Getting Started with iBeacon

Version 1.0

June 2, 2014

Getting Started with iBeacon Overview

Introduced in iOS 7, iBeacon is an exciting technology enabling new location awareness possibilities for apps. Leveraging Bluetooth Low Energy (BLE), a device with iBeacon technology can be used to establish a region around an object. This allows an iOS device to determine when it has entered or left the region, along with an estimation of proximity to a beacon . There are both hardware and software components to consider when using iBeacon technology, and this document will give an introduction to both, along with suggested uses and best practices to help ensure a highly effective deployment leading to an outstanding user experience.

iBeacon has 3 different audiences. You may fall into one, two, or possibly all three of these categories, depending on your role.

1. **App Developers**
   If you want to add new location awareness to your application, you would use the Core Location APIs in iOS to be notified when the iOS device has moved into or out of a beacon region. You can also determine approximate proximities to a device generating iBeacon advertisements. Everything you need to get started is included in the iOS SDK, no additional license is required.
2. **People Deploying Devices With iBeacon Technology**
   Whether you manage a sports arena, a museum, a retail store, or any of the myriad other physical locations where beacons could be employed, you need to be aware of how these devices work, issues surrounding signal strength and materials, and understand how to calibrate and test your deployment. If you are interested in using the iBeacon Logo on signage at a venue, but will not make devices with iBeacon technology, you will need to obtain an iBeacon logo license before using the iBeacon logo. Please visit https://developer.apple.com/ibeacon/ to apply for a license to use the iBeacon logo.
3. **People Making Devices With iBeacon Technology**
   If you are interested in manufacturing devices with iBeacon technology, you will need to obtain a license before building these devices. Please visit https://developer.apple.com/ibeacon/ to apply for an iBeacon license. Licensees receive access to technical specifications, a license to use the iBeacon logo, and the iBeacon Identity Guidelines.

Devices with iBeacon Technology

Devices with iBeacon technology can be powered using coin cell batteries for a month or longer, or operate for months at a time using larger batteries, or can be powered externally for extended periods of time. iOS devices can also be configured to generate iBeacon advertisements, although this functionality is limited in scope. This would be appropriate for uses such as a Point Of Sale or kiosk application, or for an application that wants to become an iBeacon for a short time while someone is actively using the application.

An iBeacon advertisement provides the following information via Bluetooth Low Energy:

Field	Size	Description
UUID	16 bytes	Application developers should define a UUID specific to their app and deployment use case.
Major	2 bytes	Further specifies a specific iBeacon and use case. For example, this could define a sub-region within a larger region defined by the UUID.
Minor	2 bytes	Allows further subdivision of region or use case, specified by the application developer.

The UUID, major and minor values provide the identifying information for the iBeacon. Generally speaking, this information is hierarchical in nature with the major and minor fields allowing for subdivision of the identity established by the UUID. UUIDs can be generated by using the `uuidgen` command line utility in OS X, or programmatically using the NSUUID Foundation class.

The following table shows examples of how these values may be used for a nationwide retail store. The UUID is shared by all locations. This allows an iOS device to use a single identifier to recognize *any* of the stores with a single region. Each specific store, San Francisco, Paris, and London, is then assigned a unique major value, allowing a device to identify which specific store it is in. Within each individual store, departments are given separate minor values, although these are the same across stores to make it easier for an app on a device to readily identify departments.

Store Location		San Francisco	Paris	London
UUID		D9B9EC1F-3925-43D0-80A9-1E39D4CEA95C		
Major		1	2	3
Minor	**Clothing**	10	10	10
	Housewares	20	20	20
	Automotive	30	30	30

Using this information, an iOS device could identify when it has entered or left one of the stores, which specific store it is, and what department the user might be standing in. These values are determined by the person or organization deploying the beacon device. UUIDs, and major & minor values are not registered with Apple.

iBeacon relies on BLE, and therefore require an iPhone 4S (or later), iPod touch (5th generation), iPad (3rd generation or later), or iPad mini.

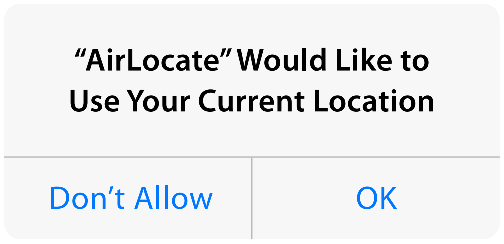
For more details about incorporating iBeacon technology in a product, you will need to obtain a license from Apple. Please visit <https://developer.apple.com/ibeacon> to apply for an iBeacon license.

iBeacon Software — Core Location APIs

Prior to iOS 7, Core Location used regions defined by a geographic location (latitude and longitude) and a radius, known as a 'geofence'. iBeacon enables a new level of flexibility by defining regions with an identifier. This allows beacons to be affixed to objects that are not tied to a single location. For example, a beacon device could be used to set a region around a movable object like a food truck or on a cruise ship. Furthermore, the same identifier can be used by multiple devices. This would enable a retail chain to use beacons in all their locations and allow an iOS device to know when it enters any one of them.

Privacy and Location

Because iBeacon is part of Core Location, the same user authorization is required in order to be used. Users will see the same location authorization alert when an application attempts to use the iBeacon APIs:



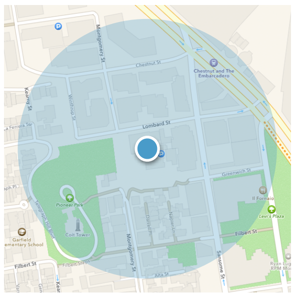
Applications that use beacon region APIs in CoreLocation will appear in the Settings app under Privacy > Location Services and users can allow or deny an application's access to iBeacon functionality at any time. Furthermore, any Bluetooth packets that are associated with iBeacon are excluded from the CoreBluetooth APIs.

As with geofence region monitoring, when in active use the status bar will show a hollow arrow. When using ranging, the status bar will show the solid location arrow.

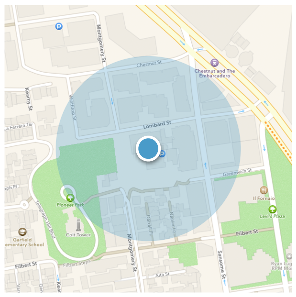
Accuracy of iBeacon

To ensure an effective user experience, it is important to consider how signals from a beacon are detected and used to determine accuracy. When an iOS device detects a beacon's signal, it uses the strength of the signal (RSSI, or Received Signal Strength Indication) to determine both proximity to the beacon, as well as the accuracy of its estimation of proximity. The stronger the signal, the more confident iOS can be about the proximity to the beacon. The weaker the signal, the less confident iOS is about the proximity to the beacon.

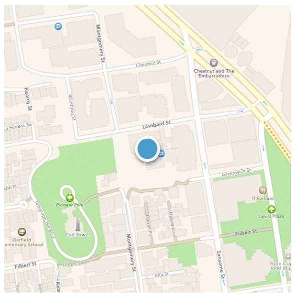
Accuracy can best be understood by relating it to how GPS works in iOS today. When an iOS device can clearly receive GPS signals, such as when a device is in the open outdoors with an unobstructed line of sight to the orbiting GPS satellites, the more accurately your location can be determined. This can easily be seen in the Maps application where the location accuracy is represented by a blue circle surrounding your current location indicator. If a device is indoors or the line of sight to the satellites is obstructed, a large blue circle indicates a lower accuracy. That is, the device could be located anywhere within the blue circle. As the line of sight to the satellites improves (e.g. the device is taken outdoor or removed from a backpack) the accuracy improves, represented by a smaller blue circle. With a better received signal strength the device can narrow the margin of error and be more confident of its location.



Device indoors, low accuracy depicted by large blue circle. The device may be anywhere in the circle.



Device outdoors, but in a backpack. Greater accuracy than indoors represented by a smaller, but still present, blue circle.



Device outdoors, held in hand with clear view to the sky. Greatest level of accuracy.

With signals are received from a device with iBeacon technology, the signal strength is generally correlated to how far away a device is from the beacon. In an ideal condition (that is, unobstructed line of sight between a device's antenna and the beacon) the closer the person is the more accurate the result.

As shown in Figure 1, when a device is far away from a beacon, the signal strength will be lower than when it is close. Due to this diminished signal strength, iOS does not have high confidence on accuracy of the proximity estimate to the beacon. This is similar to the large blue circle in the GPS example above.

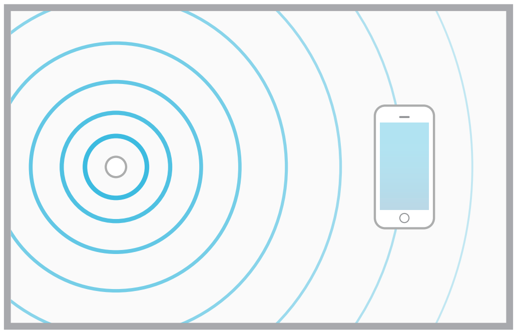


***Figure 1**: When a device is further away from the beacon, signal strength is diminished and therefore the accuracy estimate will increase.*

As the device gets closer to the beacon, the received signal strength increases and therefore the accuracy of the proximity estimate increases. This would be analogous to the smaller blue circle in the GPS example. Shown in Figure 2, a device that is closer to a beacon will have a higher confidence about its proximity to the beacon emitting the signal.

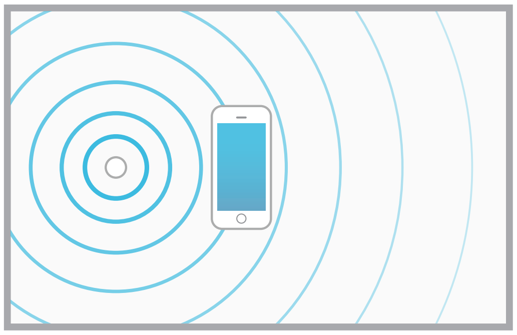


***Figure 2**: Signal strength increases as a device moves closer to the beacon, leading to a better proximity estimate.*

However, just as GPS signal strength can be diminished by physical objects like buildings or being placed in a backpack, purse or pocket, so can a beacon's signal strength. Signal attenuation, or the loss of intensity of a signal, can be caused by many factors. Physical materials surrounding the beacon, such as the wall depicted in Figure 3 between the device and the beacon will affect the received signal strength. This may cause the device to believe that the beacon is further away than it actually is.

The human body itself is an excellent attenuator of Bluetooth signals. Simply having your back to a beacon (i.e. where your body is positioned between the device and the beacon) will affect the signal strength and thereby lower the accuracy. Figure 4 shows this signal strength being diminished when somebody is physically positioned between the iOS device and the beacon.

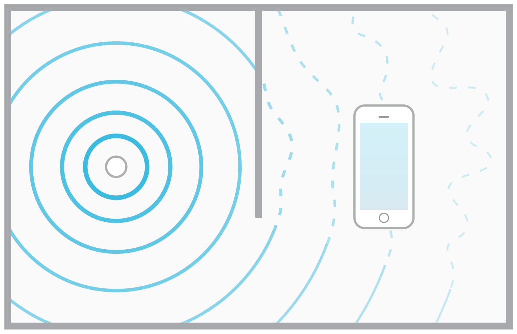


Figure 3: *Physical objects and materials can block signals, reducing the received signal strength.*

When building an application that uses either GPS or beacon, it is important to consider this accuracy. The values reported by the Core Location objects (the `horizontalAccuracy` property in the CLLocation class, or the `accuracy` property in the CLBeacon class) indicate this level of uncertainty, or the margin of error. Both are measured in meters. The higher the value, the lower the certainty of the position of the device or beacon. Keep in mind that depending on the physical surroundings a low accuracy may not be possible.

Region Monitoring

Similar to the existing geofence region monitoring, an application can request notifications when a device enters or leaves a region defined by a beacon. When an application makes this request to begin monitoring a beacon region it must specify the UUID of the iBeacon advertisement. While an app is limited to 20 regions being monitored, by using a single UUID in multiple locations, a device can easily monitor many physical locations simultaneously. Using the retail store example shown in the table earlier, a device can monitor 3 separate physical locations (San Francisco, Paris, and London) using the same UUID. The impact of this UUID-based approach compared to geofences cannot be understated: with a single line of code an application can establish monitored regions around an arbitrary number of objects or locations.

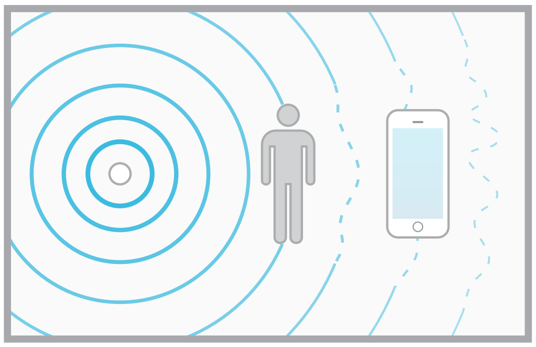


Figure 4: *Human bodies can also block signals.*

In addition to the UUID, an application can optionally supply the major and minor fields to further specify a beacon region to be monitored. Continuing with our retail chain example, if the app only specifies a UUID for the beacon region then it will be notified when the user enters or leaves *any* of the retail stores. Since the major field is being used to determine specific stores, if the user only wanted to be notified when entering a specific store, the application could configure the beacon region using the UUID + major value. Or perhaps the user is only interested in being notified when they have entered a specific department in that store. In that case the app would configure the beacon using UUID + major + minor values. This level of granularity is up to the app developer and can be specified dynamically at runtime.

As with the existing region monitoring, when the user enters or exits the beacon region, the application will be notified. If the application is not currently running (for example, if it was terminated due to memory pressure on the device), then the application is launched in the background and the notification delivered. One important consideration is in iOS 7 if the user explicitly disallows Background App Refresh (either globally or specifically for your app) then your app will no longer receive region monitoring notifications. It can continue to use the ranging APIs, however.

Being based on Bluetooth Low Energy, the typical range will be in the tens of meters, which provides a more accurate monitoring than geofence region monitoring (typically on the order of 100 meters minimum). As discussed above, geofences tend to be less accurate indoors so using iBeacon technology can dramatically improve the region monitoring results for indoor use cases. However, the accuracy can be affected by the physical positioning of a beacon, whether the user has their device in their pocket, or simply whether the beacon is in front of or behind the user can all affect the point at which a region has been entered or exited.

Ranging
iOS 7 introduces a new set of APIs for determining the approximate proximity to a device using iBeacon technology, a process known as "ranging". Based on common usage scenarios, iOS applies filters to the accuracy estimate to determine an estimated proximity to a beacon. This estimate is indicated using one of following four proximity states:

Proximity State	Description
Immediate	This represents a high level of confidence that the device is physically very close to the beacon. Very likely being held directly up to the beacon.
Near	With a clear line of sight from the device to the beacon, this would indicate a proximity of approximately 1-3 meters. As described in the section on accuracy, if there are obstructions between the device and the beacon which cause attenuation of the signal, this Near state may not be reported even though the device is in this range.
Far	This state indicates that a beacon device can be detected but the confidence in the accuracy is too low to determine either Near or Immediate. An important consideration is that the Far state does not necessarily imply "not physically near" the beacon. When Far is indicated, rely on the `accuracy` property to determine the potential proximity to the beacon.
Unknown	The proximity of the beacon cannot be determined. This may indicate that ranging has just begun, or that there are insufficient measurements to determine the state.

iBeacon User Experience Considerations
While there is a correlation between the proximity states and accuracy, the mapping is not necessarily 1:1. Consider the example of our nationwide retail store where beacons have been deployed throughout the store. An app might use region monitoring to detect the entry to the store to trigger a local notification welcoming the user to the store and inviting them to launch the app. To avoid annoying the user, the app may want to only show this notification once, only the first time the user enters the store.

Once inside the app, a custom in-store interface could be presented. If the major value contained in the iBeacon advertisement represents the specific store location then the app knows immediately what store the user is in. With the app frontmost, the device in the user's hand, and the screen turned on, this is an ideal situation to begin ranging for all the beacons in the store. Large home improvement stores tend to have many aisles and departments. With beacons positioned at the end of each aisle and within departments, an app should be able to use the proximity states for the beacons that can be seen, the app could display the user's approximate location on a map.

While many situations in this example might lead to proximity states of Near or Immediate (for example, if the user held their iPhone up to a particular display where a beacon device was positioned), due to the physical objects (typically metal shelving, large bulky display items, etc) or other customers in the store, the app may only see proximities of Far. In this situation, an app might display an interface that highlights information about nearby beacons but not lock the user into a specific beacon. Instead, the app may want to let the user choose the item that would be most relevant to them (either due to their interest or because they can readily identify which beacon actually is closest).

Passbook Integration
Passbook passes can take advantage of devices with iBeacon technology as well. By including the UUID of beacon, a Passbook pass can be made relevant when it is in the beacon's region. This works the same way specifying `latitude` and `longitude` values in the `locations` array of your pass. You can specify the UUID and, optionally, the major and minor values as an array in the `beacons` key for your pass.

Deploying iBeacon

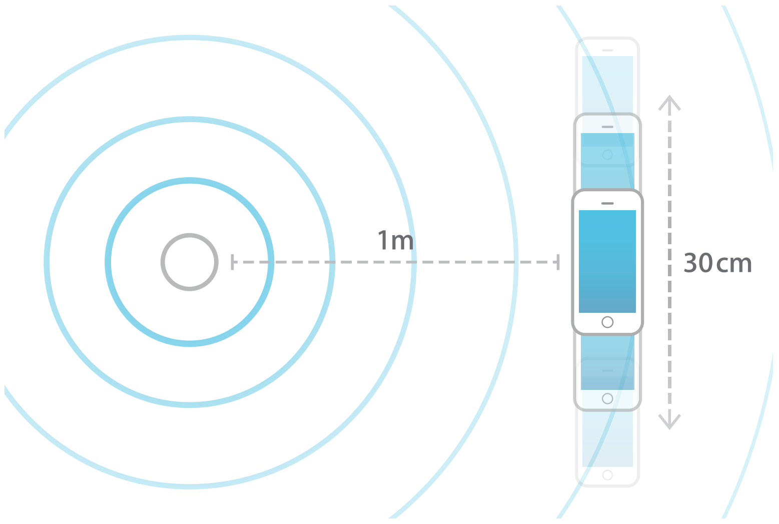
As you prepare to deploy any implementation based on iBeacon technology, you need to carefully evaluate the real world performance of your solution.

Physical Limitations
iBeacon devices use Bluetooth Low Energy to broadcast signals. BLE is based on the 2.4 GHz frequency and as such is subject to attenuation by various physical materials such as walls, doors or other physical structures. The 2.4 GHz frequency can also be affected by water, which means the human body will also affect signals. This is important to be aware of because when the Bluetooth signal is attenuated, or lessened, this affects the signal strength received by an iOS device. As discussed above, when the received signal strength is lessened, an iOS device's ability to estimate the proximity to an iBeacon device is diminished.

Calibrating iBeacon
To provide the best user experience, it is critical to perform calibration in your deployment environment. As each beacon is installed you should perform a calibration step. Core Location uses an estimation model that requires calibration at a distance of 1 meter away from the beacon. To perform this calibration you should:

- Install the beacon and have it emitting a signal.
- Using an iPhone or iPod touch running iOS 7 or later, and has a Bluetooth 4.0 radio, repeatedly sample the signal strength at a distance of 1 meter for a minimum of 10 seconds. When taking these signal strength readings you should hold the device in a portrait orientation with the top half of the device unobstructed.

- Move the device slowly back and forth on a 30cm line, maintaining orientation, and



  remaining equidistant from the measuring device (see diagram)

- For the duration of the calibration process, gather the values reported in the CLBeacon's `rssi` property.
- Average the collected `rssi` values to obtain the Measured Power value.
- Apply this Measured Power value to the beacon. Consult the details provided for the beacon used, as they may differ from manufacturer to manufacturer.

As discussed above, the physical surroundings can affect the signal strength. Since the surroundings will almost certainly vary between installation locations it is important to repeat these steps for each beacon that is installed.

Best Practices

For an optimal user experience and successful deployment, be sure to consider the following best practices:

- Ranging API are not expected to be used in the background. For best results, ranging should be used when your app is frontmost and the user is interacting with your app.
- When using the ranging APIs and multiple devices with iBeacon technology are detected, Core Location will report the beacons in an order that is a best guess of their proximity. Due to the issues of signal attenuation discussed above, this order may not be correct. For example, if two beacons are attached to objects, and are detected by the iOS device, but one signal is considerably stronger than the other but is physically further away, this may cause the farther away beacon to be reported first. Apps should carefully inspect the proximity zone reported by the beacon and if all beacons are in the Far zone then consider presenting to the

user that the two objects have been detected nearby, and allow the user to select the object that interests them.

- Leverage the optional text field within the location authorization alert to explain why the app is asking to use the user's locations. If your app has on-boarding screens, explain the benefits of why the user should agree to allowing the app to knowing their location. You can specify this optional text using the `NSLocationUsageDescription` key in your app's `Info.plist` file.
- If you are purchasing a 3rd party device with iBeacon technology, it is important to understand how these devices can be configured and who is going to do the installations, maintenance, etc.
- When deploying beacon devices in the field, be sure to train any employees that might need to interact with them. For example, if you are deploying a retail solution, make sure your retail sales associates are trained on how the iOS app interacts with the devices, what the benefits are to your customers, supported iOS device models, suggestions for troubleshooting, etc.
- If you plan to have signage in your location, it is encouraged that you have the iBeacon trademark and logo license. Please visit <https://developer.apple.com/ibeacon> to apply for an iBeacon license.
-

Common Questions and Answers

Q: Can I use iBeacon technology to precisely show a user's location on a map when indoors?
A: Due to the issues around signal strength and the variabilities in deployment environments, iBeacon technology is not intended to be used for specific location identification. It should be capable of providing room-level accuracy, but there are many factors that need to be considered to build a successful deployment. Number of beacons, where they are positioned, expected use cases, and many more factors need to be examined to provide a good user experience.

Q: How can I prevent other apps from detecting my devices with iBeacon technology?
A: In order for an app to be able to respond to a device that transmits iBeacon advertisements, it must know the UUID contained with the advertisement. Because a beacon device is advertising using BLE, it is possible for the UUID to be "sniffed" off the air and once that UUID is known, it could be used by other apps.

Q: Does using iBeacon technology put user's private data at risk?
A: iBeacon advertisements only contain UUID, major and minor values. This is a unidirectional broadcasting; there is no bidirectional communication between a beacon device and an iOS device via iBeacon technology, therefore iBeacon technology cannot be used to receive by a beacon to receive information from a user. What an app does in response to a notification triggered by an iBeacon advertisement is a separate matter, but this is no different from using existing geofencing technologies.

Q: Can I use an iOS device to issue iBeacon advertisements?
A: Yes. Any app can use the Core Bluetooth APIs to send iBeacon advertisements.

Q: Can I use an iOS device to issue iBeacon advertisements while my app is in the background?

A: No. For an iOS device to issue iBeacon advertisements, the app requesting this functionality must be frontmost, with the screen turned on and the device unlocked.

Q: If my app starts monitoring beacon regions, how will that affect battery performance?

A: iOS devices that support iBeacon can efficiently monitor iBeacon regions in the background with marginal power drain. Monitoring iBeacon regions is significantly less power demanding than running normal location updates constantly in the background.

Document Revision History

Date	Notes
2014-06-02	Initial verison