Core Location / Ranging for Beacons

Sample Code

Ranging for Beacons

Configure a device to act as a beacon and to detect surrounding beacons.

Download

iOS 13.0+ | iPadOS 13.0+ | Xcode 11.2+

Overview

Beacons make location-based products and services available to users by broadcasting information to your device. Ranging is the process of reading the characteristics of a beacon region, such as signal strength, advertising interval, and measured power.

This sample code project configures a device to act as a beacon, and configures a device to use ranging to find surrounding beacons. Use two iOS devices to run the sample, with one acting as a beacon, and the other ranging for the beacon.

> Note
>
> This project is associated with WWDC 2019 session 705: What’s New in Location.

Configure a Device to Act as a Beacon

Run the sample app on the first iOS device. Select the option to Configure a Beacon. The project hardcodes a default value for the UUID that can be changed in `ConfigureBeaconViewController.swift`.

```
let beaconUUID = UUID(uuidString: "E2C56DB5-DFFB-48D2-B060-D0F5A71096E0")
```

Optionally modify the major and minor value for the beacon, then select the Enabled switch on the configuration screen to start advertising.

`ConfigureBeaconViewController.swift` contains a view controller object that configures the iOS device running this app to act as a beacon. The `configureBeaconRegion()` method sets up the region and starts advertising itself.

```
if peripheralManager.state == .poweredOn {
    peripheralManager.stopAdvertising()
    if enabled {
        let bundleURL = Bundle.main.bundleIdentifier!

        // Defines the beacon identity characteristics the device broadcasts.
        let constraint = CLBeaconIdentityConstraint(uuid: beaconUUID!, major: major,
        region = CLBeaconRegion(beaconIdentityConstraint: constraint, identifier: bu

        let peripheralData = region.peripheralData(withMeasuredPower: nil) as? [Stri

        // Start broadcasting the beacon identity characteristics.
        peripheralManager.startAdvertising(peripheralData)
    }
```

Configure a Device to Range for Beacons

Using a second iOS device, run the sample app and tap Range for Beacons to scan for beacons. Add a UUID to range for by tapping the Add button in the upper corner of the screen. The hardcoded UUID appears by default.

`RangeBeaconViewController.swift` contains a view controller object that ranges a set of beacon regions that the user adds. As in any location-based service, first request authorization. Use a `CLLocationManager` instance to request that authorization, set up the constraint based on the hardcoded UUID, then tell the instance to start monitoring.

```
self.locationManager.requestWhenInUseAuthorization()

// Create a new constraint and add it to the dictionary.
let constraint = CLBeaconIdentityConstraint(uuid: uuid)
self.beaconConstraints[constraint] = []

/*
By monitoring for the beacon before ranging, the app is more
energy efficient if the beacon is not immediately observable.
```

```
*/
let beaconRegion = CLBeaconRegion(beaconIdentityConstraint: constraint, identifier:
self.locationManager.startMonitoring(for: beaconRegion)
```

When the device enters the specified region, the `locationManager(_ manager: CLLocationManager, didDetermineState state: CLRegionState, for region: CLRegion)` delegate method receives the region state and starts ranging beacons.

While one or more beacons are in range, the `locationManager(_ manager: CLLocationManager, didRange beacons: [CLBeacon], satisfying beaconConstraint: CLBeaconIdentityConstraint)` delegate method receives their characteristics in the passed array.

See Also

iBeacon

Determining the proximity to an iBeacon device

Detect beacons and determine the relative distance to them.

Turning an iOS device into an iBeacon device

Broadcast iBeacon signals from an iOS device.

`class CLBeacon`

Information about an observed iBeacon device and its relative distance to a person’s device.

`protocol CLCondition`

The abstract base class for all other monitor conditions.