

# Footmarks iOS SDK V3 Guide

**Patent Notice**: Various use cases and technical information provided herein are protected by a series of provisional and non-provisional patent applications.

Copyright © 2013 Footmarks, Inc. – All Rights Reserved Worldwide – Patent Pending

This document may not be reproduced without the expressed permission of Footmarks, Inc.

Any reproduction, without authorization, is an infringement of Footmarks copyright.

This document contains confidential and proprietary information.

Table of Contents

Footmarks iOS SDK V3 Guide 1

Document Overview 3

How the Footmarks Service Works 4

Footmarks SDK Compatibility 6

Step-by-step Integration 7

Description of SDK Methods and Classes 9

# Document Overview

This document will describe how to integrate, customize and use the Footmarks iOS SDK. The Footmarks SDK will enable your app to communicate with the Footmarks Service.

# How the Footmarks Service Works

The Footmarks Service is used in conjunction with the Footmarks SDK to deliver content to users when they enter the proximity of a SmartConnect Beacon. Many use cases are possible with the Footmarks Service. Below I will describe two use cases shown in Figure A and Figure B.

**Figure A Description:**

1. SmartConnect beacon broadcasts to user’s mobile handset via Bluetooth Low Energy
2. Footmarks enabled app (app which includes Footmarks SDK) receives broadcast and communicates with Footmarks Cloud Server
3. Footmarks Cloud Server sends offer details to user’s app
4. App alerts user of the Restaurant’s offer



**4**

**4**

**1**

2

2

3

**1**

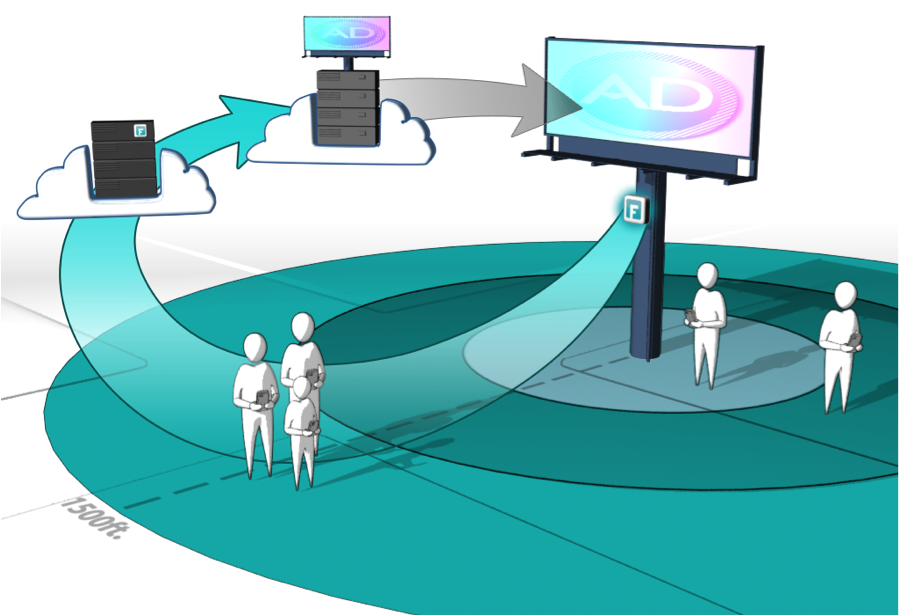
**2**

**2**

**3**

**Figure B Description:**

1. SmartConnect beacon broadcasts to user’s mobile handset via Bluetooth Low Energy (BLE).
2. Footmarks SDK communicates with Footmarks Cloud server to interpret what content to deliver to the user’s handset.
3. Footmarks Cloud server sends the necessary content to Footmarks SDK and then communicates with an “AD” cloud server any necessary information it is seeking via Footmarks Cloud API. **Note**: Bidirectional data flow.
4. “AD” cloud server determines what advertisement to display based on information received from Footmarks Cloud server.



**1**

**2**

**3**

**4**

**Figure B: Use Case 2**

# Footmarks SDK Compatibility

* The Footmarks SDK is compatible with any Apple or Android device that supports Bluetooth 4.0. These devices include iPhone 4S, iPhone 5, iPad 3 and iPad 4.
* The Footmarks SDK also requires iOS version 5.0+ to be running on the device.

# Step-by-step Integration

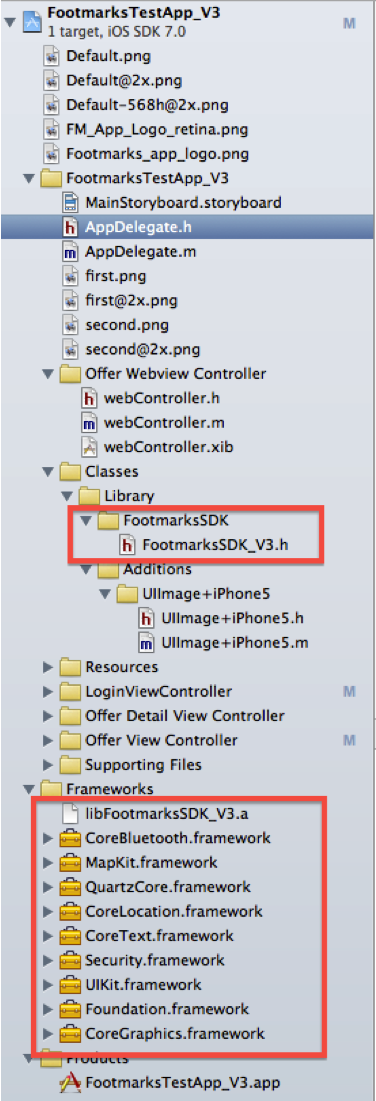
* Drag libFootmarksSDK\_VX.a into the Frameworks section of your Xcode project
  + Check “Copy items into destination group’s folder (if needed)”
  + Click the “Finish” button
* Add FootmarksSDK\_V3.h to your project

***Note****: See Figure 1 below to see how your Project Navigator should look at this point.*

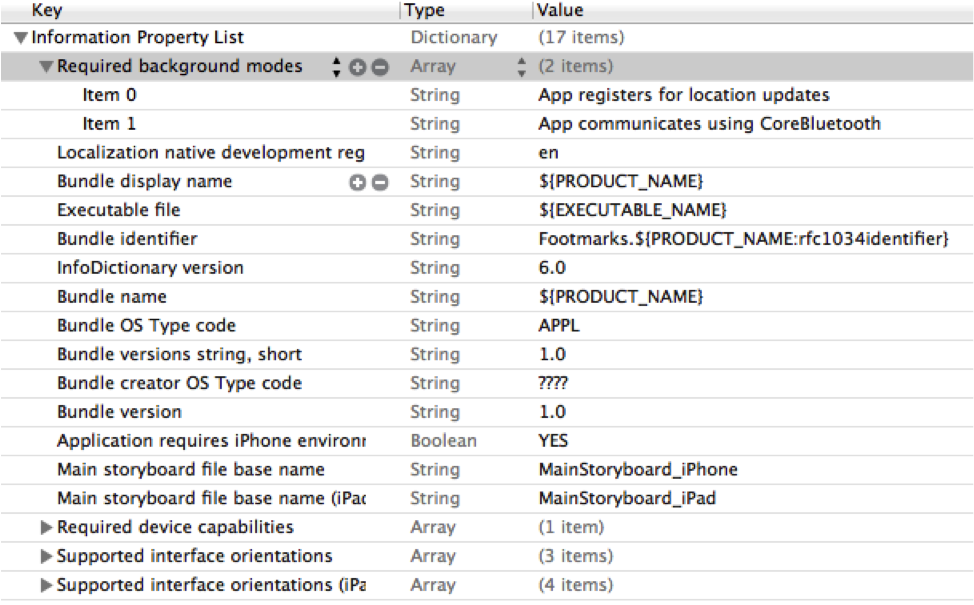
* Include CoreLocation, CoreBluetooth, UIKit, CoreGraphics, Security, Foundation, CoreText and CoreFoundation frameworks.
* Within your app’s plist file add a row for “Required background modes”. Add the following two fields under this row
  + App registers for location updates
  + App communicates using CoreBluetooth

***Note****: See Figure 2 below to see what your info.plist file should look like with the additional plist attributes.*

* Include the linker flags “-ObjC” and “-all\_load” (steps below)
  + Click your project in the Project Navigator
  + Click your projects Target
  + Click the Build Settings tab
  + Find the Setting named “Other Linker Flags” and add the 2 linker flags: “-ObjC” and “-all\_load”
* Footmarks SDK integration is now complete



**Figure 1: Project Navigator**



**Figure 2: info.plist file**

# Description of SDK Methods and Classes

/\*

LeDiscoveryDelegate is used to return information directly related to the SmartConnect Beacons and the iOS devices Bluetooth chip

\*/

@protocol LeDiscoveryDelegate <NSObject>

@optional

/\*

This delegate is invoked when the iOS devices Bluetooth is not powered on. You should ask the user to enable Bluetooth within this delegate

\*/

- (void) discoveryStatePoweredOff;

/\*

This delegate is invoked when a user comes within range of one of your SmartConnect Beacons. Perform all the necessary operations on the data within this delegate. For instance, this is the place to invoke a notification to the user in order to alert them of whatever content you would like.

@return a dictionary (dict) containing the custom JSON you entered into the Footmarks Management Console for the giving Beacon.

\*/

- (void) smartConnectDiscoveredWithInfo: (NSDictionary\*) dict;

@end

/\*

LeDiscovery is a singleton class that manages the devices Bluetooth capabilities and the beacons the app interacts with

\*/

@interface LeDiscovery : NSObject

/\*

@return a shared instance of the LeDiscovery singleton class

\*/

+ (id) sharedInstance;

/\*

Ends the devices Bluetooth discovery mode. Once called, beacons will not be detected until you start scanning again.

\*/

- (void) stopScanning;

/\*

Resumes scanning

\*/

- (void) startScanning;

/\*

Call this method to reset all internal SDK queues. This allows you to interact with Beacons you have already communicated with

\*/

- (void) resetBeaconQueues;

@property (nonatomic, assign) id<LeDiscoveryDelegate> discoveryDelegate;

@end

/\*

FootmarksAccountDelegate handles the Footmarks Service login process

\*/

@protocol FootmarksAccountDelegate <NSObject>

/\*

This delegate is invoked if the login is successful

\*/

- (void) loginSuccessful;

/\*

Invoked when login fails.

@return a string containing the reason for the login attempts failure

\*/

- (void) loginUnsuccessful: (NSString\*)error;

@end

/\*

FootmarksAccount is a singleton that handles the Footmarks Service Account processes.

\*/

@interface FootmarksAccount : NSObject

+ (id) sharedInstance;

@property (nonatomic, assign) id<FootmarksAccountDelegate> accountDelegate;

/\*

This method attempts to login to the Footmarks Service.

@param appKey

The app key provided by the Footmarks Management Console upon registration

@param appSecret

The app secret provided by the Footmarks Management Console upon registration.

@return True if no problems occurred in the process of attempting the call. Else false. Note: Success or Failure of the call will be returned to the FoomarksAccountDelegate

\*/

-(BOOL)loginToFootmarksServer: (NSString\*) appKey andAppSecret: (NSString\*) appSecret;

/\*

This method revokes the user's access token. Once revoked, the current access token will not work for protected calls to the Footmarks server.

\*/

-(void)revokeAccessToken;

@end

/\*

Necessary include to perform JSONKit operations within the SDK.

\*/

@interface NSString (JSONKitDeserializing)

- (id)objectFromJSONString;

@end