



Beacon Setup Guide



Beacons' Proximity UUID, Major, Minor, Transmission Power, and Interval values made easy

In this short guide, you'll learn which factors you need to take into account when planning your beacon deployment and setting up your beacons.

We'll cover:

1. The best way to configure your beacon identifiers (UUID, Major, and Minor)
2. Optimizing Transmission Power
3. Choosing the best Interval Value

If you're new to beacons, planning your first deployment, or wondering how to optimize your existing beacon fleet, this guide is just for you.



Beacon Identifiers

A quick recap: how beacons work

A beacon is a small Bluetooth Low Energy radio device that broadcasts small packets of data—called advertising packets—at regular intervals. The amount of the advertising packets and what they consist of depends on a communication standard that a given beacon runs on. There are a couple of such communication standards but the most popular ones are iBeacon and Eddystone.



iBeacon has one advertising packet, a beacon's ID. Eddystone has three of them—Eddystone-EID, Eddystone-URL, and Eddystone-TLM. The two latter ones are used for very specific use cases. In most deployments based on user interactions with a dedicated app, you'll use the Eddystone-EID, which is similar to iBeacons' ID.

Why is the beacon ID so crucial? Beacons have a very limited amount of memory, so they can't broadcast too much information. If you want to trigger some content to your users' devices when they're in a certain place, you need to store that content in the cloud or directly in your app and assign an ID

to it. That ID should match the ID of the beacon that will trigger your action. So any time someone with your app installed passes by the beacon, their smartphone will grab the ID, interpret it, and display the proper content.

In other words, beacons don't broadcast content, and their IDs don't carry any message by themselves. It's your app that makes the beacon ID meaningful. The more beacon triggers you want to have, the more beacons you'll need. That's why, to make beacon management and your app development easier, you need a strategic system of picking the right beacon IDs. This guide will help you create one.



Card Beacon

iBeacon identifiers

– UUID, Major and Minor



The proximity UUID (Universally Unique Identifier, UUID for short), Major, and Minor parameters – together make up your iBeacon's unique identifier, and are the key components of all advertising packets continually transmitted by your iBeacons.

Deciding how to configure these parameters is a key consideration when thinking about your beacon deployment. They'll affect your beacons' range and lifespan and help you organize your deployment.

Proximity UUID

The purpose of the proximity UUID is to distinguish beacons in your network from all other beacons in networks outside your control. When you receive your beacons, they will all have a common proximity UUID set, which will look something like:

f7826da6-4fa2-4e98-8024-bc5b71e0893e

It is the UUID that all of Kontakt.io's iBeacons sell come with. Since you want to distinguish your beacon fleet from other Kontakt.io's customers, it's recommended to change it to a custom string.

Major and Minor values

Major and Minor are unsigned integer values between 1 and 65535.

Major values are intended to identify and distinguish a group—for example all beacons on a certain floor or room in your venue could be assigned a unique Major value.

Minor values are intended to identify and distinguish an individual—for example distinguishing specific beacons within a group of beacons assigned a Major value.

So an individual iBeacon identifier could look like:

UUID	Major	Minor
f7826da6-4fa2-4e98-8024-bc5b71e0893e	59192	22271

Eddystone identifiers –Namespace, Instance



Eddystone's unique identifier is made up of Namespace and Instance.

Namespace serves a similar purpose as Proximity UUID does in iBeacon, and by default at Kontakt.io, it looks like this:

f7826da6bc5b71e0893e

Instance is Eddystone's counterpart of iBeacon's Major, or both Major and Minor mixed up together. By default, it's a randomly generated hexadecimal number, which means it contains digits (0-9) and characters (a-f).

Beacon identity and regions

Regions on iOS

An iOS application monitors regions, and a region can be one or more beacons. In case of iBeacons, a region may be specified using one of the following methods:

- a single UUID,
- UUID and Major
- UUID, Major, and Minor

If we assign a single proximity UUID to many locations, our application can monitor all these locations simultaneously. This can have a big impact in reducing the complexity of your application.

When it comes to Eddystone, there are no regions in this format's anatomy but you can artificially create ones using our iOS SDK. You can filter your Eddystones based on their Namespace, Instance, or a specific URL. By the way, the SDK will also help your app "understand" that a certain Eddystone is actually a beacon (normally, iOS treats Eddystone as any other Bluetooth device. In order to act properly, it needs a "translator", and our SDK does that for you).

Regions on Android

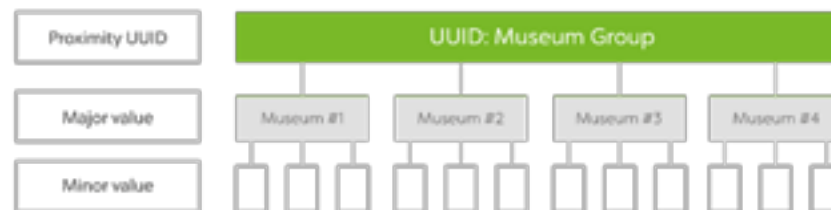
Android treats iBeacons and Eddystones the same, and it doesn't have a built-in region functionality. If you need a kind of regions, you can filter your beacons using our Android SDK.

Beacon identifiers in practice

Ok, so we covered the basics. How might that affect our beacon deployment strategy?

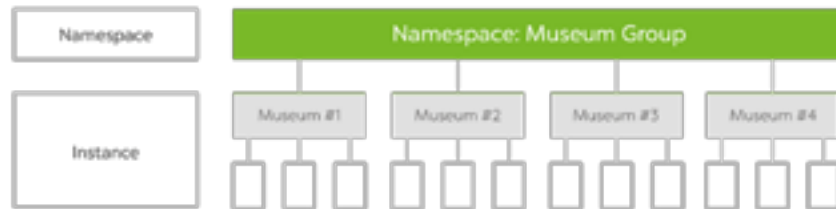
Let's look at a museum group as an example, which has a network of four Museums in distinct physical locations. Beacons are deployed to give visitors a rich interactive experience when they enter any of our Museums, and they provide users with information about individual collections as they approach each exhibit.

iBeacon



In our Museum Group example, we have assigned a single UUID to the group. A user's device can monitor our four physical locations using this single UUID. Our application 'knows' when the user has entered (or left) any of the museums in the group and notify the user appropriately just by referencing the UUID. The Major field, tells us which Museum the user is currently in so your application can act accordingly – for example by offering event suggestions, maps, audio guides, or other content. The Minor field can be used to identify entire collections within the Museum, or the individual exhibits within the collection. Your application can use this level of granularity to provide tailored information for the exhibit, a more detailed gallery map, or perhaps an audio guide through the entire exhibit.

Eddystone



For a deployment of Eddystone beacons, our network would look like this:

In this case, all beacons in our Museum Group have one Namespace. Since there's no Minor field, we use Instance to distinguish specific beacons at specific museums.

Points to consider for your beacon deployment

- Your Proximity UUID/Namespace ID should generally be the same for all beacons for your organisation.
- Depending on your the granularity of your needs, you could also (using our example) have each museum assigned with its own UUID/Namespace. Just remember that UUIDs are meant to identify all the beacons you own, or at least very large groups of them.

- A region can be defined by proximity UUID only, a UUID plus Major value, a UUID plus Major and Minor values. You can select the level of granularity you need, depending on how many venues, beacons, and groups of beacons within specific venues you have.
- Your app is going to be notified when a user enters or leaves a region, so ask yourself at what level you'll need such notifications. We recommend using your Major values to define your regions.
- Remember that once you have detected that the user has entered a region, and focused their attention with a notification, your application can start ranging all beacons in the region and provide detailed information back to the user.



Beacon Transmission Power

In this part, you'll learn about Transmission Power (Tx Power)—one of the basic configuration settings you can implement in your beacons, and how this affects your deployment and subsequent operations.



The basic configuration settings for beacons include:

1. Transmission Power (also known as Tx Power) and
2. Signal Interval (how often your beacons push out a signal)

The settings you choose for Transmission Power and Interval will have a large impact on how your beacon network operates. Choosing the most practical and efficient settings is greatly determined by your specific needs and use case.

Today, we will focus on Transmission Power only, because there are a number of factors to consider when choosing the optimal Tx Power setting for your needs. The major things the above settings will affect include:

1. Signal range (greatly influenced by Tx Power)
2. Signal stability (somewhat influenced by Tx power but more so by Interval)
3. Battery life (somewhat influenced by Tx power but more so by Interval)

Now it may be easy to say “I want the best of all three” but this is not always possible or even really necessary. The keyword here is efficiency and practicality—using too much or too little power can greatly affect the effectiveness of your beacon deployment. Let’s go over the particulars.

Beacon Transmission Power Basics

Your Transmission Power setting determines how powerfully the signal will be transmitted by your beacon. This is measured in dBm (decibel-milliwatts) and corresponds to a number rating (from 0 to 7) that you change via our Admin App (0 is least powerful – 7 is most powerful).

As you might expect, making your transmission more powerful will increase the range of your signal. It is also true, however, that the more powerful your transmission, the bigger the energy drain, and accordingly the shorter your battery life.

The table below shows how your Tx Power setting affects [Kontakt.io Beacon’s](#) range and battery life.

Admin App Setting	Tx Power	Expected Range	Expected Battery Life *
1	-30 dBm	4 m (13')	Up to 2 years
3 (default)	-12 dBm	35 m (115')	Up to 2 years
7	4 dBm	70m (230')	12 months

*Assuming 350ms default interval and optimal environmental conditions.

Determining optimal beacon Transmission Power

The best way to determine the best Tx Power for your needs is to answer the following questions:

How much beacon range do you really need?

For example, you may want to use the max Tx Power for beacons placed at a venue entrance. This is because you likely want as many people connecting with this beacon as possible (e.g. to send a welcome message.) In this case, you may need to sacrifice battery life to gain the widest range possible.

Alternatively, you may have beacons attached to a specific item (say an exhibit or shelf) where you only really want to connect to people who are standing directly in front of said item. In this case you certainly want to use a lower Tx Power to conserve battery life, and still achieve the same expected results (i.e. sending relevant information about that specific item.)

How often are you really willing to change the beacon's battery?

This is a question of cost vs. convenience vs. efficiency. Maybe you are perfectly willing to change your battery every month, or perhaps you vastly prefer getting a few years out of each battery. Since you can configure beacons individually, it is also pretty simple to get the “best of both worlds”. For example, you could

have a few entry / exit beacons with max Tx Power, and then the rest of your fleet might use a lower setting.

Keep in mind however, that to maximize the value you get out of beacons—you may end up having to replace your batteries more often than you may prefer. It's just good business sense; a little bit of bother to gain a lot of profit.

When do you really need to use a higher Tx Power?

This is a question of intended use. These are some of the common reasons our customers have given for using the maximum power setting:

1. It is really important that this beacon can be found by as many people as possible e.g. to identify a landmark, or mark a meeting point.
2. It is really important that visitors more than 5 meters away don't miss the message attached to this beacon e.g. marking entry / exit from a zone, or helping someone find their seat.
3. There is a lot of potential interference in the immediate vicinity of this particular beacon, so you need a powerful signal to cut through the interference e.g. many solid dense objects are surrounding the beacon (yes, people count as “dense heavy objects.”)

If the above situations do not apply to you, then you can likely get away with a lower Tx Power setting. This will help save your battery, and overall make your system operations more efficient.

Beacon installation recommendations

Here are some best practices of using Tx Power to get the best possible performance and battery life.

1. If you need the signal range to be as wide as possible in a high traffic area, use the max power setting and install the beacon roughly 2 meters (6.5617 feet) up from the floor, to reduce interference from people walking by.
2. If you only need the beacon to be detected a few meters or several feet away (say to identify a product sitting on a shelf,) use a lower power setting.
3. Always be mindful of “line-of-sight” i.e. that in general, there are no objects in between a beacon and the customer’s smart device.
4. If getting “line-of-sight” is impossible or impractical, try to make sure these objects are not too heavy or dense e.g. thick steel panels or brick walls between a beacon and smart device.

Also, remember that the [battery in our beacons are replaceable](#). [More info about Transmission Power](#) can be found in our [Knowledge Base](#).

ADDITIONAL TIP:

Since beacons can have their Tx Power configured individually, it is easy to mix and match beacons with different Tx Power settings. This enables you to assign an optimal value to each beacon, based on the intended use of that beacon.



Beacon Interval

The Beacon Interval setting has a tremendous effect on your signal stability and battery life. Therefore, choosing the right setting is very important for ensuring a smoothly operating and efficient beacon network.

Let's go over the particulars.



Beacon Interval Basics

The “Interval” setting determines how often your beacon will transmit its advertising packet (i.e. basically telling nearby devices “I’m here!”) This is usually measured in milliseconds (ms), but can be measured in seconds (s) at the very highest interval ranges.

In practice, however, you’ll generally only be concerned with the differences in milliseconds. There are (as of this writing) not many applications where using a second-long intervals is particularly useful. This may of course change in the future.

Choosing the Right Interval Setting

The lower your Interval rate (smaller number), the more often the beacon will transmit its advertising packet. As you might expect, this has a dramatic effect on your battery life, as seen in the table below.

Interval	Tx Power	Expected Range	Expected Battery Life *
100ms	3 (-12 dBm)	35 m (115')	Up to 7 months
300ms (default)	3 (-12 dBm)	35 m (115')	Up to 2 years
1000ms or 1s	3 (-12 dBm)	35 m (115')	Up to 4 years

*Assuming optimal environmental conditions. Refers to the [Kontakt.io Beacon](#).



You will notice immediately from the table that the current standard for the iBeacon profile (i.e. the settings recommended by Apple) actually causes dramatically reduced battery life. In reality, though, there are many cases where such a low Interval is pretty unnecessary.

It is important to note that the 100 ms interval recommended by Apple is not required. Apple's justification for using such a setting is based on achieving high signal stability (arguable) and highly accurate indoor positioning (this is actually true.)

It's worth mentioning, though, that even setting up your beacons to transmit every 100 ms doesn't mean an iPhone can respond to the signal with the same speed. iPhone calls back every 1 second due to system limitations. Decreasing your interval won't change that; it'll only increase the chance that a device manages to grab one signal out of a few transmitted within this second. This might be helpful especially when your users are moving fast or there are many obstacles within a beacon range.

On Android devices, there are no such limitations. Theoretically, you could develop an app that grabs and reacts to the beacon signal in seconds. We don't recommend that, though, because it can cause a smartphone battery drainage.

How Interval affects signal stability

In our own lab analysis, we have determined that higher interval settings (over 700 ms) actually cause major issues with signal stability. However, we have also determined that using the 100 ms interval setting, recommended for iBeacon, is actually overkill for this issue.

We've determined that 350 ms offers the perfect balance between signal stability and battery life, which is why we use this setting as our default option.

How Interval affects indoor positioning accuracy

There are some cases where you do actually want to sacrifice your battery life for the sake of indoor positioning accuracy. How do you make the right choice for your needs? It all depends on how accurately you need to determine the positioning of a device interacting with your beacon network.

The interval can make quite a big difference to positioning accuracy. When positioning a person travelling at normal walking speed, there isn't much practical difference in accuracy between the 20 ms and 350 ms setting (our default).

However, you can also see that with an Interval of 1000 ms (or 1 second), the positioning accuracy drops substantially, and

you cannot tell precisely what path the customer was taking. This may or may not be a big deal for your use case, but it is important to understand the distinction.

When should you use a low Interval setting?

We define a low Interval setting as 100 ms or less. There are a few particular cases when using a very low Interval setting is useful (to the point of sacrificing battery life.)

- The object you are tracking tends to move quickly. For example, we have some cases with auto manufacturers who need to accurately track the location of a moving car.
- You need to know the object's location by inches or centimeters, rather than by feet or meters.

Recommended Interval settings

If neither of the above two points are applicable to your use case – then we highly recommend using an interval setting between 250 ms to 400 ms for the majority of deployments. Anything lower than this will quickly drain your battery, and anything higher may have performance issues due to signal instability. Remember that we use a default Interval of 350 ms, because we find it offers the best balance of performance and battery life in most cases.

But what about iBeacon?!

Yes, it is true that Apple recommends a 100 ms Interval for the iBeacon profile. Don't worry about it.

In practice, we find this setting to be very wasteful, and it adds unnecessary hassle when deploying any significant number of beacons. Save yourself the headache and use a higher interval setting.

The 100 ms recommendation from Apple is precisely that—a recommendation, not a requirement. We believe that Apple will eventually agree with us and change the recommendation to something more practical in the real world, and based on actual implementation experience.

Your beacons will still work with iOS devices just fine, regardless of your Interval setting. Actually, they'll work better in most cases, assuming you use our recommendations.

A photograph of two men sitting at a table in a meeting. The man on the left has a beard and glasses, wearing a blue button-down shirt. The man on the right is wearing a light blue t-shirt. They are both smiling and looking towards the right. In the background, there is a glass wall covered with many pink sticky notes.

Next steps

Now that you know what beacon settings are, how they affect your deployment, and how to adjust them to your needs, it's time for you to play around with them and set up your beacons.

You can configure your beacons individually or in bulk using our [Web Panel](#) and then deliver your new settings using our Admin App for [iOS](#) or [Android](#). Here are some articles to walk you through this process:

- [Configuring Beacons in the Kontakt.io Proximity Web Panel](#)
- [Synchronizing Beacon config changes \(iOS\)](#)
- [Synchronizing Beacon config changes \(Android\)](#)

Good luck!