User Location Service in iOS

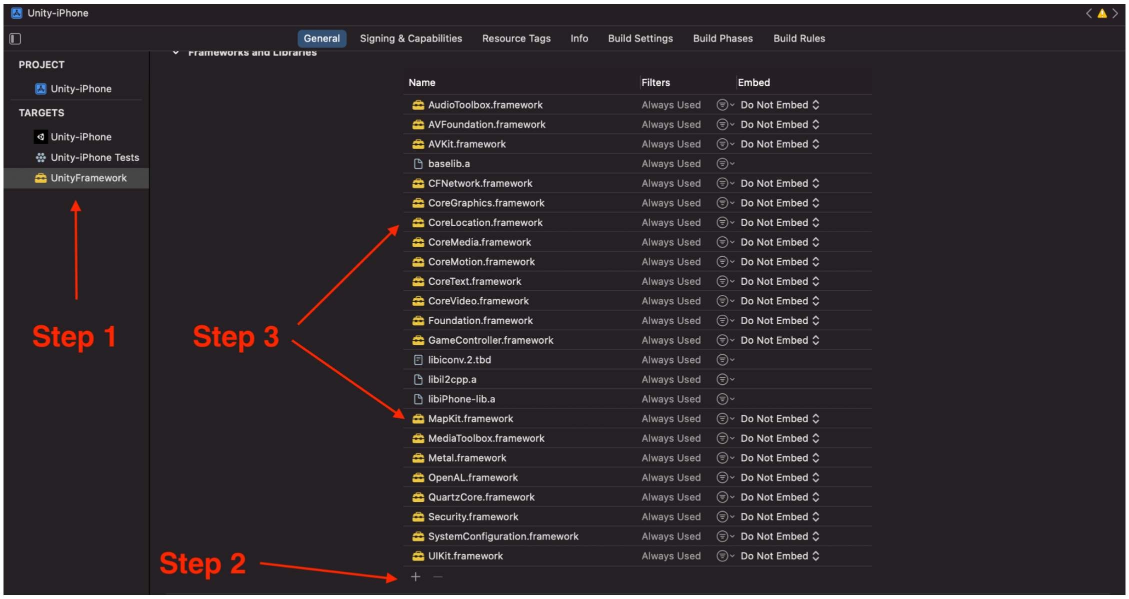
This plugin will allow developers to add GPS capability in iPhone apps and retrieve the user’s current location. This plugin will also cover tip to translate the GPS coordinate into street address. This plugin use **Core Location framework** and provides developers the necessary C# interfaces for obtaining information about the user’s location. With the GPS coordinate obtained, developers can make use of the API to decode the actual street. Also it allows developers to open maps inside the unity or can switch to apple maps. Read this document carefully. Take a look at the DemoScene in Scenes folder.

SetUp

Import the asset package in the assets folder. Ensure that following files exists in Plugins/iOS/ folder:

1. LocationManager.h
2. LocationManager.m

Open the project in Xcode. Add **CoreLocation.framework and MapKit.framework** in the project. To add the same click on '+' button at bottom under the **Linked Frameworks and Libraries** section in the **General** tab of the **Unity-Framework** Target.Please see screenshot below:



Name	Filters	Embed
AudioToolbox.framework	Always Used	Do Not Embed
AVFoundation.framework	Always Used	Do Not Embed
AVKit.framework	Always Used	Do Not Embed
baselib.a	Always Used	
CFNetwork.framework	Always Used	Do Not Embed
CoreGraphics.framework	Always Used	Do Not Embed
CoreLocation.framework	Always Used	Do Not Embed
CoreMedia.framework	Always Used	Do Not Embed
CoreMotion.framework	Always Used	Do Not Embed
CoreText.framework	Always Used	Do Not Embed
CoreVideo.framework	Always Used	Do Not Embed
Foundation.framework	Always Used	Do Not Embed
GameController.framework	Always Used	Do Not Embed
libiconv.2.tbd	Always Used	
libil2cpp.a	Always Used	
libiPhone-lib.a	Always Used	
MapKit.framework	Always Used	Do Not Embed
MediaToolbox.framework	Always Used	Do Not Embed
Metal.framework	Always Used	Do Not Embed
OpenAL.framework	Always Used	Do Not Embed
QuartzCore.framework	Always Used	Do Not Embed
Security.framework	Always Used	Do Not Embed
SystemConfiguration.framework	Always Used	Do Not Embed
UIKit.framework	Always Used	Do Not Embed

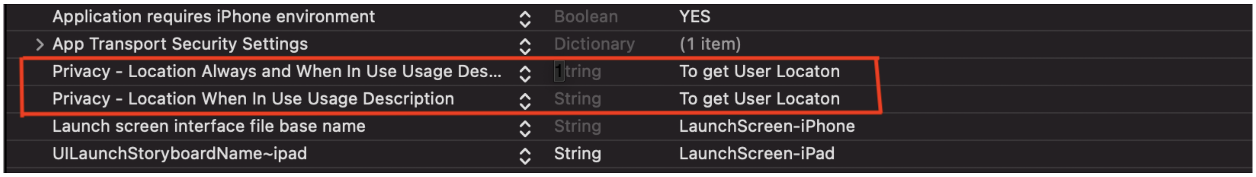
From iOS 8, Apple has made it mandatory to add following entries in the info.plist file along with a reason for using user location if your application is using **CoreLocation framework** to get user location.

Privacy - Location Always Usage Description – if you are using location service always even in the background.

Privacy - Location When In Use Usage Description - if you are using location service only when application is in foreground.

Here is a screenshot for the reference :

For being on the safe side you can add both the keys even if you are not using that option.



Application requires iPhone environment	Boolean	YES
> App Transport Security Settings	Dictionary	(1 item)
Privacy - Location Always and When In Use Usage Des...	String	To get User Locaton
Privacy - Location When In Use Usage Description	String	To get User Locaton
Launch screen interface file base name	String	LaunchScreen-iPhone
UILaunchStoryboardName~ipad	String	LaunchScreen-iPad

User Current Location

To get authorization status of location service for this application use the following API :

int getAuthrizationLevelForApplication ()

This API allows developers to get permission levels set by the user indicating whether the app is authorized to use location services. Following are constants provided by iOS devices:

1. **kCLAuthorizationStatusNotDetermined** : The user has not yet made a choice regarding whether this app can use location services.
2. **kCLAuthorizationStatusRestricted** : The user cannot change this app’s status, possibly due to active restrictions such as parental controls being in place.

3. **kCLAuthorizationStatusDenied** : The user explicitly denied the use of location services for this app or location services are currently disabled in Settings.
4. **kCLAuthorizationStatusAuthorizedAlways** : This app is authorized to start location services at any time.
5. **kCLAuthorizationStatusAuthorizedWhenInUse** : This app is authorized to start most location services while running in the foreground.

For more information regarding these values please refer to this link.

//--

To request user to grant permissions always for location services:

void requestAuthorizedAlways()

Use the above API if developers want to use location services always even when the application is in the background. This is not recommended because it will consume power from the mobile. Use this API only when the permission level for the application is **kCLAuthorizationStatusNotDetermined.** It is recommended to use **getAuthrizationLevelForApplication()** before this API to check permission level.

//--

To request user to grant permissions for location services only when app is in use:

void requestAuthorizedWhenInUse()

Use the above API if developers want to use location services only when the application is in foreground. This is recommended. Use this API only when the permission level for the application is **kCLAuthorizationStatusNotDetermined.** It is recommended to use **getAuthrizationLevelForApplication()** before this API to check permission level.

//--

To show alert to request permissions from the user :

void showAlertForPermissions(string alertTitle,string alertMessage,string defaultBtnTitle,string cancelBtnTitle)

Use the above API to display an alert to users for the location permissions for the application. It will redirect the user to the **Settings** application of the iphone, where the user can select the permission for the application. Developers need to pass following parameters in this API in the same order:

1. **alertTitle** : Title for the alert box
2. **alertMessage** : Message to display in the alert box
3. **defaultBtnTitle** : Title for the default button in the alert box.
4. **cancelBtnTitle :** Title for the button to remove the alert box.

//---

To start location monitoring:

bool startLocationMonitoring()

Developers need to use this API to start the location monitoring in iOS devices. It will return true if the monitoring has started successfully else it will return false. Reason for its failure can be the permissions provided by the user for the application so it is recommended to use **getAuthrizationLevelForApplication()** before using this API to check the permission levels.

//---

To stop location monitoring:

void stopLocationMonitoring()

Developers can use this API to stop location monitoring. It is preferred to stop location monitoring after some time of starting it. Otherwise the application will keep updating the location and it will lead to more power consumption by the application.

//---

To update message receiving object name and method name :

void setMessageReceivingObjectName(string msgReceivingGameObjectName,string msgReceivingMethodName)

Developers need to provide the name of the game object and method name which will accept the values from the plugin. It can be provided by using the above API. Need to pass following parameters in this API in the same order as mentioned:

1. **msgReceivingGameObjectName**: Name of the game object which will receive the callback.
2. **msgReceivingMethodName**: Name of the method associated with the game object which will receive value from the plugin.

Please look into the “**setLocationCallbackMessage**” method of the “**UnityReceiveMessages**” game object in the demo scene for more details.

Callbacks received by the method will be of string type separated by ‘/’. Break the string into an array and use the first element to identify the type of callback. It will return four type of callbacks :

1. **Address** : Which will contain address of location
2. **AddressError** : If address service throws some error
3. **Location** : When location api send the location values
4. **Location** Error : If some error happened in location sevice.

//--

To get address for current location:

void getAddressForCurrentLocation()

Developers can use this API to get the address for the current location from the plugin. Plugin will provide the required information to the method provided by **setMessageReceivingObjectName** API. Address provided will be “/” separated in following order:

1. **subThoroughfare** : Additional street-level information for the placemark.
2. **thoroughfare** : The street address associated with the placemark.
3. **postalCode** : The postal code associated with the placemark.
4. **locality** : The city associated with the placemark.
5. **administrativeArea** : The state or province associated with the placemark.

6. **country** : The name of the country associated with the placemark.

For more information regarding these values please refer this link.

//--

To get address for specific location:

void getAddressForLocationWithLatitudeLongitude(double locationLatitudeTemp, double locationLongitudeTemp)

To get the address of a specific location developers can use the above API providing following parameters in the same order as mentioned:

1. **locationLatitudeTemp :** Latitude of the location.
2. **locationLongitudeTemp :** Longitude of the location.

Plugin will provide the required information to the method provided by **setMessageReceivingObjectName** API. Address provided will be “/” separated as mentioned in the last API.

//--

To open Apple Map with user current location:

void openAppleMapWithCurrentLocation()

Above api will switch to the apple maps app on iphone which will be centered at the current user location.

//--

To open Apple Map with specific location:

void openAppleMapsWithLatitudeLongitude(double locationLatitudeTemp, double locationLongitudeTemp)

Above api will switch to the apple maps app on iphone which will be centered at the specified loction.

//--

To open Apple Map with driving mode to a specific location:

void openAppleMapsWithDrivingMode(double locationLatitudeTemp, double locationLongitudeTemp)

Above api will switch to the apple maps app on the iphone which will show the driving path from the current user location to the specified location.

//--

To add Map View within unity with User Location:

void openMapsViewWithCurrentLocation(double x_origin, double y_origin,double height, double width)

Above api will add the map view with current user location in unity itself which will update as the user location gets updated. This api will require the origin points, height and width of the map view. Refer the implementation in "LocationManager.cs"

//--

To add Map View within unity with Specific Location:

void openMapsViewWithCurrentLocation(double locationLatitudeTemp, double locationLongitudeTemp, double x_origin, double y_origin,double height, double width)

Above api will add the map view with specified location in unity itself . This api will require the origin points, height and width of the map view along with longitude and latitude of the location. Refer the implementation in "LocationManager.cs"

//--

To hide unhide map view :

void showMap(bool state)

Use this api to hide and unhide the map view added in the last two apis. True will show the map and false will hide the map.

Please contact us at **guptamayank516@gmail.com** in case of any query or clarifications.