`

The NMAGeoCoordinate representing the bottom-right corner of the NMAGeoBoundingBox.

[readable, weak] `NMAGeoCoordinate * center`

The NMAGeoCoordinate representing the center of the NMAGeoBoundingBox.

[readable, assign] BOOL `crossesInternationalDateLine`

Indicates whether this `NMAGeoBoundingBox` crosses the international date line

[readable, assign] double `height`

The NMAGeoBoundingBox height, in degrees.

[readable, writable, strong] `NMAGeoCoordinates * topLeft`

The NMAGeoCoordinates representing the top-left corner of the NMAGeoBoundingBox.

[readable, writable, strong] `NMAGeoCoordinates * topRight`

The NMAGeoCoordinates representing the top-right corner of the NMAGeoBoundingBox.

[readable, assign] double `width`

The NMAGeoBoundingBox width, in degrees.

Instance Method Details

- (BOOL) `containsGeoBoundingBox:(_:contains)`

Determines whether the specified NMAGeoBoundingBox is covered entirely by this NMAGeoBoundingBox (boundary included).

Parameters:

- `boundingBox`

A NMAGeoBoundingBox to check for containment within this NMAGeoBoundingBox

Returns:

YES if contained within the NMAGeoBoundingBox, NO otherwise

- (BOOL) `containsGeoCoordinates:(_:contains)`

Determines whether the specified NMAGeoCoordinates object is contained within this NMAGeoBoundingBox (boundary included).

Parameters:

- `coordinates`

A NMAGeoCoordinates object to check for containment within this NMAGeoBoundingBox



Returns:

YES if contained within the NMAGeoBoundingBox, NO otherwise

```
- (instancetype) initWithTopLeft:(NMAGeoCoordinates *) topLeft bottomRight:(NMAGeoCoordinates *) bottomRight
```

Initializes a NMAGeoBoundingBox instance with specified top-left and bottom-right NMAGeoCoordinates.

Parameters:

- **topLeft**
A top-left NMAGeoCoordinates value for the new NMAGeoBoundingBox
- **bottomRight**
A bottom-right NMAGeoCoordinates value for the new NMAGeoBoundingBox

Returns:

The NMAGeoBoundingBox

```
- (nullable NMAGeoBoundingBox *) intersectionWithGeoBoundingBox:(_:intersection)
```

Computes the intersection of this and another [NMAGeoBoundingBox](#).

Note:

The intersection may have zero area.

Parameters:

- **boundingBox**
The [NMAGeoBoundingBox](#) to intersect with

Returns:

The [NMAGeoBoundingBox](#) representing the intersection, or nil if the boxes do not uniquely intersect

```
- (BOOL) intersectsGeoBoundingBox:(_:intersects)
```

Determines whether the intersection of this NMAGeoBoundingBox and the specified NMAGeoBoundingBox is non-empty.

Parameters:

- **boundingBox**
A NMAGeoBoundingBox to check for intersection with this NMAGeoBoundingBox

Returns:

YES if the intersection of NMAGeoBoundingBox objects is non-empty, NO otherwise

`- (BOOL) isEmpty`

Determines whether the size of the enclosed NMAGeoBoundingBox is 0.

Returns:

YES if the size of the enclosed area is 0, NO otherwise

`- (BOOL) isEqualToGeoBoundingBox:(_:equals)`

Determines whether this `NMAGeoBoundingBox` is equal to another.

Returns:

YES if the bounding boxes are equal, NO otherwise

Class Method Details

`+ (instancetype) geoBoundingBoxContainingGeoBoundingBoxes:(containingBoundingBoxes:init)`

Constructs a NMAGeoBoundingBox which contains all bounding boxes in an array.

The constructed `NMAGeoBoundingBox` will be the smallest possible box which contains all the NMAGeoBoundingBoxes in the input array.

Parameters:

- **boxes**

A NSArray of `NMAGeoBoundingBox` defining the new NMAGeoBoundingBox

Returns:

The NMAGeoBoundingBox

`+ (instancetype) geoBoundingBoxContainingGeoCoordinates:(containingCoordinates:init)`

Constructs a NMAGeoBoundingBox which contains all coordinates in an array.

The constructed `NMAGeoBoundingBox` will be the smallest possible box which contains all the `NMAGeoCoordinates` in the input array.

Parameters:

- **coordinates**

A NSArray of `NMAGeoCoordinates` defining the new NMAGeoBoundingBox

Returns:

The NMAGeoBoundingBox

```
+ (instancetype) geoBoundingBoxWithCenter:(NMAGeoCoordinates *) center  
width:(double) width height:(double) height
```

Constructs a NMAGeoBoundingBox from a *NMAGeoCoordinates* center point and a height and width in degrees.

■ **Note:**

If the specified values produce a box with top edge latitude > 90 or bottom edge latitude < -90, the box will be clamped to that range. If the top edge latitude is less than the bottom edge latitude, nil will be returned.

Parameters:

- **center**
The *NMAGeoCoordinates* representing the center of the bounding box
- **width**
The width of the bounding box in degrees longitude
- **height**
The height of the bounding box in degrees latitude

Returns:

The NMAGeoBoundingBox if the inputs are valid, else nil

```
+ (instancetype) geoBoundingBoxWithTopLeft:(NMAGeoCoordinates *) topLeft  
bottomRight:(NMAGeoCoordinates *) bottomRight
```

Constructs a NMAGeoBoundingBox from *NMAGeoCoordinates* defining its top left and bottom right corners.

Parameters:

- **topLeft**
The *NMAGeoCoordinates* representing the top left corner of the bounding box
- **bottomRight**
The *NMAGeoCoordinates* representing the bottom right corner of the bounding box

Returns:

The NMAGeoBoundingBox

```
+ (instancetype) geoBoundingBoxWithTopRight:(NMAGeoCoordinates *) topRight  
bottomLeft:(NMAGeoCoordinates *) bottomLeft
```

Constructs a NMAGeoBoundingBox from *NMAGeoCoordinates* defining its top right and bottom left corners.

Parameters:

- **topRight**
The *NMAGeoCoordinates* representing the top right corner of the bounding box
- **bottomLeft**

The `NMAGeoCoordinates` representing the bottom left corner of the bounding box

Returns:

The `NMAGeoBoundingBox`

NMAGeoCoordinates

Class Summary

class `NMAGeoCoordinates`

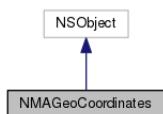
Derived from: NSObject

Represents WGS84 coordinates with double precision.

Include: `NMAKit.framework/headers/NMAGeoCoordinates.h`

Inheritance Diagrams

Figure 29: Public inheritance diagram for NMAGeoCoordinates



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 8: Public Properties

Public Properties
<code>[readable, writable, assign] float altitude</code> The altitude of the <code>NMAGeoCoordinates</code> measured in meters above sea-level.
<code>[readable, writable, assign] double latitude</code> The latitude, as measured in degrees, representing north-south coordinate (y-axis on a map)
<code>[readable, writable, assign] double longitude</code> The longitude, as measured in degrees, representing east-west coordinate (x-axis on a map)

Instance Method Summary

Table 9: Instance Methods

Instance Method Summary
<pre>- (double) <i>distanceTo:(to:distance)</i></pre> <p>Calculates (using the Haversine formula) the distance in meters between this NMAGeoCoordinates instance and the specified NMAGeoCoordinates</p>
<pre>- (double) <i>headingTo:(to:heading)</i></pre> <p>Calculates the heading in degrees from this NMAGeoCoordinates instance to the specified NMAGeoCoordinates instance</p>
<pre>- (instancetype) <i>initWithLatitude:(double) latitude longitude:(double) longitude</i></pre> <p>Initializes a NMAGeoCoordinates instance with specified latitude and longitude values and zero altitude</p>
<pre>- (instancetype) <i>initWithLatitude:(double) latitude longitude:(double) longitude altitude:(double) altitude</i></pre> <p>Initializes a NMAGeoCoordinates instance with specified values for the initial latitude, longitude and altitude</p>
<pre>- (BOOL) <i>isEqualGeoCoordinates:(:_:equals)</i></pre> <p>Determines whether this NMAGeoCoordinates is equal to another</p>

Class Method Summary

Table 10: Class Methods

Class Methods
<pre>+ (instancetype) <i>geoCoordinatesWithLatitude:(double) latitude longitude:(double) longitude</i></pre> <p>Constructs a NMAGeoCoordinates with the specified latitude and longitude</p>
<pre>+ (instancetype) <i>geoCoordinatesWithLatitude:(double) latitude longitude:(double) longitude altitude:(double) altitude</i></pre> <p>Constructs a NMAGeoCoordinates with the specified latitude, longitude, and altitude</p>

Class Details

Represents WGS84 coordinates with double precision. A NMAGeoCoordinates encapsulates a latitude and longitude value, plus an optional altitude value.

Note:

Although NMAGeoCoordinates are accepted and handled with double precision (about 15 significant digits), the real precision of rendering operations such as NMAMapView::setCenterAt might be lower due to internal conversions. Overall precision of NMAGeoCoordinates can be assured for only six digits following the decimal point.

Public Property Details

[readable, writable, assign] float **altitude**

The altitude of the *NMAGeoCoordinates* measured in meters above sea-level.

By default, the plane of the map is at 0 meters. Thus, geocoordinates with positive altitudes represent points above the plane of the map, while negative altitudes are below the plane of the map.

■ **Note:**

Valid values are in the range [-10000.0, 10000.0]. If an invalid value is specified, it will be adjusted to the closest possible valid value (e.g. a specified value of -10200.0 will be set to -10000.0).

[readable, writable, assign] double **latitude**

The latitude, as measured in degrees, representing north-south coordinate (y-axis on a map).

■ **Note:**

The latitude must be in the range [-90, 90]. Values outside this range will be clamped.

[readable, writable, assign] double **longitude**

The longitude, as measured in degrees, representing east-west coordinate (x-axis on a map).

■ **Note:**

The longitude must be in the range [-180, 180]. Values outside this range will be wrapped. For example, a value of 200 degrees will be wrapped to -160.

Instance Method Details

-(double) **distanceTo:(to:distance)**

Calculates (using the Haversine formula) the distance in meters between this NMAGeoCoordinates instance and the specified NMAGeoCoordinates.

Parameters:

- **coordinates**

A second NMAGeoCoordinates some distance away

Returns:

The distance between the NMAGeoCoordinates, or -DBL_MAX if the coordinates parameter is nil

-(double) **headingTo:(to:heading)**

Calculates the heading in degrees from this NMAGeoCoordinates instance to the specified NMAGeoCoordinates instance.

Parameters:

- **coordinates**

NMAGeoCoordinates to which to the heading is calculated.

Returns:

Heading from this coordinate to the given coordinate, in degrees. 0 = north, increasing clockwise. If coordinates is nil or invalid a negative result will be returned to indicate an error.

- (instancetype) initWithLatitude:(double) latitude longitude:(double) longitude

Initializes a NMAGeoCoordinates instance with specified latitude and longitude values and zero altitude.

Parameters:

- **latitude**
An initial latitude value. Values outside of [-90, 90] will be clamped to that range (100 becomes 90).
- **longitude**
An initial longitude value. Values outside of [-180, 180] will be wrapped to that range (200 becomes -160).

Returns:

The NMAGeoCoordinates instance, or nil if initialization failed

- (instancetype) initWithLatitude:(double) latitude longitude:(double) longitude altitude:(double) altitude

Initializes a NMAGeoCoordinates instance with specified values for the initial latitude, longitude and altitude.

Parameters:

- **latitude**
An initial latitude value
- **longitude**
An initial longitude value
- **altitude**
An initial altitude value

Returns:

The NMAGeoCoordinates instance, or nil if initialization failed

- (BOOL) isEqualToGeoCoordinates:(_:equals)

Determines whether this *NMAGeoCoordinates* is equal to another.

Returns:

YES if the coordinates are equal, NO otherwise

Class Method Details

```
+ (instancetype) geoCoordinatesWithLatitude:(double) latitude longitude:(double) longitude
```

Constructs a [NMAGeoCoordinates](#) with the specified latitude and longitude.

Parameters:

- **latitude**
An initial latitude value
- **longitude**
An initial longitude value

```
+ (instancetype) geoCoordinatesWithLatitude:(double) latitude longitude:(double) longitude altitude:(double) altitude
```

Constructs a [NMAGeoCoordinates](#) with the specified latitude, longitude, and altitude.

Parameters:

- **latitude**
An initial latitude value
- **longitude**
An initial longitude value
- **altitude**
An initial altitude value

NMAGeoPosition

Class Summary

class NMAGeoPosition

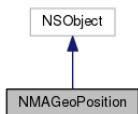
Derived from: NSObject

Represents position, speed, and course information as provided by a positioning source for a distinct moment in time.

Include: NMKit.framework/headers/NMAGeoPosition.h

Inheritance Diagrams

Figure 30: Public inheritance diagram for NMAGeoPosition



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 11: Public Properties

Public Properties
<p>[readable, assign] double <i>altitudeAccuracy</i> The accuracy of the altitude, in meters</p>
<p>[readable, assign] <i>NMAGeoCoordinates</i> * <i>coordinates</i> The geographical location of the position</p>
<p>[readable, assign] double <i>course</i> The course (direction of travel) of the position, in degrees.</p>
<p>[readable, assign] BOOL <i>isValid</i> Indicates the validity of the NMAGeoPosition</p>
<p>[readable, assign] double <i>latitudeAccuracy</i> The accuracy of the latitude coordinate, in meters</p>
<p>[readable, assign] double <i>longitudeAccuracy</i> The accuracy of the longitude coordinate, in meters</p>
<p>[readable, assign] double <i>speed</i> The movement speed of the position</p>
<p>[readable, strong] NSDate * <i>timestamp</i> The time at which the position measurements were taken</p>

Class Method Summary

Table 12: Class Methods

Class Methods
<p>+(instancetype) <i>geoPositionWithCLLocation:(CLLocation *)location init</i> A convenience constructor for creating NMAGeoPosition instances with the specified CLLocation</p>
<p>+(instancetype) <i>geoPositionWithCoordinates:(NMAGeoCoordinates *) coordinates speed:(double) speed course:(double) course accuracy:(double) accuracy</i> A convenience constructor for creating NMAGeoPosition instances with specified coordinates, speed, course, and accuracy</p>
<p>+(instancetype) <i>geoPositionWithCoordinates:(NMAGeoCoordinates *) coordinates speed:(double) speed course:(double) course accuracy:(double) accuracy timestamp:(NSDate *) timestamp</i> A convenience constructor for creating NMAGeoPosition instances with specified coordinates, speed, course, and accuracy</p>

Class Methods

```
+ (instancetype) geoPositionWithCoordinates:(NMAGeoCoordinates *) coordinates speed:(double) speed  
course:(double) course latitudeAccuracy:(double) latitudeAccuracy longitudeAccuracy:(double)  
longitudeAccuracy altitudeAccuracy:(double) altitudeAccuracy timestamp:(NSDate *) timestamp
```

A convenience constructor for creating NMAGeoPosition instances with specified coordinates, speed, course, latitudeAccuracy, longitudeAccuracy and altitudeAccuracy

Class Details

Represents position, speed, and course information as provided by a positioning source for a distinct moment in time.

Public Property Details

[readable, assign] double **altitudeAccuracy**

The accuracy of the altitude, in meters.

■ Note:

Will be NMAGeoPositionUnknownValue if unknown.

[readable, assign] NMAGeoCoordinates * **coordinates**

The geographical location of the position.

[readable, assign] double **course**

The course (direction of travel) of the position, in degrees.

Valid course values are in the range [0, 360), with 0 degrees representing north and values increasing clockwise. Thus, east is 90 degrees, south is 180 degrees, and so on.

■ Note:

Will be NMAGeoPositionUnknownValue if unknown.

[readable, assign] BOOL **isValid**

Indicates the validity of the NMAGeoPosition. The position is valid if its coordinates are valid.

[readable, assign] double **latitudeAccuracy**

The accuracy of the latitude coordinate, in meters.

■ Note:

Will be NMAGeoPositionUnknownValue if unknown.

[readable, assign] double longitudeAccuracy

The accuracy of the longitude coordinate, in meters.

■ **Note:**

Will be NMAGeoPositionUnknownValue if unknown.

[readable, assign] double speed

The movement speed of the position.

■ **Note:**

Will be NMAGeoPositionUnknownValue if unknown.

[readable, strong] NSDate * timestamp

The time at which the position measurements were taken.

Class Method Details

+ (instancetype) geoPositionWithCLLocation:(CLLocation *)location init

A convenience constructor for creating *NMAGeoPosition* instances with the specified CLLocation.

+ (instancetype) geoPositionWithCoordinates:(NMAGeoCoordinates *)coordinates speed:(double) speed course:(double) course accuracy:(double) accuracy

A convenience constructor for creating *NMAGeoPosition* instances with specified coordinates, speed, course, and accuracy.

■ **Note:**

The accuracy parameter will be used for all of latitudeAccuracy, longitudeAccuracy, and altitudeAccuracy.

■ **Note:**

The timestamp will be set to the current time.

+ (instancetype) geoPositionWithCoordinates:(NMAGeoCoordinates *)coordinates speed:(double) speed course:(double) course accuracy:(double) accuracy timestamp:(NSDate *) timestamp

A convenience constructor for creating *NMAGeoPosition* instances with specified coordinates, speed, course, and accuracy.

■ **Note:**

The accuracy parameter will be used for all of latitudeAccuracy, longitudeAccuracy, and altitudeAccuracy.

```
+ (instancetype) geoPositionWithCoordinates:(NMAGeoCoordinates *) coordinates speed:(double) speed course:(double) course latitudeAccuracy:(double) latitudeAccuracy longitudeAccuracy:(double) longitudeAccuracy altitudeAccuracy:(double) altitudeAccuracy timestamp:(NSDate *) timestamp
```

A convenience constructor for creating [NMAGeoPosition](#) instances with specified coordinates, speed, course, latitudeAccuracy, longitudeAccuracy and altitudeAccuracy.

NMAPositioningManager

Class Summary

class **NMAPositioningManager**

Derived from: [NSObject](#)

Manages positioning services for the NMA SDK.

Include: [NMAKit.framework/headers/NMAPositioningManager.h](#)

Inheritance Diagrams

Figure 31: Public inheritance diagram for NMAPositioningManager



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 13: Public Properties

Public Properties
<p>[readable, assign] BOOL active Indicates whether the NMAPositioningManager is currently broadcasting position updates</p>
<p>[readable, assign] NMAGeoPosition * currentPosition The best estimate for the user's current position.</p>
<p>[readable, assign] NMAGeoPosition * rawPosition The latest unprocessed position</p>

Instance Method Summary

Table 14: Instance Methods

Instance Method Summary
<code>- (BOOL) <i>startPositioning</i></code> Enables position broadcasting.
<code>- (void) <i>stopPositioning</i></code> Disables position broadcasting.

Class Method Summary

Table 15: Class Methods

Class Methods
<code>+ (nonnull <i>NMAPositioningManager</i> *) <i>sharedPositioningManager</i></code> Returns the NMAPositioningManager singleton instance

Class Details

Manages positioning services for the NMA SDK.

This is a singleton class and thus should never be retained, released, or instantiated. The singleton instance is accessed via the `sharedPositioningManager` method.

Use the `startPositioning` method to begin tracking position, and `stopPositioning` to stop. By default, position data comes from `CLLocationManager`.

Only one type of position is available from the positioning manager in Starter Edition: the raw position. The raw position is simply passed straight through from the positioning manager's data source. If the position is lost, a `NMAPositioningManagerDidLosePositionNotification` is sent.

Note:

Certain other SDK features may require the use of the positioning manager; if these features are accessed, the positioning manager will be enabled automatically (for example, the `NMAPositionIndicator` of a `NMAMapView`).

Note:

`NMAPositioningManager` requires valid authentication credentials to be set via `NMAApplicationContext`. If valid credentials are not present the API may not function correctly or may stop functioning correctly in the future when server side configurations change.

Public Property Details

[readable, assign] BOOL `active`

Indicates whether the `NMAPositioningManager` is currently broadcasting position updates.

[readable, assign] `NMAGeoPosition * currentPosition`

The best estimate for the user's current position.

The currentPosition will be equal to the rawPosition in the Starter Edition.

■ Note:

Will be nil if the positioning manager is not active or does not have a valid position.

[readable, assign] `NMAGeoPosition * rawPosition`

The latest unprocessed position.

■ Note:

Will be nil if the latest update did not return a valid position.

Instance Method Details

`- (BOOL) startPositioning`

Enables position broadcasting.

Returns:

YES if positioning was successfully started, NO if positioning was already in progress or if positioning failed to start.

`- (void) stopPositioning`

Disables position broadcasting.

Class Method Details

`+ (nonnull NMAPositioningManager *) sharedPositioningManager`

Returns the `NMAPositioningManager` singleton instance.

■ Note:

Use this method to obtain a `NMAPositioningManager` instance. Do not call init directly.

Returns:

shared `NMAPositioningManager` instance

NMARoadElement

Class Summary

class **NMARoadElement**

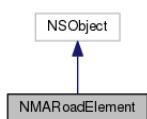
Derived from: `NSObject`

Represents a section of a road.

Include: `NMAKit.framework/headers/NMARoadElement.h`

Inheritance Diagrams

Figure 32: Public inheritance diagram for NMARoadElement



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 16: Public Properties

Public Properties
<code>[readable, assign] NSString * <i>abbreviatedName</i></code> The abbreviated name of the road element
<code>[readable, assign]NSUInteger <i>attributes</i></code> The attributes of the NMARoadElement
<code>[readable, assign] NSUInteger <i>averageSpeed</i></code> The average speed of traffic on the NMARoadElement in m/s, or 0 if the information is not available
<code>[readable, assign] NSUInteger <i>numberOfLanes</i></code> The number of lanes of the road element, or 0 if the data is unavailable
<code>[readable, assign] NMARoadElementPluralType <i>pluralType</i></code> The plural type of the road element
<code>[readable, assign] NSString * <i>roadName</i></code> The name of the road element, or an empty string if the data is not available
<code>[readable, assign] NSUInteger <i>speedLimit</i></code> The speed limit of the NMARoadElement in m/s, or 0 if the information is not available
<code>[readable, assign] NMARoadElementType <i>type</i></code> The type of the road element

Public Properties

[readable, assign] `NSString * uniqueId`

The unique identifier of the NMARoadElement

Class Details

Represents a section of a road.

Public Property Details

[readable, assign] `NSString * abbreviatedName`

The abbreviated name of the road element. For example, I5 for the Interstate 5 in the US. This property will contain an empty string if the abbreviated name is unavailable.

[readable, assign] `NSUInteger attributes`

The attributes of the `NMARoadElement`. Can be a combination of any values from `NMARoadElementAttribute`.

[readable, assign] `NSUInteger averageSpeed`

The average speed of traffic on the `NMARoadElement` in m/s, or 0 if the information is not available.

[readable, assign] `NSUInteger numberOfLanes`

The number of lanes of the road element, or 0 if the data is unavailable.

[readable, assign] `NMARoadElementPluralType pluralType`

See also:

`NMARoadElementPluralType`

[readable, assign] `NSString * roadName`

The name of the road element, or an empty string if the data is not available.

[readable, assign] `NSUInteger speedLimit`

The speed limit of the `NMARoadElement` in m/s, or 0 if the information is not available.

[readable, assign] `NMARoadElementType type`

See also:

[NMARoadElementType](#)

[readable, assign] `NSString * uniqueId`

The unique identifier of the [NMARoadElement](#).

Compare uniqueids of two road elements if you only want to know whether they are the same road elements with the same geometries.

NMA Mapping

Group Summary

group NMA Mapping

The Mapping group provides classes, protocols, and enumerations that can be used to enable an interactive map and related functionality, such as creating and adding map objects.

[For complete information, see the section [Group Details](#)]

Nested Classes

Table 17: Public Nested Classes

Public Nested Classes
<code>NMAMapCircle</code>
<code>NMAMapContainer</code>
<code>NMAMapGestureDelegate [p]</code>
<code>NMAMapInfoBubbleCustomizationContext</code>
<code>NMAMapInfoBubbleCustomizer</code>
<code>NMAMapMarker</code>
<code>NMAMapObject</code>
<code>NMAMapPolygon</code>
<code>NMAMapPolyline</code>
<code>NMAMapRoute</code>
<code>NMAMapScheme</code>
<code>NMAMapTileLayer</code>
<code>NMAMapTileLayerDataSource [p]</code>
<code>NMAMapView</code>
<code>NMAMapViewDelegate [p]</code>

Public Nested Classes

NMAPositionIndicator

Public Typedef Summary

Table 18: Public Typedefs

Public Typedefs

`(void) (^NMAInfoBubbleEventBlock) (void)`

The tap handler block called when the info bubble is tapped.

Public Enumeration Summary

Table 19: Public Enumerations

Public Enumerations

NMAMapAnimation

Defines types of animations that can illustrate on-screen map movements.

NMAMapGestureType

The types of gesture that can be performed on an NMAMapView.

NMAMapObjectType

Defines types of NMAMapObject objects that an application can add to a map.

NMAMapPPI

Defines PPI of the map tiles.

Public Variable Summary

Table 20: Public Variables

Public Variables

`FOUNDATION_EXPORT NSString *const NMAMapSchemeHybridDay`

Satellite tile scheme presented in "day" colors, with roads

`FOUNDATION_EXPORT NSString *const NMAMapSchemeNormalDay`

Normal scheme presented in "day" colors

`FOUNDATION_EXPORT NSString *const NMAMapSchemeNormalNight`

Normal scheme presented in "night" colors

`FOUNDATION_EXPORT NSString *const NMAMapSchemePedestrianDay`

Pedestrian navigation scheme presented in "day" colors

`FOUNDATION_EXPORT NSString *const NMAMapSchemePedestrianNight`

Pedestrian navigation scheme presented in "night" colors

Public Variables

```
FOUNDATION_EXPORT NSString *const NMAMapSchemeReducedDay
```

Normal scheme presented in "day" with less color palette

```
FOUNDATION_EXPORT NSString *const NMAMapSchemeReducedNight
```

Normal scheme presented in "night" with less color palette

```
FOUNDATION_EXPORT NSString *const NMAMapSchemeSatelliteDay
```

Satellite tile scheme presented in "day" colors

```
FOUNDATION_EXPORT NSString *const NMAMapSchemeTerrainDay
```

Terrain bitmap scheme presented in "day" colors

```
FOUNDATION_EXPORT const float NMAMapViewPreserveValue
```

This value may be passed into the map move method setGeoCenter:zoomLevel:withAnimation: (NMAMapView) in place of zoom in order to preserve the current value of that property

Group Details

The Mapping group provides classes, protocols, and enumerations that can be used to enable an interactive map and related functionality, such as creating and adding map objects. Some key classes and protocols in this group are: [NMAMapView](#), [NMAMapGestureDelegate](#), [NMAMapObject](#), and [NMAMapScheme](#)

Public Typedef Details

```
(void) (^NMAInfoBubbleEventBlock) (void)
```

Include: NMAKit.framework/headers/NMAMapMarker.h

The tap handler block called when the info bubble is tapped.

Public Enumeration Details

NMAMapAnimation

Include: NMAKit.framework/headers/NMAMapView.h

Defines types of animations that can illustrate on-screen map movements.

Enumeration Members:

- **NMAMapAnimationLinear**

Animation moves in a linear manner as the map resets to a new position.

- **NMAMapAnimationNone**

No animation is performed as the map resets to a new position.

NMAMapGestureType

Include: NMAKit.framework/headers/NMAMapGesture.h

The types of gesture that can be performed on an [NMAMapView](#).

Enumeration Members:

- **NMAMapGestureTypeDoubleTap**
A double tap.
- **NMAMapGestureTypeLongPress**
A long press.
- **NMAMapGestureTypePan**
A pan.
- **NMAMapGestureTypePinch**
A pinch.
- **NMAMapGestureTypeTap**
A tap.
- **NMAMapGestureTypeTwoFingerTap**
A two-finger tap.
- **NMAMapGestureTypeAll**
All geture types.

NMAMapObjectType

Include: NMKit.framework/headers/NMAMapObject.h

Defines types of [NMAMapObject](#) objects that an application can add to a map.

Enumeration Members:

- **NMAMapObjectTypeMarker**
A map marker.
- **NMAMapObjectTypePolygon**
A polygon.
- **NMAMapObjectTypePolyline**
A polyline.
- **NMAMapObjectTypeRoute**
A route.
- **NMAMapObjectTypeContainer**
A map container to hold other map objects.
- **NMAMapObjectTypeCircle**
A circle.
- **NMAMapObjectTypeReserved**
A reserved object.
- **NMAMapObjectTypeUnknown**
An unknown object.

NMAMapPPI

Include: NMAKit.framework/headers/NMAMapView.h

Defines PPI of the map tiles.

Enumeration Members:

- **NMAMapPPILow**
Default PPI, suitable for mobile devices.
- **NMAMapPPIHigh**
High PPI.
- **NMAMapPPIReserved**
Internal use only.

Public Variable Details

FOUNDATION_EXPORT NSString *const NMAMapSchemeHybridDay

Include: NMAKit.framework/headers/NMAMapScheme.h

Satellite tile scheme presented in "day" colors, with roads.

FOUNDATION_EXPORT NSString *const NMAMapSchemeNormalDay

Include: NMAKit.framework/headers/NMAMapScheme.h

Normal scheme presented in "day" colors.

FOUNDATION_EXPORT NSString *const NMAMapSchemeNormalNight

Include: NMAKit.framework/headers/NMAMapScheme.h

Normal scheme presented in "night" colors.

FOUNDATION_EXPORT NSString *const NMAMapSchemePedestrianDay

Include: NMAKit.framework/headers/NMAMapScheme.h

Pedestrian navigation scheme presented in "day" colors.

■ Note:

When using this scheme it is recommended to enable pedestrian features on the map view.

FOUNDATION_EXPORT NSString *const NMAMapSchemePedestrianNight

Include: NMAKit.framework/headers/NMAMapScheme.h

Pedestrian navigation scheme presented in "night" colors.

■ **Note:**

When using this scheme it is recommended to enable pedestrian features on the map view.

FOUNDATION_EXPORT NSString *const NMAMapSchemeReducedDay

Include: NMAKit.framework/headers/NMAMapScheme.h

Normal scheme presented in "day" with less color palette.

FOUNDATION_EXPORT NSString *const NMAMapSchemeReducedNight

Include: NMAKit.framework/headers/NMAMapScheme.h

Normal scheme presented in "night" with less color palette.

FOUNDATION_EXPORT NSString *const NMAMapSchemeSatelliteDay

Include: NMAKit.framework/headers/NMAMapScheme.h

Satellite tile scheme presented in "day" colors.

■ **Note:**

No roads are shown. For satellite and roads use [*NMAMapSchemeHybridDay*](#)

FOUNDATION_EXPORT NSString *const NMAMapSchemeTerrainDay

Include: NMAKit.framework/headers/NMAMapScheme.h

Terrain bitmap scheme presented in "day" colors.

FOUNDATION_EXPORT const float NMAMapViewPreserveValue

Include: NMAKit.framework/headers/NMAMapView.h

This value may be passed into the map move method [*setGeoCenter:zoomLevel:withAnimation:*](#) in place of zoom in order to preserve the current value of that property.

See also:

[*setGeoCenter:zoomLevel:withAnimation:*](#)

<NMAMapGestureDelegate>

Protocol Summary

protocol <NMAMapGestureDelegate>

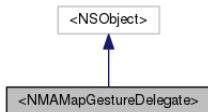
Derived from: <NSObject>

Delegate protocol for map gesture handling.

Include: NMAMapKit.framework/headers/NMAMapGesture.h

Inheritance Diagrams

Figure 33: Public inheritance diagram for <NMAMapGestureDelegate> – <NMAMapGestureDelegate>



[For complete information, see the section [Protocol Details](#)]

Instance Method Summary

Table 21: Instance Methods

Instance Method Summary
<pre>- @optional (void) mapView:(nonnull NMAMapView *) mapView didReceiveDoubleTapAtLocation: (_:@at:mapViewDidReceiveDoubleTap)</pre> <p>Delegate callback for an NMAMapView double-tap gesture</p>
<pre>- @optional (void) mapView:(nonnull NMAMapView *) mapView didReceiveLongPressAtLocation: (_:@at:mapViewDidReceiveLongPress)</pre> <p>Delegate callback for an NMAMapView long press gesture</p>
<pre>- @optional (void) mapView:(nonnull NMAMapView *) mapView didReceivePan:(CGPoint) translation atLocation:(:_:translation:@at:mapViewDidReceivePan)</pre> <p>Delegate callback for an NMAMapView pan gesture.</p>
<pre>- @optional (void) mapView:(nonnull NMAMapView *) mapView didReceivePinch:(float) pinch atLocation:(:_:pinch:@at:mapViewDidReceivePinch)</pre> <p>Delegate callback for an NMAMapView pinch gesture.</p>
<pre>- @optional (void) mapView:(nonnull NMAMapView *) mapView didReceiveTapAtLocation: (_:@at:mapViewDidReceiveTap)</pre> <p>Delegate callback for an NMAMapView tap gesture</p>
<pre>- @optional (void) mapView:(nonnull NMAMapView *) mapView didReceiveTwoFingerTapAtLocation: (_:@at:mapViewDidReceiveTwoFingerTap)</pre> <p>Delegate callback for an NMAMapView two finger-tap gesture</p>

Protocol Details

Delegate protocol for map gesture handling.

This protocol should be implemented by any class which acts as the gesture delegate for a map view or any map subview which needs to receive gestures from the map view.

The `NMAMapView` may optionally be provided with a gesture delegate, set via the `gestureDelegate` property. This delegate can be used to receive events about gestures.

■ **Note:**

Methods of this protocol will be called on the main queue.

Instance Method Details

- `@optional (void) mapView:(nonnull NMAMapView *) mapView didReceiveDoubleTapAtLocation:(_:at:mapViewDidReceiveDoubleTap)`

Delegate callback for an `NMAMapView` double-tap gesture.

Parameters:

- **mapView**

The `NMAMapView` on which the gesture occurred.

- **location**

The screen location in points at which the gesture occurred.

- `@optional (void) mapView:(nonnull NMAMapView *) mapView didReceiveLongPressAtLocation:(_:at:mapViewDidReceiveLongPress)`

Delegate callback for an `NMAMapView` long press gesture.

Parameters:

- **mapView**

The `NMAMapView` on which the gesture occurred.

- **location**

The screen location in points at which the gesture occurred.

- `@optional (void) mapView:(nonnull NMAMapView *) mapView didReceivePan:(CGPoint) translation atLocation:(_:translation:at:mapViewDidReceivePan)`

Delegate callback for an `NMAMapView` pan gesture.

The translation of a pan gesture is reported as the distance covered by the gesture since the last callback or since the beginning of the gesture. Note that this differs from the behaviour of `UIPanGestureRecognizer`, which reports the total translation since the beginning of the gesture.

Parameters:

- **mapView**

The `NMAMapView` on which the gesture occurred.

- **translation**

The current translation of the pan gesture, in points.

- **location**

The screen location in points at which the gesture occurred.

```
- @optional (void) mapView:(nonnull NMAMapView *) mapView didReceivePinch:(float) pinch atLocation:(_:pinch:at:mapViewDidReceivePinch)
```

Delegate callback for an [NMAMapView](#) pinch gesture.

The current pinch change is measured as the relative distance change of the two touch points since the last pinch callback or the beginning of the gesture. For example, if the pinch value is 1.2, the two touches are 20% farther apart than they were at the time of the last update. Note that this differs from the behaviour of UIPinchGestureRecognizer, which reports the total pinch since the beginning of the gesture.

Parameters:

- **mapView**

The [NMAMapView](#) on which the gesture occurred.

- **pinch**

The current relative pinch change of the gesture.

- **location**

The screen location in points at which the gesture occurred.

```
- @optional (void) mapView:(nonnull NMAMapView *) mapView didReceiveTapAtLocation:(_:at:mapViewDidReceiveTap)
```

Delegate callback for an [NMAMapView](#) tap gesture.

Parameters:

- **mapView**

The [NMAMapView](#) on which the gesture occurred.

- **location**

The screen location in points at which the gesture occurred.

```
- @optional (void) mapView:(nonnull NMAMapView *) mapView didReceiveTwoFingerTapAtLocation:(_:at:mapViewDidReceiveTwoFingerTap)
```

Delegate callback for an [NMAMapView](#) two finger-tap gesture.

Parameters:

- **mapView**

The [NMAMapView](#) on which the gesture occurred.

- **location**

The screen location in points at which the gesture occurred.

<NMAMapTileLayerDataSource>

Protocol Summary

protocol <NMAMapTileLayerDataSource>

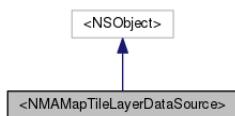
Derived from: <NSObject>

An interface for providing tile bitmap data to an NMAMapTileLayer instance.

Include: NMAKit.framework/headers/NMAMapTileLayer.h

Inheritance Diagrams

Figure 34: Public inheritance diagram for <NMAMapTileLayerDataSource> – <NMAMapTileLayerDataSource>



[For complete information, see the section [Protocol Details](#)]

Instance Method Summary

Table 22: Instance Methods

Instance Method Summary
<pre>- @optional (BOOL) mapTileLayer:(nonnull NMAMapTileLayer *) mapTileLayer hasTileAtX:(NSUInteger) x y:(NSUInteger) y zoomLevel:(NSUInteger) zoomLevel</pre> <p>Indicates if the specified tile should be rendered.</p>
<pre>- @optional (nonnull NSData *) mapTileLayer:(nonnull NMAMapTileLayer *) mapTileLayer requestDataForTileAtX:(NSUInteger) x y:(NSUInteger) y zoomLevel:(NSUInteger) zoomLevel</pre> <p>Requests raw tile bitmap data for the specified tile.</p>
<pre>- @optional (nonnull NSString *) mapTileLayer:(nonnull NMAMapTileLayer *) mapTileLayer urlForTileAtX:(NSUInteger) x y:(NSUInteger) y zoomLevel:(NSUInteger) zoomLevel</pre> <p>Returns a URL from which the tile bitmap can be downloaded.</p>

Protocol Details

An interface for providing tile bitmap data to an [NMAMapTileLayer](#) instance.

■ Note:

The methods of this protocol will be called on an internal NMA map rendering queue. Blocking this queue for extended periods of time will adversely effect map rendering performance.

Instance Method Details

- @optional (BOOL) mapTileLayer:(nonnull [NMAMapTileLayer](#) *) mapTileLayer hasTileAtX:(NSUInteger) x y:(NSUInteger) y zoomLevel:(NSUInteger) zoomLevel

Indicates if the specified tile should be rendered.

If this method returns NO then a further call to request tile bitmap data or tile URL will not be made. Not implementing this protocol method is equivalent to returning YES.

At each zoomLevel the framework expects the "world" to be rendered on $(2^{\text{zoomLevel}})^2$ tiles.

The x and y parameters indicate which tile is being requested for the given zoom level:

- At zoom level 3 the framework will ask you for 8 x 8 tiles
- At zoom level 4 the framework will ask you for 16 x 16 tiles
- ...

■ Note:

This method is called on an internal NMA map rendering queue. Blocking for extended periods of time will impact map rendering performance.

Parameters:

- **mapTileLayer**
NMAMapTileLayer instance that is requesting the tile.
- **x**
X coordinate of the tile being requested in the range 1.. $(2^{\text{zoomLevel}})$.
- **y**
Y coordinate of the tile being requested in the range 1.. $(2^{\text{zoomLevel}})$.
- **zoomLevel**

Zoom level of the tile being requested in the range NMAMapViewMinimumZoomLevel..NMAMapViewMaximumZoomLevel. The zoomLevel parameter value may NOT have the same value as the zoomLevel property of the [NMAMapView](#) to which the [NMAMapTileLayer](#) has been added.

Returns:

YES if the requested tile should be rendered, NO otherwise.

- @optional (nonnull NSData *) mapTileLayer:(nonnull [NMAMapTileLayer](#) *) mapTileLayer requestDataForTileAtX:(NSUInteger) x y:(NSUInteger) y zoomLevel:(NSUInteger) zoomLevel

Requests raw tile bitmap data for the specified tile.

Implement this method if you wish to supply raw tile bitmap data. This method will be called whenever the map view needs bitmap data to render the specified tile.

See [mapTileLayer:hasTileAtX:y:zoomLevel:](#) for a detailed description of the x, y and zoomLevel parameters.

Implementing this method will prevent `mapTileLayer:urlForTileAtX:y:zoomLevel:` and `mapTileLayer:requestTileAtX:y:zoomLevel:tileRequest` from being called.

■ Note:

This method is called on an internal NMA map rendering queue. Blocking for extended periods of time will impact map rendering performance.

Parameters:

- **mapTileLayer**

NMAMapTileLayer instance that is requesting the tile bitmap.

- **x**

X coordinate of the tile being requested in the range 1..(2^{zoomLevel}).

- **y**

Y coordinate of the tile being requested in the range 1..(2^{zoomLevel}).

- **zoomLevel**

Zoom level of the tile being requested in the range

`NMAMapViewMinimumZoomLevel..NMAMapViewMaximumZoomLevel`. The `zoomLevel` parameter value may NOT have the same value as the `zoomLevel` property of the *NMAMapView* to which the *NMAMapTileLayer* has been added.

Returns:

Raw tile bitmap data.

```
- @optional (nonnull NSString *) mapTileLayer:(nonnull NMAMapTileLayer *) mapTileLayer urlForTileAtX:(NSUInteger) x y:(NSUInteger) y zoomLevel:(NSUInteger) zoomLevel
```

Returns a URL from which the tile bitmap can be downloaded.

Implement this method if you wish to have the framework download tile bitmap data from a URL you provide. This method will be called whenever the map view needs bitmap data to render the specified tile.

This method is provided as a convenience for the common case of downloading bitmaps tiles from a URL. The data returned from the URL must be in either png and jpg format.

See `mapTileLayer:hasTileAtX:y:zoomLevel:` for a detailed description of the x, y and zoomLevel parameters.

■ Note:

This method will not be called if you also implement
`mapTileLayer:requestTileAtX:y:zoomLevel:tileRequest` or `mapTileLayer:requestTileAtX:y:zoomLevel`

■ Note:

This method is called on an internal NMA map rendering queue. Blocking for extended periods of time will impact map rendering performance.

Parameters:

- **mapTileLayer**

[NMAMapTileLayer](#) instance that is requesting the tile URL.

- **x**
X coordinate of the tile being requested in the range 1..(2^{zoomLevel}).
- **y**
Y coordinate of the tile being requested in the range 1..(2^{zoomLevel}).
- **zoomLevel**
Zoom level of the tile being requested in the range NMAMapViewMinimumZoomLevel..NMAMapViewMaximumZoomLevel. The zoomLevel parameter value may NOT have the same value as the zoomLevel property of the [NMAMapView](#) to which the [NMAMapTileLayer](#) has been added.

Returns:

URL to fetch the requested tile from.

<NMAMapViewDelegate>

Protocol Summary

protocol <NMAMapViewDelegate>

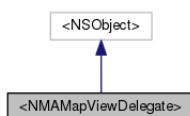
Derived from: <NSObject>

Represents a delegate that offers listeners and callback methods related to visible actions that result from user interaction.

Include: NMKit.framework/headers/NMAMapView.h

Inheritance Diagrams

Figure 35: Public inheritance diagram for <NMAMapViewDelegate> – <NMAMapViewDelegate>



[For complete information, see the section [Protocol Details](#)]

Instance Method Summary

Table 23: Instance Methods

Instance Method Summary
<pre>- @optional (void) mapView:(NMAMapView *) mapView didSelectObjects:(_:objects:mapViewDidSelectObjects)</pre> <p>A callback indicating that NMAMapObject map objects have been selected</p>

Instance Method Summary

- @optional (void) `mapViewDidBeginAnimation:(NMAMapView *) mapView`

A callback indicating the beginning of an animation on the map

- @optional (void) `mapViewDidBeginMovement:(NMAMapView *) mapView`

A callback indicating that a map movement has begun due to user interaction

- @optional (void) `mapViewDidDraw:(NMAMapView *) mapView`

A callback indicating that map data has been drawn

- @optional (void) `mapViewDidEndAnimation:(NMAMapView *) mapView`

A callback indicating the end of an animation on the map

- @optional (void) `mapViewDidEndMovement:(NMAMapView *) mapView`

A callback indicating that a map movement has ended

Protocol Details

Represents a delegate that offers listeners and callback methods related to visible actions that result from user interaction.

■ Note:

Methods of this protocol are called on the main queue.

Instance Method Details

- @optional (void) `mapView:(NMAMapView *) mapView didSelectObjects:(NSArray<NMAMapObject> *) objects`

A callback indicating that NMAMapObject map objects have been selected.

Parameters:

• **mapView**

The `NMAMapView` instance sending the callback

• **objects**

A NSArray of selected map objects

- @optional (void) `mapViewDidBeginAnimation:(NMAMapView *) mapView`

A callback indicating the beginning of an animation on the map.

Note that this callback is invoked after one of the following operations starts:

- setting the map center (with animation), for example with `setGeoCenter:zoomLevel:withAnimation:` or `setGeoCenter:withAnimation:`
- setting the map zoom level (with animation). E.g. with `setZoomLevel:withAnimation:`

- @optional (void) `mapViewDidBeginMovement:(NMAMapView *) mapView`

A callback indicating that a map movement has begun due to user interaction.

■ **Note:**

This callback will be sent at the start of the gesture when the map is panned or pinch zoomed(only for user gestures).

■ **Note:**

If the map receives more than one of these gestures at the same time, only a single callback will be sent until all map movement has stopped.

- `@optional (void) mapViewDidDraw:(NMAMapView *) mapView`

A callback indicating that map data has been drawn.

- `@optional (void) mapViewDidEndAnimation:(NMAMapView *) mapView`

A callback indicating the end of an animation on the map.

This is fired after one of the following operations ends:

- setting the map center (with animation). E.g. with `NMAMapView setCenter:`
- setting the map zoom level (with animation). E.g. with `setZoomLevel:withAnimation:`

- `@optional (void) mapViewDidEndMovement:(NMAMapView *) mapView`

A callback indicating that a map movement has ended.

■ **Note:**

There will be a one-to-one correspondence with this callback and the `mapViewDidBeginMovement` callback.

NMAMapCircle

Class Summary

class `NMAMapCircle`

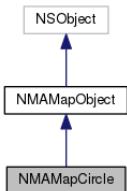
Derived from: `NMAMapObject`

Represents a NMAMapObject in the shape of a circle.

Include: `NMAKit.framework/headers/NMAMapCircle.h`

Inheritance Diagrams

Figure 36: Public inheritance diagram for NMAMapCircle



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 24: Public Properties

Public Properties
<p>[readable, writable, strong] <code>NMAGeoCoordinates * center</code> The NMAGeoCoordinates representing the center of the NMAMapCircle</p>
<p>[readable, writable, strong] <code>UIColor * fillColor</code> The fill color for the NMAMapCircle</p>
<p>[readable, writable, strong] <code>UIColor * lineColor</code> The line color for the NMAMapCircle</p>
<p>[readable, writable, assign] <code>NSUInteger lineWidth</code> The line width, in pixels, for the NMAMapCircle border</p>
<p>[readable, writable, assign] <code>double radius</code> The radius of the NMAMapCircle, in meters</p>

Instance Method Summary

Table 25: Instance Methods

Instance Method Summary
<p><code>- (instancetype) initWithGeoCoordinates:(NMAGeoCoordinates *) coordinates init:</code> Initializes a NMAMapCircle instance with the specified NMAGeoCoordinates as the center and a radius of 1 meter</p>
<p><code>- (instancetype) initWithGeoCoordinates:(NMAGeoCoordinates *) coordinates radius:</code> <code>(coordinates:radius:init)</code> Initializes a NMAMapCircle instance with the specified NMAGeoCoordinates as the center and the specified double as the radius (in meters)</p>

Class Method Summary

Table 26: Class Methods

Class Methods
<pre>+ (instancetype) mapCircleWithGeoCoordinates:(NMAGeoCoordinates *) coordinates</pre> <p>A convenience method to construct a NMAMapCircle at specified coordinates</p>
<pre>+ (instancetype) mapCircleWithGeoCoordinates:(NMAGeoCoordinates *) coordinates radius:(double) radius</pre> <p>A convenience method to construct a NMAMapCircle at specified coordinates and with a given radius</p>

Class Details

Represents a [NMAMapObject](#) in the shape of a circle. NMAMapCircle is helper class for creating and updating a [NMAMapPolygon](#) representing a circle.

Public Property Details

[readable, writable, strong] `NMAGeoCoordinates * center`

The NMAGeoCoordinates representing the center of the NMAMapCircle.

■ Note:

The NMAViewObject location property will return the same coordinates.

[readable, writable, strong] `UIColor * fillColor`

The fill color for the NMAMapCircle.

■ Note:

The default value (in RGBA) is 0x00498A55

[readable, writable, strong] `UIColor * lineColor`

The line color for the NMAMapCircle.

■ Note:

The default value (in RGBA) is 0xC00066FF.

[readable, writable, assign] `NSUInteger lineWidth`

The line width, in pixels, for the NMAMapCircle border. The supported value range is [0..100].

■ Note:

The default value is 0 pixels

[readable, writable, assign] double **radius**

The radius of the NMAMapCircle, in meters.

■ **Note:**

The default value is 1 meter

Instance Method Details

- (instancetype) initWithGeoCoordinates:(NMAGeoCoordinates *) coordinates radius:(double) radius init)

Initializes a NMAMapCircle instance with the specified NMAGeoCoordinates as the center and a radius of 1 meter.

Parameters:

- **coordinates**

A NMAGeoCoordinates representing the NMAMapCircle center

Returns:

The NMAMapCircle

- (instancetype) initWithGeoCoordinates:(NMAGeoCoordinates *) coordinates radius:(double) radius init)

Initializes a NMAMapCircle instance with the specified NMAGeoCoordinates as the center and the specified double as the radius (in meters).

Parameters:

- **coordinates**

A NMAGeoCoordinates representing the NMAMapCircle center

- **radius**

The radius of the circle

Returns:

The NMAMapCircle

Class Method Details

+ (instancetype) mapCircleWithGeoCoordinates:(NMAGeoCoordinates *) coordinates

A convenience method to construct a *NMAMapCircle* at specified coordinates.

Parameters:

- **coordinates**

The location of the map circle center

```
+ (instancetype) mapCircleWithGeoCoordinates:(NMAGeoCoordinates *)  
coordinates radius:(double) radius
```

A convenience method to construct a *NMAMapCircle* at specified coordinates and with a given radius.

Parameters:

- **coordinates**
The location of the map circle center
- **radius**
The radius (in meters) of the map circle

NMAMapContainer

Class Summary

class **NMAMapContainer**

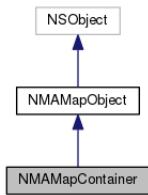
Derived from: *NMAMapObject*

A container of *NMAMapObject* instances, containers can be added to a *NMAMapView* instance.

Include: *NMAKit.framework/headers/NMAMapContainer.h*

Inheritance Diagrams

Figure 37: Public inheritance diagram for NMAMapContainer



[For complete information, see the section *Class Details*]

Public Property Summary

Table 27: Public Properties

Public Properties
<pre>[readable, copy] NSArray< __kindof <i>NMAMapObject</i> * > * <i>mapObjects</i></pre> <p>The NSArray of <i>NMAMapObject</i> objects within the <i>NMAMapContainer</i></p>

Instance Method Summary

Table 28: Instance Methods

Instance Method Summary
<p><code>- (BOOL) <i>addMapObject:</i>(_:add)</code></p> <p>Adds a NMAMapObject to the NMAMapContainer.</p>
<p><code>- (BOOL) <i>removeAll</i></code></p> <p>Removes all NMAMapObject objects from the NMAMapContainer</p>
<p><code>- (BOOL) <i>removeMapObject:</i>(_:remove)</code></p> <p>Removes the specified NMAMapObject from the NMAMapContainer</p>

Class Details

A container of [NMAMapObject](#) instances, containers can be added to a [NMAMapView](#) instance.

Applications can, as necessary, add or remove certain types of NMAMapObject objects to or from a NMAMapContainer (refer to [addMapObject:](#) for details). A NMAMapContainer is also useful for grouping together NMAMapObject objects to allow application operations on entire groups of map objects without requiring an iterative approach to inspecting or changing the state of each individual NMAMapObject (e.g. for changing their visibility).

You can use the `visible` property to control whether the objects in the container are visible. If it's set to YES, then the visibility settings of each map object in the container are used. But unlike other map objects, NMAMapContainer does not support the use of visibility masks, so you cannot set container visibility on a zoom level basis.

Note:

The `zOrder` of an [NMAMapObject](#) instance in a container is relative only to other [NMAMapObject](#) instances within the same container. Note that containers are derived from [NMAMapObject](#) also have `zOrder`.

Public Property Details

[readable, copy] NSArray< __kindof [NMAMapObject](#) * > * `mapObjects`

The NSArray of [NMAMapObject](#) objects within the NMAMapContainer.

Instance Method Details

`- (BOOL) addMapObject:(_:add)`

Adds a NMAMapObject to the NMAMapContainer.

NMAMapObject types that can be added to a NMAMapContainer are:

- [NMAMapMarker](#)
- [NMAMapCircle](#)

- [*NMAMapPolygon*](#)
- [*NMAMapPolyline*](#)

■ Note:

Attempts to add an object twice to the same container will be ignored.

Parameters:

- **aMapObject**

A NMAMapObject to be added to the NMAMapContainer.

Returns:

YES if the NMAMapObject was added successfully, NO otherwise.

See also:

[*removeMapObject*](#):

- (BOOL) removeAll

Removes all [*NMAMapObject*](#) objects from the NMAMapContainer.

Returns:

YES if all NMAMapObject objects were removed successfully, NO otherwise.

- (BOOL) removeMapObject:(_:remove)

Removes the specified NMAMapObject from the NMAMapContainer.

■ Note:

If the specified NMAMapObject is not contained within the NMAMapContainer, the container will be unchanged.

Parameters:

- **aMapObject**

A NMAMapObject to remove from the NMAMapContainer.

Returns:

YES if the NMAMapObject was removed successfully, NO otherwise.

See also:

[*removeAllMapObjects*](#)

NMAMapInfoBubbleCustomizationContext

Class Summary

class NMAMapInfoBubbleCustomizationContext

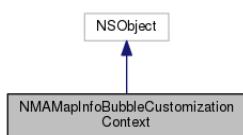
Derived from: NSObject

Responsible for all the customizable values for an info bubble.

Include: NMAKit.framework/headers/NMAMapInfoBubbleCustomizationContext.h

Inheritance Diagrams

Figure 38: Public inheritance diagram for NMAMapInfoBubbleCustomizationContext



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 29: Public Properties

Public Properties
<p>[readable, writable, assign] CGFloat bubbleAlpha The alpha of the info bubble</p>
<p>[readable, writable, assign] CGFloat bubbleCornerRadius The radius to use when drawing the corners</p>
<p>[readable, writable, strong] UIColor * bubbleFillColor The default value is white</p>
<p>[readable, writable, assign] CGFloat bubbleLeftRightMargin Outside margin for the whole bubble shape</p>
<p>[readable, writable, assign] CGFloat bubbleLineWidth The width of the border line</p>
<p>[readable, writable, assign] CGFloat bubbleMaxWidth The maximum width of the info bubble</p>
<p>[readable, writable, assign] CGFloat bubblePadding Left, top, right and bottom margin for the text area,</p>
<p>[readable, writable, strong] UIColor * bubbleStrokeColor The color of the border line</p>



Public Properties

[readable, writable, assign] CGFloat **bubbleTailHeight**

The default value is 8

[readable, writable, assign] CGFloat **bubbleTailWidth**

The default value is 8

[readable, writable, strong] UILabel * **descriptionLabel**

The default description title label uses 11

[readable, writable, strong] UILabel * **titleLabel**

The default title label uses 12

Class Details

Responsible for all the customizable values for an info bubble.

Public Property Details

[readable, writable, assign] CGFloat **bubbleAlpha**

The alpha of the info bubble. The default value is 1.0.

[readable, writable, assign] CGFloat **bubbleCornerRadius**

The radius to use when drawing the corners. For rounded corners use a nonzero value. The default value is 0.0, i.e no rounded corners.

[readable, writable, strong] UIColor * **bubbleFillColor**

The default value is white.

[readable, writable, assign] CGFloat **bubbleLeftRightMargin**

Outside margin for the whole bubble shape. When the info bubble is on the map view there should be left and right margin at least equal to this value. If not, the text will be multiline. The default value is 25.0.

[readable, writable, assign] CGFloat **bubbleLineWidth**

The width of the border line. The default value is 1.0.

[readable, writable, assign] CGFloat **bubbleMaxWidth**

The maximum width of the info bubble. If the text don't fit into this width, the text will be multiline. The default value is 150.0.

[readable, writable, assign] `CGFloat bubblePadding`

Left, top, right and bottom margin for the text area, i.e. title & description. The default value is 4.0.

[readable, writable, strong] `UIColor * bubbleStrokeColor`

The color of the border line. The default value is black.

[readable, writable, assign] `CGFloat bubbleTailHeight`

The default value is 8.0.

[readable, writable, assign] `CGFloat bubbleTailWidth`

The default value is 8.0.

[readable, writable, strong] `UILabel * descriptionLabel`

The default description title label uses 11.0 system font, has black text and clear background colors and its text alignment is justified. All the other settings keeps the UILabel default values.

[readable, writable, strong] `UILabel * titleLabel`

The default title label uses 12.0 bold system font, has black text and clear background colors and its text alignment is justified. All the other settings keeps the UILabel default values.

NMAMapInfoBubbleCustomizer

Class Summary

class **NMAMapInfoBubbleCustomizer**

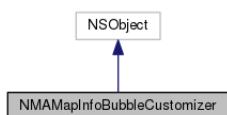
Derived from: `NSObject`

Responsible for all the customization possible for the map info bubbles.

Include: `NMAKit.framework/headers/NMAMapInfoBubbleCustomizer.h`

Inheritance Diagrams

Figure 39: Public inheritance diagram for `NMAMapInfoBubbleCustomizer`



[For complete information, see the section [Class Details](#)]

Class Method Summary

Table 30: Class Methods

Class Methods
<pre>+ (void) customize:(nullable NMAMapInfoBubbleCustomizationContext *) context</pre> <p>Customizes all the map info bubbles with the specified context</p>

Class Details

Responsible for all the customization possible for the map info bubbles.

Class Method Details

```
+ (void) customize:(nullable NMAMapInfoBubbleCustomizationContext *) context
```

Customizes all the map info bubbles with the specified context.

Parameters:

- **context**

The customization context. When it is nil, reverts back to the default map info bubbles.

NMAMapMarker

Class Summary

class NMAMapMarker

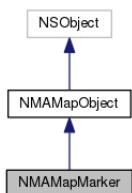
Derived from: [NMAMapObject](#)

Represents a marker used to display an icon on a geographical position on a map.

Include: NMAMapKit.framework/headers/NMAMapMarker.h

Inheritance Diagrams

Figure 40: Public inheritance diagram for NMAMapMarker



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 31: Public Properties

Public Properties
<p>[readable, writable, assign] <code>CGPoint anchorOffset</code> The <code>CGPoint</code> offset of the <code>NMAMapMarker</code> from its anchored location.</p>
<p>[readable, writable, strong] <code>NMAGeoCoordinates * coordinates</code> The geographical coordinates at which the <code>NMAMapMarker</code> is displayed on a map</p>
<p>[readable, writable, strong] <code>UIImage * icon</code> The <code>UIImage</code> displayed at the marker location</p>
<p>[readable, writable, assign] <code>NMAInfoBubbleEventBlock infoBubbleEventBlock</code> This block is called when the info bubble is tapped</p>
<p>[readable, writable, assign] <code>NSString * textDescription</code> Description text for the <code>NMAMapMarker</code></p>
<p>[readable, writable, assign] <code>NSString * title</code> The current title for the default info bubble</p>

Instance Method Summary

Table 32: Instance Methods

Instance Method Summary
<p><code>- (BOOL) hideInfoBubble</code> Hide the info bubble from <code>NMAMapView</code>.</p>
<p><code>- (nonnull instancetype) initWithGeoCoordinates:(nonnull NMAGeoCoordinates *) coordinates</code> Initializes a <code>NMAMapMarker</code> instance with the specified <code>NMAGeoCoordinates</code></p>
<p><code>- (nonnull instancetype) initWithGeoCoordinates:(nonnull NMAGeoCoordinates *) coordinates image:(nullable UIImage *) image</code> Initializes a <code>NMAMapMarker</code> instance with the specified <code>NMAGeoCoordinates</code> and <code>UIImage</code> for the displayed icon</p>
<p><code>- (CGPoint) setAnchorOffsetUsingLayoutPosition:(NMALayoutPosition) position</code></p>
<p><code>- (BOOL) showInfoBubble</code> Display the default info bubble if the <code>NMAMapMarker</code>'s title has been set to something non-nil. The info bubble is displayed only after the <code>NMAMapMarker</code> is added onto a <code>NMAMapView</code>, otherwise making this call would make no effects. Also a non-zero length of the title or the <code>textDescription</code> is required.</p>
<p><code>- (BOOL) showInfoBubble:(nonnull UIView *) view</code> Display the <code>view</code> passed as an info bubble.</p>

Class Method Summary

Table 33: Class Methods

Class Methods
<pre>+ (nonnull instancetype) mapMarkerWithGeoCoordinates:(nonnull NMAGeoCoordinates *) coordinates</pre> <p>A convenience method to construct an NMAMapMarker at the specified coordinates</p>
<pre>+ (nonnull instancetype) mapMarkerWithGeoCoordinates:(nonnull NMAGeoCoordinates *) coordinates image:(nullable UIImage *) image</pre> <p>A convenience method to construct an NMAMapMarker at the specified coordinates and with a UIImage icon</p>

Class Details

Represents a marker used to display an icon on a geographical position on a map. The map handles proper placement of icons on the screen as well as panning.

■ **Note:**

Application users can select NMAMapMarker objects.

Public Property Details

[readable, writable, assign] `CGPoint anchorOffset`

The CGPoint offset of the NMAMapMarker from its anchored location.

Use this property to reposition a NMAMapMarker as needed. The x and y offset values are measured in points. Positive offset values move the NMAMapMarker down and to the right, while negative values move it up and to the left.

If no offset is provided, the NMAMapMarker is centered on its anchor; that is, the center of the marker image is located at the map-space representation of the marker's world-space location (its anchor, an NMAGeoCoordinates).

[readable, writable, strong] `NMAGeoCoordinates * coordinates`

The geographical coordinates at which the NMAMapMarker is displayed on a map.

[readable, writable, strong] `UIImage * icon`

The UIImage displayed at the marker location.

[readable, writable, assign] `NMAInfoBubbleEventBlock infoBubbleEventBlock`

This block is called when the info bubble is tapped. Default is nil.

[readable, writable, assign] `NSString * textDescription`

Description text for the NMAMapMarker. This is shown below the title. Default is nil.

[readable, writable, assign] **NSString * title**

The current title for the default info bubble. Default is nil.

Instance Method Details

- (BOOL) hideInfoBubble

Hide the info bubble from NMAMapView.

Returns:

YES If the info bubble is successfully hidden from NMAMapView. NO if the info bubble is not attached to this NMAMapMarker.

- (nonnull instancetype) initWithGeoCoordinates:(nonnull NMAGeoCoordinates *) coordinates

Initializes a NMAMapMarker instance with the specified NMAGeoCoordinates.

Parameters:

• **coordinates**

A NMAGeoCoordinates representing the map coordinates for displaying the NMAMapMarker

Returns:

The NMAMapMarker

- (nonnull instancetype) initWithGeoCoordinates:(nonnull NMAGeoCoordinates *) coordinates image:(nullable UIImage *) image

Initializes a NMAMapMarker instance with the specified NMAGeoCoordinates and UIImage for the displayed icon.

Parameters:

• **coordinates**

A NMAGeoCoordinates representing the map coordinates for displaying the NMAMapMarker

• **image**

A UIImage the icon.

Returns:

The NMAMapMarker

`- (CGPoint) setAnchorOffsetUsingLayoutPosition:(NMALayoutPosition) position`

A convenience function to set where the NMAMapMarker is attached to its anchor

Parameters:

- **position**

A NMAnchorLocation for anchoring the marker icon.

Returns:

The CGPoint offset value that was applied to the anchorOffset property

`- (BOOL) showInfoBubble`

Display the default info bubble if the NMAMapMarker's title has been set to something non-nil. The info bubble is displayed only after the NMAMapMarker is added onto a NMAMapView, otherwise making this call would make no effects. Also a non-zero length of the title or the textDescription is required.

As only one info bubble can be displayed on the map at a time, thus when calling showInfoBubble on a different NMAMapMarker, the current info bubble on the display will automatically be closed before the newly selected one is shown.

Returns:

YES If the MapMarker meets all the requirements to show the bubble.

`- (BOOL) showInfoBubble:(nonnull UIView *) view`

Display the view passed as an info bubble.

Parameters:

- **view**

The view to be used as the info bubble view.

Returns:

YES If the MapMarker meets all the requirements to show the bubble.

See also:

[showInfoBubble](#)

Class Method Details

`+ (nonnullinstancetype) mapMarkerWithGeoCoordinates:(nonnull NMAGeoCoordinates *) coordinates`

A convenience method to construct an [NMAMapMarker](#) at the specified coordinates.

Parameters:

- **coordinates**

The `NMAGeoCoordinates` at which to create the map marker.

```
+ (nonnull instancetype) mapMarkerWithGeoCoordinates:(nonnull  
NMAGeoCoordinates *) coordinates image:(nullable UIImage *) image
```

A convenience method to construct an `NMAMapMarker` at the specified coordinates and with a `UIImage` icon.

Parameters:

- **coordinates**
The `NMAGeoCoordinates` at which to create the map marker.
- **image**
The `UIImage` to use for the marker icon.

NMAMapObject

Class Summary

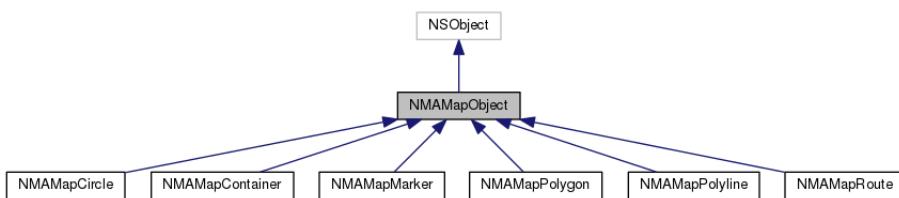
class `NMAMapObject`

Derived from: `NSObject`

Include: `NMAKit.framework/headers/NMAMapObject.h`

Inheritance Diagrams

Figure 41: Public inheritance diagram for `NMAMapObject`



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 34: Public Properties

Public Properties
<code>[readable, weak] NMAMapContainer * parent</code> The map object's container

Public Properties

[readable, assign] `NMAMapObjectType` `type`

The map object's type

[readable, writable, assign] `BOOL` `visible`

Indicates whether the visibility of the `NMAMapObject` on the map is enabled

[readable, writable, assign] `NSUInteger` `zIndex`

The `NSUInteger` z-index (stacking order) for the `NMAMapObject` within its containing `NMAMapContainer`, or within the `NMAMapView` if the object was added to `NMAMapView` directly.

Instance Method Summary

Table 35: Instance Methods

Instance Method Summary

`- (BOOL) isVisibleAtZoomLevel:(atZoomLevel:isVisible)`

Determines whether the map object is set to visible at the specified zoom level when the map object's visible property is set to YES

`- (void) setVisibility:(BOOL) visible atZoomLevel:(_:atZoomLevel:setVisibility)`

Sets the visibility of the map object at a specific zoom level.

`- (void) setVisibility:(BOOL) visible fromZoomLevel:(float) fromZoomLevel toZoomLevel:(_:fromZoomLevel:toZoomLevel:setVisibility)`

Sets the visibility of the map object for a range of zoom levels.

`- (NSUInteger) uniqueId`

Returns the unique ID value of the `NMAMapObject`

Class Details

Public Property Details

[readable, weak] `NMAMapContainer * parent`

The map object's container. This will be nil if the map object does not belong to a container or was added to an `NMAMapView` directly.

[readable, assign] `NMAMapObjectType` `type`

The map object's type.

[readable, writable, assign] `BOOL` `visible`

Indicates whether the visibility of the NMAMapObject on the map is enabled. Whether the NMAMapObject is actually visible depends on whether the visible mask is set for the desired zoom level.

■ **Note:**

The getter is `isVisible`.

■ **Note:**

This property is independent but takes priority over the per-zoom level settings. For example, you can call `[setVisible:YES fromZoomLevel:0 toZoomLevel:6]` and then set `visible` to NO, and the object will not appear on the map. Setting `visible` to YES will then cause the object to be visible in zoom levels 0 to 6.

[readable, writable, assign] NSUInteger `zIndex`

The NSUInteger z-index (stacking order) for the `NMAMapObject` within its containing `NMAMapContainer`, or within the `NMAMapView` if the object was added to `NMAMapView` directly.

Must be in the range `NMAMapObjectMinimumZIndex..NMAMapObjectMaximumZIndex`. The property will be clamped to this range if invalid values are specified.

■ **Note:**

Objects with the highest value are placed at the top of the stacking order.

Instance Method Details

`- (BOOL) isVisibleAtZoomLevel:(atZoomLevel:isVisible)`

Determines whether the map object is set to visible at the specified zoom level when the map object's `visible` property is set to YES.

Parameters:

- `zoomLevel`

for checking whether the map object is visible. Will be clamped to the range `NMAMapViewMinimumZoomLevel..NMAMapViewMaximumZoomLevel`. This is rounded off to the nearest integer. e.g. 14.1 means [14..15]

`- (void) setVisibility:(BOOL) visible atZoomLevel:(_:atZoomLevel:setVisibility)`

Sets the visibility of the map object at a specific zoom level.

This method can only be used after the `NMAMapObject` has been added to the map. By default, the visibility for all zoom levels are set to YES.

■ **Note:**

This will take effect if/when the map object's `visible` property is set YES.

Parameters:

- **visible**

indicates whether the map object should be visible.

- **zoomLevel**

for applying the specified visibility. Will be clamped to the range

NMAMapViewMinimumZoomLevel..NMAMapViewMaximumZoomLevel. This is rounded off to the nearest integer. e.g. 14.1 means [14..15)

```
- (void) setVisibility:(BOOL) visible fromZoomLevel:(float) fromZoomLevel  
toZoomLevel:(_:fromZoomLevel:toZoomLevel:setVisibility)
```

Sets the visibility of the map object for a range of zoom levels.

This method can only be used after the [NMAMapObject](#) has been added to the map. By default, the map object is visible at all zoom levels. Each zoom level retains its own visibility state.

If you make the following calls:

```
#1. visible property is set to YES  
|11111111111111111111|  
#2. setVisible:NO fromZoomLevel:10 toZoomLevel:16  
      10      16  
      |0000000|  
#3. setVisible:YES fromZoomLevel:5 toZoomLevel:13  
      5      13  
      |111111111|  
#4. setVisible:NO atZoomLevel:20
```

The result is a map object that is invisible at zoom levels 14-16 and 20 and visible at the others.

```
      5      14 17 20  
|111111111111110001110|
```

■ **Note:**

This will take effect if/when the map object's **visible** property is set YES.

Parameters:

- **visible**

indicates whether the map object should be visible.

- **fromZoomLevel**

clamped to range NMAMapViewMinimumZoomLevel..NMAMapViewMaximumZoomLevel, rounded off to the nearest integer.

- **toZoomLevel**

clamped to range NMAMapViewMinimumZoomLevel..NMAMapViewMaximumZoomLevel, rounded off to the nearest integer.

```
- (NSUInteger) uniqueId
```

Returns the unique ID value of the NMAMapObject.

■ **Note:**

This value will be consistent for the lifetime of the object.

Returns:

The unique ID

NMAMapPolygon

Class Summary

class NMAMapPolygon

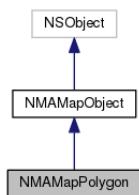
Derived from: NMAMapObject

Represents a NMAMapObject in the shape of a polygon.

Include: NMAKit.framework/headers/NMAMapPolygon.h

Inheritance Diagrams

Figure 42: Public inheritance diagram for NMAMapPolygon



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 36: Public Properties

Public Properties
<p>[readable, writable, strong] UIColor * <i>fillColor</i> The fill color for the NMAMapPolygon</p>
<p>[readable, writable, strong] UIColor * <i>lineColor</i> The color for the NMAMapPolygon border line</p>
<p>[readable, writable, assign]NSUInteger <i>lineWidth</i> The line width, in pixels, for the NMAMapPolygon border</p>
<p>[readable, copy] NSArray< NMAGeoCoordinates * > * <i>vertices</i> The NSArray of NMAGeoCoordinates objects comprising the path that defines the NMAMapPolygon shape</p>

Instance Method Summary

Table 37: Instance Methods

Instance Method Summary
<code>- (void) addVertex:(NMAGeoCoordinates *) vertex atIndex:(NSInteger)at add:</code> Insert a NMAGeoCoordinates point at the specified index of the NMAMapPolygon path
<code>- (void) appendVertex:(NMAGeoCoordinates *) vertex</code> Append a NMAGeoCoordinates path point to the end of the NMAMapPolygon
<code>- (void) clearVertices</code> Removes all vertices from the NMAMapPolygon's path
<code>- (BOOL) containsGeoCoordinates:(NMAGeoCoordinates *) coordinates</code> Determines if the NMAMapPolygon contains the input NMAGeoCoordinates
<code>- (instancetype) initWithVertices:(NSArray<NMAGeoCoordinates * > *) vertices</code> Initializes a NMAMapPolygon instance with the specified NSArray of NMAGeoCoordinates objects
<code>- (BOOL) isValid</code> Determines whether the NMAMapPolygon object is valid
<code>- (void) removeLast</code>
<code>- (void) removeVertexAtIndex:(NSInteger)at remove:</code> Remove a NMAGeoCoordinates point at the specified index of the NMAMapPolygon path

Class Method Summary

Table 38: Class Methods

Class Methods
<code>+ (instancetype) mapPolygonWithVertices:(NSArray<NMAGeoCoordinates * > *) vertices</code> A convenience method for constructing an NMAMapPolygon

Class Details

Represents a [NMAMapObject](#) in the shape of a polygon. In comparison to a [NMAMapPolyline](#), it is assumed that the last coordinates within the path is connected with the first coordinates, thereby constructing an enclosed geometry.

Note:

The NMAMapObject location property for an [NMAMapPolygon](#) returns the coordinates of the first vertex (if available).

Public Property Details

[readable, writable, strong] UIColor * **fillColor**

The fill color for the NMAMapPolygon.

[readable, writable, strong] UIColor * **lineColor**

The color for the NMAMapPolygon border line.

■ **Note:**

Attempts to set this property to nil are ignored.

[readable, writable, assign] NSUInteger **lineWidth**

The line width, in pixels, for the NMAMapPolygon border.

The line width must be in the range [0, 100]. The default width is 1 pixel.

[readable, copy] NSArray< *NMAGeoCoordinates* * > * **vertices**

The NSArray of NMAGeoCoordinates objects comprising the path that defines the NMAMapPolygon shape.

Instance Method Details

- (void) addVertex:(*NMAGeoCoordinates* *) vertex atIndex:(*_*:at:add)

Insert a NMAGeoCoordinates point at the specified index of the NMAMapPolygon path.

■ **Note:**

If index is equal or greater than the total number of vertices, the vertex will be appended to the end of the polygon path.

Parameters:

- **vertex**
A NMAGeoCoordinates path point to add.
- **index**
Position to insert. Index starts with 0.

- (void) appendVertex:(*_*:append)

Append a NMAGeoCoordinates path point to the end of the NMAMapPolygon.

Parameters:

- **vertex**
A NMAGeoCoordinates path point to append

- (void) clearVertices

Removes all vertices from the *NMAMapPolygon*'s path.

`- (BOOL) containsGeoCoordinates:(:_contains)`

Determines if the NMAMapPolygon contains the input NMAGeoCoordinates. Points at the peaks or corners of the polygon are considered outside the polygon.

Parameters:

- **coordinates**

NMAGeoCoordinates to test. The altitude is ignored.

Returns:

YES if the NMAGeoCoordinates is inside the NMAMapPolygon, NO otherwise

`- (instancetype) initWithVertices:(NSArray< NMAGeoCoordinates * > *) vertices`

Initializes a NMAMapPolygon instance with the specified NSArray of *NMAGeoCoordinates* objects.

Parameters:

- **vertices**

A NSArray of NMAGeoCoordinates points defining the NMAMapPolygon shape

Returns:

The NMAMapPolygon

`- (BOOL) isValid`

Determines whether the NMAMapPolygon object is valid. A polygon is valid as long as it does not self-intersect.

Returns:

YES if the NMAMapPolygon is valid, NO otherwise

`- (void) removeLast`

Remove the last NMAGeoCoordinates point of the NMAMapPolygon

`- (void) removeVertexAtIndex:(at:)remove`

Remove a NMAGeoCoordinates point at the specified index of the NMAMapPolygon path.

Parameters:

- **index**

Position to remove. Index starts with 0 and must be smaller than the total number of vertices.

Class Method Details

```
+ (instancetype) mapPolygonWithVertices:(NSArray *) vertices
```

A convenience method for constructing an [NMAMapPolygon](#).

Parameters:

- **vertices**

An array of [NMAGeoCoordinates](#) which define the polygon

NMAMapPolyline

Class Summary

class **NMAMapPolyline**

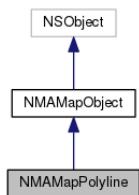
Derived from: [NMAMapObject](#)

Represents a NMAMapObject in the shape of a polyline.

Include: `NMAKit.framework/headers/NMAMapPolyline.h`

Inheritance Diagrams

Figure 43: Public inheritance diagram for NMAMapPolyline



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 39: Public Properties

Public Properties
<code>[readable, writable, strong] UIColor * <i>lineColor</i></code> The color for the NMAMapPolyline
<code>[readable, writable, assign]NSUInteger <i>lineWidth</i></code> The width, in pixels, for the NMAMapPolyline
<code>[readable, copy] NSArray< NMAGeoCoordinates * > * <i>vertices</i></code> The NSArray of NMAGeoCoordinates that comprise the path of the NMAMapPolyline

Instance Method Summary

Table 40: Instance Methods

Instance Method Summary
<pre>- (void) addVertex:(NMAGeoCoordinates *) vertex atIndex:(NSInteger)at add</pre> <p>Insert a NMAGeoCoordinates point at the specified index of the NMAMapPolyline path</p>
<pre>- (void) appendVertex:(NMAGeoCoordinates *) vertex</pre> <p>Append a NMAGeoCoordinates point to the end of the NMAMapPolyline path</p>
<pre>- (void) clearVertices</pre> <p>Removes all vertices from the NMAMapPolyline's path</p>
<pre>- (NSInteger) indexOfNearestVertexToGeoCoordinates:(NMAGeoCoordinates *) vertex</pre> <p>Determines the index of the NMAMapPolyline path array that is closest to the specified NMAGeoCoordinates object</p>
<pre>- (instancetype) initWithVertices:(NSArray<NMAGeoCoordinates * > *) vertices</pre> <p>Initializes a NMAMapPolyline instance with the specified NSArray of NMAGeoCoordinates objects</p>
<pre>- (NMAGeoCoordinates *) nearestVertexToGeoCoordinates:(NMAGeoCoordinates *) vertex</pre> <p>Determines the NMAGeoCoordinates along the NMAMapPolyline path that is closest to the specified NMAGeoCoordinates object</p>
<pre>- (void) removeLastVertex</pre> <p>Remove the last NMAGeoCoordinates point of the NMAMapPolyline path</p>
<pre>- (void) removeVertexAtIndex:(NSInteger)at remove</pre> <p>Remove a NMAGeoCoordinates point at the specified index of the NMAMapPolyline path</p>

Class Method Summary

Table 41: Class Methods

Class Methods
<pre>+ (instancetype) mapPolylineWithVertices:(NSArray *) vertices</pre> <p>A convenience method for constructing an NMAMapPolyline</p>

Class Details

Represents a [NMAMapObject](#) in the shape of a polyline.

A NMAMapPolyline has multiple points that combine to create its path and which are traversed in order. The line between two consecutive points is always straight and is defined by the shortest navigable way to move between them.

Note:

The NMAViewObject location property for an [NMAMapPolyline](#) returns the coordinates of the first vertex (if available).

Public Property Details

[readable, writable, strong] `UIColor * lineColor`

The color for the NMAMapPolyline.

■ **Note:**

Attempts to set this property to nil are ignored.

[readable, writable, assign] `NSUInteger lineWidth`

The width, in pixels, for the NMAMapPolyline.

The line width must be in the range [0, 100]. The default width is 1 pixel.

[readable, copy] `NSArray< NMAGeoCoordinates * > * vertices`

The NSArray of *NMAGeoCoordinates* that comprise the path of the NMAMapPolyline.

Instance Method Details

`- (void) addVertex:(NMAGeoCoordinates *) vertex atIndex:(NSInteger)atIndex`

Insert a NMAGeoCoordinates point at the specified index of the NMAMapPolyline path.

■ **Note:**

If index is equal or greater than the total number of vertices, the vertex will be appended to the end of the polyline path.

Parameters:

- **vertex**
A NMAGeoCoordinates path point to add.
- **index**
Position to insert. Index starts with 0.

`- (void) appendVertex:(NMAGeoCoordinates *) vertex`

Append a NMAGeoCoordinates point to the end of the NMAMapPolyline path.

Parameters:

- **vertex**
A NMAGeoCoordinates path point to append

`- (void) clearVertices`

Removes all vertices from the *NMAMapPolyline*'s path.

`- (NSInteger) indexOfNearestVertexToGeoCoordinates:(to:NSInteger) nearestVertex`

Determines the index of the NMAMapPolyline path array that is closest to the specified NMAGeoCoordinates object.

Parameters:

- **coordinates**

A NMAGeoCoordinates to test for nearness to the NMAMapPolyline

Returns:

The index of the NMAGeoCoordinates vertex

`- (instancetype) initWithVertices:(NSArray< NMAGeoCoordinates * > *) vertices`

Initializes a NMAMapPolyline instance with the specified NSArray of *NMAGeoCoordinates* objects.

Parameters:

- **vertices**

A NSArray of NMAGeoCoordinates points defining the NMAMapPolyline

Returns:

The NMAMapPolyline

`- (NMAGeoCoordinates *) nearestVertexToGeoCoordinates:(to:nearestVertex)`

Determines the NMAGeoCoordinates along the NMAMapPolyline path that is closest to the specified NMAGeoCoordinates object.

Parameters:

- **coordinates**

A NMAGeoCoordinates to test for nearness to the NMAMapPolyline

Returns:

The NMAGeoCoordinates vertex along the NMAMapPolyline path

`- (void) removeLastVertex`

Remove the last NMAGeoCoordinates point of the NMAMapPolyline path.

`- (void) removeVertexAtIndex:(at:remove)`

Remove a NMAGeoCoordinates point at the specified index of the NMAMapPolyline path.

Parameters:

- **index**

Position to remove. Index starts with 0 and must be smaller than the total number of vertices.

Class Method Details

`+ (instancetype) mapPolylineWithVertices:(NSArray *) vertices`

A convenience method for constructing an [NMAMapPolyline](#).

Parameters:

- **vertices**

An array of [NMAGeoCoordinates](#) which define the polyline

NMAMapRoute

Class Summary

class **NMAMapRoute**

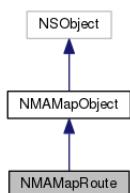
Derived from: [NMAMapObject](#)

Represents a [NMAMapObject](#) in the shape of a route that can be displayed on a map.

Include: NMAKit.framework/headers/NMAMapRoute.h

Inheritance Diagrams

Figure 44: Public inheritance diagram for NMAMapRoute



[For complete information, see the section [Class Details](#)]

See also:

[NMARoute](#)

Public Property Summary

Table 42: Public Properties

Public Properties
<code>[readable, writable, strong] UIColor * color</code> The UIColor representing the color of the NMAMapRoute

Public Properties

[readable, writable, assign] BOOL ***maneuverNumberDisplayed***

Indicates whether the maneuver number is displayed on the NMAMapRoute

[readable, writable, strong] ***NMARoute * route***

The NMARoute represented by the NMAMapRoute

Instance Method Summary

Table 43: Instance Methods

Instance Method Summary

- (instancetype) ***UNAVAILABLE_ATTRIBUTE***

Use mapRouteWithRoute to instantiate a NMAMapRoute

Class Method Summary

Table 44: Class Methods

Class Methods

+ (instancetype) ***mapRouteWithRoute:(NMARoute *) route***

A convenience constructor for NMAMapRoute

Class Details

Represents a ***NMAMapObject*** in the shape of a route that can be displayed on a map.

In order to display the route object on the map, the route object needs to be added to an ***NMAMapView*** using its addMapObject: method. The NMAViewObject location property for a ***NMAMapRoute*** returns the coordinates of the first waypoint.

See also:

[NMARoute](#)

Public Property Details

[readable, writable, strong] UIColor * ***color***

The UIColor representing the color of the NMAMapRoute.

■ Note:

Alpha-value transparency is supported and will also be set when setting this property.

[readable, writable, assign] BOOL ***maneuverNumberDisplayed***

Indicates whether the maneuver number is displayed on the NMAMapRoute.

[readable, writable, strong] `NMARoute * route`

The `NMARoute` represented by the `NMAMapRoute`.

Instance Method Details

`- (instancetype) UNAVAILABLE_ATTRIBUTE`

Use `mapRouteWithRoute` to instantiate a `NMAMapRoute`.

Class Method Details

`+ (instancetype) mapRouteWithRoute:(NMARoute *) route`

A convenience constructor for `NMAMapRoute`.

Parameters:

- `route`

The route object from which to construct the `NMAMapRoute`.

NMAMapScheme

Class Summary

class `NMAMapScheme`

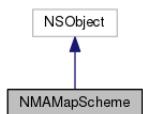
Derived from: `NSObject`

Represents a collection of schemes that a `NMAMapView` supports.

Include: `NMAKit.framework/headers/NMAMapScheme.h`

Inheritance Diagrams

Figure 45: Public inheritance diagram for `NMAMapScheme`



[For complete information, see the section [Class Details](#)]

Class Method Summary

Table 45: Class Methods

Class Methods
<code>+ (BOOL) isValid:(NSString *) scheme</code> Determines whether the specified scheme is supported (is valid for the application)

Class Details

Represents a collection of schemes that a [NMAMapView](#) supports. Each of the schemes in this class can be set by way of the [NMAMapView::mapScheme](#) property.

Class Method Details

`+ (BOOL) isValid:(NSString *) scheme`

Determines whether the specified scheme is supported (is valid for the application).

Parameters:

- `scheme`

A NSString representing the scheme to check

Returns:

YES if the scheme is valid, NO otherwise

NMAMapTileLayer

Class Summary

class NMAMapTileLayer

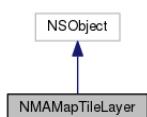
Derived from: NSObject

A layer of custom raster tiles for display in an [NMAMapView](#).

Include: NMAKit.framework/headers/NMAMapTileLayer.h

Inheritance Diagrams

Figure 46: Public inheritance diagram for NMAMapTileLayer



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 46: Public Properties

Public Properties
[readable, writable, assign] <code>NMAGeoBoundingBox *</code> <i>boundingBox</i> Specifies the <code>NMAGeoBoundingBox</code> within which tiles will be requested and rendered.
[readable, assign] BOOL <i>cacheEnabled</i> Specifies whether tile bitmaps will be cached to disk by the framework.
[readable, assign] NSString * <i>cacheldentifier</i> Identifier used to associate disk cached tile data with a <code>NMAMapTileLayer</code> instance to prevent clashes in the cache folder.
[readable, writable, assign]NSUInteger <i>cacheSizeLimit</i> Specifies the maximum size the cache will consume on disk.
[readable, writable, assign] NSTimeInterval <i>cacheTimeToLive</i> Specifies the cache expiration time in seconds.
[readable, writable, weak] id< <code>NMAMapTileLayerDataSource</code> > <i>dataSource</i>
[readable, assign] BOOL <i>locked</i> Indicates if the tile layer is locked
[readable, writable, assign] NSUInteger <i>zIndex</i> Specifies the Z-index (stacking order) for the bitmap tiles within the map layer specified by the <code>mapLayerType</code> property.

Instance Method Summary

Table 47: Instance Methods

Instance Method Summary
<code>- (void) clearCache</code> Clear the file system cache identified by the <code>cacheldentifier</code> property
<code>- (void) hideAtZoomLevel:(atZoomLevel:hide)</code> Hide the raster tiles at the specified zoom level.
<code>- (void) hideFromZoomLevel:(int) fromLevel toZoomLevel:(fromZoomLevel:toZoomLevel:hide)</code> Hide the raster tiles at the specified zoom level range.
<code>- (BOOL) isShownAtZoomLevel:(atZoomLevel:isShown)</code> Returns whether tiles are visible at the specified zoom level.
<code>- (void) setCacheEnabled:(BOOL) enabled withIdentifier:(enabled:identifier:setCache)</code> /brief Enable/Disable caching of tile data to disk
<code>- (void) showAtZoomLevel:(atZoomLevel:show)</code> Show the raster tiles at the specified zoom level.

Instance Method Summary

```
- (void) showFromZoomLevel:(int) fromLevel toZoomLevel:(fromZoomLevel:toZoomLevel:show)
```

Show the raster tiles at the specified zoom level range.

Class Details

A layer of custom raster tiles for display in an [NMAMapView](#).

Raster tiles are supplied as bitmap data and can be supplied synchronously or simply by providing a URL from which to download the tiles from.

To use this class create an instance, configure the properties and call `NMAMapView::addMapTileLayer`.

■ Note:

IMPORTANT! The properties of this interface should not be modified after the instance has been added to an [NMAMapView](#). See the [NMAMapTileLayer::locked](#) property.

Public Property Details

[readable, writable, assign] [NMAGeoBoundingBox](#) * **boundingBox**

Specifies the [NMAGeoBoundingBox](#) within which tiles will be requested and rendered.

Tiles falling fully outside the `boundingBox` will not be requested.

This property defaults to an unbounded value - i.e. tiles are rendered across the world.

■ Note:

This property cannot be set when the instance is locked. See [NMAMapTileLayer::locked](#).

[readable, assign] BOOL **cacheEnabled**

Specifies whether tile bitmaps will be cached to disk by the framework.

File system caching is provided as a convenience. If caching is enabled tiles will not be re-requested via `NMAMapTileLayerDelegate` until they have expired.

This property defaults to NO. Caching is enabled by calling `setCacheEnabled:withIdentifier:`.

[readable, assign] NSString * **cacheIdentifier**

Identifier used to associate disk cached tile data with a [NMAMapTileLayer](#) instance to prevent clashes in the cache folder.

You must use unique cache identifiers to ensure there will be no filename clashes in the cache folder structure. It also allows you to easily identify the cache in the filesystem.

This property is set when you enable caching. See `setCacheEnabled:withIdentifier:`.

[readable, writable, assign]NSUInteger **cacheSizeLimit**

Specifies the maximum size the cache will consume on disk.

The framework may allow the cache size to grow an additional 5MB to the size specified in order to reduce disk access.

This property defaults to 0 which means the cache size is not limited.

■ **Note:**

This property has no effect if the cacheEnabled property is set to NO. Also, this property cannot be set when the instance is locked. See [NMAMapTileLayer::locked](#).

[readable, writable, assign] NSTimeInterval **cacheTimeToLive**

Specifies the cache expiration time in seconds.

Tile bitmaps will expire cacheTimeToLive seconds after they have been downloaded causing them to be re-requested via [NMAMapTileLayerDataSource](#) when the map needs to render them.

This property defaults to 0 which means the cached tiles never expire.

■ **Note:**

This property has no effect if the cacheEnabled property is set to NO. Negative values will be reset to 0.

[readable, writable, weak] id< [NMAMapTileLayerDataSource](#) > **dataSource**

Data source that provides tile bitmap data for the [NMAMapTileLayer](#)

■ **Note:**

This property cannot be set when the instance is locked. See [NMAMapTileLayer::locked](#).

[readable, assign] BOOL **locked**

Indicates if the tile layer is locked.

The tile layer is locked when it is added to an [NMAMapView](#) instance. While the tile layer is locked attempts to set any properties will be ignored. The tile layer is unlocked when it is removed from an [NMAMapView](#) instance.

[readable, writable, assign] NSUInteger **zIndex**

Specifies the Z-index (stacking order) for the bitmap tiles within the map layer specified by the mapLayerType property.

All objects within a map layer have a Z-index associated with them. Objects with the highest value are placed at the top of the stacking order. If two or more objects within a map layer have the same z-index value their stacking order is undefined.

Z-index values range from NMAMapObjectMinimumZIndex to NMAMapObjectMaximumZIndex. The property will be clamped to this range if invalid values are specified.

This property defaults to NMAMapObjectMinimumZIndex.

■ **Note:**

This property cannot be set when the instance is locked. See [NMAMapTileLayer::locked](#).

Instance Method Details

`-(void) clearCache`

Clear the file system cache identified by the cacheIdentifier property.

■ **Note:**

To guarantee the cache is cleared this method should be called after the [NMAMapTileLayer](#) instance has been removed from the [NMAMapView](#)

`-(void) hideAtZoomLevel:(atZoomLevel:hide)`

Hide the raster tiles at the specified zoom level.

You can control the tile visibility for each zoom level independently. For example, tiles may be visible at levels 0, 5, 7 only. Tiles are shown at all zoom levels by default.

■ **Note:**

This method does nothing when the instance is locked. See [NMAMapTileLayer::locked](#).

Parameters:

- **zoomLevel**

Zoom level at which to hide the tiles. Values outside the range NMAMapViewMinimumZoomLevel..NMAMapViewMaximumZoomLevel will be ignored

`-(void) hideFromZoomLevel:(int) fromLevel toZoomLevel:(fromZoomLevel:toZoomLevel:hide)`

Hide the raster tiles at the specified zoom level range.

You can control the tile visibility for each zoom level independently. This method allows you set the visibility for a range of zoom levels in one call. Tiles are shown at all zoom levels by default.

The method will do nothing if ANY parameters fall outside the range

NMAMapViewMinimumZoomLevel..NMAMapViewMaximumZoomLevel or if fromLevel > toLevel.

■ **Note:**

This method does nothing when the instance is locked. See [NMAMapTileLayer::locked](#).

Parameters:

- **fromLevel**

Lower zoom level index at which to hide the tiles. Values outside the range NMAMapViewMinimumZoomLevel..NMAMapViewMaximumZoomLevel will be ignored

- **toLevel**

Upper zoom level index at which to hide the tiles. Values outside the range NMAMapViewMinimumZoomLevel..NMAMapViewMaximumZoomLevel will be ignored

- (BOOL) isShownAtZoomLevel:(atZoomLevel:isShown)

Returns whether tiles are visible at the specified zoom level.

You can control the tile visibility for each zoom level independently. This method allows you set the visibility for a range of zoom levels in one call. Tiles are shown at all zoom levels by default.

Parameters:

- **zoomLevel**

Zoom level. Values outside the range NMAMapViewMinimumZoomLevel..NMAMapViewMaximumZoomLevel will return NO.

Returns:

YES if tiles are shown at the specified zoom level, NO otherwise

- (void) setCacheEnabled:(BOOL) enabled withIdentifier:(enabled:identifier:setCache)

/brief Enable/Disable caching of tile data to disk.

File system caching is provided as a convenience. If caching is enabled tiles will not be re-requested via NMAMapTileLayerDelegate until they have expired.

It's important to always use the same cache identifier for your tile data. Otherwise, multiple disassociated cache folders will be created in the file system. This wastes disk space, and you will not benefit from persistent disk caching across sessions.

Note:

Calling this method changes the values of the cacheEnabled and cacheIdentifier properties.

Parameters:

- **cacheEnabled**

YES to enable caching, NO to disable.

- **cacheIdentifier**

You must use a unique cache identifier to ensure there will be no filename clashes in the cache folder structure. Passing a nil or empty string will result in caching NOT being enabled. The identifier is set the first time you call this method and will not be changed by subsequent calls to this method for the lifetime of the instance. If the identifier you supply contains non alphanumeric characters they will be stripped from the identifier, with the exception of "_" and "-".

- (void) showAtZoomLevel:(atZoomLevel:show)

Show the raster tiles at the specified zoom level.

You can control the tile visibility for each zoom level independently. For example, tiles may be visible at levels 0, 5, 7 only. Tiles are shown at all zoom levels by default.

■ **Note:**

This method does nothing when the instance is locked. See [NMAMapTileLayer::locked](#).

Parameters:

- **zoomLevel**

Zoom level at which to show the tiles. Values outside the range

NMAMapViewMinimumZoomLevel..NMAMapViewMaximumZoomLevel will be ignored

```
-(void) showFromZoomLevel:(int) fromLevel toZoomLevel:  
(fromZoomLevel:toZoomLevel:show)
```

Show the raster tiles at the specified zoom level range.

You can control the tile visibility for each zoom level independently. For example, tiles may be visible at levels 0, 5, 7 only. Tiles are shown at all zoom levels by default.

The method will do nothing if ANY parameters fall outside the range

NMAMapViewMinimumZoomLevel..NMAMapViewMaximumZoomLevel or if fromLevel > toLevel.

■ **Note:**

This method does nothing when the instance is locked. See [NMAMapTileLayer::locked](#).

Parameters:

- **fromLevel**

Lower zoom level index at which to show the tiles. Values outside the range

NMAMapViewMinimumZoomLevel..NMAMapViewMaximumZoomLevel will be ignored

- **toLevel**

Upper zoom level index at which to show the tiles. Values outside the range

NMAMapViewMinimumZoomLevel..NMAMapViewMaximumZoomLevel will be ignored

NMAMapView

Class Summary

class **NMAMapView**

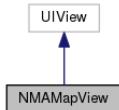
Derived from: `UIView`

A `UIView` subclass used by an application to display a geographical map.

Include: `NMAKit.framework/headers/NMAMapView.h`

Inheritance Diagrams

Figure 47: Public inheritance diagram for NMAMapView



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 48: Public Properties

Public Properties
<p>[readable, copy] <code>NMAGeoBoundingBox</code> * <code>boundingBox</code> The NMAGeoBoundingBox representing the current screen area of the NMAMapView</p>
<p>[readable, writable, assign] float <code>copyrightLogoHorizontalMargin</code> The distance in points that the copyright logo will be from the left or right edge of the <code>NMAMapView</code>.</p>
<p>[readable, writable, assign] <code>NMALayoutPosition</code> <code>copyrightLogoPosition</code> The copyright logo position for the NMAMapView.</p>
<p>[readable, writable, assign] float <code>copyrightLogoVerticalMargin</code> The distance in points that the copyright logo will be from the top or bottom edge of the <code>NMAMapView</code>.</p>
<p>[readable, writable, weak] id< <code>NMAMapViewDelegate</code> > <code>delegate</code> An event handler for the NMAMapView</p>
<p>[readable, assign] <code>NSString</code> * <code>displayLanguage</code> The language which the map is set to render</p>
<p>[readable, writable, assign] <code>NMAGeoCoordinates</code> * <code>geoCenter</code> The NMAGeoCoordinates of the world location corresponding to the current screen position of the transformCenter</p>
<p>[readable, writable, weak] id< <code>NMAMapGestureDelegate</code> > <code>gestureDelegate</code> An optional delegate to receive gestures events</p>
<p>[readable, writable, assign] BOOL <code>kineticPanningEnabled</code> Indicates the kinetic panning behaviour of the NMAMapView</p>
<p>[readable, writable, assign] <code>NSTimeInterval</code> <code>longPressDuration</code> The touch duration required to trigger a long press gesture</p>
<p>[readable, writable, assign] BOOL <code>mapCenterFixedOnZoom</code> Indicates whether the transform center is used on zoom gestures</p>
<p>[readable, writable, assign] BOOL <code>mapInteractionEnabled</code> Indicates the state of enabling and modifying map gesture interaction for the NMAMapView</p>

Public Properties

[readable, writable, assign] `NMAMapPPI` `mapPPI`

[readable, writable, strong] `NSString *` `mapScheme`

The scheme for the NMAMapView

[readable, assign] `NMAPositionIndicator *` `positionIndicator`

The position indicator for the NMAMapView

[readable, writable, assign] `BOOL` `renderAllowed`

A flag which can be used to allow or disallow map rendering

[readable, assign] `NSString *` `secondaryDisplayLanguage`

The secondary language which the map is set to render

[readable, writable, assign] `CGPoint` `transformCenter`

The point at which map movements and animations are centered

[readable, writable, assign] `BOOL` `useHighResolutionMap`

[readable, writable, assign] `float` `zoomLevel`

The zoom level for the NMAMapView

Instance Method Summary

Table 49: Instance Methods

Instance Method Summary

`- (BOOL) addMapObject:(__:)add`

Adds a NMAMapObject to the NMAMapView

`- (BOOL) addMapObjects:(objects:)add`

Adds a NSArray of NMAMapObject objects to the NMAMapView

`- (void) addMapTileLayer:(mapTileLayer:)add`

Adds a NMAMapTileLayer to the map view for rendering custom raster tiles

`- (void) disableMapGestures:(NSUInteger) gestures`

Disable one or more NMAMapView gestures

`- (void) enableMapGestures:(NSUInteger) gestures`

Enable one or more NMAMapView gestures

`- (nullable NMAGeoCoordinates *) geoCoordinatesFromPoint:(from:)geoCoordinates`

Converts a specified on-screen point to an equivalent NMAGeoCoordinates object

`- (BOOL) isMapGestureEnabled:(NMAMapGestureType) gesture`

Queries whether or not a gesture type is enabled on the NMAMapView

`- (NSArray< NMAMapTileLayer * > *) mapTileLayers`

Returns an array of NMAMapTileLayer instances that have been added to the map view

HERE iOS SDK Developer's Guide

► API reference



Instance Method Summary

`- (NSArray< __kindof NMAMapObject * > *) objectsAtPoint:(at:objects)`

Returns a NSArray of all NMAMapObject objects that are selected at a specified point on the screen

`- (double) pointDistanceFromGeoCoordinates:(NMAGeoCoordinates *) startCoordinates toGeoCoordinates:(from:to:pointDistance)`

Calculates the point distance between two geo coordinates

`- (CGPoint) pointFromGeoCoordinates:(from:point)`

Converts a specified NMAGeoCoordinates object to its equivalent point in screen space.

`- (BOOL) removeMapObject:(:_:remove)`

Removes an existing NMAMapObject from the NMAMapView

`- (BOOL) removeMapObjects:(objects:remove)`

Removes a NSArray of NMAMapObject objects from the NMAMapView

`- (void) removeMapTileLayer:(mapTileLayer:remove)`

Removes a NMAMapTileLayer from the map view

`- (void) setBoundingBox:(NMAGeoBoundingBox *) boundingBox insideRect:(CGRect) screenRect withAnimation:(boundingBox:inside:animation:set)`

Transforms the map to fit an arbitrary geo bounding box with the bounds of a specific screen region with optional animation

`- (void) setBoundingBox:(NMAGeoBoundingBox *) boundingBox withAnimation:(boundingBox:animation:set)`

Sets the geo bounding box of the map view with optional animation

`- (void) setGeoCenter:(NMAGeoCoordinates *) geoCenter withAnimation:(geoCenter:animation:set)`

Sets the geoCenter of the map with optional animation

`- (void) setGeoCenter:(NMAGeoCoordinates *) geoCenter zoomLevel:(float) zoomLevel withAnimation:(geoCenter:zoomLevel:animation:set)`

Simultaneously sets one or more map transformation properties with optional animation

`- (void) setGeoCoordinates:(NMAGeoCoordinates *) coordinates toPoint:(CGPoint) point withAnimation:(coordinates:to:animation:set)`

Positions the NMAMapView so that the specified world location coincides with the specified screen location

`- (void) setGeoCoordinates:(NMAGeoCoordinates *) coordinates toPoint:(CGPoint) point withAnimation:(NMAMapAnimation) animation zoomLevel:(coordinates:to:animation:zoomLevel:set)`

Positions the NMAMapView so that the specified world location coincides with the specified screen location

`- (void) setZoomLevel:(float) zoomLevel withAnimation:(zoomLevel:animation:set)`

Sets the zoomLevel of the map with optional animation

`- (BOOL) useDefaultDisplayLanguage`

Set the map display language using the system default locale

`- (BOOL) useDisplayLanguageFromLocale:(NSLocale *) locale`

Set the map display language using the locale specified



Instance Method Summary

`- (BOOL) useSecondaryDisplayLanguageFromLocale:(NSLocale *_Nullable) locale`

Set the secondary map display language using the locale specified

`- (NSArray< __kindof NMAMapObject * > *) visibleObjectsAtPoint:(at:visibleObjects)`

Returns a NSArray of all visible NMAMapObject objects that are selected at a specified point on the screen

Class Method Summary

Table 50: Class Methods

Class Methods

`+ (void) shouldBeginRenderingAutomatically:(BOOL) enabled`

Used to control the default NMAMapView rendering behaviour

Class Details

A UIView subclass used by an application to display a geographical map.

Public Property Details

[readable, copy] `NMAGeoBoundingBox * boundingBox`

The NMAGeoBoundingBox representing the current screen area of the `NMAMapView`.

The boundingBox of the map is the smallest possible `NMAGeoBoundingBox` which contains all of the map area currently visible on the screen.

■ Note:

The bounding box may be slightly inaccurate if the method is invoked while the map is moving.

[readable, writable, assign] `float copyrightLogoHorizontalMargin`

The distance in points that the copyright logo will be from the left or right edge of the `NMAMapView`.

■ Note:

Values less than 10.0 will be ignored.

■ Note:

Does not apply when the copyrightLogoPosition is NMALayoutPositionTopCenter or NMALayoutPositionBottomCenter.

[readable, writable, assign] `NMALayoutPosition copyrightLogoPosition`

The copyright logo position for the NMAMapView.

■ Note:

Valid values are NMALayoutPositionTopLeft, NMALayoutPositionTopCenter, NMALayoutPositionTopRight, NMALayoutPositionBottomLeft, NMALayoutPositionBottomCenter, and NMALayoutPositionBottomRight.

[readable, writable, assign] float **copyrightLogoVerticalMargin**

The distance in points that the copyright logo will be from the top or bottom edge of the [NMAMapView](#).

■ **Note:**

Values less than 10.0 will be ignored.

[readable, writable, weak] id< [NMAMapViewDelegate](#) > **delegate**

An event handler for the [NMAMapView](#).

An object may be installed as the [NMAMapView](#)'s delegate in order to respond to certain map events, such as object selection or map movement. See the [NMAMapViewDelegate](#) protocol for the full list of events.

[readable, assign] NSString * **displayLanguage**

The language which the map is set to render.

[readable, writable, assign] [NMAGeoCoordinates](#) * **geoCenter**

The NMAGeoCoordinates of the world location corresponding to the current screen position of the transformCenter.

[readable, writable, weak] id< [NMAMapGestureDelegate](#) > **gestureDelegate**

An optional delegate to receive gestures events.

See also:

[NMAMapGestureDelegate](#)

[readable, writable, assign] BOOL **kineticPanningEnabled**

Indicates the kinetic panning behaviour of the NMAMapView.

When kinetic panning is enabled, at the conclusion of a pan gesture (when the user's finger is removed from the device), the map will continue moving in the direction and with the speed of the original pan before gradually coming to a stop.

[readable, writable, assign] NSTimeInterval **longPressDuration**

The touch duration required to trigger a long press gesture.

■ **Note:**

The default value is 1 second.

[readable, writable, assign] BOOL `mapCenterFixedOnZoom`

Indicates whether the transform center is used on zoom gestures.

By default disabled. If enabled, zoom will always be applied using the current transform center, instead of a point relative the touch interaction. See the `transformCenter` property.

■ **Note:**

The transform center is usually the same as the screen's geometric center, except in cases where it makes sense to have it slightly modified (example: in guidance mode, the transform center is lowered a bit so that the route ahead is more visible in the map view).

[readable, writable, assign] BOOL `mapInteractionEnabled`

Indicates the state of enabling and modifying map gesture interaction for the `NMAMapView`.

If enabled, gesture handlers can be enabled to begin gesture handling. If disabled, all gesture handling is disabled and the gesture handler state cannot be modified.

[readable, writable, assign] `NMAMapPPI` `mapPPI`

[readable, writable, strong] `NSString *` `mapScheme`

The scheme for the `NMAMapView`.

See [NMAMapScheme](#) for supported schemes.

[readable, assign] `NMAPositionIndicator *` `positionIndicator`

The position indicator for the `NMAMapView`.

[readable, writable, assign] BOOL `renderAllowed`

A flag which can be used to allow or disallow map rendering.

■ **Note:**

Setting `renderAllowed` to NO guarantees that the map will not draw again (but it will finish if a frame is in the middle of rendering), but the map may not redraw with `renderAllowed` set to YES if it is prevented from drawing for another reason.

■ **Note:**

The map can be hinted to redraw (if allowed) by calling `setNeedsDisplay`.

[readable, assign] NSString * secondaryDisplayLanguage

The secondary language which the map is set to render.

[readable, writable, assign] CGPoint transformCenter

The point at which map movements and animations are centered.

The transformCenter is a point within the bounds of the map that is used to orient the map during transformations. For example, when one of the `setGeoCenter:zoomLevel:withAnimation:` and `setGeoCenter:withAnimation:` methods is used, the specified geocoordinates will be aligned with the transformCenter on the screen.

The transformCenter is specified in relative coordinates in the range {[0, 1], [0, 1]}, with {0, 0} representing the upper left corner of the map and {1, 1} the lower right corner.

■ **Note:**

The default value is {0.5, 0.5}.

■ **Note:**

The center point of map movements caused by gestures will depend on the touch locations of the gestures and not on the transformCenter.

■ **Note:**

Values on or very near the edge of the map will be ignored.

■ **Note:**

Values above the horizon will be ignored.

[readable, writable, assign] BOOL useHighResolutionMap

[readable, writable, assign] float zoomLevel

The zoom level for the NMAMapView.

■ **Note:**

Values must be in the range [NMAMapViewMinimumZoomLevel, NMAMapViewMaximumZoomLevel]; invalid values will be clamped to this range.

Instance Method Details

- (BOOL) addMapObject:(_:add)

Adds a NMAMapObject to the NMAMapView.

■ **Note:**

Returns NO if the object has already been added (duplicates are not allowed).

Parameters:

- **object**

A NMAMapObject to add

Returns:

YES if the map object was added successfully, NO otherwise

`- (BOOL) addMapObjects:(NSArray<NMAMapObject> *)objects add:(BOOL)add`

Adds a NSArray of NMAMapObject objects to the NMAMapView.

Parameters:

- **objects**

A NSArray of NMAMapObject objects to add

Returns:

YES if all map objects were added successfully, NO otherwise

`- (void) addMapTileLayer:(NMAMapTileLayer *)mapTileLayer add:(BOOL)add`

Adds a *NMAMapTileLayer* to the map view for rendering custom raster tiles.

Note:

Attempting to add a single *NMAMapTileLayer* instance to multiple *NMAMapView* instances is not supported and will result in undefined behavior.

Parameters:

- **tileLayer**

NMAMapTileLayer instance to be added to the map view. Ignores attempts to add the same *NMAMapTileLayer* instance twice.

`- (void) disableMapGestures:(NSUInteger) gestures`

Disable one or more *NMAMapView* gestures.

Note:

Pass NMAMapGestureTypeAll to disable all gestures.

Note:

Disabling a gesture type will also disable that type for the *NMAMapView*'s gestureDelegate.

Parameters:

- **gestures**

A bit-mask combination of one or more NMAMapGestureType values.

`- (void) enableMapGestures:(NSUInteger) gestures`

Enable one or more [NMAMapView](#) gestures.

■ **Note:**

All gestures are enabled by default.

■ **Note:**

Pass NMAMapGestureTypeAll to enable all gestures.

Parameters:

- **gestures**

A bit-mask combination of one or more NMAMapGestureType values.

`- (nullable NMAGeoCoordinates *) geoCoordinatesFromPoint:(from:geoCoordinates)`

Converts a specified on-screen point to an equivalent NMAGeoCoordinates object.

Parameters:

- **point**

A point on the screen to convert

Returns:

The NMAGeoCoordinates, or nil if the conversion failed

`- (BOOL) isMapGestureEnabled:(NMAMapGestureType) gesture`

Queries whether or not a gesture type is enabled on the NMAMapView.

Parameters:

- **gesture**

The gesture type to query.

`- (NSArray< NMAMapTileLayer * > *) mapTileLayers`

Returns an array of [NMAMapTileLayer](#) instances that have been added to the map view.

Returns:

NSArray of [NMAMapTileLayer](#) instances.

`- (NSArray< __kindof NMAMapObject * > *) objectsAtPoint:(at:objects)`

Returns a NSArray of all NMAMapObject objects that are selected at a specified point on the screen.

■ **Note:**

Use `isKindOfClass:` to further determine `NMAMapObject` type instead of `isMemberOfClass:`.

Parameters:

- **point**

An on-screen `CGPoint` specified in logical coordinates

Returns:

The `NSArray` of selected `NMAMapObject` objects

```
- (double) pointDistanceFromGeoCoordinates:(NMAGeoCoordinates *)  
startCoordinates toGeoCoordinates:(from:to:pointDistance)
```

Calculates the point distance between two geo coordinates.

Note:

The calculation will fail if the coordinates are too far apart or too far from the current location of the map.

Parameters:

- **startCoordinates**

Starting `NMAGeoCoordinates`

- **endCoordinates**

Ending `NMAGeoCoordinates`

Returns:

The point distance between the `NMAGeoCoordinates`, or `-CGFLOAT_MAX` if either of the `coordinates` parameters is `nil` or the calculation cannot be performed.

```
- (CGPoint) pointFromGeoCoordinates:(from:point)
```

Converts a specified `NMAGeoCoordinates` object to its equivalent point in screen space.

Valid geo coordinates will produce a valid point, even if the coordinates are not on the screen, as long as the coordinates are not too far from the current map location. To check if coordinates are within the bounds of the screen, use the `CGRectContainsPoint` method after making the conversion.

Parameters:

- **coordinates**

A `NMAGeoCoordinates` object to convert

Returns:

The on-screen `CGPoint`, or `CGPoint.x = CGPoint.y = -CGFLOAT_MAX` if the `NMAGeoCoordinates` object is `nil`

```
- (BOOL) removeMapObject:(_:remove)
```

Removes an existing `NMAMapObject` from the `NMAMapView`.

Parameters:

- **object**

A NMAMapObject to remove

Returns:

YES if the map object was removed successfully, NO otherwise

`- (BOOL) removeMapObjects:(NSArray<NMAMapObject> *) objects:remove`

Removes a NSArray of NMAMapObject objects from the NMAMapView.

Parameters:

- **objects**

A NSArray of NMAMapObject objects to remove

Returns:

YES if all map objects were removed successfully, NO otherwise

`- (void) removeMapTileLayer:(NMAMapTileLayer *) tileLayer:remove`

Removes a *NMAMapTileLayer* from the map view.

Parameters:

- **tileLayer**

NMAMapTileLayer instance to be removed from the map view. Does nothing if tileLayer has not been added to the map view yet.

`- (void) setBoundingBox:(NMAGeoBoundingBox *) boundingBox insideRect:(CGRect) screenRect withAnimation:(boundingBox:inside:animation:set)`

Transforms the map to fit an arbitrary geo bounding box with the bounds of a specific screen region with optional animation.

Passing the bounds of the map view to this method is equivalent to calling `setBoundingBox:withAnimation:`.

Parameters:

- **boundingBox**

A NMAGeoBoundingBox to display after the transformation

- **screenRect**

The region of the view inside which the bounding box will be located.

- **animationType**

The animation method used to perform the transformation.

```
- (void) setBoundingBox:(NMAGeoBoundingBox *) boundingBox withAnimation:(boundingBox:animation:set)
```

Sets the geo bounding box of the map view with optional animation.

After this method completes, the full extents of the specified geo bounding box will be visible in the map view. Depending on map orientation, this means that the boundingBox property may return a different value than the one passed to this method, because the map may be displaying some areas outside of the input bounding box.

Parameters:

- **boundingBox**
A NMAGeoBoundingBox to display after the transformation
- **animationType**
A NMAMapAnimation to illustrate the transformation

```
- (void) setGeoCenter:(NMAGeoCoordinates *) geoCenter withAnimation:(geoCenter:animation:set)
```

Sets the geoCenter of the map with optional animation.

Parameters:

- **geoCenter**
The *NMAGeoCoordinates* at which the map is centered.
- **animationType**
The animation method used to display the transformation.

```
- (void) setGeoCenter:(NMAGeoCoordinates *) geoCenter zoomLevel:(float) zoomLevel withAnimation:(geoCenter:zoomLevel:animation:set)
```

Simultaneously sets one or more map transformation properties with optional animation.

Any combination of new properties may be passed to this method. To leave a property unchanged, pass nil (geoCenter) or NMAMapViewPreserveValue (zoomLevel) for that property.

Parameters:

- **geoCenter**
The new geoCenter.
- **zoomLevel**
The new zoomLevel.
- **animationType**
The animation method used to display the transformation.

```
- (void) setGeoCoordinates:(NMAGeoCoordinates *) coordinates toPoint:  
(CGPoint) point withAnimation:(coordinates:to:animation:set)
```

Positions the *NMAMapView* so that the specified world location coincides with the specified screen location.

■ **Note:**

The screen point must be within the bounds of the map view.

■ **Note:**

Due to the curvature of the Earth, positioning may be inaccurate at low zoom levels.

Parameters:

- **coordinates**
The world location
- **point**
The screen location
- **animation**
The NMAMapAnimation type to use while moving the map

```
- (void) setGeoCoordinates:(NMAGeoCoordinates *) coordinates toPoint:  
(CGPoint) point withAnimation:(NMAMapAnimation) animation zoomLevel:  
(coordinates:to:animation:zoomLevel:set)
```

Positions the *NMAMapView* so that the specified world location coincides with the specified screen location.

■ **Note:**

The screen point must be within the bounds of the map view.

■ **Note:**

Due to the curvature of the Earth, positioning may be inaccurate at low zoom levels.

Parameters:

- **coordinates**
The world location
- **point**
The screen location
- **animation**
The NMAMapAnimation type to use while moving the map
- **zoomLevel**
Desired zoom level of the newly-centered NMAMapView (pass NMAMapViewPreserveValue to keep the current zoom level)

```
- (void) setZoomLevel:(float) zoomLevel withAnimation:  
(zoomLevel:animation:set)
```

Sets the zoomLevel of the map with optional animation.

Parameters:

- **zoomLevel**
The zoomLevel at which to set the map.
- **animationType**
The animation method used to display the transformation.

- (BOOL) useDefaultDisplayLanguage

Set the map display language using the system default locale.

 **Note:**

If it fails, the map display language is set to English.

Returns:

True if the language is accepted by the map, false otherwise.

- (BOOL) useDisplayLanguageFromLocale:(NSLocale *) locale

Set the map display language using the locale specified.

Parameters:

- **locale**
The NSLocale as described in https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Classes/NSLocale_Class/.

Returns:

True if the locale is accepted by the map, false otherwise.

- (BOOL) useSecondaryDisplayLanguageFromLocale:(NSLocale *_Nullable) locale

Set the secondary map display language using the locale specified.

Setting a secondary display language will cause the language to be displayed below the primary language for some map labels (e.g. countries).

Parameters:

- **locale**
Pass nil if you want to remove the secondary map display language. The NSLocale as described in https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/Classes/NSLocale_Class/.

Returns:

True if the locale is accepted by the map, false otherwise.

```
- (NSArray< __kindof NMAMapObject * > *) visibleObjectsAtPoint:  
(at:visibleObjects)
```

Returns a NSArray of all visible NMAMapObject objects that are selected at a specified point on the screen.

Parameters:

- **point**

An on-screen CGPoint specified in logical coordinates

Returns:

The NSArray of selected NMAMapObject objects

Class Method Details

```
+ (void) shouldBeginRenderingAutomatically:(BOOL) enabled
```

Used to control the default NMAMapView rendering behaviour.

This class method is used to set whether or not instances of NMAMapView will begin rendering immediately upon creation. Any map view created after passing NO to this method must have its renderAllowed property set to YES to begin rendering.

NMAPositionIndicator

Class Summary

class NMAPositionIndicator

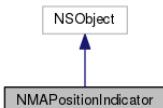
Derived from: NSObject

Used to provide a visual indication of the user's current position.

Include: NMKit.framework/headers/NMAPositionIndicator.h

Inheritance Diagrams

Figure 48: Public inheritance diagram for NMAPositionIndicator



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 51: Public Properties

Public Properties
<p>[readable, writable, assign] UIColor * accuracyIndicatorColor The color of the accuracy indicator circle</p>
<p>[readable, writable, assign] BOOL accuracyIndicatorVisible Indicates whether the <i>NMAPositionIndicator</i> accuracy indicator (the surrounding circle) is visible</p>
<p>[readable, writable, assign] BOOL visible Indicates whether the <i>NMAPositionIndicator</i> is visible</p>

Class Details

Used to provide a visual indication of the user's current position.

Each instance of *NMAMapView* automatically has a *NMAPositionIndicator* created for it. The position indicator is hidden by default; it can be displayed by change its visible property to YES.

By default, the position indicator displays an *NMAMapMarker* showing a green circle.

The accuracy of the current position is indicated by a circle surrounding the indicator object. The visibility of the accuracy indicator is controlled via the accuracyIndicatorVisible property, and its color may be changed by setting the accuracyIndicatorColor.

■ **Note:**

The *NMAPositionIndicator* requires *NMAPositioningManager* position updates to function correctly, and will automatically start the positioning service when required.

Public Property Details

[readable, writable, assign] UIColor * **accuracyIndicatorColor**

The color of the accuracy indicator circle.

[readable, writable, assign] BOOL **accuracyIndicatorVisible**

Indicates whether the *NMAPositionIndicator* accuracy indicator (the surrounding circle) is visible.

■ **Note:**

The getter is `isAccuracyIndicatorVisible`.

[readable, writable, assign] BOOL **visible**

Indicates whether the *NMAPositionIndicator* is visible.

■ **Note:**

The getter is `isVisible`.

■ Note:

Setting visible to YES will start *NMAPositioningManager* positioning if it is not already active.

NMA Routing

Group Summary

group NMA Routing

The Routing group contains classes, protocols, and enumerations for describing and calculating routes.

[For complete information, see the section *Group Details*]

Nested Classes

Table 52: Public Nested Classes

Public Nested Classes
<i>NMAManeuver</i>
<i>NMARoute</i>
<i>NMARouteElement</i>
<i>NMARouteManager</i>
<i>NMARouteManagerDelegate [p]</i>
<i>NMARouteTta</i>
<i>NMARoutingMode</i>
<i>NMASignpost</i>
<i>NMAWaypoint</i>

Public Enumeration Summary

Table 53: Public Enumerations

Public Enumerations
<i>NMAManeuverAction</i> Defines types of actions for a NMAManeuver.
<i>NMAManeuverIcon</i> Defines types of icons for a NMAManeuver.
<i>NMAManeuverTrafficDirection</i> Defines types of traffic directions, the side of the road on which one must drive.

Public Enumerations

[NMAManeuverTurn](#)

Defines types of turns for a NMAManeuver.

[NMARouteDurationDetail](#)

Identifiers for detailed explanation of travel duration (NMARoute::duration) estimates relating to a route.

[NMARouteElementType](#)

Defines types of route element.

[NMARouteManagerError](#)

Defines types of error codes that can be returned from route calculation operations.

[NMARouteViolatedOption](#)

Implicit routing options that may be violated by routes returned from [NMARouteManager](#).

[NMARoutingOption](#)

Represents values describing routing options that can be used to enforce special conditions on a calculated route.

[NMARoutingType](#)

Represents values describing different routing types.

[NMATransportMode](#)

Represents values describing different transport modes, the mode of transportation a person will be using to travel a route (e.g. a car).

Public Variable Summary

Table 54: Public Variables

Public Variables
<code>FOUNDATION_EXPORT NSInteger const NMARouteSublegWhole</code> A constant used to indicate the whole route should be used in route leg selection.

Group Details

The Routing group contains classes, protocols, and enumerations for describing and calculating routes. The key class to this group is [NMARouteManager](#).

Public Enumeration Details

[NMAManeuverAction](#)

Include: NMAKit.framework/headers/NMAManeuver.h

Defines types of actions for a [NMAManeuver](#).

Enumeration Members:

- [NMAManeuverActionUndefined](#)

An undefined action.

- **NMAManeuverActionNone**

An indication there is no action associated with the maneuver.

- **NMAManeuverActionEnd**

An action that indicates the end of a route.

- **NMAManeuverActionStopover**

An action that indicates a stopover.

- **NMAManeuverActionJunction**

An action that indicates a junction.

- **NMAManeuverActionRoundabout**

An action that indicates a roundabout.

- **NMAManeuverActionUTurn**

An action that indicates a u-turn.

- **NMAManeuverActionEnterHighwayFromRight**

An action that indicates entering a highway from the right.

- **NMAManeuverActionEnterHighwayFromLeft**

An action that indicates entering a highway from the left.

- **NMAManeuverActionEnterHighway**

An action that indicates entering a highway.

- **NMAManeuverActionLeaveHighway**

An action that indicates leaving a highway.

- **NMAManeuverActionChangeHighway**

An action that indicates changing from one highway to another.

- **NMAManeuverActionContinueHighway**

An action that indicates continuing along a highway.

- **NMAManeuverActionFerry**

An action that indicates boarding a ferry.

- **NMAManeuverActionPassJunction**

An action that indicates passing a junction.

- **NMAManeuverActionInvalid**

An invalid action.

NMAManeuverIcon

Include: `NMAKit.framework/headers/NMAManeuver.h`

Defines types of icons for a [NMAManeuver](#).

■ **Note:**

`NMAManeuverIcon` enum is simplified version of all valid `NMAManeuverAction` and `NMAManeuverTurn` combinations. It can be used to provide navigation directions, but for more detailed representation, use `NMAManeuverAction` in conjunction with `NMAManeuverTurn`.

Enumeration Members:

- **NMAManeuverIconUndefined**
An undefined icon.
- **NMAManeuverIconGoStraight**
An icon that indicates a straight heading.
- **NMAManeuverIconUTurnRight**
An icon that indicates a right u-turn.
- **NMAManeuverIconUTurnLeft**
An icon that indicates a left u-turn.
- **NMAManeuverIconKeepRight**
An icon that indicates keeping to the right.
- **NMAManeuverIconLightRight**
An icon that indicates a light right turn.
- **NMAManeuverIconQuiteRight**
An icon that indicates a normal right turn.
- **NMAManeuverIconHeavyRight**
An icon that indicates a heavy right turn.
- **NMAManeuverIconKeepMiddle**
An icon that indicates keeping to the left.
- **NMAManeuverIconKeepLeft**
An icon that indicates keeping to the middle line.
- **NMAManeuverIconLightLeft**
An icon that indicates a light left turn.
- **NMAManeuverIconQuiteLeft**
An icon that indicates a normal left turn.
- **NMAManeuverIconHeavyLeft**
An icon that indicates a heavy left turn.
- **NMAManeuverIconEnterHighwayRightLane**
An icon that indicates entering a highway into the right lane.
- **NMAManeuverIconEnterHighwayLeftLane**
An icon that indicates entering a highway into the left lane.
- **NMAManeuverIconLeaveHighwayRightLane**
An icon that indicates leaving a highway from the right lane.
- **NMAManeuverIconLeaveHighwayLeftLane**
An icon that indicates from the left lane.
- **NMAManeuverIconHighwayKeepRight**
An icon that indicates keeping to the right.
- **NMAManeuverIconHighwayKeepLeft**

An icon that indicates keeping to the left.

- **NMAManeuverIconRoundabout1**

An icon that indicates using the first exit encountered while navigating a roundabout in a counter-clockwise direction.

- **NMAManeuverIconRoundabout2**

An icon that indicates using the second exit encountered while navigating a roundabout in a counter-clockwise direction.

- **NMAManeuverIconRoundabout3**

An icon that indicates using the third exit encountered while navigating a roundabout in a counter-clockwise direction.

- **NMAManeuverIconRoundabout4**

An icon that indicates using the fourth exit encountered while navigating a roundabout in a counter-clockwise direction.

- **NMAManeuverIconRoundabout5**

An icon that indicates using the fifth exit encountered while navigating a roundabout in a counter-clockwise direction.

- **NMAManeuverIconRoundabout6**

An icon that indicates using the sixth exit encountered while navigating a roundabout in a counter-clockwise direction.

- **NMAManeuverIconRoundabout7**

An icon that indicates using the seventh exit encountered while navigating a roundabout in a counter-clockwise direction.

- **NMAManeuverIconRoundabout8**

An icon that indicates using the eighth exit encountered while navigating a roundabout in a counter-clockwise direction.

- **NMAManeuverIconRoundabout9**

An icon that indicates using the ninth exit encountered while navigating a roundabout in a counter-clockwise direction.

- **NMAManeuverIconRoundabout10**

An icon that indicates using the tenth exit encountered while navigating a roundabout in a counter-clockwise direction.

- **NMAManeuverIconRoundabout11**

An icon that indicates using the eleventh exit encountered while navigating a roundabout in a counter-clockwise direction.

- **NMAManeuverIconRoundabout12**

An icon that indicates using the twelfth exit encountered while navigating a roundabout in a counter-clockwise direction.

- **NMAManeuverIconRoundabout1LH**

An icon that indicates using the first exit encountered while navigating a roundabout in a counter-clockwise direction.

- **NMAManeuverIconRoundabout2LH**
An icon that indicates using the second exit encountered while navigating a roundabout in a clockwise direction.
- **NMAManeuverIconRoundabout3LH**
An icon that indicates using the third exit encountered while navigating a roundabout in a clockwise direction.
- **NMAManeuverIconRoundabout4LH**
An icon that indicates using the fourth exit encountered while navigating a roundabout in a clockwise direction.
- **NMAManeuverIconRoundabout5LH**
An icon that indicates using the fifth exit encountered while navigating a roundabout in a clockwise direction.
- **NMAManeuverIconRoundabout6LH**
An icon that indicates using the sixth exit encountered while navigating a roundabout in a clockwise direction.
- **NMAManeuverIconRoundabout7LH**
An icon that indicates using the seventh exit encountered while navigating a roundabout in a clockwise direction.
- **NMAManeuverIconRoundabout8LH**
An icon that indicates using the eighth exit encountered while navigating a roundabout in a clockwise direction.
- **NMAManeuverIconRoundabout9LH**
An icon that indicates using the ninth exit encountered while navigating a roundabout in a clockwise direction.
- **NMAManeuverIconRoundabout10LH**
An icon that indicates using the tenth exit encountered while navigating a roundabout in a clockwise direction.
- **NMAManeuverIconRoundabout11LH**
An icon that indicates using the eleventh exit encountered while navigating a roundabout in a clockwise direction.
- **NMAManeuverIconRoundabout12LH**
An icon that indicates using the twelfth exit encountered while navigating a roundabout in a clockwise direction.
- **NMAManeuverIconStart**
An icon that indicates the start point (displayed when route navigation has not yet begun).
- **NMAManeuverIconEnd**
An icon that indicates the destination point.
- **NMAManeuverIconFerry**
An icon that indicates boarding a ferry.

NMAManeuverTrafficDirection

Include: `NMAKit.framework/headers/NMAManeuver.h`

Defines types of traffic directions, the side of the road on which one must drive.

Enumeration Members:

- **NMAManeuverTrafficDirectionLeft**
Traffic flows on the left side of the road, as in the UK.
- **NMAManeuverTrafficDirectionRight**
Traffic flows on the right side of the road, as in the USA.

NMAManeuverTurn

Include: `NMAKit.framework/headers/NMAManeuver.h`

Defines types of turns for a [NMAManeuver](#).

Enumeration Members:

- **NMAManeuverTurnUndefined**
An undefined turn.
- **NMAManeuverTurnNone**
Indicates that no turn is necessary.
- **NMAManeuverTurnKeepMiddle**
A turn that indicates keeping to the middle when a road forks.
- **NMAManeuverTurnKeepRight**
A turn that indicates keeping to the right when a road forks.
- **NMAManeuverTurnLightRight**
A turn that indicates making a light right turn.
- **NMAManeuverTurnQuiteRight**
A turn that indicates making a normal right turn.
- **NMAManeuverTurnHeavyRight**
A turn that indicates making a heavy right turn.
- **NMAManeuverTurnKeepLeft**
A turn that indicates keeping to the left when a road forks.
- **NMAManeuverTurnLightLeft**
A turn that indicates making a light left turn.
- **NMAManeuverTurnQuiteLeft**
A turn that indicates making a normal left turn.
- **NMAManeuverTurnHeavyLeft**
A turn that indicates making a heavy left turn.
- **NMAManeuverTurnReturn**
A turn that indicates a u-turn, turning around.

- **NMAManeuverTurnRoundabout1**
A turn that indicates taking the first exit of a roundabout.
- **NMAManeuverTurnRoundabout2**
A turn that indicates taking the second exit of a roundabout.
- **NMAManeuverTurnRoundabout3**
A turn that indicates taking the third exit of a roundabout.
- **NMAManeuverTurnRoundabout4**
A turn that indicates taking the fourth exit of a roundabout.
- **NMAManeuverTurnRoundabout5**
A turn that indicates taking the fifth exit of a roundabout.
- **NMAManeuverTurnRoundabout6**
A turn that indicates taking the sixth exit of a roundabout.
- **NMAManeuverTurnRoundabout7**
A turn that indicates taking the seventh exit of a roundabout.
- **NMAManeuverTurnRoundabout8**
A turn that indicates taking the eighth exit of a roundabout.
- **NMAManeuverTurnRoundabout9**
A turn that indicates taking the ninth exit of a roundabout.
- **NMAManeuverTurnRoundabout10**
A turn that indicates taking the tenth exit of a roundabout.
- **NMAManeuverTurnRoundabout11**
A turn that indicates taking the eleventh exit of a roundabout.
- **NMAManeuverTurnRoundabout12**
A turn that indicates taking the twelfth exit of a roundabout.

NMARouteDurationDetail

Include: `NMAKit.framework/headers/NMARouteTta.h`

Identifiers for detailed explanation of travel duration (`NMARoute::duration`) estimates relating to a route.

Note:

This is a bitwise enum. If none of the bit is set, `NMARouteDurationDetailAccurate` is assumed

Enumeration Members:

- **NMARouteDurationDetailAccurate**
The estimate is accurate.
- **NMARouteDurationDetailBlockedRoad**
The route uses a road that is blocked.
- **NMARouteDurationDetailCarPool**
The route uses a road with carpool restriction.

- **NMARouteDurationDetailRestrictedTurn**

The route uses a restricted turn.

NMARouteElementType

Include: NMAKit.framework/headers/NMARouteElement.h

Defines types of route element.

Enumeration Members:

- **NMARouteElementRoad**
Road Element
- **NMARouteElementInvalid**
invalid

NMARouteManagerError

Include: NMAKit.framework/headers/NMARouteManager.h

Defines types of error codes that can be returned from route calculation operations.

Enumeration Members:

- **NMARouteManagerErrorNone**
There was no error, route calculation succeeded.
- **NMARouteManagerErrorUnknown**
There was an unknown error.
- **NMARouteManagerErrorOutOfMemory**
There was an out-of-memory error.
- **NMARouteManagerErrorInvalidParameters**
There was an error due to invalid parameters.
- **NMARouteManagerErrorInvalidOperation**
There was an error due to another request already being processed.
- **NMARouteManagerErrorGraphDisconnected**
There was an error because no route could be found.
- **NMARouteManagerErrorGraphDisconnectedCheckOptions**
There was an error because no route could be found, possibly due to some option (e.g. disabled highways) preventing it.
- **NMARouteManagerErrorNoStartPoint**
There was an error because no starting waypoint could be found.
- **NMARouteManagerErrorNoEndPoint**
There was an error because no destination waypoint could be found.
- **NMARouteManagerErrorNoEndPointCheckOptions**

There was an error because the destination point is unreachable, possibly due to some option (e.g. disabled highways) preventing it.

- **NMARouteManagerErrorCannotDoPedestrian**

There was an error because pedestrian mode was specified yet is not practical (e.g. the route is too long).

- **NMARouteManagerErrorRoutingCancelled**

There was an error because the application user cancelled the route calculation.

- **NMARouteManagerErrorViolatesOptions**

There was an error because the route calculation request included options that prohibit successful completion.

- **NMARouteManagerErrorRouteCorrupted**

There was an error because the service could not digest the requested route parameters.

- **NMARouteManagerErrorInvalidCredentials**

There was an error due to invalid or missing HERE Developer credentials.

NMARouteViolatedOption

Include: `NMAKit.framework/headers/NMARouteManager.h`

Implicit routing options that may be violated by routes returned from [NMARouteManager](#).

Enumeration Members:

- **NMARouteViolatedOptionNone**

The returned route does not violate any options

- **NMARouteViolatedOptionBlockedRoad**

The route passes through a blocked road (e.g. due to construction)

- **NMARouteViolatedOptionTurnRestriction**

The route passes through a road with a time-based turn restriction

NMARoutingOption

Include: `NMAKit.framework/headers/NMARoutingMode.h`

Represents values describing routing options that can be used to enforce special conditions on a calculated route.

Enumeration Members:

- **NMARoutingOptionAvoidBoatFerry**

The routing engine will avoid links that are part of a boat ferry passage.

- **NMARoutingOptionAvoidDirtRoad**

The routing engine will avoid links that are part of a dirt road.

- **NMARoutingOptionAvoidHighway**

The routing engine will avoid links that are part of a highway.

■ Note:

Effective only when used with [NMATransportModeCar](#).

- **NMARoutingOptionAvoidPark**

The routing engine will avoid links that go through a park.

■ Note:

Effective only when used with [NMATransportModePedestrian](#).

- **NMARoutingOptionAvoidTollRoad**

The routing engine will avoid links that are part of a toll road.

■ Note:

Effective only when used with [NMATransportModeCar](#).

- **NMARoutingOptionAvoidTunnel**

The routing engine will avoid links that are part of a tunnel.

■ Note:

Effective only when used with [NMATransportModeCar](#).

- **NMARoutingOptionAvoidCarShuttleTrain**

The routing engine will avoid links that are part of a motorail train or car shuttle train passage.

- **NMARoutingOptionAvoidCarpool**

The routing engine will avoid HOV/Carpool roads.

NMARoutingType

Include: `NMAKit.framework/headers/NMARoutingMode.h`

Represents values describing different routing types.

Enumeration Members:

- **NMARoutingTypeShortest**

Search for the shortest route, taking road conditions and restrictions into account (minimizes travel distance).

- **NMARoutingTypeFastest**

Search for the fastest route, taking road conditions and restrictions into account (minimizes travel time).

NMATransportMode

Include: `NMAKit.framework/headers/NMARoutingMode.h`

Represents values describing different transport modes, the mode of transportation a person will be using to travel a route (e.g. a car).

Enumeration Members:

- **NMATransportModeCar**

A car is being used as the mode of transportation.

- **NMATransportModePedestrian**

Walking is being used as the mode of transportation.

Public Variable Details

`FOUNDATION_EXPORT NSInteger const NMARouteSublegWhole`

Include: `NMAKit.framework/headers/NMARoute.h`

A constant used to indicate the whole route should be used in route leg selection.

<NMARouteManagerDelegate>

Protocol Summary

protocol <**NMARouteManagerDelegate**>

Derived from: <`NSObject`>

Represents a delegate to handle `NMARouteManager` route calculation updates.

Include: `NMAKit.framework/headers/NMARouteManager.h`

Inheritance Diagrams

Figure 49: Public inheritance diagram for <**NMARouteManagerDelegate**> – <**NMARouteManagerDelegate**>



[For complete information, see the section [Protocol Details](#)]

Instance Method Summary

Table 55: Instance Methods

Instance Method Summary
<pre>- (void) routeManager:(NMARouteManager *) routeManager didCalculateRoutes:(nullable NSArray<__kindof NMARoute * *>) routes withError:(NMARouteManagerError) error violatedOptions:(_:routes:error:violatedOptions:routeManagerDidCalculate)</pre> <p>Callback upon completion of route calculation</p>
<pre>- @optional (void) routeManager:(NMARouteManager *) routeManager didUpdateProgress:(_:progress:routeManagerDidUpdate)</pre> <p>Callback for reporting the progress of route calculation; values range from 0</p>

Protocol Details

Represents a delegate to handle [NMARouteManager](#) route calculation updates.

Methods of this protocol are called on the main queue.

Instance Method Details

```
- (void) routeManager:(NMARouteManager *) routeManager  
didCalculateRoutes:(nullable NSArray< __kindof NMARoute * > *)  
routes withError:(NMARouteManagerError) error violatedOptions:  
(_:routes:error:violatedOptions:routeManagerDidCalculate)
```

Callback upon completion of route calculation.

Parameters:

- **routeManager**

The [NMARouteManager](#) instance which calculated the route.

- **routes**

An NSArray of [NMARoute](#) objects representing alternate route solutions, or nil if route calculation failed. The route at index 0 is the best result found. The number of routes in the array will be less than or equal to the resultCount property specified in the [NMARoutingMode](#) used for the route request.

- **error**

A routing calculation error. See [NMARouteManagerError](#) for possible error codes. If NMARouteManagerErrorRouteUseDisabledRoads is returned then violatedOptions is not empty (the returned [NMARoute](#) is valid and can be used for navigation although it uses roads which were disabled in the original [NMARoutingMode::routingOptions](#) options).

- **violatedOptions**

An NSArray of NSNumber objects, one per route calculated, representing which options (if any) were violated for the corresponding route. The NSNumber objects should be converted to NSUInteger before checking. The value will be an OR combination of [NMARoutingOption](#) and [NMARouteViolatedOption](#) values, or [NMARouteViolatedOptionNone](#) if no options were violated. If route calculation failed, the array will contain a single object containing all the routing options (if any).

```
- @optional (void) routeManager:(NMARouteManager *) routeManager  
didUpdateProgress:(_:progress:routeManagerDidUpdate)
```

Callback for reporting the progress of route calculation; values range from 0.0 (starting) to 1.0 (complete).

 **Note:**

In certain circumstances a recalculation is required and this percentage will go from 1.0 to 0.0

Parameters:

- **routeManager**

The [NMARouteManager](#) singleton instance.

- **progress**

The progress of the current routing operation.

NMAManeuver

Class Summary

class **NMAManeuver**

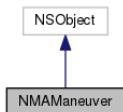
Derived from: `NSObject`

Represents a maneuver, which is the action required to leave one street segment and enter the next in the linked chain of directions that comprises a calculated NMARoute.

Include: `NMAKit.framework/headers/NMAManeuver.h`

Inheritance Diagrams

Figure 50: Public inheritance diagram for NMAManeuver



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 56: Public Properties

Public Properties
<p>[readable, assign] <code>NMAManeuverAction</code> <i>action</i> The NMAManeuverAction to take for the maneuver</p>
<p>[readable, assign] <code>NMAGeoCoordinates</code> * <i>coordinates</i> The NMAGeoCoordinates of the maneuver</p>
<p>[readable, assign] <code>NSUInteger</code> <i>distanceFromPreviousManeuver</i> The distance from the previous maneuver along the route to the NMAManeuver, in meters</p>
<p>[readable, assign] <code>NSUInteger</code> <i>distanceFromStart</i> The distance from the start of the route to the maneuver, in meters</p>
<p>[readable, assign] <code>NSUInteger</code> <i>distanceToNextManeuver</i> The distance to the next maneuver along the route to the NMAManeuver, in meters</p>
<p>[readable, assign] <code>NMAManeuverIcon</code> <i>icon</i> The NMAManeuverIcon for the NMAManeuver</p>

Public Properties

[readable, assign] `NSUInteger mapOrientation`

The angle (from north) at the start of the maneuver, in degrees

[readable, assign] `NSString * nextRoadName`

The name of the road to which the NMAManeuver leads

[readable, assign] `NSString * nextRoadNumber`

The number of the road to which the maneuver leads, a short label for the road, (e

[readable, assign] `NSString * roadName`

The name of the road on which the NMAManeuver takes place, nil if not available

[readable, assign] `NSString * roadNumber`

The number of the road on which the NMAManeuver takes place, a short label for the road, (e

[readable, assign] `NSArray< __kindof NMARouteElement * > * routeElements`

The list of NMARouteElement (or instances of classes derived from it) within the maneuver object

[readable, assign] `NMASignpost * signpost`

Gets the signpost information for this maneuver

[readable, assign] `NSDate * startTime`

The (estimated) time at which the maneuver starts in UTC

[readable, assign] `NMATransportMode transportMode`

Gets the transport mode for the maneuver

[readable, assign] `NMAManeuverTurn turn`

The NMAManeuverTurn to take for the maneuver

Class Details

Represents a maneuver, which is the action required to leave one street segment and enter the next in the linked chain of directions that comprises a calculated NMARoute.

Public Property Details

[readable, assign] `NMAManeuverAction action`

The NMAManeuverAction to take for the maneuver.

Note:

If the action for the maneuver is undefined, attempts to read this property will return NMAManeuverActionUndefined.

[readable, assign] `NMAGeoCoordinates * coordinates`

The NMAGeoCoordinates of the maneuver.

[readable, assign] `NSUInteger distanceFromPreviousManeuver`

The distance from the previous maneuver along the route to the NMAManeuver, in meters.

[readable, assign] `NSUInteger distanceFromStart`

The distance from the start of the route to the maneuver, in meters.

[readable, assign] `NSUInteger distanceToNextManeuver`

The distance to the next maneuver along the route to the NMAManeuver, in meters.

[readable, assign] `NMAManeuverIcon icon`

The NMAManeuverIcon for the NMAManeuver.

■ **Note:**

If the icon type for the maneuver is undefined, attempts to read this property will return NMAManeuverIconUndefined.

[readable, assign] `NSUInteger mapOrientation`

The angle (from north) at the start of the maneuver, in degrees.

■ **Note:**

Zero represents true-north, with increasing values representing a clockwise progression of map orientation.

[readable, assign] `NSString * nextRoadName`

The name of the road to which the NMAManeuver leads. nil if not available. If not available, it should be left blank. It's erroneous to assume that that may be the same as some maneuver along the route.

■ **Note:**

It is localized to the locale of the region.

[readable, assign] `NSString * nextRoadNumber`

The number of the road to which the maneuver leads, a short label for the road, (e.g. 5 for Interstate 5). nil if not available. If not available, it should be left blank. It's erroneous to assume that that may be the same as some maneuver along the route.

■ **Note:**

It is localized to the locale of the region.

[readable, assign] `NSString * roadName`

The name of the road on which the NMAManeuver takes place, nil if not available. If not available, it should be left blank. It's erroneous to assume that that may be the same as prior maneuvers.

▀ Note:

It is localized to the locale of the region.

[readable, assign] `NSString * roadNumber`

The number of the road on which the NMAManeuver takes place, a short label for the road, (e.g. 5 for the Interstate 5).

▀ Note:

If the road number for the maneuver is unknown, attempts to read this property will return an empty NSString. nil if not available. If not available, it should be left blank. It's erroneous to assume that that may be the same as prior maneuvers.

▀ Note:

It is localized to the locale of the region.

[readable, assign] `NSArray< __kindof NMARouteElement * > * routeElements`

The list of `NMARouteElement` (or instances of classes derived from it) within the maneuver object.

[readable, assign] `NMASignpost * signpost`

Gets the signpost information for this maneuver.

▀ Note:

nil if no signpost information for this maneuver.

[readable, assign] `NSDate * startTime`

The (estimated) time at which the maneuver starts in UTC.

▀ Note:

The departure time must be set in `NMARoutingMode` before routing. nil if not available.

[readable, assign] `NMATransportMode transportMode`

Gets the transport mode for the maneuver. This might differ from the transport mode used for route calculation. For example, in the case where a route is calculated using public transport, the overall route is a public transport route, but some individual maneuvers may be pedestrian (e.g. walking to a bus stop, transfers which involve walking to a new stop, etc.).

[readable, assign] `NMAManeuverTurn` `turn`

The NMAManeuverTurn to take for the maneuver.

■ **Note:**

If the turn for the maneuver is undefined, attempts to read this property will return NMAManeuverTurnUndefined.

NMARoute

Class Summary

class `NMARoute`

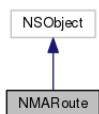
Derived from: `NSObject`

Represents a path (a collection of maneuvers) connecting two or more waypoints.

Include: `NMAKit.framework/headers/NMARoute.h`

Inheritance Diagrams

Figure 51: Public inheritance diagram for NMARoute



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 57: Public Properties

Public Properties

[readable, assign] `NMAGeoBoundingBox` * `boundingBox`

The smallest NMAGeoBoundingBox that contains the entire NMARoute

[readable, assign] `NMAWaypoint` * `destination`

The destination NMAWaypoint for the NMARoute

[readable, assign] `NSUInteger` `length`

The length of the NMARoute, in meters

[readable, assign] `NSArray< NMAManeuver >` * `maneuvers`

Array of NMAManeuver to represent all the maneuvers that travelers will encounter along the NMARoute

[readable, assign] `NMAMapPolyline` * `mapPolyline`

The NMAMapPolyline representation of the route

Public Properties

[readable, assign] `NMARoutingMode * routingMode`

The NMARoutingMode for the NMARoute

[readable, assign] `NMAWaypoint * start`

The starting `NMAWaypoint` for the NMARoute.

[readable, assign] `NSUInteger sublegCount`

Returns the number of sub-legs the route has a subleg is the part of a route between two stop waypoints

[readable, assign] `NMARouteTta * tta`

The `NMARouteTta` object of traveling the whole `NMARoute`.

[readable, writable, strong] `NSString * userTag`

[readable, assign] `NSArray< NMAWaypoint * > * waypoints`

Array of `NMAWaypoint` for all waypoints of the NMARoute

Instance Method Summary

Table 58: Instance Methods

Instance Method Summary

`-(nullable NMARouteTta *) ttaForSubleg:(NSUInteger) subleg`

The `NMARouteTta` object of traveling the specified subLeg of the `NMARoute`.

Class Details

Represents a path (a collection of maneuvers) connecting two or more waypoints.

Waypoints may be thought of as the input to a route calculation whereas maneuvers are the results of calculating a route.

Public Property Details

[readable, assign] `NMAGeoBoundingBox * boundingBox`

The smallest `NMAGeoBoundingBox` that contains the entire NMARoute.

[readable, assign] `NMAWaypoint * destination`

The destination `NMAWaypoint` for the NMARoute.

[readable, assign] `NSUInteger length`

The length of the NMARoute, in meters. This is the actual distance covered if you were to travel the route.

[readable, assign] NSArray< *NMAManeuver* * > * **maneuvers**

Array of *NMAManeuver* to represent all the maneuvers that travelers will encounter along the NMARoute.

[readable, assign] *NMAMapPolyline* * **mapPolyline**

The *NMAMapPolyline* representation of the route.

[readable, assign] *NMARoutingMode* * **routingMode**

The *NMARoutingMode* for the NMARoute.

[readable, assign] *NMAWaypoint* * **start**

The starting *NMAWaypoint* for the NMARoute.

[readable, assign]NSUInteger **sublegCount**

Returns the number of sub-legs the route has a subleg is the part of a route between two stop waypoints.

[readable, assign] *NMARouteTta* * **tta**

The *NMARouteTta* object of traveling the whole *NMARoute*.

[readable, writable, strong] NSString * **userTag**

A user-defined tag to identify the NMARoute

■ **Note:**

The default value for userTag is empty string.

[readable, assign] NSArray< *NMAWaypoint* * > * **waypoints**

Array of *NMAWaypoint* for all waypoints of the NMARoute.

Instance Method Details

- (nullable *NMARouteTta* *) **ttaForSubleg:(NSUInteger) subleg**

The *NMARouteTta* object of traveling the specified subLeg of the *NMARoute*.

Parameters:

- **subleg**

The index of the subleg on the `NMARoute` to be used for this calculation. If `NMARouteSublegWhole` is provided, it is the same as `tta` property.

NMARouteElement

Class Summary

class `NMARouteElement`

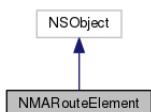
Derived from: `NSObject`

Represents a generic container for an element of a route.

Include: `NMAKit.framework/headers/NMARouteElement.h`

Inheritance Diagrams

Figure 52: Public inheritance diagram for `NMARouteElement`



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 59: Public Properties

Public Properties
<code>[readable, assign] NMARoadElement * roadElement</code> Road element of the route element
<code>[readable, assign] NMARouteElementType type</code> The route element type

Class Details

Represents a generic container for an element of a route.

Public Property Details

`[readable, assign] NMARoadElement * roadElement`

Road element of the route element.

■ **Note:**

`nil` if type of the route element is not `NMARouteElementRoad`

[readable, assign] `NMARouteElementType` `type`

The route element type.

NMARouteManager

Class Summary

class `NMARouteManager`

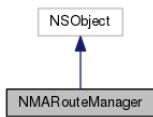
Derived from: `NSObject`

Represents a manager responsible for calculating a `NMARoute` from a list of stops and (optionally) a `NMARoutingMode`.

Include: `NMAKit.framework/headers/NMARouteManager.h`

Inheritance Diagrams

Figure 53: Public inheritance diagram for `NMARouteManager`



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 60: Public Properties

Public Properties
[readable, assign] <code>BOOL</code> <code>busy</code>
Indicates whether or not the <code>NMARouteManager</code> is currently calculating a route
[readable, writable, weak] <code>id< NMARouteManagerDelegate ></code> <code>delegate</code>
The <code>NMARouteManagerDelegate</code> that listens for messages from the <code>NMARouteManager</code>

Instance Method Summary

Table 61: Instance Methods

Instance Method Summary
<code>- (BOOL) calculateRouteWithStops:(stops:calculateRoute)</code>
Starts a route calculation with the given stop list and the default <code>NMARoutingMode</code> .

Instance Method Summary

`- (BOOL) calculateRouteWithStops:(NSArray *) stops routingMode:(stops:mode:calculateRoute)`

Starts a route calculation with the give stop list and routing mode

`- (BOOL) cancel`

Cancels the current route calculation.

Class Method Summary

Table 62: Class Methods

Class Methods

`+ (NMARouteManager *) sharedRouteManager`

Returns the NMARouteManager singleton instance

Class Details

Represents a manager responsible for calculating a [NMARoute](#) from a list of stops and (optionally) a [NMARoutingMode](#).

[NMARouteManagerDelegate](#) is used to monitor route calculation progress and route calculation completion. Set the [NMARouteManager](#) delegate property in order to receive callbacks through [NMARouteManagerDelegate](#) protocol.

The current status of the [NMARouteManager](#) can be checked via the `busy` property.

[NMARouteManager](#) only supports one routing request at a time. Attempts to make another request before the current request has finished will fail.

[NMARouteManager](#) requires valid authentication credentials to be set via [NMApplicationContext](#). If valid credentials are not present the API may not function correctly or may stop functioning correctly in the future when server side configurations change.

Routing within China is supported, but routes may not cross the Chinese border. Thus, any route calculation which begins, ends, or passes through China and involves one or more other countries will fail.

Public Property Details

[\[readable, assign\] BOOL busy](#)

Indicates whether or not the [NMARouteManager](#) is currently calculating a route.

■ **Note:**

Any requests made while the [NMARouteManager](#) is busy will fail.

[\[readable, writable, weak\] id< NMARouteManagerDelegate > delegate](#)

The NMARouteManagerDelegate that listens for messages from the NMARouteManager.

Instance Method Details

- (BOOL) calculateRouteWithStops:(stops:calculateRoute)

Starts a route calculation with the given stop list and the default [NMARoutingMode](#).

The array of stops used to calculate the route must have at least two items. Any of the following may be included in any combination:

1. [NMAGeoCoordinates](#)
2. [NMAPlace](#)
3. [NMAPlaceLocation](#)
4. [NMAWaypoint](#)

If a request is made while a previous request is still in progress, this method will return NO and no callback will be sent to the delegate. If there is not another request in progress but the parameters of the current request are invalid, this method will return NO and a callback will be sent to the delegate.

■ **Note:**

See [NMARoutingMode](#) for the default routing mode properties.

■ **Note:**

[NMARouteManagerDelegate](#) `routeManager:didCalculateRoutes:withError:violatedOptions:` will be called upon success. The returned route will contain a list of NMAWayPoints based on the stop locations; the original stops will be not be returned.

Parameters:

- **stops**

A list of stops to route between

Returns:

YES if the route calculation request was made successfully, NO otherwise

- (BOOL) calculateRouteWithStops:(NSArray *) stops routingMode:(stops:mode:calculateRoute)

Starts a route calculation with the give stop list and routing mode.

■ **Note:**

[NMARouteManagerDelegate](#) `routeManager:didCalculateRoutes:withError:violatedOptions:` will be called upon completion.

Parameters:

- **stops**

A list of stops to route between

- **mode**

The [NMARoutingMode](#) to use for route calculation

Returns:

YES if route calculation was successful, NO otherwise

See also:

calculateRouteWithStops:

-(BOOL) cancel

Cancels the current route calculation.

If route calculation has been started, calling this method cancels the calculation and then notifies the delegate by way of calling `routeManager:didCalculateRoutes:withError:violatedOptions:` with error code `NMARouteManagerErrorRoutingCancelled`. If routing calculation could not be cancelled (e.g. no existing calculation is being performed), the delegate will not be notified and NO will be returned from this method.

Returns:

YES if the cancellation can be processed, NO otherwise

Class Method Details

+(*NMARouteManager* *) sharedRouteManager

Returns the `NMARouteManager` singleton instance.

 **Note:**

Use this method to obtain a `NMARouteManager` instance. Do not call `init` directly.

Returns:

shared `NMARouteManager` instance

NMARouteTta

Class Summary

class `NMARouteTta`

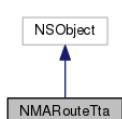
Derived from: `NSObject`

Describes Time to Arrival details of a `NMARoute` or a subleg of a `NMARoute`.

Include: `NMAKit.framework/headers/NMARouteTta.h`

Inheritance Diagrams

Figure 54: Public inheritance diagram for `NMARouteTta`



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 63: Public Properties

Public Properties
<code>[readable, assign] BOOL <i>blocked</i></code> Whether a blocked road is present
<code>[readable, assign] <i>NMARouteDurationDetail</i> <i>details</i></code>
<code>[readable, assign] NSTimeInterval <i>duration</i></code> Duration in seconds

Class Details

Describes Time to Arrival details of a [NMARoute](#) or a subleg of a [NMARoute](#).

Public Property Details

`[readable, assign] BOOL blocked`

Whether a blocked road is present.

`[readable, assign] NMARouteDurationDetail details`

`[readable, assign] NSTimeInterval duration`

Duration in seconds. -1 if not available.

NMARoutingMode

Class Summary

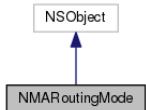
class **NMARoutingMode**

Derived from: `NSObject`

Include: `NMAKit.framework/headers/NMARoutingMode.h`

Inheritance Diagrams

Figure 55: Public inheritance diagram for NMARoutingMode



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 64: Public Properties

Public Properties
<code>[readable, writable, strong] NSDate * departureTime</code> The desired departure time
<code>[readable, writable, assign]NSUInteger resultLimit</code> The maximum number of alternative routes to calculate
<code>[readable, writable, assign]NSUInteger routingOptions</code> The OR-ed NMARoutingOption values for the NMARoutingMode
<code>[readable, writable, assign] NMARoutingType routingType</code> The NMARoutingType for the NMARoutingMode
<code>[readable, writable, assign] NMATransportMode transportMode</code> The NMATransportMode for the NMARoutingMode

Instance Method Summary

Table 65: Instance Methods

Instance Method Summary
<code>- (id) init</code> Initializes a NMARoutingMode instance with default values
<code>- (id) initWithRoutingType:(NMARoutingType) routingType transportMode:(NMATransportMode) transportMode routingOptions:(NSUInteger) routingOptions</code> Initializes a NMARoutingMode instance with the specified routing parameters

Class Details

Represents a model of the parameters required for one route calculation, encapsulating parameters such as:

- The `NMARoutingType`
- The `NMATransportMode`
- The `NMARoutingOption` (specify 0 for none)

Public Property Details

[readable, writable, strong] NSDate * **departureTime**

The desired departure time.

[readable, writable, assign]NSUInteger **resultLimit**

The maximum number of alternative routes to calculate. The actual number calculated may be less. The default value is 1.

■ **Note:**

Only one result will be returned for routes with more than two waypoints.

[readable, writable, assign] NSUInteger **routingOptions**

The OR-ed *NMARoutingOption* values for the NMARoutingMode.

■ **Note:**

The default value is 0 (no options selected).

[readable, writable, assign] *NMARoutingType* **routingType**

The NMARoutingType for the NMARoutingMode.

■ **Note:**

The default value is NMARoutingTypeShortest.

[readable, writable, assign] *NMATransportMode* **transportMode**

The NMATransportMode for the NMARoutingMode.

■ **Note:**

The default value is NMATransportModeCar.

Instance Method Details

-(id) **init**

Initializes a NMARoutingMode instance with default values.

-(id) **initWithRoutingType:(*NMARoutingType*) routingType transportMode:(*NMATransportMode*) transportMode routingOptions:(NSUInteger) routingOptions**

Initializes a NMARoutingMode instance with the specified routing parameters.

Parameters:

- **routingType**
 - A routing type (see [NMARoutingType](#))
- **transportMode**
 - A transport mode (see [NMATransportMode](#))
- **routingOptions**
 - An OR-ed combination of routing options (see [NMARoutingOption](#)).

Returns:

The NMARoutingMode object, or nil if initialization failed

NMASignpost

Class Summary

class **NMASignpost**

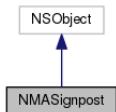
Derived from: `NSObject`

This class stores signpost information along the Route.

Include: `NMAKit.framework/headers/NMASignpost.h`

Inheritance Diagrams

Figure 56: Public inheritance diagram for NMASignpost



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 66: Public Properties

Public Properties
<code>[readable, assign] NSString * exitText</code> The exit text of the signpost

Class Details

This class stores signpost information along the Route.

Public Property Details

[readable, assign] `NSString * exitText`

The exit text of the signpost.

NMAWaypoint

Class Summary

class `NMAWaypoint`

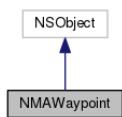
Derived from: `NSObject`

Waypoints define a route's stopovers, including its start point it's destination point and any points in between.

Include: `NMAKit.framework/headers/NMAWaypoint.h`

Inheritance Diagrams

Figure 57: Public inheritance diagram for NMAWaypoint



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 67: Public Properties

Public Properties
[readable, assign] <code>NMAGeoCoordinates * mappedPosition</code> The NMAGeoCoordinates representing the mapped position of the NMAWaypoint - the coordinates adjusted by the routing request
[readable, assign] <code>NMAGeoCoordinates * originalPosition</code> The NMAGeoCoordinates representing the original position of the NMAWaypoint - typically the coordinates passed into a routing request

Class Details

Waypoints define a route's stopovers, including its start point it's destination point and any points in between.

Each waypoint represents a position along an `NMARoute`, this includes:

- The original (input) coordinates specified as part of a search request or

- The adjusted coordinates used to calculate the route. For example, the routing engine may adjust the input coordinates so that they fall upon a road. The process of adjusting the coordinates is known as map matching.

■ Note:

A waypoint can represent the result of map matching.

Public Property Details

[readable, assign] `NMAGeoCoordinates * mappedPosition`

The NMAGeoCoordinates representing the mapped position of the NMAWaypoint - the coordinates adjusted by the routing request.

[readable, assign] `NMAGeoCoordinates * originalPosition`

The NMAGeoCoordinates representing the original position of the NMAWaypoint - typically the coordinates passed into a routing request.

NMA Search

Group Summary

group NMA Search

The Search group provides classes, protocols, and enumerations for performing place and geocoder searches.

[For complete information, see the section [Group Details](#)]

Nested Classes

Table 68: Public Nested Classes

Public Nested Classes
<code>NMAAddress</code>
<code>NMAAutoSuggest</code>
<code>NMAAutoSuggestPlace</code>
<code>NMAAutoSuggestQuery</code>
<code>NMAAutoSuggestSearch</code>
<code>NMAAutoSuggestionRequest</code>
<code>NMACategory</code>
<code>NMACategoryFilter</code>

Public Enumeration Summary

Table 70: Public Enumerations

Public Enumerations
NMAAutoSuggestType Represents available NMAAutoSuggest derived objects.
NMACategoryFilterType Defines types of categories filter.
NMAMediaType Represents available NMAMedia derived objects.
NMATextFormat Search result text output formats.

Public Variable Summary

Table 71: Public Variables

Public Variables
FOUNDATION_EXPORT NSString *const _Nonnull NMAPlacesContentWikipedia Places Wikipedia Content.
FOUNDATION_EXPORT NSString *const _Nonnull NMAPlacesSourceBuilding Places Reference Source for building IDs.
NS_ASSUME_NONNULL_BEGIN FOUNDATION_EXPORT NSString *const _Nonnull NMAPlacesSourcePVID Places Reference Source for core POI IDs.
FOUNDATION_EXPORT NSString *const _Nonnull NMAPlacesSourceSharing Places Reference Source for sharing IDs.
FOUNDATION_EXPORT NSString *const _Nonnull NMAPlacesSourceVenuesAll Places Reference Source for all types of venue IDs.
FOUNDATION_EXPORT NSString *const _Nonnull NMAPlacesSourceVenuesContent Places Reference Source for venue content IDs.
FOUNDATION_EXPORT NSString *const _Nonnull NMAPlacesSourceVenuesDestination Places Reference Source for venue destination IDs.
FOUNDATION_EXPORT NSString *const _Nonnull NMAPlacesSourceVenuesVenue Places Reference Source for venue IDs.
FOUNDATION_EXPORT NSString *_Nonnull const NMA_PLACES_ERROR_DOMAIN

Group Details

The Search group provides classes, protocols, and enumerations for performing place and geocoder searches. Some important classes in this group are [NMAPlaces](#) and [NMAGeocoder](#).

Public Typedef Details

```
(void) (^NMAPlacesCategoriesCompletionBlock) (NSArray< NMACategory * > *_Nullable categories, NSError *_Nullable error)
```

Include: `NMAKit.framework/headers/NMAPlaces.h`

A typedef of a block parameter signature used with [NMAPlaces::refreshTopLevelCategoriesWithCompletion:](#).

■ **Note:**

The block will be called on the main queue.

Parameters:

- **categories**

An NSArray of localized [NMACategory](#) if available, nil otherwise.

- **error**

The error if the refresh failed, or nil if the refresh was successful.

See also:

[NMAPlaces::refreshTopLevelCategoriesWithCompletion:](#)

```
(void) (^NMAResponseCompletionBlock) (NMAResponse *_Nonnull response, id _Nullable data, NSError *_Nullable error)
```

Include: `NMAKit.framework/headers/NMAResponse.h`

A typedef of a block parameter signature used with [NMAResponse::startWithBlock:](#).

The following table shows corresponding request and result types.

- [NMAResponse](#) - Expected Search Results
- [NMAGeocodeResponse](#) - NSArray([NMAGeocodeResult](#))
- [NMAResponse](#) - NSArray([NMAResponse](#))
- [NMADiscoveryResponse](#) - [NMADiscoveryPage](#)
- [NMAResponse](#) - [NMAResponse](#)
- [NMAMediaCollectionPageResponse](#) - [NMAMediaCollectionPage](#)
- [NMASuggestionResponse](#) - NSArray([NSString](#))
- [NMACategoryGraphResponse](#) - NSArray([NMACategory](#))

■ **Note:**

The block will be called on the main queue.

Parameters:

- **request**

The search request being completed.

- **data**

Search results. Search result type varies according to the search request initiated. It can be nil if no results are found or an error is encountered.

- **error**

The error if the request failed, or nil if the request was successful.

See also:

[*NMARequest::startWithBlock:*](#)

[NMASearchResultType](#)

Public Enumeration Details

NMAAutoSuggestType

Include: NMAKit.framework/headers/NMAAutoSuggest.h

Represents available [*NMAAutoSuggest*](#) derived objects.

Enumeration Members:

- **NMAAutoSuggestTypeUnknown**

[*NMAAutoSuggest*](#) object unknown.

- **NMAAutoSuggestTypePlace**

A link to a suggested place.

- **NMAAutoSuggestTypeSearch**

A link to a suggested search.

- **NMAAutoSuggestTypeQuery**

A link to a suggested query.

NMACategoryFilterType

Include: NMAKit.framework/headers/NMACategoryFilter.h

Defines types of categories filter.

Enumeration Members:

- **NMACategoryFilterTypeNone**

No category.

- **NMACategoryFilterTypeAccommodation**

The accommodation category.

- **NMACategoryFilterTypeAdministrativeAreasBuildings**

The administrative-areas-buildings category.

- **NMACategoryFilterTypeEatDrink**

The eat-drink category.

- **NMACategoryFilterTypeGoingOut**

The going-out category.

- **NMACategoryFilterTypeLeisureOutdoor**

The leisure-outdoor category.

- **NMACategoryFilterTypeNaturalGeographical**

The natural-geographical category.

- **NMACategoryFilterTypeShopping**

The shopping category.

- **NMACategoryFilterTypeSightsMuseums**

The sights-museums category.

- **NMACategoryFilterTypeTransport**

The transport category.

- **NMACategoryFilterTypePetrolStation**

The petrol-station category.

- **NMACategoryFilterTypeAtmBankExchange**

The atm-bank-exchange category.

- **NMACategoryFilterTypeToiletRestArea**

The toilet-rest-area category.

- **NMACategoryFilterTypeHospitalHealthCareFacility**

The hospital-health-care-facility category.

NMAMediaType

Include: NMKit.framework/headers/NMAMedia.h

Represents available [NMAMedia](#) derived objects.

Enumeration Members:

- **NMAMediaTypeUnknown**

NMAMedia object unknown.

- **NMAMediaTypeEditorial**

NMAMediaEditorial.

- **NMAMediaTypeImage**

NMAMediaImage.

- **NMAMediaTypeReview**

NMAMediaReview.

- **NMAMediaTypeRating**

NMAMediaRating.

NMATextFormat

Include: NMAKit.framework/headers/NMARequest.h

Search result text output formats.

Enumeration Members:

- **NMATextFormatHTML**
The request results will be rendered as HTML.
- **NMATextFormatPlain**
The request results will be rendered as plain text.

Public Variable Details

FOUNDATION_EXPORT NSString *const _Nonnull NMAPPlacesContentWikipedia

Include: NMAKit.framework/headers/NMAPPlaces.h

Places Wikipedia Content.

FOUNDATION_EXPORT NSString *const _Nonnull NMAPPlacesSourceBuilding

Include: NMAKit.framework/headers/NMAPPlaces.h

Places Reference Source for building IDs.

**NS_ASSUME_NONNULL_BEGIN FOUNDATION_EXPORT NSString *const
_Nonnull NMAPPlacesSourcePVID**

Include: NMAKit.framework/headers/NMAPPlaces.h

Places Reference Source for core POI IDs.

FOUNDATION_EXPORT NSString *const _Nonnull NMAPPlacesSourceSharing

Include: NMAKit.framework/headers/NMAPPlaces.h

Places Reference Source for sharing IDs.

FOUNDATION_EXPORT NSString *const _Nonnull NMAPPlacesSourceVenuesAll

Include: NMAKit.framework/headers/NMAPPlaces.h

Places Reference Source for all types of venue IDs.

FOUNDATION_EXPORT NSString *const _Nonnull NMAPPlacesSourceVenuesContent

Include: NMAKit.framework/headers/NMAPPlaces.h

Places Reference Source for venue content IDs.

FOUNDATION_EXPORT NSString *const _Nonnull NMAPPlacesSourceVenuesDestination

Include: NMAKit.framework/headers/NMAPPlaces.h

Places Reference Source for venue destination IDs.

FOUNDATION_EXPORT NSString *const _Nonnull NMAPPlacesSourceVenuesVenue

Include: NMAKit.framework/headers/NMAPPlaces.h

Places Reference Source for venue IDs.

FOUNDATION_EXPORT NSString *_Nonnull const NMA_PLACES_ERROR_DOMAIN

Include: NMAKit.framework/headers/NMAResultListener.h

Defines the name for Places API error domain

<NMAResultListener>

Protocol Summary

protocol <NMAResultListener>

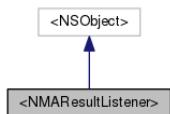
Derived from: <NSObject>

Represents an event listener that reports information about query progress changes and query completion.

Include: NMAKit.framework/headers/NMAResultListener.h

Inheritance Diagrams

Figure 58: Public inheritance diagram for <NMAResultListener> – <NMAResultListener>



[For complete information, see the section [Protocol Details](#)]

Instance Method Summary

Table 72: Instance Methods

Instance Method Summary
<pre>- (void) request:(NMARequest *_Nonnull) request didCompleteWithData:(id _Nullable) data error:(NSError *_Nonnull) error requestDidComplete:(void)</pre>

Protocol Details

Represents an event listener that reports information about query progress changes and query completion.

Methods of this protocol are called on the main queue.

Instance Method Details

```
- (void) request:(NMARequest *_Nonnull) request didCompleteWithData:(id _Nullable) data error:(NSError *_Nonnull) error requestDidComplete:(void)
```

NMAAddress

Class Summary

class **NMAAddress**

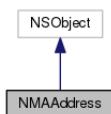
Derived from: NSObject

Represents address information for a geographic location.

Include: NMAKit.framework/headers/NMAAddress.h

Inheritance Diagrams

Figure 59: Public inheritance diagram for NMAAddress



[For complete information, see the section [Class Details](#)]

See also:

[NMAPlaceLocation](#)

Public Property Summary

Table 73: Public Properties

Public Properties
[readable, writable, strong] <code>NSString * city</code> The city name
[readable, writable, strong] <code>NSString * countryCode</code> The country code
[readable, writable, strong] <code>NSString * countryName</code> The country name
[readable, writable, strong] <code>NSString * county</code> The county name
[readable, writable, strong] <code>NSString * district</code> The district name
[readable, writable, strong] <code>NSString * floorNumber</code> The floor number in a multi-story building
[readable, writable, strong] <code>NSString * formattedAddress</code> The NSString representation of the address formatted according to local standards
[readable, writable, strong] <code>NSString * houseNumber</code> The house number
[readable, writable, strong] <code>NSString * label</code> The descriptive label for the NMAAddress
[readable, writable, strong] <code>NSString * postalCode</code> The postal code
[readable, writable, strong] <code>NSString * state</code> The state name
[readable, writable, strong] <code>NSString * street</code> The street name
[readable, writable, strong] <code>NSString * suiteNumberOrName</code> The suite number or unit name

Instance Method Summary

Table 74: Instance Methods

Instance Method Summary
<code>- (nonnull NSDictionary< NSString *, NSString *> *) additionalData</code> Gets the NSDictionary containing all additional key-value NMAAddress data

Class Details

See also:

NMAPlaceLocation

Public Property Details

[readable, writable, strong] `NSString * city`

The city name.

[readable, writable, strong] `NSString * countryCode`

The country code.

▀ Note:

The country code defined in "ISO 3166-1 alpha-3" three letter format.

[readable, writable, strong] `NSString * countryName`

The country name.

[readable, writable, strong] `NSString * county`

The county name.

[readable, writable, strong] `NSString * district`

The district name.

[readable, writable, strong] `NSString * floorNumber`

The floor number in a multi-story building.

[readable, writable, strong] `NSString * formattedAddress`

The NSString representation of the address formatted according to local standards.

[readable, writable, strong] `NSString * houseNumber`

The house number.

[readable, writable, strong] `NSString * label`

The descriptive label for the NMAAddress.

[readable, writable, strong] `NSString * postalCode`

The postal code.

[readable, writable, strong] `NSString * state`

The state name.

[readable, writable, strong] `NSString * street`

The street name.

[readable, writable, strong] `NSString * suiteNumberOrName`

The suite number or unit name.

Instance Method Details

-`(nonnull NSDictionary< NSString *, NSString *> *) additionalData`

Gets the NSDictionary containing all additional key-value NMAAddress data.

Returns:

The NSDictionary of additional key-value data

NMAAutoSuggest

Class Summary

class `NMAAutoSuggest`

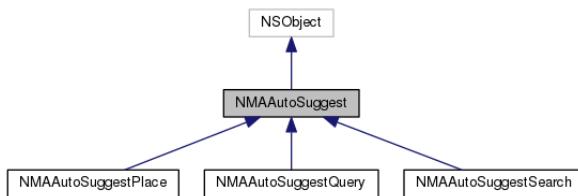
Derived from: NSObject

Represents the base interface for suggested places, searches and query.

Include: `NMAKit.framework/headers/NMAAutoSuggest.h`

Inheritance Diagrams

Figure 60: Public inheritance diagram for NMAAutoSuggest



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 75: Public Properties

Public Properties
<p>[readable, strong] <code>NSString * highlightedTitle</code> The NSString representation of the title for this place to be displayed to the user</p>
<p>[readable, strong] <code>NSString * title</code> The NSString representation of the title for this place to be displayed to the user</p>
<p>[readable, assign] <code>NMAAutoSuggestType type</code> The specialized AutoSuggest type</p>

Class Details

Represents the base interface for suggested places, searches and query.

The following specialized type might be available:

- place
- search
- query

Public Property Details

[readable, strong] `NSString * highlightedTitle`

The NSString representation of the title for this place to be displayed to the user. It contains markup highlighting the parts of the string that were matched.

For example: User performs an AutoSuggest search of "Rest" will return the following. title:Joey Restaurant highlightedTitle:Joey Rest aurant

[readable, strong] `NSString * title`

The NSString representation of the title for this place to be displayed to the user.

[readable, assign] `NMAAutoSuggestType` `type`

The specialized AutoSuggest type.

NMAAutoSuggestPlace

Class Summary

class `NMAAutoSuggestPlace`

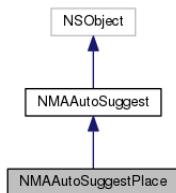
Derived from: [NMAAutoSuggest](#)

Represents a suggested place.

Include: `NMAKit.framework/headers/NMAAutoSuggestPlace.h`

Inheritance Diagrams

Figure 61: Public inheritance diagram for `NMAAutoSuggestPlace`



[For complete information, see the section [Class Details](#)]

See also:

[NMAAutoSuggestSearch](#)

Public Property Summary

Table 76: Public Properties

Public Properties
[readable, strong] <code>NMAGeoBoundingBox</code> * <code>boundingBox</code> The enclosing NMAGeoBoundingBox describing a range of coordinates that correspond to the NMAPlace
[readable, strong] <code>NSString</code> * <code>category</code> The NSString representation of the category for the NMAPlace
[readable, strong] <code>NMAGeoCoordinates</code> * <code>position</code> The NMAGeoCoordinates representing the geographical position of the NMAPlace
[readable, strong] <code>NSString</code> * <code>vicinityDescription</code> The NSString description of the location of the NMAPlace

Instance Method Summary

Table 77: Instance Methods

Instance Method Summary
<pre>-(nullable NMAPlaceRequest *) placeDetailsRequest</pre> <p>Gets the NMAPlaceRequest to retrieve the NMAPlace details</p>

Class Details

Represents a suggested place.

See also:

[NMAAutoSuggestSearch](#)

Public Property Details

[readable, strong] [NMAGeoBoundingBox](#) * **boundingBox**

The enclosing NMAGeoBoundingBox describing a range of coordinates that correspond to the [NMAPlace](#). Typically, bounding boxes are associated with places such as cities and countries.

Note:

Bounding box information for a place is optional. Attempts to read this property could return nil.

[readable, strong] [NSString](#) * **category**

The NSString representation of the category for the [NMAPlace](#).

[readable, strong] [NMAGeoCoordinates](#) * **position**

The NMAGeoCoordinates representing the geographical position of the [NMAPlace](#).

[readable, strong] [NSString](#) * **vicinityDescription**

The NSString description of the location of the [NMAPlace](#).

Typically, this property is derived from the address, but could also contain any other description that helps an application user understand where the place is located.

Note:

Vicinity description for a place is optional. Attempts to read this property could return nil.

Instance Method Details

-(nullable [NMAPlaceRequest](#) *) [placeDetailsRequest](#)

Gets the NMAPlaceRequest to retrieve the [NMAPlace](#) details.

NMAAutoSuggestQuery

Class Summary

class **NMAAutoSuggestQuery**

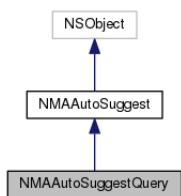
Derived from: [NMAAutoSuggest](#)

Represents a suggested query.

Include: `NMAKit.framework/headers/NMAAutoSuggestQuery.h`

Inheritance Diagrams

Figure 62: Public inheritance diagram for NMAAutoSuggestQuery



[For complete information, see the section [Class Details](#)]

See also:

[NMAAutoSuggestQuery](#)

Public Property Summary

Table 78: Public Properties

Public Properties
<code>[readable, strong] NSString * completion</code>

Instance Method Summary

Table 79: Instance Methods

Instance Method Summary
<code>-(nullable NMAAutoSuggestionRequest *) autoSuggestionRequest</code>

Class Details

Represents a suggested query. Query can be used to trigger next Suggest request.

See also:

NMAAutoSuggestQuery

Public Property Details

[readable, strong] `NSString * completion`

The suggested query completion of the given string

Instance Method Details

-`(nullable NMAAutoSuggestionRequest *) autoSuggestionRequest`

Gets the NMAAutoSuggestionRequest to perform the next AutoSuggestion request with the user's chosen completed query term

NMAAutoSuggestSearch

Class Summary

class `NMAAutoSuggestSearch`

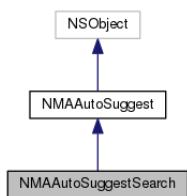
Derived from: `NMAAutoSuggest`

Represents a suggested search.

Include: `NMAKit.framework/headers/NMAAutoSuggestSearch.h`

Inheritance Diagrams

Figure 63: Public inheritance diagram for `NMAAutoSuggestSearch`



[For complete information, see the section [Class Details](#)]

See also:

[NMAAutoSuggestPlace](#)

Public Property Summary

Table 80: Public Properties

Public Properties
<p>[readable, strong] <code>NMAGeoBoundingBox * boundingBox</code> The enclosing NMAGeoBoundingBox describing a range of coordinates that correspond to the <code>NMAPlace</code></p>
<p>[readable, strong] <code>NSString * category</code> The NSString representation of the category for the <code>NMAPlace</code></p>
<p>[readable, strong] <code>NMAGeoCoordinates * position</code> The NMAGeoCoordinates representing the geographical position of the <code>NMAPlace</code></p>

Instance Method Summary

Table 81: Instance Methods

Instance Method Summary
<p><code>-(nullable NMADiscoveryRequest *) suggestedSearchRequest</code> Gets the NMADiscoveryRequest to perform a suggested search e</p>

Class Details

Represents a suggested search.

See also:

[NMAAutoSuggestPlace](#)

Public Property Details

[readable, strong] `NMAGeoBoundingBox * boundingBox`

The enclosing NMAGeoBoundingBox describing a range of coordinates that correspond to the `NMAPlace`. Typically, bounding boxes are associated with places such as cities and countries.

▀ Note:

Bounding box information for a place is optional. Attempts to read this property could return nil.

[readable, strong] `NSString * category`

The NSString representation of the category for the `NMAPlace`.

▀ Note:

category for a place is optional. Attempts to read this property could return nil.

[readable, strong] `NMAGeoCoordinates * position`

The NMAGeoCoordinates representing the geographical position of the *NMAPlace*.

■ **Note:**

position for a place is optional. Attempts to read this property could return nil.

Instance Method Details

`- (nullable NMADiscoveryRequest *) suggestedSearchRequest`

Gets the NMADiscoveryRequest to perform a suggested search e.g. category search.

NMAAutoSuggestionRequest

Class Summary

class **NMAAutoSuggestionRequest**

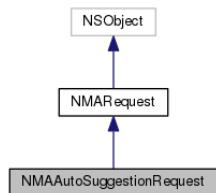
Derived from: *NMAResponse*

Represents a request to retrieve a list of suggested search terms, instant results and refined search links related to a specified location context and partial search term.

Include: `NMAKit.framework/headers/NMAAutoSuggestionRequest.h`

Inheritance Diagrams

Figure 64: Public inheritance diagram for NMAAutoSuggestionRequest



[For complete information, see the section *Class Details*]

Public Property Summary

Table 82: Public Properties

Public Properties
<p>[readable, writable, assign] <code>NMATextFormat</code> <code>textFormat</code> The text format of the request results</p>

Class Details

Represents a request to retrieve a list of suggested search terms, instant results and refined search links related to a specified location context and partial search term.

Public Property Details

[readable, writable, assign] `NMATextFormat` `textFormat`

The text format of the request results.

■ **Note:**

The default value is `NMATextFormatHTML`.

NMACategory

Class Summary

class `NMACategory`

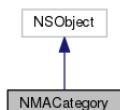
Derived from: `NSObject`

Represents a category with which a `NMAPlace` can be associated.

Include: `NMAKit.framework/headers/NMACategory.h`

Inheritance Diagrams

Figure 65: Public inheritance diagram for `NMACategory`



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 83: Public Properties

Public Properties
[readable, assign] <code>NSString *</code> <code>iconUrl</code> Gets the URL of the category icon
[readable, assign] <code>NSString *</code> <code>name</code> Gets the display name for the category
[readable, assign] <code>NMACategory *</code> <code>parentCategory</code> Gets the parent of this category

Public Properties

[readable, assign] NSArray< *NMACategory* * > * **subCategories**

Gets the list of subcategories NMACategory of this category

[readable, assign] NSString * **uniqueId**

Gets the unique identifier for the category

Instance Method Summary

Table 84: Instance Methods

Instance Method Summary

-((unavailable("Default initializer not available.")) **__attribute__**

Default initializer not available.

Class Details

Represents a category with which a *NMAPlace* can be associated.

Categories can be used to search for places (e.g. a search that is based on a category with which certain places are associated).

Public Property Details

[readable, assign] NSString * **iconUrl**

Gets the URL of the category icon.

■ **Note:**

Application can download the icon online with this URL, but it needs to use its own icon for offline.

[readable, assign] NSString * **name**

Gets the display name for the category. The name is localized to the current locale of the device.

[readable, assign] *NMACategory* * **parentCategory**

Gets the parent of this category.

Note that an *NMACategory* might have no parent category, in which case this method would return nil.

[readable, assign] NSArray< *NMACategory* * > * **subCategories**

Gets the list of subcategories NMACategory of this category.

Note an *NMACategory* might have no subcategories, in which case this method would return empty array.

[readable, assign] `NSString * uniqueId`

Gets the unique identifier for the category.

Instance Method Details

-((unavailable("Default initializer not available.")) `__attribute__`

Default initializer not available.

NMACategoryFilter

Class Summary

class **NMACategoryFilter**

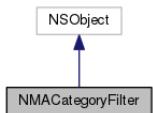
Derived from: `NSObject`

Represents a filter used when performing a search for popular places within a specific location (exploring).

Include: `NMAKit.framework/headers/NMACategoryFilter.h`

Inheritance Diagrams

Figure 66: Public inheritance diagram for NMACategoryFilter



[For complete information, see the section [Class Details](#)]

Instance Method Summary

Table 85: Instance Methods

Instance Method Summary
<p><code>- (void) <i>addCategoryFilterFromType:(fromType:add)</i></code></p> <p>Adds a <code>NMACategoryFilterType</code> to the <code>NMACategoryFilter</code></p>
<p><code>- (void) <i>addCategoryFilterFromUniqueId:(fromUniqueId:add)</i></code></p> <p>Adds an <code>NMACategory</code>'s <code>uniqueId</code> to the <code>NMACategoryFilter</code></p>
<p><code>- (nonnull NSString *) <i>toString</i></code></p> <p>Returns a stringified <code>NMACategoryFilter</code>, within which individual categories are separated by commas</p>

Class Details

Represents a filter used when performing a search for popular places within a specific location (exploring). The filter limits a search to specified categories.

Instance Method Details

- (void) addCategoryFilterFromType:(fromType:add)

Adds a NMACategoryFilterType to the [NMACategoryFilter](#).

Parameters:

- **categoryFilterType**

An NMACategoryFilterType filter

- (void) addCategoryFilterFromUniqueId:(fromUniqueId:add)

Adds an [NMACategory](#)'s uniqueId to the [NMACategoryFilter](#).

Parameters:

- **uniqueId**

An NSString for [NMACategory](#)'s uniqueId

- (nonnull NSString *) toString

Returns a stringified [NMACategoryFilter](#), within which individual categories are separated by commas.

Returns:

The stringified, comma-delimited set of categories.

NMACategoryGraphRequest

Class Summary

class **NMACategoryGraphRequest**

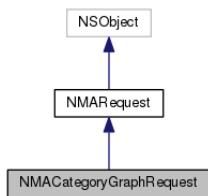
Derived from: [NMARequest](#)

Represents a request to retrieve an NSArray of NMACategory objects.

Include: `NMAKit.framework/headers/NMACategoryGraphRequest.h`

Inheritance Diagrams

Figure 67: Public inheritance diagram for NMACategoryGraphRequest



[For complete information, see the section [Class Details](#)]

Class Details

Represents a request to retrieve an NSArray of [NMACategory](#) objects.

NMAContactDetail

Class Summary

class **NMAContactDetail**

Derived from: NSObject

Represents detailed information about a contact for a NMAPlace.

Include: NMAKit.framework/headers/NMAContactDetail.h

Inheritance Diagrams

Figure 68: Public inheritance diagram for NMAContactDetail



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 86: Public Properties

Public Properties
<code>[readable, assign] NSString * label</code>
The localized label describing the mechanism by which application end users can contact the place
<code>[readable, assign] NSString * type</code>
The type (email, fax, phone, website, etc)

Public Properties

[readable, assign] `NSString * value`

The string value appropriate to the localized label

Class Details

Represents detailed information about a contact for a `NMAPlace`.

For example, if a `NMAPlace` has among its known contacts array the phone number 555-1234, use this interface to access details about that particular contact's label ("Phone") and value ("555-1234").

Public Property Details

[readable, assign] `NSString * label`

The localized label describing the mechanism by which application end users can contact the place.

[readable, assign] `NSString * type`

The type (email, fax, phone, website, etc.) of the `NMAContactDetail`.

[readable, assign] `NSString * value`

The string value appropriate to the localized label.

NMADiscoveryLink

Class Summary

class `NMADiscoveryLink`

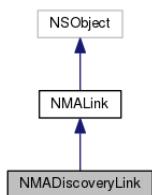
Derived from: `NMALink`

Represents a discovery search results link that can be used to perform another discovery search.

Include: `NMAKit.framework/headers/NMADiscoveryLink.h`

Inheritance Diagrams

Figure 69: Public inheritance diagram for `NMADiscoveryLink`



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 87: Public Properties

Public Properties
<p>[readable, assign] <code>NMADiscoveryRequest * request</code> An NMADiscoveryRequest object to perform another NMADiscoveryPage request</p>

Class Details

Represents a discovery search results link that can be used to perform another discovery search.

Public Property Details

[readable, assign] `NMADiscoveryRequest * request`

An NMADiscoveryRequest object to perform another NMADiscoveryPage request.

NMADiscoveryPage

Class Summary

class **NMADiscoveryPage**

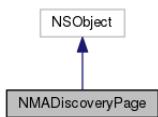
Derived from: NSObject

Represents the paginated result of a NMADiscoveryRequest request.

Include: NMAKit.framework/headers/NMADiscoveryPage.h

Inheritance Diagrams

Figure 70: Public inheritance diagram for NMADiscoveryPage



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 88: Public Properties

Public Properties
[readable, assign] NSInteger available The total number of available NMAPlaceLink or NMADiscoveryLink objects for this discovery page
[readable, assign] NSArray< __kindof NMALink * > * discoveryResults Array of NMAPlaceLink or NMADiscoveryLink objects for the current page of results
[readable, assign] NMADiscoveryRequest * nextPageRequest The NMADiscoveryRequest for requesting the next page of results
[readable, assign] NSInteger offsetCount Result offset of the current results page into the total number of results available

Class Details

Represents the paginated result of a [NMADiscoveryRequest](#) request.

[NMADiscoveryRequest](#) objects are typically created using the [NMAPlaces](#) interface to initialize search, explore or here requests.

Results may be split into multiple page. Each results page contains a collection of [NMAPlaceLink](#) and [NMADiscoveryLink](#) objects, each of which represents either an actual place or a further discovery request.

Public Property Details

[readable, assign] NSInteger **available**

The total number of available [NMAPlaceLink](#) or [NMADiscoveryLink](#) objects for this discovery page.

[readable, assign] NSArray< __kindof [NMALink](#) * > * **discoveryResults**

Array of [NMAPlaceLink](#) or [NMADiscoveryLink](#) objects for the current page of results.

■ **Note:**

The [discoveryResults](#) may contain more results than the maximum number specified in [NMAResponse::collectionSize](#). These extra results link to different resources and contain further links via which the user will be linked to a place.

[readable, assign] [NMADiscoveryRequest](#) * **nextPageRequest**

The [NMADiscoveryRequest](#) for requesting the next page of results.

■ **Note:**

This will be nil if there is no next page.

[readable, assign] NSInteger offsetCount

Result offset of the current results page into the total number of results available.

■ Note:

The number of results on the current page is the number of elements in *discoveryResults*.

NMADiscoveryRequest

Class Summary

class **NMADiscoveryRequest**

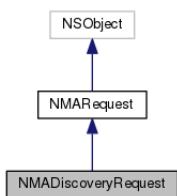
Derived from: *NMARequest*

Represents a request to retrieve an *NMADiscoveryPage* object.

Include: *NMAKit.framework/headers/NMADiscoveryRequest.h*

Inheritance Diagrams

Figure 71: Public inheritance diagram for **NMADiscoveryRequest**



[For complete information, see the section *Class Details*]

Public Property Summary

Table 89: Public Properties

Public Properties
<p>[readable, writable, assign] <i>NMATextFormat</i> <i>textFormat</i> The text format of the request results</p>

Class Details

Represents a request to retrieve an *NMADiscoveryPage* object.

Public Property Details

[readable, writable, assign] *NMATextFormat* *textFormat*

The text format of the request results.

■ Note:

The default value is NMATextFormatHTML.

NMAExtendedAttribute

Class Summary

class **NMAExtendedAttribute**

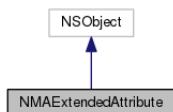
Derived from: NSObject

Represents additional detailed information about a [NMAPlace](#), an extensible collection of attributes that can include:

Include: NMAKit.framework/headers/NMAExtendedAttribute.h

Inheritance Diagrams

Figure 72: Public inheritance diagram for NMAExtendedAttribute



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 90: Public Properties

Public Properties
<p>[readable, assign] NSString * label The localized display label for the extended attribute, e.g. "Payment", which can be displayed directly by the client application.</p>
<p>[readable, assign] NSString * text The localized text for the extended attribute as NSString, e.g. "VISA", which can be displayed directly by the client application.</p>
<p>[readable, assign] NSString * uniqueid Gets the unique identifier for the NMAExtendedAttribute.</p>

Class Details

Represents additional detailed information about a [NMAPlace](#), an extensible collection of attributes that can include:

- Payment - a list of available payment methods (such as cash, credit card, or direct debit, etc)

- OpeningHours - a list of hours during which the place is open for business
- AnnualClosings - a description of annual closing dates such as holidays or other special occasions
- Price - a price list
- NearestLandmark - a description of the nearest landmark
- LanguagesSpoken - a list of the languages that are spoken at the place
- AvailableParking - a list of parking options available nearby
- Smoking - a notification of whether smoking is allowed
- DisabledAccess - a notification of whether disabled access is available

Public Property Details

[readable, assign] NSString * label

The localized display label for the extended attribute, e.g. "Payment", which can be displayed directly by the client application.

[readable, assign] NSString * text

The localized text for the extended attribute as NSString, e.g. "VISA", which can be displayed directly by the client application.

If the text represents a list of items, the items are separated by a line break entity (
 if the text format is HTML-encoded or newline if the text format is plain).

[readable, assign] NSString * uniqueId

Gets the unique identifier for the NMAExtendedAttribute.

 **Note:**

See the list of attributes in the class description.

NMAGeocodeRequest

Class Summary

class **NMAGeocodeRequest**

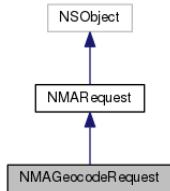
Derived from: [NMARequest](#)

Represents a request to retrieve NMAPlaceLocation data by way of NMAGeocoder search services.

Include: `NMAKit.framework/headers/NMAGeocodeRequest.h`

Inheritance Diagrams

Figure 73: Public inheritance diagram for NMAGeocodeRequest



[For complete information, see the section [Class Details](#)]

Class Details

Represents a request to retrieve [NMAPlaceLocation](#) data by way of [NMAGocoder](#) search services.

■ Note:

A NMAGeocodeRequest is a kind of [NMAResult](#), and its response is an instance of [NMAGeocodeResult](#).

NMAGeocodeResult

Class Summary

class NMAGeocodeResult

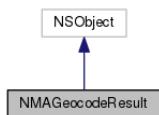
Derived from: NSObject

Represents the result of a geocode request.

Include: NMAKit.framework/headers/NMAGeocodeResult.h

Inheritance Diagrams

Figure 74: Public inheritance diagram for NMAGeocodeResult



[For complete information, see the section [Class Details](#)]

See also:

[NMAGeocodeRequest](#)

Public Property Summary

Table 91: Public Properties

Public Properties
<p>[readable, assign] <code>NMAPlaceLocation</code> * <code>location</code> The location data of the geocode result.</p>
<p>[readable, assign] <code>NSDictionary< NSString *, NSNumber *> *</code> <code>matchQuality</code> Details about the quality of the result.</p>
<p>[readable, assign] <code>float</code> <code>relevance</code> The relevance of the result location to the original search parameters.</p>

Class Details

Represents the result of a geocode request.

The data of a geocode result is represented by an instance of `NMAPlaceLocation`, accessed through the `location` property. The quality of the result can be assessed in a broad sense with the `relevance` property, or in more detail using the `matchQuality` dictionary.

See also:

[NMAGeocodeRequest](#)

Public Property Details

[readable, assign] `NMAPlaceLocation` * `location`

The location data of the geocode result.

[readable, assign] `NSDictionary< NSString *, NSNumber *> *` `matchQuality`

Details about the quality of the result.

Additional information about the accuracy of the result is available in the `matchQuality` dictionary. It contains one or more `NSNumber` values representing how well each individual search parameter was matched. The values will be in the range [0, 1] with a value of 1 indicating a perfect match.

The possible keys are "State", "County", "City", "District", "Street", "SecondaryStreet", "HouseNumber", and "PostalCode".

■ Note:

The "SecondaryStreet" key will be present if an intersection was specified in the geocode request. For example, if the request contained "1st St at 2nd Ave", the "Street" key would access the match value for "1st St" and the "SecondaryStreet" key would access the match value for "2nd Ave".

[readable, assign] `float` `relevance`

The relevance of the result location to the original search parameters.

The relevance of a result provides a measure of how accurate or reliable it is. The valid range is [0, 1], with a value of 1 representing a perfect match. If relevancy data is unavailable (such as for an offline search), this property will have the value NMAGeocodeResultInvalidRelevance.

NMAGeocoder

Class Summary

class **NMAGeocoder**

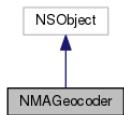
Derived from: `NSObject`

Represents a factory used to instantiate geocoder requests.

Include: `NMAKit.framework/headers/NMAGeocoder.h`

Inheritance Diagrams

Figure 75: Public inheritance diagram for `NMAGeocoder`



[For complete information, see the section [Class Details](#)]

Instance Method Summary

Table 92: Instance Methods

Instance Method Summary
<pre>- (NMAGeocodeRequest *) createGeocodeRequestWithQuery:(NSString *) query searchArea:(NMAGeoBoundingBox *) searchArea locationContext:(query:searchArea:locationContext:makeGeocodeRequest)</pre> <p>Creates a geocoder request that resolves a free text query into an array of <code>NMAPlaceLocation</code></p>
<pre>- (NMAGeocodeRequest *) createGeocodeRequestWithQuery:(NSString *) query searchRadius:(NSInteger) searchRadius searchCenter:(query:searchRadius:searchCenter:makeGeocodeRequest)</pre> <p>Creates a geocoder request that resolves a free text query into an array of <code>NMAPlaceLocation</code></p>
<pre>- (NMAResponseGeocodeRequest *) createReverseGeocodeRequestWithGeoCoordinates:(geoCoordinates:makeReverseGeocodeRequest)</pre> <p>Creates a reverse geocoder request that resolves a <code>NMAGeoCoordinates</code> context into a <code>NMAPlaceLocation</code></p>

Class Method Summary

Table 93: Class Methods

Class Methods
<pre>+(< NMAGeocoder *>) sharedGeocoder</pre> <p>Returns the NMAGeocoder singleton instance</p>

Class Details

Represents a factory used to instantiate geocoder requests.

A NMAGeocoder enables searching for location data by way of the following search services:

- One box Geocode - matches a free text query (and location context) to its correct *NMAPlaceLocation* on the map (its latitude and longitude).
- Reverse Geocode - retrieves a *NMAPlaceLocation* based on a given *NMAGeoCoordinates*.

■ Note:

NMAgeocoder requires valid authentication credentials to be set via *NMAApplicationContext*. If valid credentials are not present an attempt to call *sharedNMAGeocoder* will result in a runtime assert.

Instance Method Details

```
-(< NMAGeocodeRequest *>) createGeocodeRequestWithQuery:(NSString *)query searchArea:(NMAGeoBoundingBox *)searchArea locationContext:(query:searchArea:locationContext:makeGeocodeRequest)
```

Creates a geocoder request that resolves a free text query into an array of *NMAPlaceLocation*. After creating the request, [request startWithListener:(id<NMAResultListener>)] needs to be called to start the search.

Parameters:

- **query**
Query text specifying the search item to locate
- **searchArea**
NMAGeoBoundingBox representing the search area (this is an optional parameter and should not be set unless such functionality is explicitly intended in the application)
- **geoCoordinates**
NMAGeoCoordinates object representing the location context used to search for results that are appropriate to the query parameter

Returns:

The *NMAgeocodeRequest*

```
-(< NMAGeocodeRequest *>) createGeocodeRequestWithQuery:(NSString *)query searchRadius:(NSInteger)searchRadius searchCenter:(query:searchRadius:searchCenter:makeGeocodeRequest)
```

Creates a geocoder request that resolves a free text query into an array of [NMAPlaceLocation](#). After creating the request, [request startWithListener:(id<NMAResultListener>)] needs to be called to start the search.

Parameters:

- **query**
Query text specifying the search item to locate
- **searchRadius**
Radius in meters, in which the search should be performed
- **geoCoordinates**
[NMAGeoCoordinates](#) object representing the location context used to search for results that are appropriate to the query parameter. Note that this search location can be overridden if the query text already specifies a search area, such as a state.

Returns:

The [NMAGeocodeRequest](#)

```
- ( NMAResponseGeocodeRequest * ) createReverseGeocodeRequestWithGeoCoordinates:  
(geoCoordinates:makeReverseGeocodeRequest)
```

Creates a reverse geocoder request that resolves a [NMAGeoCoordinates](#) context into a [NMAPlaceLocation](#). After creating the request, [request startWithListener:(id<NMAResultListener>)] needs to be called to start the search.

Parameters:

- **geoCoordinates**
A query [NMAGeoCoordinates](#) context

Returns:

The [NMAResponseGeocodeRequest](#)

Class Method Details

```
+ ( NMAGeocoder * ) sharedGeocoder
```

Returns the NMAGeocoder singleton instance.

■ Note:

Use this method to obtain a NMAGeocoder instance. Do not call init directly.

Returns:

shared NMAGeocoder instance

NMALink

Class Summary

class **NMALink**

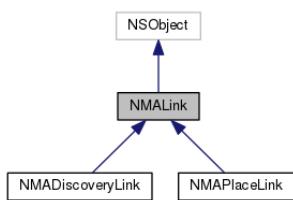
Derived from: `NSObject`

Represents a HERE Places resource link which requires another request to be made in order to retrieve the full details of the resource.

Include: `NMAKit.framework/headers/NMALink.h`

Inheritance Diagrams

Figure 76: Public inheritance diagram for NMALink



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 94: Public Properties

Public Properties
<code>[readable, assign] NSString * iconUrl</code> Gets an URL of the icon for the resource to which associated with this NMALink
<code>[readable, assign] NSString * name</code> Gets the localized name for the resource to which the NMALink refers
<code>[readable, assign] NSString * uniqueId</code> Gets the unique identifier for the resource to which the NMALink refers
<code>[readable, assign] NSString * url</code> Gets the NSString representation of the link URL

Class Details

Represents a HERE Places resource link which requires another request to be made in order to retrieve the full details of the resource. The link contains basic metadata about the linked resource.

The interface is typically subclassed to provide methods to make the appropriate resource request.

Public Property Details

[readable, assign] `NSString * iconUrl`

Gets an URL of the icon for the resource to which associated with this [NMALink](#).

■ **Note:**

An icon URL for a [NMALink](#) object is optional, so this property could be nil. Application can download the icon online with this URL, but it needs to use its own icon for offline.

[readable, assign] `NSString * name`

Gets the localized name for the resource to which the NMALink refers.

Client applications can safely display this name to the user.

■ **Note:**

A title for a linked object is optional, so this property could be nil.

[readable, assign] `NSString * uniqueId`

Gets the unique identifier for the resource to which the NMALink refers.

■ **Note:**

An identifier for a linked object is optional, so this property could be nil.

[readable, assign] `NSString * url`

Gets the NSString representation of the link URL.

This URL may be used to perform an HTTP GET request to the HERE Places REST service to retrieve the full metadata of the linked resources.

The request may return a HTML or JSON response. If url contains "places.hybrid.api.here.com", it will return a JSON response as specified at <http://developer.here.com/places>. If URL contains other domains, it will most likely return a HTML response.

■ **Note:**

For convenience it is recommended to use request methods in derived classes when available.

NMAMedia

Class Summary

class **NMAMedia**

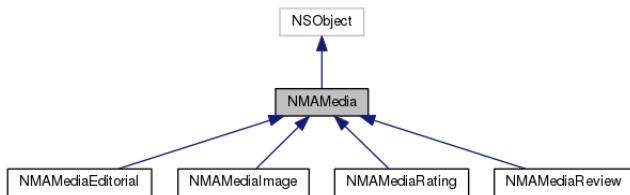
Derived from: NSObject

Represents the base interface for additional rich content about a [NMAPlace](#).

Include: NMAKit.framework/headers/NMAMedia.h

Inheritance Diagrams

Figure 77: Public inheritance diagram for NMAMedia



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 95: Public Properties

Public Properties
<p>[readable, assign] <code>NSString * attributionText</code> The NSString representation of the attribution text for the NMAMedia, according to the terms and conditions of the originating source</p>
<p>[readable, assign] <code>NMALink * supplierLink</code> The NMALink to the resource representing the supplier of the content (the object provides details on the origin of the information)</p>
<p>[readable, assign] <code>NMAMediaType type</code> The specialized content type</p>
<p>[readable, assign] <code>NMALink * viaLink</code> The NMALink to the origin of the information, usually a website of the supplier</p>

Class Details

Represents the base interface for additional rich content about a [NMAPlace](#).

The following specialized content might be available:

- images
- editorials
- reviews

Public Property Details

`[readable, assign] NSString * attributionText`

The NSString representation of the attribution text for the NMAMedia, according to the terms and conditions of the originating source.

[readable, assign] `NMALink * supplierLink`

The NMALink to the resource representing the supplier of the content (the object provides details on the origin of the information).

[readable, assign] `NMAMediaType type`

The specialized content type.

[readable, assign] `NMALink * viaLink`

The NMALink to the origin of the information, usually a website of the supplier. The type attribute of the link might need to be checked to see if there is a better user agent than a web browser for handling the media type of the linked-to resource.

NMAMediaCollectionPage

Class Summary

class **NMAMediaCollectionPage**

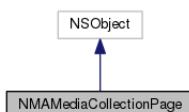
Derived from: `NSObject`

Represents a paginated collection of NMAMedia objects of a specific type.

Include: `NMAKit.framework/headers/NMAMediaCollectionPage.h`

Inheritance Diagrams

Figure 78: Public inheritance diagram for `NMAMediaCollectionPage`



[For complete information, see the section [Class Details](#)]

See also:

[NMAMedia](#)

[NMAMediaCollectionPageRequest](#)

Public Property Summary

Table 96: Public Properties

Public Properties
<p>[readable, assign] <code>NSInteger available</code> The total number of available NMAMedia objects in this given media collection</p>
<p>[readable, assign] <code>NSArray< __kindof NMAMedia * > * mediaContents</code></p>
<p>[readable, assign] <code>NMAMediaCollectionPageRequest * nextPageRequest</code> The NMAMediaCollectionPageRequest for requesting the next page of the NMAMediaCollectionPage</p>
<p>[readable, assign] <code>NSInteger offsetCount</code> The offset count for the current page of the completed NMAMediaCollectionPageRequest</p>
<p>[readable, assign] <code>NMAMediaType type</code> The specialized content type of the objects on the page</p>

Class Details

Represents a paginated collection of `NMAMedia` objects of a specific type.

■ **Note:**

Each collection contains objects of a single NMAMediaType

See also:

[NMAMedia](#)

[NMAMediaCollectionPageRequest](#)

Public Property Details

[readable, assign] `NSInteger available`

The total number of available `NMAMedia` objects in this given media collection.

[readable, assign] `NSArray< __kindof NMAMedia * > * mediaContents`

Array of `NMAMedia` objects for the current page

[readable, assign] `NMAMediaCollectionPageRequest * nextPageRequest`

The `NMAMediaCollectionPageRequest` for requesting the next page of the `NMAMediaCollectionPage`.

■ **Note:**

This will be nil if no more pages are available.

[readable, assign] `NSInteger offsetCount`

The offset count for the current page of the completed [NMAMediaCollectionPageRequest](#).

[readable, assign] [NMAMediaType](#) type

The specialized content type of the objects on the page.

NMAMediaCollectionPageRequest

Class Summary

class [NMAMediaCollectionPageRequest](#)

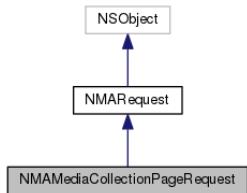
Derived from: [NMAResponse](#)

Represents a request to retrieve an [NMAMediaCollectionPage](#) object.

Include: `NMAKit.framework/headers/NMAMediaCollectionPageRequest.h`

Inheritance Diagrams

Figure 79: Public inheritance diagram for [NMAMediaCollectionPageRequest](#)



[For complete information, see the section [Class Details](#)]

Class Details

Represents a request to retrieve an [NMAMediaCollectionPage](#) object.

NMAMediaEditorial

Class Summary

class [NMAMediaEditorial](#)

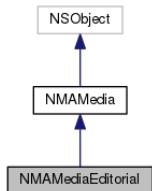
Derived from: [NMAMedia](#)

Represents editorial content about a [NMAPlace](#).

Include: `NMAKit.framework/headers/NMAMediaEditorial.h`

Inheritance Diagrams

Figure 80: Public inheritance diagram for NMAMediaEditorial



[For complete information, see the section [Class Details](#)]

See also:

[NMAMediaImage](#)

[NMAMediaReview](#)

[NMAMediaRating](#)

Public Property Summary

Table 97: Public Properties

Public Properties
<p>[readable, assign] <code>NSString * description</code> The <code>NSString</code> for this editorial content</p>
<p>[readable, assign] <code>NSString * languageCode</code> The BCP-47 language code for the editorial text</p>

Class Details

Represents editorial content about a [NMAPlace](#).

See also:

[NMAMediaImage](#)

[NMAMediaReview](#)

[NMAMediaRating](#)

Public Property Details

[readable, assign] `NSString * description`

The `NSString` for this editorial content.

[readable, assign] `NSString * languageCode`

The BCP-47 language code for the editorial text.

■ Note:

A language code for an editorial is optional and so attempts to read this property could return empty string. In this scenario the language of editorial is unknown but it is likely to be English.

NMAMedialimage

Class Summary

class **NMAMedialimage**

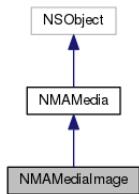
Derived from: [NMAMedia](#)

Represents an image associated with an NMAPlace.

Include: `NMAKit.framework/headers/NMAMediaImage.h`

Inheritance Diagrams

Figure 81: Public inheritance diagram for NMAMedialimage



[For complete information, see the section [Class Details](#)]

See also:

[NMAMediaEditorial](#)

[NMAMediaReview](#)

[NMAMediaRating](#)

Public Property Summary

Table 98: Public Properties

Public Properties
<code>[readable, assign] NSString * <i>imageSource</i></code> The external URL of the image source
<code>[readable, assign] NSString * <i>uniqueId</i></code> The unique identifier of the image
<code>[readable, assign] NMALink * <i>user</i></code> The details of the user who contributed the image

Instance Method Summary

Table 99: Instance Methods

Instance Method Summary
<pre>-(nullable NSString *) URLForImageWithWidth:(NSInteger) width height:(width:height:imageURL)</pre> <p>Gets a URL for an image with specific width-height dimensions</p>

Class Details

Represents an image associated with an [NMAPlace](#). Images are uploaded by the HERE user community.

See also:

[NMAMediaEditorial](#)

[NMAMediaReview](#)

[NMAMediaRating](#)

Public Property Details

[readable, assign] NSString * imageSource

The external URL of the image source.

[readable, assign] NSString * uniqueId

The unique identifier of the image.

■ Note:

An ID for an image is optional. Attempts to read this property could return empty string.

[readable, assign] NMALink * user

The details of the user who contributed the image.

■ Note:

A user-contributor for an image is optional. Attempts to read this property could return nil.

Instance Method Details

-(nullable NSString *) URLForImageWithWidth:(NSInteger) width height:(width:height:imageURL)

Gets a URL for an image with specific width-height dimensions.

The provided width-height are used as upper boundaries for the dimensions of the returned images. The solution will never scale up small images and will maintain the aspect ratio of the original image.

A URL of image with specific width-height dimensions will not be returned if it has not been queried with `NMAResponse addImageSizeWithWidth:height`

For example:

```
image_dimensions=w32-h32,w64-h64  
image_dimensions=w32-h32,w300  
image_dimensions=h200
```

Parameters:

- **width**
The image width.
- **height**
The image height.

Returns:

The URL to retrieve the image with the specified width-height dimensions. Can be nil if an image with the specified dimensions is unavailable.

NMAMediaRating

Class Summary

class **NMAMediaRating**

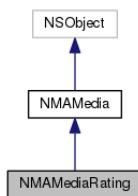
Derived from: [NMAMedia](#)

Represents rating content about a NMAPlace.

Include: `NMAKit.framework/headers/NMAMediaRating.h`

Inheritance Diagrams

Figure 82: Public inheritance diagram for NMAMediaRating



[For complete information, see the section [Class Details](#)]

See also:

[NMAMediaImage](#)

[NMAMediaReview](#)

[NMAMediaEditorial](#)

Public Property Summary

Table 100: Public Properties

Public Properties
<code>[readable, assign] double average</code> The average rating for this <i>NMAPlace</i>
<code>[readable, assign]NSUInteger count</code> The number of individual votes that users provide for rating this <i>NMAPlace</i>

Class Details

Represents rating content about a *NMAPlace*.

See also:

[NMAMediaImage](#)

[NMAMediaReview](#)

[NMAMediaEditorial](#)

Public Property Details

`[readable, assign] double average`

The average rating for this *NMAPlace*.

`[readable, assign] NSUInteger count`

The number of individual votes that users provide for rating this *NMAPlace*.

NMAMediaReview

Class Summary

class **NMAMediaReview**

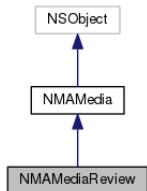
Derived from: [NMAMedia](#)

Represents review content that provides access to user reviews and ratings about a *NMAPlace*.

Include: `NMAKit.framework/headers/NMAMediaReview.h`

Inheritance Diagrams

Figure 83: Public inheritance diagram for NMAMediaReview



[For complete information, see the section [Class Details](#)]

See also:

[NMAMediaEditorial](#)

[NMAMedialmage](#)

[NMAMediaRatings](#)

Public Property Summary

Table 101: Public Properties

Public Properties
<p>[readable, assign] NSDate * <i>contributionDate</i> The NSDate specifying the date when the user contributed the review</p>
<p>[readable, assign] NMALink * <i>contributor</i> Details of the user who contributed the review</p>
<p>[readable, assign] NSString * <i>description</i> The review content for the NMAPlace</p>
<p>[readable, assign] NSString * <i>languageCode</i> The BCP-47 language code in which the review is available</p>
<p>[readable, assign] double <i>rating</i> The rating that the review contributor gave for the place</p>
<p>[readable, assign] NSString * <i>title</i> The title of the review that the user contributed</p>

Class Details

Represents review content that provides access to user reviews and ratings about a [NMAPlace](#).

See also:

[NMAMediaEditorial](#)

[NMAMedialmage](#)

NMAMediaRatings

Public Property Details

[readable, assign] NSDate * **contributionDate**

The NSDate specifying the date when the user contributed the review.

[readable, assign] NMALink * **contributor**

Details of the user who contributed the review.

■ **Note:**

The contributor may be unknown. Attempts to read this property could return nil.

[readable, assign] NSString * **description**

The review content for the [NMAPlace](#).

■ **Note:**

Depending on the NMASupplier, the full content might not be displayed and the full review might be available only within the resource to which the optional *via* attribute links.

[readable, assign] NSString * **languageCode**

The BCP-47 language code in which the review is available.

■ **Note:**

Attempts to read this property could return empty string as the language code for a review is optional. If nil the *description* language is unknown but is likely to be English.

[readable, assign] double **rating**

The rating that the review contributor gave for the place.

■ **Note:**

The rating may come from various systems with different rating schemes. The value of this attribute may not reflect the rating scheme of the supplier's service, but is adjusted to the value range of the Places API [1..5].

[readable, assign] NSString * **title**

The title of the review that the user contributed.

■ **Note:**

A title for a review is optional. Attempts to read this property could return empty string.

NMAPlace

Class Summary

class **NMAPlace**

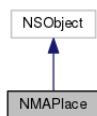
Derived from: `NSObject`

Represents a set of data about a physical place.

Include: `NMAKit.framework/headers/NMAPlace.h`

Inheritance Diagrams

Figure 84: Public inheritance diagram for NMAPlace



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 102: Public Properties

Public Properties
<code>[readable, assign] NSDictionary< NSString *, NSString * > * alternativeNames</code>
<code>[readable, assign] NSString * attributionText</code> The string containing the attribution string for this NMAPlace
<code>[readable, assign] NSArray< NMACategory * > * categories</code> The NSArray of NMACategory objects linking to categories that have been assigned to the NMAPlace
<code>[readable, assign] NSArray< NMAContactDetail * > * contacts</code> Array of NMAContactDetail objects that provide information on how to contact the place
<code>[readable, assign] NMAMediaCollectionPage * editorials</code> NMAMediaCollectionPage of NMAMediaEditorial objects
<code>[readable, assign] NSArray< NMAExtendedAttribute * > * extendedAttributes</code> NSArray of NMAExtendedAttribute objects representing additional information about an NMAPlace
<code>[readable, assign] NSString * iconUrl</code> An URL of icon that represents the NMAPlace
<code>[readable, assign] NMAMediaCollectionPage * images</code> NMAMediaCollectionPage of NMAMediaImage objects

Public Properties

[readable, assign] `NMAPlaceLocation` * `location`

The NMAPlaceLocation of the NMAPlace

[readable, assign] `NSString` * `name`

[readable, assign] `NMAMediaCollectionPage` * `ratings`

NMAMediaCollectionPage of NMAMediaRating objects

[readable, assign] `NSDictionary< NSString *, __kindof NMALink * >` * `related`

The NSDictionary of related places (where available) that might also interest an application user

[readable, assign] `NMALink` * `reportingLink`

Gets the link for getting options for reporting a place because, for example, it contains inappropriate content or the place does not exist

[readable, assign] `NMAMediaCollectionPage` * `reviews`

NMAMediaCollectionPage of NMAMediaReview objects

[readable, assign] `NMALink` * `supplier`

Link to the supplier of the `NMAPlace`.

[readable, assign] `NSString` * `uniqueId`

[readable, assign] `NMARatings` * `userRatings`

The user ratings for the NMAPlace

[readable, assign] `NSString` * `viewUrl`

URL to an external web page that represents the NMAPlace

Instance Method Summary

Table 103: Instance Methods

Instance Method Summary

`- (nullable NSArray< NSString * > *)
referenceIdentifiersForSource:(forSource:referenceIdentifiers)`

Get reference identifiers for a specific source

Class Details

Represents a set of data about a physical place.

A NMAPlace acts as a container for various information about a place, which itself is a point of interest such as a popular restaurant, a park, or someone's home.

Note:

A `NMAPlace` can contain attributes, collections of media about the place, and key-value pairs of related places.

Public Property Details

[readable, assign] `NSDictionary< NSString *, NSString * > * alternativeNames`

Gets the NSDictionary for the alternative names of `NMAPlace` in a list of mapping languages and alternative names

■ **Note:**

An alternative name for a place is optional. Attempts to read this property could return nil.

[readable, assign] `NSString * attributionText`

The string containing the attribution string for this `NMAPlace`.

example : Provided by Qype user Max Mustermann

■ **Note:**

Attribution text for a place is optional. Attempts to read this property could return nil.

[readable, assign] `NSArray< NMACategory * > * categories`

The NSArray of `NMACategory` objects linking to categories that have been assigned to the `NMAPlace`.

■ **Note:**

A `NMACategory` object extends the standard link object with an optional icon link pointing to an icon appropriate for that category. Categories for a place is optional. Attempts to read this property could return nil.

[readable, assign] `NSArray< NMAContactDetail * > * contacts`

Array of `NMAContactDetail` objects that provide information on how to contact the place.

■ **Note:**

Contact information for a place is optional. Attempts to read this property could return nil.

[readable, assign] `NMAMediaCollectionPage * editorials`

`NMAMediaCollectionPage` of `NMAMediaEditorial` objects. See also `NMAMediaCollectionPage::mediaContents` related to this `NMAPlace`.

■ **Note:**

Editorial content for a place is optional. Attempts to read this property could return nil.

[readable, assign] `NSArray< NMAExtendedAttribute * > * extendedAttributes`

NSArray of `NMAExtendedAttribute` objects representing additional information about an `NMAPlace`.

An extensible collection of attributes can include:

- payment - [NMExtendedAttribute](#)
- openingHours - [NMExtendedAttribute](#)
- annualClosings - [NMExtendedAttribute](#)
- price - [NMExtendedAttribute](#)
- nearestLandmark - [NMExtendedAttribute](#)
- languagesSpoken - [NMExtendedAttribute](#)
- availableParking - [NMExtendedAttribute](#)
- smoking - [NMExtendedAttribute](#)
- disabledAccess - [NMExtendedAttribute](#)

■ **Note:**

Extended attribute for a place is optional. Attempts to read this property could return nil.

[readable, assign] `NSString * iconUrl`

An URL of icon that represents the [NMAPlace](#).

■ **Note:**

Application can download the icon online with this URL, but it needs to use its own icon for offline.

[readable, assign] `NMAMediaCollectionPage * images`

[NMAMediaCollectionPage](#) of [NMAMediaImage](#) objects. See also [NMAMediaCollectionPage::mediaContents](#) related to this [NMAPlace](#).

■ **Note:**

Image content for a place is optional. Attempts to read this property could return nil.

[readable, assign] `NMAPlaceLocation * location`

The [NMAPlaceLocation](#) of the [NMAPlace](#).

[readable, assign] `NSString * name`

The name of the [NMAPlace](#) and localized to language preference defined in [NMRequest](#) (if possible)

■ **Note:**

Attempts to read this property could return an empty string.

[readable, assign] `NMAMediaCollectionPage * ratings`

[NMAMediaCollectionPage](#) of [NMAMediaRating](#) objects. See also [NMAMediaCollectionPage::mediaContents](#) related to this [NMAPlace](#).

■ **Note:**

Rating content for a place is optional. Attempts to read this property could return nil.

[readable, assign] NSDictionary< NSString *, __kindof NMALink * > * related

The NSDictionary of related places (where available) that might also interest an application user.

The NSDictionary is keyed by a localized title to language preference defined in [NMAResponse](#) (if possible) of the related places (for example, "Nearby", "public-transport"), while the associated value is a link presented in class derived from [NMALink](#)(i.e.NMADiscoveryLink, [NMAPlaceLink](#),etc ...).

■ **Note:**

Attempts to read this property could return nil.

[readable, assign] NMALink * reportingLink

Gets the link for getting options for reporting a place because, for example, it contains inappropriate content or the place does not exist.

■ **Note:**

A reporting link for a place is optional. Attempts to read this property could return nil.

[readable, assign] NMAMediaCollectionPage * reviews

[NMAMediaCollectionPage](#) of [NMAMediaReview](#) objects. See also [NMAMediaCollectionPage::mediaContents](#) related to this [NMAPlace](#).

■ **Note:**

Review content for a place is optional. Attempts to read this property could return nil.

[readable, assign] NMALink * supplier

Link to the supplier of the [NMAPlace](#).

The supplier link extends the standard link object with an optional icon link that points to the supplier's brand icon.

Example:

```
"name": "Qype",
"url": "http://...",
"icon": "http://..."
```

■ **Note:**

A supplier link for a place is optional. Attempts to read this property could return nil.

[readable, assign] NSString * uniqueId

Unique consistent identifier for the place represented by this [NMAPlace](#) instance

■ **Note:**

The identifier value is consistent across application sessions. Applications that want to keep a reference to a specific place can store this identifier for subsequent linking to additional resources.

[readable, assign] NMARatings * userRatings

The user ratings for the [NMAPlace](#).

■ **Note:**

User ratings for a place are optional. Attempts to read this property could return nil.

[readable, assign] NSString * viewUrl

URL to an external web page that represents the [NMAPlace](#).

■ **Note:**

Applications are encouraged to provide web links to places they display information about. Attempts to read this property could return an empty string.

Instance Method Details

- (nullable NSArray< NSString * > *) referenceIdentifiersForSource:(forSource:referenceIdentifiers)

Get reference identifiers for a specific source. For example, a place can have a reference to an extruded building object in the map. The reference identifiers can be retrieved by calling this method with source `NMAPlacesSourceBuilding`.

Index 0 is the primary reference identifier for the place.

■ **Note:**

A reference identifier will not be returned if it has not been queried with the [NMAResponse::addSource](#) method.

Parameters:

- **source**

NSString specifying the source

Returns:

An NSArray of reference identifiers. Return nil if reference identifier does not exist.

NMAPlaceLink

Class Summary

class **NMAPlaceLink**

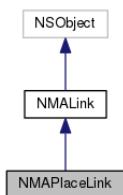
Derived from: [NMALink](#)

Represents metadata about a place, this object is usually returned as a response to places search.

Include: NMAKit.framework/headers/NMAPlaceLink.h

Inheritance Diagrams

Figure 85: Public inheritance diagram for NMAPlaceLink



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 104: Public Properties

Public Properties
<p>[readable, assign] double averageRating The average rating for the NMAPlace if available, otherwise -1</p>
<p>[readable, assign] NMAGeoBoundingBox * boundingBox The enclosing NMAGeoBoundingBox describing a range of coordinates that correspond to the NMAPlace</p>
<p>[readable, assign] NMACategory * category The NMACategory to the primary category for the NMAPlace</p>
<p>[readable, assign] NSInteger distance The distance to the NMAPlace, in meters</p>
<p>[readable, assign] BOOL isSponsored Indicates whether this search result is sponsored</p>
<p>[readable, assign] NMAGeoCoordinates * position The NMAGeoCoordinates representing the geographical position of the NMAPlace</p>
<p>[readable, assign] NSString * vicinityDescription The NSString description of the vicinity surrounding the NMAPlaceLink</p>

Instance Method Summary

Table 105: Instance Methods

Instance Method Summary
<pre>- (nullable <i>NMAPlaceRequest</i> *) <i>detailsRequest</i></pre> <p>Gets the request to retrieve the <i>NMAPlace</i> details</p>
<pre>- (nullable NSArray< NSString * &gt; *) <i>referenceIdentifiersForSource:(forSource:referenceIdentifiers)</i></pre> <p>Get reference identifiers for a specific source. For example, a place can have a reference to an extruded building object in the map. The reference identifiers can be retrieved by calling this method with source <i>NMAPlacesSourceBuilding</i>.</p>

Class Details

Represents metadata about a place, this object is usually returned as a response to places search.

■ Note:

To retrieve the actual *NMAPlace* object use the *detailsRequest* property.

Public Property Details

[readable, assign] double **averageRating**

The average rating for the *NMAPlace* if available, otherwise -1.0 to indicate the value is not available.

[readable, assign] *NMAGeoBoundingBox* * **boundingBox**

The enclosing *NMAGeoBoundingBox* describing a range of coordinates that correspond to the *NMAPlace*. Typically, bounding boxes are associated with places such as cities and countries.

This bounding box does not necessarily have the location from the position property as its center. For example, if the search was performed with a street name, this bounding box may be one that contains the entire street, while position can be any point along the street.

■ Note:

Bounding box information for a place is optional. Attempts to read this property could return nil.

[readable, assign] *NMACategory* * **category**

The *NMACategory* to the primary category for the *NMAPlace*.

[readable, assign] NSInteger **distance**

The distance to the *NMAPlace*, in meters.

■ Note:

Distance information for a place is optional. Attempts to read this property could return be zero.

[readable, assign] BOOL **isSponsored**

Indicates whether this search result is sponsored.

Applications should provide some visual differentiation between sponsored search results and regular search results.

■ Note:

Sponsorship is an optional attribute for a place.

[readable, assign] *NMAGeoCoordinates* * **position**

The NMAGeoCoordinates representing the geographical position of the *NMAPlace*.

[readable, assign] *NSString* * **vicinityDescription**

The NSString description of the vicinity surrounding the *NMAPlaceLink*.

Typically, this property is derived from the address, but could also contain any other description that helps an application user understand where the place is located.

Instance Method Details

-*(nullable NMAPlaceRequest *) detailsRequest*

Gets the request to retrieve the *NMAPlace* details. Attempts to read this property could return nil.

-*(nullable NSArray< NSString * > *) referenceIdentifiersForSource:(forSource:referenceIdentifiers)*

Get reference identifiers for a specific source. For example, a place can have a reference to an extruded building object in the map. The reference identifiers can be retrieved by calling this method with source *NMAPlacesSourceBuilding*.

Index 0 is the primary reference identifier for the place.

■ Note:

A reference identifier will not be returned if it has not been queried with the *NMARequest::addSource:* function.

Parameters:

- **source**

NSString specifying the source

Returns:

An NSArray of reference identifiers. Return nil if reference identifier does not exist.

NMAPlaceLocation

Class Summary

class **NMAPlaceLocation**

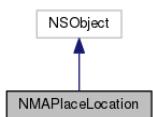
Derived from: `NSObject`

Represents a location on a map in terms of a street address and a geographical position.

Include: `NMAKit.framework/headers/NMAPlaceLocation.h`

Inheritance Diagrams

Figure 86: Public inheritance diagram for `NMAPlaceLocation`



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 106: Public Properties

Public Properties
<code>[readable, writable, strong] NMAAddress * <i>address</i></code> The <code>NMAAddress</code> for the <code>NMAPlaceLocation</code>
<code>[readable, writable, strong] NMAGeoBoundingBox * <i>boundingBox</i></code> The bounding box of the place location if available
<code>[readable, writable, strong] NSString * <i>label</i></code> The descriptive label for the <code>NMAPlaceLocation</code>
<code>[readable, writable, strong] NSString * <i>locationId</i></code> The unique ID of the place location
<code>[readable, writable, strong] NMAGeoCoordinates * <i>position</i></code> <code>NMAGeoCoordinates</code> for the <code>NMAPlaceLocation</code> if available

Instance Method Summary

Table 107: Instance Methods

Instance Method Summary
<pre>-(nullable NSArray< NSString * &gt; *) referenceIdentifiersForSource:(forSource:referenceIdentifiers)</pre> <p>Get reference identifiers for a specific source</p>

Class Details

Represents a location on a map in terms of a street address and a geographical position.

Public Property Details

[readable, writable, strong] `NMAAddress * address`

The `NMAAddress` for the `NMAPlaceLocation`.

[readable, writable, strong] `NMAGeoBoundingBox * boundingBox`

The bounding box of the place location if available.

[readable, writable, strong] `NSString * label`

The descriptive label for the `NMAPlaceLocation`.

[readable, writable, strong] `NSString * locationId`

The unique ID of the place location.

[readable, writable, strong] `NMAGeoCoordinates * position`

`NMAGeoCoordinates` for the `NMAPlaceLocation` if available.

Instance Method Details

`-(nullable NSArray< NSString * > *) referenceIdentifiersForSource:(forSource:referenceIdentifiers)`

Get reference identifiers for a specific source. For example, a place can have a reference to an extruded building object in the map. The reference identifiers can be retrieved by calling this method with source `NMAPlacesSourceBuilding`.

Index 0 is the primary reference identifier for the place.

■ Note:

A reference identifier will not be returned if it has not been queried with the (see [NMAResponse::addSource:](#)).

Parameters:

- **source**

NSString specifying the source.

Returns:

An NSArray of reference identifiers. Return nil if reference identifier does not exist.

NMAPlaceRequest

Class Summary

class **NMAPlaceRequest**

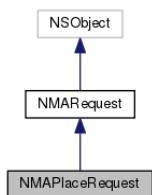
Derived from: [NMAResponse](#)

Represents a request to retrieve an NMAPlace object.

Include: NMKit.framework/headers/NMAPlaceRequest.h

Inheritance Diagrams

Figure 87: Public inheritance diagram for NMAPlaceRequest



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 108: Public Properties

Public Properties
<p>[readable, writable, assign] NMATextFormat textFormat The text format of the request results</p>

Instance Method Summary

Table 109: Instance Methods

Instance Method Summary
<pre>- (void) addContent:(nonnull NSString *) content</pre> <p>To request additional content</p>
<pre>- (void) addImageSizeWithWidth:(NSInteger) width height:(width:height:addImageSize)</pre> <p>To request an image media with specific width-height dimensions</p>

Class Details

Represents a request to retrieve an [NMAPlace](#) object.

Public Property Details

[readable, writable, assign] [NMATextFormat](#) **textFormat**

The text format of the request results.

■ **Note:**

The default value is NMATextFormatHTML.

Instance Method Details

- (void) addContent:(nonnull NSString *) content

To request additional content. Result will be provided in editorials property of [NMAPlace](#) object

Parameters:

- **content**

The content name. Currently only NMAPlacesContentWikipedia is supported. Unsupported values are ignored.

- (void) addImageSizeWithWidth:(NSInteger) width height:(width:height:addImageSize)

To request an image media with specific width-height dimensions. At least one of the sizes (width or height) needs to be valid (greater than 0).

Note taht the provided width-height are used as upper boundaries for the dimensions of the returned images.

For example:

```
image_dimensions=w32-h32,w64-h64  
image_dimensions=w32-h32,w300  
image_dimensions=h200
```

Parameters:

- **width**
The image width.
- **height**
The image height.

NMAPlaces

Class Summary

class **NMAPlaces**

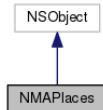
Derived from: NSObject

Represents a factory for creating places requests.

Include: NMAKit.framework/headers/NMAPlaces.h

Inheritance Diagrams

Figure 88: Public inheritance diagram for NMAPlaces



[For complete information, see the section [Class Details](#)]

Instance Method Summary

Table 110: Instance Methods

Instance Method Summary
<pre>-(nullable NMAAutoSuggestionRequest *) createAutoSuggestionRequestWithLocation:(NMAGeoCoordinates *) location partialTerm:(location:partialTerm:makeAutoSuggestionRequest)</pre> <p>Creates a NMAPlaces request to return lists of suggested search terms, instant results and refined search links related to a specified location context and partial search term</p>
<pre>-(nullable NMADiscoveryRequest *) createExploreRequestWithLocation:(NMAGeoCoordinates *) location searchArea:(NMAGeoBoundingBox *_Nullable) geoArea filters:(location:geoArea:filters:makeExploreRequest)</pre> <p>Creates a NMAPlaces explore request using a specified location context and category filter</p>
<pre>-(nullable NMADiscoveryRequest *) createHereRequestWithLocation:(NMAGeoCoordinates *) location filters:(location:filters:makeHereRequest)</pre> <p>Creates a NMAPlaces "What's here?" request using a specified location context and category filter</p>

Instance Method Summary

```
- (nullable NMPlaceRequest *) createLookupRequestWithReferenceIdentifier:(NSString *)  
    referenceIdentifier inSource:(referenceIdentifier:source:makeLookupRequest)
```

Creates a NMPlaceRequest based on a reference identifier and source

```
- (nullable NMADiscoveryRequest *) createSearchRequestWithLocation:( NMAGeoCoordinates *) location  
    query:(location:query:makeSearchRequest)
```

Creates a NMAPlaces search request using a specified location and query

```
- (nullable NMASuggestionRequest *) createSuggestionRequestWithLocation:( NMAGeoCoordinates *) location  
    partialTerm:(location:partialTerm:makeSuggestionRequest)
```

Creates a NMAPlaces request to return a list of suggested search terms that are related to a specified location context and partial search term

```
- (void) refreshTopLevelCategoriesWithCompletion:(completionBlock:refreshTopLevelCategories)
```

Refresh the localized top level categories for Places Search

```
- (nullable NSArray< NMCategory * > *) topLevelCategories
```

Return the localized top level categories for Places Search

Class Method Summary

Table 111: Class Methods

Class Methods
+ (<i>NMAPlaces</i> *) <i>sharedPlaces</i> Returns the NMAPlaces singleton instance

Class Details

Represents a factory for creating places requests.

The NMAPlaces service supports a variety of use cases that center around two major features:

- Place discovery - helps users find places that are relevant to them.
- Place information retrieval - provides detailed information about places in which users are interested.

For place discovery, the NMAPlaces API supports two use cases from which to choose according to the requirements for the application:

- Search - finds places that match user-provided search terms.
- Explore - guides users to relevant places nearby.

The NMAPlaces API defines resources that represent entry points for supported use cases, employing links to describe the application flow within a particular use case.

For example, the discovery request is an entry point to the NMAPlaces API resource. When an application user submits a discovery request to the search resource, the response contains, among other items, a list of links to places or to place resources. Each referenced place might have rich content associated with it, including ratings, images, reviews, editorials, and owner content. Each place also contains references to related places, which allow users to discover places relevant to or otherwise related to their original search.

■ Note:

NMAPlaces requires valid authentication credentials to be set via *NMAApplicationContext*. If valid credentials are not present the API may not function correctly or may stop functioning correctly in the future when server side configurations change.

Instance Method Details

```
- (nullable NMAAutoSuggestionRequest *)  
createAutoSuggestionRequestWithLocation:(NMAGeoCoordinates *) location  
partialTerm:(location:partialTerm:makeAutoSuggestionRequest)
```

Creates a NMAPlaces request to return lists of suggested search terms, instant results and refined search links related to a specified location context and partial search term.

Note that this Autosuggest request supports online only. Please use *createSuggestionRequestWithLocation:partialTerm* for offline.

In addition to the requested location this search also considers the current device position as returned by *NMAPositioningManager*.

A search location context must be provided by setting a search location. Failing to set a location context will result in an error *NMARequestErrorInvalidParameter* when executing the request.

Parameters:

- **location**
A NMAGeoCoordinates representing the query location context
- **partialTerm**
An NSString specifying the partial search term

Returns:

The NMAAutoSuggestionRequest if the location is valid, nil otherwise

```
- (nullable NMADiscoveryRequest *) createExploreRequestWithLocation:(NMAGeoCoordinates *) location searchArea:(NMAGeoBoundingBox *_Nullable) geoArea filters:(location:geoArea:filters:makeExploreRequest)
```

Creates a NMAPlaces explore request using a specified location context and category filter. If a category filter is provided, the created request will limit results to include only items with categories specified in the filter.

The results of the explore request is a list of nearby relevant places for a given positon. It answers the question "What interesting places are near a location?" The results returned are confined to those located in the current search area and are ordered by popularity.

■ Note:

In addition to the requested location this search also considers the current device position as returned by *NMAPositioningManager*.

Parameters:

- **location**

A NMAGeoCoordinates representing the location context used to search for nearby places

- **geoArea**

A NMAGeoBoundingBox representing the query's search area (this is an optional parameter and should not be set unless such functionality is explicitly intended in the application)

- **filters**

A NMACategoryFilter representing the category filter

Returns:

The NMADiscoveryRequest if parameters are valid

```
- (nullable NMADiscoveryRequest *) createHereRequestWithLocation:(  
NMAGeoCoordinates *) location filters:(location:filters:makeHereRequest)
```

Creates a NMAPlaces "What's here?" request using a specified location context and category filter. If a category filter is provided, the created request will limit results to include only items with categories specified in the filter.

The results of the here request is a list of places with addresses that lie within the vicinity of the search location. The feature is typically used by applications that include "check-in" or "click on map to get more information" options.

 **Note:**

In addition to the requested location this search also considers the current device position as returned by [NMAPositioningManager](#).

Parameters:

- **location**

A NMAGeoCoordinates representing the location context used to search for nearby places

- **filters**

A NMACategoryFilter representing the category filter

Returns:

The NMADiscoveryRequest if both parameters are valid

```
- (nullable NMAPlaceRequest *) createLookupRequestWithReferenceIdentifier:  
(NSString *) referenceIdentifier inSource:  
(referenceIdentifier:source:makeLookupRequest)
```

Creates a NMAPlaceRequest based on a reference identifier and source.

The supported sources are "pvid", "sharing", "venues", "venues.content", "venues.venue", "venues.destination", "building". See also Resource look-up .

"sharing" example:

The 'referenceIdentifier' argument to use is found in [NMAPlace:viewUrl](#). The 'referenceIdentifier' is the last path segment of the View URI.

Using the following example View URI:

```
http://here.com/p/s-
aWQ9Mjc2dTMzzGMtMzk0NGE0NzQ4YTYyNDk5Y2I3OTVjZjhizWUzNDljOWM7bj1BbnRpcG9kZXM7Yz1jb2ZmZWUtdGVhO
3BoPSUyQjQ5MTc2MzgzNDAxMTg7bGF0PTUyLjUzMDkz02xvbj0xMy40MDk5Mg
```

The sharing reference identifier is:

```
s-aWQ9Mjc2dTMzzGMtMzk0NGE0NzQ4YTYyNDk5Y2I3OTVjZjhizWUzNDljOWM7bj1BbnRpcG9kZXM7Yz1jb2ZmZWUtdGVhO
3BoPSUyQjQ5MTc2MzgzNDAxMTg7bGF0PTUyLjUzMDkz02xvbj0xMy40MDk5Mg
```

The reference source is [NMAPlacesSourceSharing](#).

For example:

```
NMAPlaceRequest* request =
[[NMAPlaces sharedPlaces] createLookupRequestWithReferenceIdentifier: @"s-
aWQ9Mjc2dTMzzGMtMzk0NGE0NzQ4YTYyNDk5Y2I3OTVjZjhizWUzNDljOWM7bj1BbnRpcG9kZXM7Yz1jb2ZmZWUtdGVhO
3BoPSUyQjQ5MTc2MzgzNDAxMTg7bGF0PTUyLjUzMDkz02xvbj0xMy40MDk5Mg"
inSource:NMAPlacesSourceSharing];
```

"venues.venue" example:

The venue reference identifier for Alexa Shopping Center is DM_7171. The reference source is NMAPlacesSourceVenuesVenue.

For example:

```
NMPlaceRequest* request = [[NMAPlaces sharedPlaces]
createLookupRequestWithReferenceIdentifier:@"DM_7171" inSource:NMAPlacesSourceVenuesVenue];
```

Parameters:

- **referenceIdentifier**
The reference identifier of the requested [NMPlace](#) in a specified source (ie Core POI, Venue).
- **source**
The NSString of the source in lower case (for example pvid).

Returns:

The NMPlaceRequest if the reference identifier is valid, nil otherwise

```
- (nullable NMADiscoveryRequest *) createSearchRequestWithLocation:(  
NMAGeoCoordinates *) location query:(location:query:makeSearchRequest)
```

Creates a NMAPlaces search request using a specified location and query.

The results of the search request are sets of places that match a user's search term in a specific location. Additional user's current position will also be sent to ensure you get the best results possible. The user's current position is only available if [[NMAPositioningManager active](#)] is true.

The viewport is a bounding box used to perform the search. To ensure you get the best results possible, you should match the viewport to your map if there is a map visible to the user. The viewport is optional (see also [NMAResponse::viewport](#)).

Parameters:

- **location**

A NMAGeoCoordinates representing the query's location context used to search for results that are appropriate to the query parameter

- **query**

NSString specifying the kind of places to locate

Returns:

The NMADiscoveryRequest if both parameters are valid, nil otherwise.

```
- (nullable NMASuggestionRequest *) createSuggestionRequestWithLocation:(NMAGeoCoordinates *) location partialTerm:(location:partialTerm:makeSuggestionRequest)
```

Creates a NMAPlaces request to return a list of suggested search terms that are related to a specified location context and partial search term.

■ **Note:**

In addition to the requested location this search also considers the current device position as returned by [NMAPositioningManager](#).

Parameters:

- **location**

A NMAGeoCoordinates representing the query location context

- **partialTerm**

A NSString specifying the partial search term

Returns:

The NMASuggestionRequest if the location is valid, nil otherwise

```
- (void) refreshTopLevelCategoriesWithCompletion:(completionBlock:refreshTopLevelCategories)
```

Refresh the localized top level categories for Places Search.

```
- (nullable NSArray< NMACategory * > *) topLevelCategories
```

Return the localized top level categories for Places Search.

User can manually refresh the categories with `refreshTopLevelCategoriesWithCompletion:`

■ **Note:**

The list of categories is cached. The update request to Places backend is made periodically and when device locale is changed. If there is no cache or cache is being updated, a nil is returned. User should try again later.

Returns:

An NSArray of localized [NMACategory](#) if available, nil otherwise

Class Method Details

`+ (NMAPlaces *) sharedPlaces`

Returns the NMAPlaces singleton instance.

 **Note:**

Use this method to obtain a NMAPlaces instance. Do not call init directly.

Returns:

shared NMAPlaces instance

NMARatings

Class Summary

class NMARatings

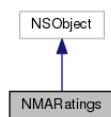
Derived from: NSObject

Represents a summary of the user-supplied ratings for a NMAPlace.

Include: `NMAKit.framework/headers/NMARatings.h`

Inheritance Diagrams

Figure 89: Public inheritance diagram for NMARatings



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 112: Public Properties

Public Properties
<code>[readable, assign] double average</code> The average rating for the NMAPlace
<code>[readable, assign]NSUInteger count</code> The number of individual votes that users provide for rating the NMAPlace

Class Details

Represents a summary of the user-supplied ratings for a *NMAPlace*.

Public Property Details

[readable, assign] double **average**

The average rating for the *NMAPlace*.

[readable, assign] NSUInteger **count**

The number of individual votes that users provide for rating the *NMAPlace*.

NMARequest

Class Summary

class **NMARequest**

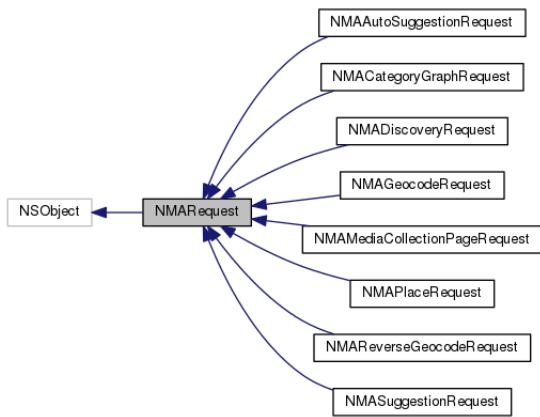
Derived from: `NSObject`

Represents a base interface for a search request.

Include: `NMAKit.framework/headers/NMARequest.h`

Inheritance Diagrams

Figure 90: Public inheritance diagram for **NMARequest**



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 113: Public Properties

Public Properties
[readable, writable, assign] <code>NSUInteger collectionSize</code> The maximum number of items in the response
[readable, writable, strong] <code>NSString * languagePreference</code> NSString representing the preferred language for request responses
[readable, assign] <code>NSArray< NSString * > * sources</code> The NSArray of sources have been added for a NMPlace
[readable, writable, strong] <code>NSString * userTag</code> User-defined Tag as convenience method to identify NMAResult
[readable, writable, assign] <code>NMAGeoBoundingBox * viewport</code> The bounding box in which to perform the search

Instance Method Summary

Table 114: Instance Methods

Instance Method Summary
<code>- (void) addSource:(source:add)</code> This function adds the a reference identifier to be returned in the request results
<code>- (BOOL) cancel</code> Cancels any pending query request
<code>- (nullable NSError *) startWithBlock:(block:start)</code> Invokes an asynchronous query request with a specified block
<code>- (nullable NSError *) startWithListener:(listener:start)</code> Invokes an asynchronous query request with a specified listener

Class Details

Represents a base interface for a search request.

Public Property Details

[readable, writable, assign] `NSUInteger collectionSize`

The maximum number of items in the response.

■ **Note:**

The default value is 10. The maximum value is 100, if over, it's capped to 100.



[readable, writable, strong] `NSString * languagePreference`

`NSString` representing the preferred language for request responses.

Default Value: `[[NSLocale preferredLanguages] objectAtIndex:0]`.

■ Note:

`languagePreference` should be a valid code according to the IETF BCP-47 standard (see <http://tools.ietf.org/html/bcp47>).

[readable, assign] `NSArray< NSString * > * sources`

The `NSArray` of sources have been added for a [`NMPlace`](#).

See also:

[`NMARequest::addSource:`](#)

[readable, writable, strong] `NSString * userTag`

User-defined Tag as convenience method to identify `NMARequest`.

■ Note:

The default value for `userTag` is empty string and user should assign any tag value with their own convention.

[readable, writable, assign] `NMAGeoBoundingBox * viewport`

The bounding box in which to perform the search.

The `viewport` can act as an implicit location context in the absence of an explicit location context. To ensure you get the best results possible, you should match the `viewport` to your map if there is a map visible to the user.

Instance Method Details

`- (void) addSource:(source:add)`

This function adds the a reference identifier to be returned in the request results. See also [`NMPlace::referenceIdentifiersForSource:`](#).

Examples of reference identifiers are: "NMAPlacesSourcePVID" for HERE's core content product which provides map and POI data, "NMAPlacesSourceVenuesAll" for HERE's venue maps product whcih provides indoor maps for thousands of venues, "NMAPlacesSourceBuilding" for extruded building.

Parameters:

- **source**

A string containing the source identifier to add.

`-(BOOL) cancel`

Cancels any pending query request.

`-(nullable NSError *) startWithBlock:(block:start)`

Invokes an asynchronous query request with a specified block.

Parameters:

- **aBlock**

A block to be executed upon completion of the request

Returns:

The appropriate NSError code

See also:

NMARequestCompletionBlock

startWithListener:

`-(nullable NSError *) startWithListener:(listener:start)`

Invokes an asynchronous query request with a specified listener.

Parameters:

- **aSearchEventListener**

A listener to listen for search results

Returns:

The appropriate NSError code

See also:

startWithBlock:

NMAResponseGeocodeRequest

Class Summary

class **NMAResponseGeocodeRequest**

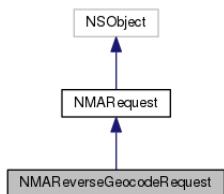
Derived from: *NMARequest*

Represents a request to retrieve NMAPlaceLocation data by way of NMAGeocoder search services.

Include: `NMAKit.framework/headers/NMAResponseGeocodeRequest.h`

Inheritance Diagrams

Figure 91: Public inheritance diagram for NMAReverseGeocodeRequest



[For complete information, see the section [Class Details](#)]

Class Details

Represents a request to retrieve [NMAPlaceLocation](#) data by way of [NMAGeocoder](#) search services.

■ Note:

A NMAReverseGeocodeRequest is a kind of [NMARequest](#), and its response is an instance of [NMAReverseGeocodeResult](#).

NMAReverseGeocodeResult

Class Summary

class NMAReverseGeocodeResult

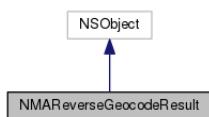
Derived from: NSObject

Represents the result of a reverse geocoding request.

Include: NMKit.framework/headers/NMAReverseGeocodeResult.h

Inheritance Diagrams

Figure 92: Public inheritance diagram for NMAReverseGeocodeResult



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 115: Public Properties

Public Properties
<p>[readable, assign] float <i>distance</i> The distance between the result and the search input coordinates</p>

Public Properties

[readable, assign] `NMAPlaceLocation * location`

The location data of the reverse geocode result

Class Details

Represents the result of a reverse geocoding request.

The data of a reverse geocoding result is represented by an instance of `NMAPlaceLocation`, accessed via the `location` property. The distance between the result and the original search location is give by the `distance` property.

Public Property Details

[readable, assign] `float distance`

The distance between the result and the search input coordinates.

[readable, assign] `NMAPlaceLocation * location`

The location data of the reverse geocode result.

NMASuggestionRequest

Class Summary

class `NMASuggestionRequest`

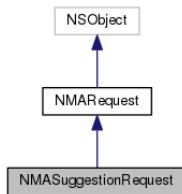
Derived from: `NMARequest`

Represents a request to retrieve a list of search terms.

Include: `NMAKit.framework/headers/NMASuggestionRequest.h`

Inheritance Diagrams

Figure 93: Public inheritance diagram for `NMASuggestionRequest`



[For complete information, see the section [Class Details](#)]

Public Property Summary

Table 116: Public Properties

Public Properties
[readable, writable, assign] <code>NMATextFormat</code> <code>textFormat</code> The text format of the request results

Class Details

Represents a request to retrieve a list of search terms.

Public Property Details

[readable, writable, assign] `NMATextFormat` `textFormat`

The text format of the request results.

■ **Note:**

The default value is NMATextFormatHTML.