

beacondo

USER GUIDE

Overview

What is Beacondo and why should you care?	5
How does it work?	5
Who can use Beacondo?	5
How much control do I have over the app?	5
How much does it cost?	6

Concepts

Apple Xcode 5	8
Installing Beacondo Designer	9
Working with projects	9
Your project folder	9
How your app fits together	10
Settings: basic app configuration	10
Screens: navigation layouts for your app	10
Pages: the core of your content	10
Navigating between screens, sections and pages	11
Releasing your app	11
About device sizes	11

Settings

Basic app settings	13
Default grid settings	13
Advanced app settings	14
Working with beacons	14
Limits on iBeacons	15

Screens

Working with screens	17
Adding a screen	17
Tabs vs menus	17
Screen buttons	18
Carousels and slideshows	18
Maps	18
Table rows	19
Text entry	19
Tips for screens	20

Pages

Sections vs Pages	22
Adding a section	22
Adding a page	22
Writing page content	23
Working with page templates	23

Actions

Making your app respond to the user and beacons	25
List of all actions you can use in Beacondo	25
Working with properties	27
Working with local beacons	28

Baskets

What are they used for?	30
Basic usage	30
Checking out with a basket	31
Checking out with a local beacon	31

Building

What happens on the build screen	33
How to test your builds	34
Testing with Beacondo Viewer	34
Testing with Xcode	34
Testing your iBeacons	35
If things don't work... ..	35
Configuring Xcode	36
Creating an icon and splash screen	36
Setting your app name	37
Configuring your iBeacon message and adding your license key	37
Miscellaneous settings, including device orientation	38
Submitting your app to Apple	38

Tips

Getting your image sizes right	42
Working with lots of iBeacons	42
Working with barcodes and QR codes	42
Working with iTunes Connect	42

Overview

Overview

What is Beacondo and why should you care?

Beacondo is a software development kit (SDK) that lets anyone build iPhone apps with no coding. At the heart of Beacondo lies our support for real-world interaction, which means we make great use of Apple's innovative new iBeacon technology so that your app can respond to a user's exact location – even down to knowing which department of a store they are in. We also add in the ability for you to let users scan barcodes and QR codes inside your store, and you get to decide what information is shown when they do so.

Beacondo also gives you a range of more familiar app functionality, such as scrolling tables, embedded movies and sounds, 3D carousels that users can flick through, and the all-important ability to post messages to Facebook and Twitter to let users do your marketing for you.

How does it work?

You know your brand, and you know your audience, but you might not have the skills or cash to pay a development studio to build you a custom app that can entice and inform your customers. That's where Beacondo comes in: our unique Beacondo Designer tool is an app that runs right on your Mac's desktop, and lets you construct your entire app with no coding skills.

That means you decide which images go in and what happens when they are tapped. It means you decide which products you list and how they look. And it means you decide what happens when a user walks into your store – do you want a message to appear? Do you want them to see the latest sales you have? It's your choice, thanks to Beacondo.

Who can use Beacondo?

Beacondo is a great way to build mobile apps for retail, but it's equally good for museums, coffee shops, hotels and anywhere users can benefit from getting location-specific information. You could, for example, place iBeacons around your museum and show guidebook information as they walk around – it's so easy to do that you'll be tempted to put iBeacons everywhere!

How much control do I have over the app?

One of the unique aspects of Beacondo is that you have *complete* control over the app. We give you an Xcode project, which is the same tool professional iOS developers use to make their own apps. You can then customize that project to your heart's content, and submit to Apple only when you're ready.

How much does it cost?

We have three license options, depending on your needs. You can start with our Basic license then upgrade to either Premium or Enterprise when you're ready. This helps take the risk out of app development – you can start for free, and upgrade only when it makes sense to do so.

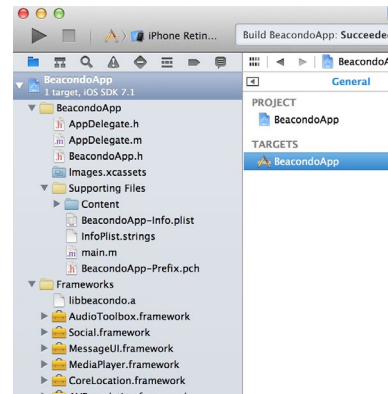
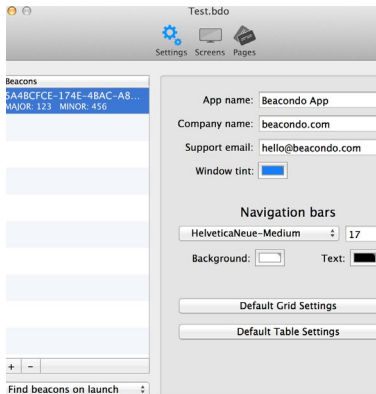


	<i>Basic</i>	<i>Premium</i>	<i>Enterprise</i>
Includes “Powered by Beacondo” watermark	×		
Up to 20 iBeacons	×	×	
Over 20 iBeacons			×
Customizable branding	×	×	×
3D carousels and slideshows	×	×	×
Scan barcodes and QR codes	×	×	×
Play sounds and movies	×	×	×
Post to Facebook and Twitter	×	×	×
Maps and KML support	×	×	×
Native performance	×	×	×
User baskets and wish lists	×	×	×
Local beacon configuration			×
Dynamic app configuration			×
Custom user properties (e.g. “Hello <name>!”)			×
Custom app notifications			×

Concepts

Concepts

There are three parts to Beacondo: Beacondo Designer, Beacondo Viewer, and Beacondo SDK. You'll be using all three in the process of making your app.



Beacondo Designer

Our creation tool is where you create screens, sections and pages for users to navigate around, drop in pictures where you want them, attach actions to respond to user interactivity, and tell your app which beacons it should look for.

Beacondo Viewer

Our testing tool lets you drop in apps for testing either using iTunes on your Mac or using a website. Viewer is also the easiest way to let others test your app remotely, because you can just share the URL to your app for them to download.

Beacondo SDK

Once you've built your app, our coding tools take over. We provide you with several pre-made Xcode projects for you to configure however fits your needs. You'll need to use this to build your finished app ready to go to the App Store.



Apple Xcode 5

When you're ready to submit your app to Apple for distribution on the App Store, you'll need to install Apple's Xcode development environment. This available for free from the Mac App Store, and contains everything you need to build and submit your Beacondo app to Apple. Beacondo supports Xcode 5.0 or later.

Installing Beacondo Designer

To begin with, the most important thing is to get up and running with Beacondo Designer, which is the tool that lets you create apps from your Mac OS X desktop. Beacondo Designer is compatible with OS X Mountain Lion (10.8) and Mavericks (10.9), but has no system requirements beyond that.

To install Beacondo Designer, go to www.beacondo.com in your web browser and click the Download link in the top-right corner of the site. You'll see three download options, and the one on the far left is for Beacondo Designer, so click that now. Your computer will download a zip file containing Beacondo Designer as well as a README file that explains any last-minute notices – double-click on the zip file to extract it, then look inside the folder to see the extracted files. We recommend you read the README before continuing.

You can run Beacondo Designer either from the folder you extracted it to, or by dragging that app into your Mac's Applications folder.

Working with projects

When you double-click on the Beacondo Designer icon, the app will launch and you'll see three buttons: Create Empty Project, Create From Template, and Open Existing. We recommend starting with the template to begin with, because it creates a simple app for you with some example pictures to jump start your own work. Once you have more experience, you should be able to start with the empty project option.

Regardless of whether you choose Empty or Template for your new project, you'll immediately be asked where you want to save your work. This is important, because Beacondo creates your project directory straight away so that you can copy images, movies, sound and HTML pages in there without waiting.

Your project folder

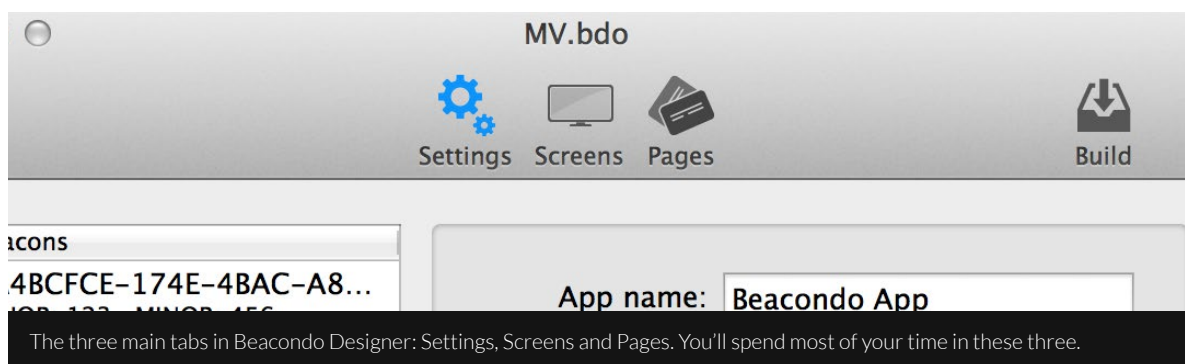
Every Beacondo project has a project folder, which is where all your assets should go when you want to use them. Once they are in the project folder, you'll be able to use them in your designs in Beacondo Designer. So, the first step in any Beacondo project is to open up the project folder and copy in a variety of assets that you intend to use – pictures, movies, sounds, and so on. You can add more later, of course – it's just a matter of dragging more files into your project folder as you need them.

When you build your app, Beacondo copies only the assets that you actually used. Any files not used are ignored when it comes to a build, so don't worry if your project folder gets big.

One last thing to remember is that you should put full-size images into your project folder, preferably without much compression. Beacondo resizes and compresses your images when you make a build, so it's best to keep the quality high until build time.

How your app fits together

A Beacondo app is made up of three distinct parts: settings, screens and pages. These are split into the three tabs you'll see at the top of the app editing window, shown below. To the right is the Build button, which we'll come onto later –that's where you finalise your app for real devices.



Settings: basic app configuration

The settings tab lets you accomplish three key tasks:

- Set basic information about your app, such as its name, its color scheme, and how you want the navigation bar to look – that's the bar across the top of each screen.
- Set defaults for items inside grids and tables. More advanced users might want to have custom designs for every grid and table that override these default settings, but most people will be happy just adjusting the defaults.
- Configure which iBeacons you want to listen for, and what should happen when they are triggered.

Screens: navigation layouts for your app

A screen in Beacondo is a list of layout items grouped together, presented in a view the user can scroll through and tap on to trigger actions. You get to decide what goes in each screen and in what order, as well as what buttons should appear on the left and right of the screen. You can add as many layout items as you want, choosing from four basic types: carousel, slideshow, row and map.

Pages: the core of your content

A page in Beacondo is a scrolling text area, optionally with a header image at the top. If you're making an app for retail, you'll want a page for each one of your products, and you'll see that you can attach values such as price (e.g. \$499) and EAN (a barcode number, e.g. 12345678) to each page. Pages can be grouped into sections (for example, "Groceries", "Pharmacy", etc), but you can have just one section if you want.

Navigating between screens, sections and pages

Beacondo gives you actions to direct users to specific screens, sections and pages inside your app. That's done by giving each screen, section and page a unique ID that you can reference elsewhere in Beacondo Designer. This ID can be something you type, but by default we use the title of the screen/section/page you're working on – as you type the title, Designer will fill in the ID for you.

In situations where you have multiple things with the same title, you should change the ID to be something unique. The reason for this is down to navigation: if you want to send your user to the page called Awesome Product and there are two pages with that same ID, Beacondo won't know which one you meant. These IDs are used everywhere you want to navigate to a particular screen, section or page, so the rule is simple: if the ID Beacondo gives you isn't unique and you want to link to that thing, change the ID.

Releasing your app

Once you've finished in Beacondo Designer and configured your app in Xcode, you need to send it off to Apple for review and ultimately distribution around the world to your users. This is done entirely through a system that Apple controls, called iTunes Connect, and you're going to need to become quite familiar with it because it's not something we can do for you. Using Apple's systems you will need to:

- Create a developer account for your company.
- Create a certificate, an app ID, and a provisioning profile
- Optionally register test devices with Apple so you can run your app on real iPhones before submission.
- Create your app on iTunes Connect, and upload screenshots and descriptions.

Apple has a large collection of documentation describing how to do all the above, but a much easier solution is to read the tutorial at <http://bit.ly/1e1vtMt> because it walks you through from start to finish. All of the above needs to be done only once, when you initial set up your app.

About device sizes

Beacondo supports iPhone 4, 4s, 5, 5s and 5c, which means it supports two screen sizes: 640x960 (iPhone 4 and 4s) and 640x1136

(iPhone 5, 5s and 5c). As you can see, both device families are 640 pixels wide, so when you're asked to set heights in your app you should make them relative to that width.

Settings

Settings

Basic app settings

The settings tab asks you to fill in a selection of basic app settings, most of which are required in order for the app to function properly. These are:

- App name: this is used inside your app if any message is shown to the user. It's also used inside Beacondo Viewer so you can see which test apps you have installed.
- Company name: this should be set to the name of the company the app belongs to.
- Support email: this is used as the default email recipient if you select to send a support email from inside the app, and should be set to the email address of someone who can respond to user questions.
- Window tint: this is the color that will be used for icons in tab bars and navigation bars.
- Status bar: whether you want it to have light text, dark text, or be hidden entirely.

You can also fill in the “Navigation bars” section of this tab, although it's pre-filled with system defaults so you can safely ignore it. If you want to customize your navigation bar, here is what the options do:

- The first two fields ask you to specify a font name and size in points. This defaults to 17pt HelveticaNeue-Medium, but iOS supports a variety of other fonts. The navigation bar in iOS is fixed at a certain height, so you shouldn't go too large with the size.
- The Background field sets the background color of the bar. This defaults to white.
- The Text field sets the text color inside bars, used when you have no title image. This defaults to black.

Default grid settings

NOTE: This sets the grid settings for all sections in your app. You can add custom settings for each section as you see fit, but any without custom settings will use these values.

Grid settings are used to show a sections' pages in a grid format. Beacondo fits as many items in each row as possible, based on the size you specify. There are seven options for a grid, all pre-filled with defaults.

- Layout: there are five layout variations for each item in your grid: “Image”, “Image, Title”, “Image, Price”, “Title, Image”, and “Title, Price”. Each of these are stacked vertically, meaning that if you choose “Image, Price” the thumbnail image for a page will appear first, with its price beneath it.
- Image and text height: how high the image and text (either Title or Price) should appear in the grid.
- Cell width: how wide each grid square should be.
- Cell spacing: how much spacing should be between each grid square.
- Title font and size: if you have selected to include either Title or Price in your layout, this is the font name and size that will be used to draw that text.
- Border color and width: this will draw a colored line around each page thumbnail image.
- Background: the background color of the grid.

Advanced app settings

There are a handful of advanced app settings that will be of interest to a subset of users. These settings are split into two types: basket and side menu. The basket options are:

User can adjust quantities: when this is checked, the user can swipe to the left on any item in their basket to see a Quantity button. When they tap this, they have the option to change the quantity of that item in their basket. Setting the quantity to zero will delete the item.

Show item quantities: when this is checked, the user will see “<Quantity> x <Product>” in their basket. For example, “1 x Jacket”. If your basket or wishlist doesn’t have quantities (e.g. if you’re making a tourism app) then you will want to turn this off.

Show prices and total: when this is checked, each item in the basket will have its price shown beneath its name, and a total will be shown at the end of the list. The total field will automatically be given a currency symbol based on the items in the basket.

The side menu options let you configure the appearance of the side menu, if you are using it. By default, if you have multiple screens with tab icons attached to them, Beacondo will arrange the screens as tabs across the bottom of the screen. If you would rather use the “Hamburger” options style list, you can go to the Screens menu and change “Show screens as tabs” to be “Show with left menu”. Any screen that has a tab icon attached will now be shown in a menu on the left. To activate the menu, you should either add a button with the action “Show screens menu” or the user can swipe from left to right anywhere in the app.

The side menu options let you configure the background color of the table, the background color of the item that is currently selected, as well as the text color.

Working with beacons

The left-hand side of the Settings tab is where you configure how your app should respond to the presence of iBeacons. At the bottom of the table is a popup menu where you can select whether you want your apps to start finding beacons as soon as it launches, or whether you want to trigger looking for beacons at a specific time. This setting is useful because the app needs to ask each user for permission to read their location, and if you choose “Find beacons on demand” it means you can explain to the user why you want access to their location before showing the request.

To add a new iBeacon, click the + button at the bottom of the beacons table then double-click the table row that gets added. This will bring up the Edit Beacon window, which contains a number of options that you will need to fill in:

- Identifier: every iBeacon has a unique identifier attached to it, which is a long series of letters and numbers. This identifier, often called a Universally Unique Identifier (UUID), must be entered exactly into the text field provided.

- Major and Minor: when you have reprogrammed several iBeacons to share the same UUID, you can distinguish between them by giving them different Major and Minor numbers. For example, major value could be used to distinguish between stores around the world, and minor value could be used to distinguish between different parts of each store. If you use -1 for both major and minor value, Beacondo will trigger your action if any beacon with the correct UUID comes into range.
- Max range: how close the user should be to the iBeacon for the app to trigger the action attached to the beacon. This applies only when the app is running – if the app isn't running, Beacondo will show a message on the screen as soon as the beacon is detected regardless of range.
- Action: this dictates what should happen when the user is close enough to the iBeacon with the UUID, Major value and Minor value that you specified.
- Prompt: if you type some text in here, this will be shown to the user before the action is executed. The user will be shown two buttons alongside the message: OK and Cancel. The action you attached to the beacon will be executed only if the user taps OK. If no prompt is set, the beacon's action will be executed immediately.
- Frequency: how often the beacon should be triggered. "Trigger every time the user enters range" means that if the user exits and enters the iBeacon region multiple times while running the app the action will be executed several times. The alternative, "Trigger once per app run," is more common, because it means users won't see the prompt until they terminate the app and come back again later.

Limits on iBeacons

There are two restrictions to using iBeacons. First, users of Basic and Premium can monitor no more than 20 iBeacon configurations, whereas users of Enterprise can monitor over 20 iBeacon configurations. For maximum

compatibility, the recommended way to configure iBeacons is to give them all the same UUID and identify them by Major and Minor values. Using this method, Beacondo Enterprise can track hundreds if not thousands of iBeacons around the world.

Screens

Screens

Working with screens

A screen in Beacondo is a collection of options shown to users. Screens scroll vertically as you add more information, so you don't need to worry about supporting the different heights of iPhone 4 and iPhone 5 – Beacondo will just scroll a bit further on iPhone 4. There are five types of item you can create:

- Carousels: these are horizontally scrolling 3D flipbooks that contain images. Users can swipe through the images until they find the one they want, and tap on it to trigger an action.
- Slideshows: these are created in the same way as carousels, but scrolling in a flat horizontal motion. Like carousels, slideshows can also trigger actions when tapped.
- Maps: this will embed a map into your screen, which can show the user's location and/or other locations you have specified in a KML file.
- Row: this is a standard table row, containing a picture, a title and a subtitle. You can attach actions to rows, and they will be triggered when the row is tapped.
- Text entry: this lets the users enter some text into your app. The text is stored for the entire app run, and may be sent to a server for processing using the "Submit form to URL" action.

You can place as many items as you want and they stack up vertically. Each can have their own height, although rows default to a height of zero – meaning they use the default table height from your app settings.

Adding a screen

To create a new screen, click the + button in the bottom-left corner of the window. Each screen has four options that you'll need to consider:

- ID: this should be a unique identifier for this screen so that you can link to it from elsewhere.
- Title: the text you want to appear in the navigation bar when this screen is being shown.
- Title Image: the image you want to appear in the navigation bar when this screen is being shown. If you set a title image, the title text won't appear.
- Tab icon: if you have at least two screens with tab icons attached, Beacondo will automatically create a tab bar for you, where the user can select which screen they want to see.

You can re-order screens by dragging them around in the table.

Tabs vs menus

When you have at least two screens with tab icons attached, the default behavior of Beacondo is to display the screens as tabs across the bottom of the screen. If you would rather use a left-hand options menu instead (sometimes called a "Hamburger" menu because of the icon commonly used for it), you can request this layout mode by going to the Screens view and changing "Show screens as tabs" in the bottom left to be "Show with left menu."

Screen buttons

Each screen can have a left and a right button attached to it, each of which can perform an action when they are tapped. You can always rely on the right-hand button being there, but if your screen is being presented from another screen then the left-hand button will be replaced by a “Back” button so that the user can return to the previous screen. A button can have a title or a title image (if the image exists, the text is not shown) as well as an action.

Carousels and slideshows

Carousels and slideshows work almost identically, so for the purpose of this documentation we’ll talk about “carousels” to mean them both. To add a new carousel, select a screen from the left-hand menu in the Screens tab, then click the + button towards the middle bottom of the window. You’ll be asked to choose what type of layout item you want to add, so please choose either Carousel or Slideshow.

When you add a carousel, a new row will appear in the “Items on this screen” table, saying “Carousel: 0 states”. That means you’ve created the carousel, but it doesn’t yet have any images attached to it. To attach some images, double-click the new row to bring up the Edit Carousel window. This screen asks you to set three options for the entire carousel: its height, whether to show it as a carousel or a slideshow, and whether you want to show paging dots. Paging dots can get in the way sometimes, but they do provide a great hint to users that they can swipe for more.

To add images to your carousel, click the + button in the Edit Carousel window. Each carousel state has three options: the image you want it to show, as well as its action and its action parameter. To get started, copy an image file into your project folder, then click the down arrow next to “Image” and choose the picture you just dragged in. You can use images in JPEG and PNG format, or movies in MOV or M4V format.

Maps

To add a new map, select a screen from the left-hand menu in the Screens tab, then click the + button towards the middle bottom of the window. You’ll be asked to choose what type of layout item you want to add, so please choose Map.

When you add a map, a new row will appear in the “Items on this screen” table, saying “Map.” To modify the way the map looks, double-click that row to bring up the Edit Map window. This screen asks you to enter various details about your map:

- Height: how height the map should appear on your screen. Maps are unique in that if they are the only item on your screen and you set this value to be 0 they will automatically take up the entire screen. This is great for when you want to show users lots of map data all at once.
- Map mode: this determines the appearance of the map. “Standard” gives you an illustrative line-art

style with street names, “Satellite” gives you a photographic map, and “Hybrid” blends the two.

- Locations: this lets you import an existing KML file of locations and regions. You will want to use this if you have lots of stores you want to display. This is created outside of Beacondo Designer.
- Show location: if you’re using a KML file, this instructs Beacondo to focus the user’s view on one or several of the locations in your KML file.
- Show user: you can choose whether to show the user’s location (the default), show the user’s location and follow as they move, or not show their location at all.
- Scrolling: when enabled, the user can scroll around the map freely.

Table rows

Table rows are the most basic way to show information to users. To add a new row, select a screen from the left-hand menu in the Screens tab, then click the + button towards the middle bottom of the window. You’ll be asked to choose what type of layout item you want to add, so please choose Table Row.

When you add a table row, a new item will appear in the “Items on this screen” table, saying “Row” and the current title of the row. To modify the way the row looks, double-click it to bring up the Edit Row window. This screen asks you to enter various details about your row:

- Height: if you want this row to have a specific height, enter it here. If you leave the number as 0, the row will inherit the row height used in your app’s default table settings.
- Title: the main title for this row, shown in a larger, bolder font.
- Subtitle: the subtitle for this row, shown beneath the title text.
- Thumbnail: a picture to appear on the left of the row.
- Background: a picture to appear behind the row. If you intend to support landscape orientation as well as portrait, you should add spacing to the left and right of your image so that it can be centered. In portrait, this will be 640 pixels wide, but in landscape either 960 or 1136 depending on the device.
- Action: define what should happen when the user taps the row.
- Arrow: when enabled, Beacondo will shown an arrow on the right edge of your table row, encouraging users to tap it.

Text entry

Text entries let users enter text into your app for you to then process as you see fit. Any text they type is preserved for the current app run (i.e., if they navigate away from the screen then return their text will still be there), but it will be cleared when they finish using the app so you should make sure you send it off somewhere beforehand.

To add a new text entry, select a screen from the left-hand menu in the Screens tab, then click the + button towards the middle bottom of the window. You’ll be asked to choose what type of layout item you want, so

please choose Text Entry.

When you add a text entry, a new item will appear in the “Items on this screen” table saying “Entry” and the current title of the entry. To modify the way the entry looks, double-click it to bring up the Edit Entry window. This screen asks you to enter various details about your text entry:

- Height: if you want this entry to have a specific height, enter it here. If you leave the number as 0, the entry will inherit the row height used in your app’s default table settings.
- Title width: once you have entered all your text entries, you should adjust the title width of each of them so that the text lines up. Your title will be right-aligned and the text input area will be left-aligned, so if you give your entries the same title width the input areas will all line up.
- Prompt: the text prompt that will appear to the left of the field. For example, “Username”.
- Field name: the name used for this entry when submitted to a server using the “Submit form to URL” action. This name is not visible to users. For example, “username”.
- Placeholder: the text that will be shown inside the input area when no text is present. For example, “Type your username here.”
- Initial text: the text that will be pre-filled in the input area. For example, “Bilbo Baggins.”
- Required: when checked, this means the “Submit form to URL” action will fail unless some text has been entered in this field.

Tips for screens

- Each item on your screen always occupies the full width of the screen. In portrait mode, that means things are 640 pixels wide, so if you’re using carousels you should ensure your images are at least that wide. If they are wider, Beacondo will scale them down at build time.
- If you wish to place a map onto a screen that has other items, we recommend you disable scrolling for the map. This is because the rest of the screen will also be scrollable, so it can confuse the user to have two scrollable things.
- You can have as many layout items as you want, but keep in mind that users don’t want to scroll around too much. It’s better to have multiple screens that link to each other rather than having one screen that tries to do everything.
- If you don’t want your application to have tabs, you should make sure that none of your screens have a tab icon set.
- If you want to fit in with the designs of other Apple apps, you should choose tab icons that use line art with thin lines. Beacondo will tint the icon according to the Window Tint option in your app’s settings. We can highly recommend buying a pre-made icon set that lets you select from existing artwork – you should check out www.glyphish.com as they are particularly good.

Pages

Pages

Sections vs Pages

Beacondo lets you create as many pages as you want, and you can jump straight to them from screens. However, organizing your pages into sections gives you the advantage of letting users browse through your pages section-by-section as if they were walking down an aisle in a grocery store.

Adding a section

To add a section, click the + button on the bottom-left of the Pages tab and choose Add Section. Your new section will be created and selected automatically, and you'll see the options to customize it on the right half of the window. They are:

- ID: this should be a unique identifier for this section so that you can link to it from elsewhere.
- Title: the text you want to appear in the navigation bar when this section is being shown.
- Title Image: the image you want to appear in the navigation bar when this section is being shown. If you set a title image, the title text won't appear.
- Grid Settings and Table Settings: if you want to give this section different settings from the defaults you configured in the Settings tab, click these buttons. You can customize neither, one or both depending on your needs, and you can revert back to default if you change your mind later.
- Right button: a button that will appear in the top-right corner when this section is being shown.

You'll notice there is no left button option for sections as there is for screens. This is because sections must always be shown from elsewhere in your app, so the left button area is taken up by the Back button that lets users navigate to the previous screen.

Adding a page

To add a page, first select which section you want it to go into, then click the + button in the bottom-left corner and choose Add Page. Your new page will be created and selected automatically, and you'll see the options to customize it on the right half of the window. For products, the customization is split into two parts, so you'll see two headings at the top: layout and content. Here's what's in the layout screen:

- ID: this should be a unique identifier for this screen so that you can link to it from elsewhere.
- Title: the text you want to appear in the navigation bar when this screen is being shown.
- EAN: if this is a product with a barcode, enter its barcode number here.
- Price: if this is a product with a price, enter that price here, including currency symbol.
- Title Image: the image you want to appear in the navigation bar when this screen is being shown. If you set a title image, the title text won't appear.
- Thumbnail: a small picture illustrating the page when shown inside a section view – i.e., a table or grid.
- Header: this can be a carousel or slideshow. Click the button to customize its states.
- Right button: a button that will appear in the top-right corner when this section is being shown.

Writing page content

Beacondo gives you three ways to deliver page content: using inline text that you've typed straight into Designer, by loading HTML files from your computer, or by loading web pages that are online. The last one is self-explanatory, so we're going to focus on the first two. There are advantages to both of them:

Typing straight into Beacondo Designer means:

- You don't have to worry about external tools.
- You can use keyboard shortcuts such as Cmd+B, Cmd+I and Cmd+U to make text bold, italic and underline like you do in other apps.

Creating a separate file and linking it means:

- You can have a team of people creating different content files simultaneously.
- You can use dedicated HTML creation tools that you know and trust.
- You can use as much advanced HTML as you want.

Beginners are definitely best off typing straight into the text box.

Beacondo uses HTML for page content because it's an industry standard way to display formatted text, and by blending it with the native code carousel in the header of pages you'll get the best of both worlds.

Working with page templates

Page templates let you achieve consistent designs across your pages without having to copy and paste large amounts of HTML. If you start with the built-in project, you'll get a basic page template called simply pageTemplate.html. If you want to customize this template or make other ones for different kinds of pages, we recommend you copy pageTemplate.html and use it as the basis for other templates.

The relationship between page content and page templates is simple: a page template can contain any amount of HTML that you want, as well as JavaScript and CSS, but it looks for a special marker: `<!--CONTENT-->`. When Beacondo loads a page, it dynamically replaces that `<!--CONTENT-->` line with the content for the current page. This means you shouldn't include any HTML header and footer data in your page HTML, because that should all be provided by your template. Of course, there might be some pages where you want everything to be custom, so in that case just set an empty page template and put all your HTML into the page file.

Actions

Actions

Making your app respond to the user and beacons

We put actions into their own sections because they can be used in several places:

- When your user walks into range of a beacon you have configured.
- When they tap the left or right button of a screen.
- When they tap the right button of a page.
- When they tap an image inside a carousel or slideshow in a screen or page.
- When they tap a table row in a screen.

Regardless of where the action is triggered, the set of actions you'll see don't change. However, two actions can be used only in specific circumstances: "Switch section style" can be used only when attached to a right button in a section, and "Add single item to basket" and "Add multiple items" to basket" can be used only when attached to a right button in a page.

The "Post app notification" action can be used only by Beacondo Enterprise licensees.

List of all actions you can use in Beacondo

- Show page: this will show the page that has the ID specified in the action parameter. You might want to trigger this if you have written special pages that describe basic app information, such as welcome text or special offers.
- Show screen: this will show the screen that has the ID specified in the action parameter. This is used to navigate between screens, so for example your main screen might contain a menu of other screens the user can navigate to.
- Show section (table): this will show the section that has the ID specified in the action parameter, and will show it in table format.
- Show section (grid): this is the same "Show section (table)" except it shows the section in grid layout.
- Switch section style: this action is applicable only to the right button of sections, and will automatically switch a section between grid and table layout. This lets the user choose which style they prefer.
- Show screens menu: if you enabled "Show with left menu" on the Screens menu, this action will display the left-hand menu showing available screens.
- Add single item to basket: this adds the current page to the basket. The current page must have an EAN attached to it, even if it's just one you've made up. If you put text into the parameter for this action, that text will be shown to the user as confirmation that the item was added.
- Add multiple items to basket: this prompts the user to enter how many of the current item they wish to add to their basket. The current page must have an EAN attached to it, even if it's just one you've made up. If you put text into the parameter for this action, that text will be shown to the user when they are asked to enter a quantity.
- Show current basket: this loads a section in table style showing the user's current basket. You must first have created a section that you intend to use for your basket, and configured it as you want. You should then use the ID of that section with the "Show current basket" action to bring up the basket. Do

not use “Show section” for showing baskets, because the user will not see their basket’s contents.

- Send basket to URL: if you specify a website URL here, that will be loaded inside the internal web browser along with the contents of the basket. The basket will be sent to you in an HTTP GET request by appending ? to your URL then a JSON-encoded dictionary of the basket under the name “basket”. This dictionary will contain the item’s EAN and the quantity the user added.
- Clear basket: this removes all items from the basket.
- Share with email: this will bring up an email message pre-filled with the message specified in the action parameter. The user can then address the email to whomever they wish, and send it.
- Share with Facebook: this will bring up a text area where the user can write a Facebook status message, pre-filled with the message specified in the action parameter. The user can amend the text before submitting it to Facebook.
- Share with Twitter: this will bring up a text area where the user can write a tweet, pre-filled with the message specified in the action parameter. The user can amend the text before submitting it to Twitter.
- Display message: this will show a message box on the screen, pre-filled with the text specified in the action parameter. The user can tap OK to clear the message.
- Open website (internal): this will open an internal web browser window pointing at the website specified in the action parameter.
- Open website (external): this will open an external web browser window pointing at the website specified in the action parameter.
- Play local sound: this will play a sound on the current screen, playing the sound file specified in the action parameter. You should use either MP3, M4A or WAV file formats. If the user navigates away from the screen, the sound will stop. If you trigger another “Play local sound” action with the same parameter, the sound will be stopped.
- Play global sound: this works identically to “Play local sound” except the sound will not stop when the user leaves the current screen.
- Play movie: this will play a full-screen movie as specified in the action parameter. You should use a M4V or MOV movie, and keep the resolution fairly low – the iPhone is capable of showing a maximum of 1136x640 pixels in landscape mode, so if you try to feed it 1080p you’re just wasting data.
- Post app notification: this will send an NSNotification with the name specified in the action parameter. This is for advanced users only, and requires custom development to catch the notification. If you specify {\$BASKET} as your parameter, the notification that you’re sent will be named BeacondoBasket and will be a dictionary containing the current user’s basket. Please note: this feature is available only in Beacondo Enterprise.
- Submit form to URL: this will open an internal web browser pointing at the URL you specify in the parameter field. The URL will be sent as an HTTP GET request by appending ? to your URL then URL-encoded values from whatever input fields are on the current screen. The “field name” property of each input field is used, and will be sent to you as individual parts of the request. For example, www.yoursite.com?foo=bar&baz=wombat.
- Scan barcode / QR code: this will bring up a barcode scanning camera window that will recognize any barcode or QR code. When a valid code is found, the camera window will close and Beacondo will search for and show any product that has the EAN found in the code.

- Show page search: this will show a window where the user can type search criteria and see a list of matching pages. Beacondo searches the title of each product to match against what the user has typed, and automatically groups the results into the sections you have created.
- Start finding beacons: if you have set your beacon mode to be “Find beacons on demand” this will trigger Beacondo to start looking for iBeacons.
- Stop finding beacons: this will force Beacondo to stop looking for iBeacons. This lets users opt out of location monitoring if they find it annoying.
- Show beacon distance: this shows an approximate distance between the user and the iBeacon you specify. Please note that this is extremely approximate, which is why no measurements are shown. Instead, you get a color from blue (far away) and red (very near) plus a text description. The reason the distance is so approximate is because radio noise (lots of devices with mobile phone signals, WiFi, Bluetooth and such) and physical obstacles (walls, people, etc) make it impossible to measure distance with any precision. You should set the parameter of this action to be the beacon UUID, its major value and its minor value, separated by commas. For example: **5A4BCFCE-174E-4BAC-A814-092E77F6B7E5, 123, 456**.

Working with properties

Enterprise licensees can take advantage of an advanced text replacement system called properties. These properties are set remotely using a web server, but stored locally so that they can be used anywhere in the app – in screen titles, in row text, in page text that is loaded from a file or inline text, and even inside action parameters. Properties are stored permanently attached to your app, but will be deleted if the user deletes the app. Please note: this feature is available only to enterprise licensees.

Properties are set using special HTML tags that are read in by the internal web browser. It is important that you understand you must use the internal web browser to set properties, because they cannot be read from the external web browser.

You can set as many properties as you want in a given a web page. Each one should look like this:

```
<beacondoProperty>foo=bar</beacondoProperty>
```

That line of HTML can appear anywhere in your HTML document, but it’s probably best to put it inside your <head> element. Depending on your site’s layout and design, you may find your property setting appears in the output of your page. If this happens, you should use something like this:

```
<beacondoProperty style="display: none;">foo=bar</beacondoProperty>
```

Using `foo=bar` will make Beacondo replace `{foo}` with `bar` everywhere it appears. For example, you could use a table row to say, **Hello {name}** and have that replaced with **Hello Bob** if the `$name` property was set. If a property is used without being set, it will be empty. In the example above, it would read **Hello**.

Working with local beacons

Local beacons are one of the most powerful and flexible technologies inside Beacondo, and bring with them a whole world of extra technology possibilities. Each user can have one local beacon attached to their phone, and when it's configured their phone becomes an iBeacon in its own right, sending out its own unique ID for other devices to track. This means you can, for example:

- Monitor the location of customers in your store.
- Check school attendance at a glance.
- Trigger actions when you user touches their phone against one of your devices.
- Let staff locate users who have requested help, or a waiter attend a table where a diner wants to order.

What's more, you can configure and reconfigure your local beacon remotely, which means you can have a user enter text in your app, submit that form to a URL, and set a local beacon on their device as a result of that form – for example, once they've checked out with a basket of items.

Please note: local beacons are available only to enterprise licensees.

Configuring your user's local beacon is done entirely using the internal web browser, by placing special HTML elements that are read in and activated. You must not use the external web browser, because the special elements cannot be read.

To create a local beacon, add these three tags to a web page that is loaded using the internal web browser:

```
<beacondoLocalUUID>5A4BCFCE-174E-4BAC-A814-092E77F6B7E5</beacondoLocalUUID>  
<beacondoLocalMajor>123</beacondoLocalMajor>  
<beacondoLocalMinor>456</beacondoLocalMinor>  
<beacondoLocalDuration>permanent</beacondoLocalDuration>
```

You should replace the values with whatever you want for that user. The UUID should be the correct length and format – the **uuidgen** tool on Mac OS X can generate one for you if you want. The major and minor values should be whole numbers between 0 and 65536, so with 65536 x 65536 unique combinations to choose from that's over 4 billion unique devices able to use the same UUID.

The **beacondoLocalDuration** value can be either "permanent" or "temporary". A temporary beacon will transmit only while the app is running this time; if the user quits the app and relaunches, the beacon will be deleted. A permanent beacon will restart when the app is run subsequent times. Regardless of whether you choose permanent or temporary, the beacon can transmit only while the app is actively running.

Depending on the layout and design of your site, the three local beacon HTML elements may appear in your page's layout. If this happens, you should use **style="display: none;"** to ensure they are hidden. For example, **<beacondoLocalMajor style="display: none;">123</beacondoLocalMajor>**.

Baskets

Baskets

What are they used for?

Beacondo baskets have been designed to be flexible enough for all kinds of purchases – not just a simple shopping basket like you might see in a store, although clearly that’s possible too. For example, it could be used to build wish list of books to read or a collection of favorite sights to visit, or indeed any grouping where you want users to be able to pages to a list and view that list later. For ease of description we refer to these customizable lists as baskets.

The way your basket is displayed is customized by creating a section inside the standard Pages tab, but without any pages inside it – Beacondo automatically adds pages to the section for you. When you want to show someone their basket, you should use the “Show current basket” action and specify the ID of the section design that you want to show. You should *not* use “Show section”, because that will not cause Beacondo to add pages to the section.

Basic usage

To make your basket work, you should follow these steps:

- From the Pages tab, create a new section. Call it “Basket” or “Wishlist” or whatever you please, as long as it has a unique ID.
- Add a custom title image, configure its right button, or adjust its table settings as you see fit.
- Elsewhere in your app, for example on your main menu, add a button that has the action “Show current basket” and set its parameter to be the unique ID from step 1.
- On pages that you want users to be able to add to their basket, set the right button to be either “Add single item to basket” or “Add multiple items to basket.”
- If you choose “Add single item to basket”, any text you put into the parameter box will be shown to the user as confirmation that the item was added.
- If you choose “Add multiple items to basket”, any text you put into the parameter box will be shown to the user when they are asked how many items they want to add.
- Make sure you attach an EAN to each of the pages that can be added to a basket. This EAN could be a real barcode number for the item or it could be some text you made up, but it does need to identify the page uniquely. For example, “The Whitehouse” is a good enough EAN if you’re making a tourist app for Washington DC.

That’s it! Users can now go ahead and use your app, add things to the basket as they see fit, then view and edit the basket at their leisure. When they tap to view their basket, the settings from the section you created will be used – this is where you should put an “Empty basket” button if you want one.

On the Settings tab there are two configuration options for your basket that affect the way it looks and behaves. When “User can adjust quantities” is checked, it means the user can swipe to the left on any item on their basket and adjust its quantity. When “Show prices and total” is checked, it means that users will see how much each item costs as well as a total for their basket at the end of the list.

Checking out with a basket

Once your basket is ready to go, you have two options for checking out: sending it to a website for processing or posting it as an app notification and catching it with your own code.

To use the former option, set the right button of your basket screen to have the action “Send basket to URL” and provide your web server’s URL as the parameter. The basket will be sent to you in an HTTP GET request by appending ? to your URL then a JSON-encoded dictionary of the basket under the name “basket”. This dictionary will contain the item’s EAN and the quantity the user added, and you can process the order.

To use the latter option, set the right button of your basket screen to have the action “Post app notification” and use the parameter **{BASKET}**. When Beacondo detects this, it will create a dictionary out of the basket with EAN as key and quantity as value, then post it to the app with the notification name **BeacondoBasket**. You should write code to catch this notification and complete the transaction. Please note: this feature is available only to Enterprise licensees.

Checking out with a local beacon

Enterprise licensees can take advantage of the local beacon system in Beacondo to do much more advanced check out and delivery. A local beacon means that your user’s device becomes a beacon too, transmitting its own signal that other users and devices can find. This lets you do two things once your user has checked out:

1. Use a “touch to authenticate” system whereby the user’s beacon must be touched against another device to authenticate that user. This might be where you show that user’s order so that you can collect payment, or perhaps you’re using the local beacon to identify that someone has attended class.
2. Use a “find this beacon” system whereby the user’s beacon broadcasts its existence, and one of your store assistants can collect their order and deliver it to that user by locating the beacon.

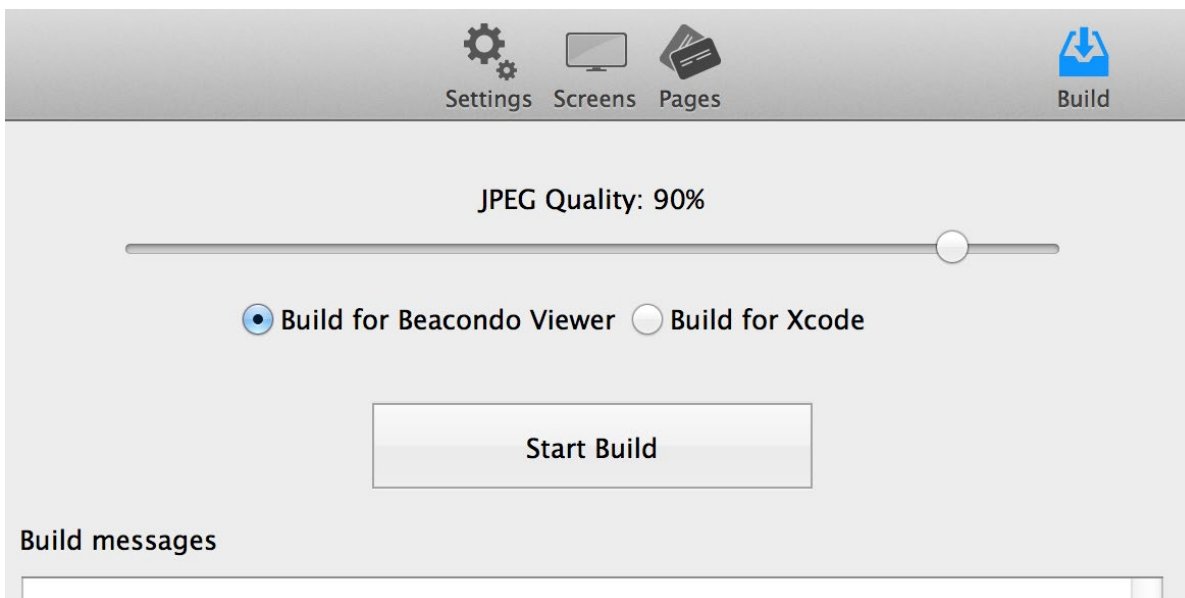
For more information about local beacons, see “Working with local beacons” under “Actions”.

Building

Building

What happens on the build screen

Up to this point, you've been working in Designer to configure your settings, layout your screens, and enter in your products, and you've hopefully been using high-resolution versions of your artwork so that you can make changes without having to redo work. The next step is to test on a real device, and that's where the build screen comes in: you choose what level of JPEG compression you want, then decide whether you want to test in Beacondo Viewer or in your own Xcode project, and finally hit the button marked Start Build.



During a build process, Beacondo Designer resizes your images to the correct size as specified in your layouts, to make sure they take up as little space as possible. When you've used JPEGs for images, it will also compress those JPEGs to the quality you set on the Build screen – PNG files will be resized down to fit the screen but not compressed, and movies will be copied as-is from your filesystem into the build.

What happens when you press the Start Build button depends on which build option you chose:

- If you chose Build for Beacondo Viewer then a zip file will be created, so you'll be asked to choose where to save that zip file.
- If you chose Build for Xcode, you should have downloaded the Beacondo SDK project from beacondo.com and extracted it somewhere. When you press Start Build, you should look for the Content folder inside the app template you want to use. For most people, this will be either be BeacondoApp or BeacondoApp-No-iBeacons.

As the build happens, any warnings or errors will be written to the Build Messages window; you should read these carefully, because it will report things like missing images that need to be corrected.

How to test your builds

Testing your app thoroughly is key to making sure it does what you planned, and that there aren't any typos lurking on your pages. There are two ways to test, but most people will want to use both because they serve different purposes.

Testing with Beacondo Viewer means:

- You can just drag zip files onto your iPhone using iTunes.
- You can distribute the test app to anyone in the world by emailing your zip or putting it onto a web server where they can download it.
- You don't need to worry about installing Xcode yet.

Testing with Xcode means:

- You get the "real" experience of having your own app.
- If you have in-house coders, you can add your own code that augments your Beacondo app.
- You're most of the way there to sending your app to Apple.

The usual process is to test with Beacondo Viewer several times until you're happy with your content, then start testing with Xcode as you approach a final build. The actual submission to Apple does require you to use Xcode, so there's no harm getting familiar with it.

Testing with Beacondo Viewer

To test your app with Beacondo Viewer, first install Beacondo Viewer from the App Store. You can either search for Beacondo or just follow the link on beacondo.com. You should then do a build using Beacondo Designer, with the "Build for Beacondo Viewer" option set – this will output a zip file.

Once you have your app zip file, you should connect your iPhone to your Mac then launch iTunes on your Mac. Near the top-right of the iTunes window will be a button that says "iPhone" – select that, then choose the Apps tab and scroll down until you see Beacondo. You can now drag your built app zip into Beacondo. Once the copy has finished, switch to your iPhone, launch Beacondo Viewer, tap the + button in the top-right then tap "Copy from iTunes."

When you're testing in Beacondo Viewer, a back button will automatically be added to the top-level screens in your app. This won't appear when you build your own app, but we need it in Beacondo Viewer so that you can get back to the main list of installed apps without having to quit each time.

Testing with Xcode

When you test with Xcode, it means loading up one of the sample SDK apps from beacondo.com and replacing its Content with yours. We provide several examples apps for you to test with, but the one most people will want to use is called BeacondoApp – it's an empty shell that just needs to be filled with your

content in order to be ready to ship to Apple. *Please note: if you do not intend to use iBeacons in your app, you should use the BeacondoApp-No-iBeacons project.* You can of course make as many changes to the code as you want, but to begin with it's best just to make sure that everything works.

When you're doing a build, choose the "Build for Xcode" option and click Start Build. You'll be asked to select where your Content directory is, so browse to the example SDK project you want to use and look for its Content folder. All the files necessary to run your app will be copied into that directory.

The next step is to look for the file BeacondoApp.xcodeproj, which will be in the root directory of the example you chose. When you double-click this it will launch Xcode, and you should be able to just click the play button in the top-left corner of Xcode to build your iOS app. By default, that button builds for iPhone Simulator, which comes with Xcode and lets you test on a virtual iPhone. It might take up to 30 seconds for Xcode to build and run the app the first time, particularly because it takes some time to launch iPhone Simulator. But don't worry, it's faster next time - you don't need to close and re-launch iPhone Simulator each time, just click Stop and Play in Xcode.

Working in iPhone Simulator is almost identical to working on a real device, which means you can see exactly how your app will look to real users. However, there is one big difference: the iPhone Simulator does not support iBeacons, which means you can only test that your iBeacon actions are working using a real device. The simulator does support location tracking on maps, but you need to enable it - select the Simulator window, then look in the menu bar at the top for Debug > Location and enter a test location.

Testing your iBeacons

If you're using iBeacons you'll want to ensure that everything has been configured correctly, and the easiest way to do that is by using a second iPhone. On the first iPhone you'll run your app as built using Xcode or using Beacondo Viewer, and on the second iPhone you should install an app called Locate for iBeacon - it's available from the App Store, and is free.

Locate for iBeacon lets an iPhone pretend to be an iBeacon, and you can configure it to match the settings you had configured in your app. This will let you trigger your iBeacon actions in a test environment to make sure they all work correctly.

If things don't work...

If your iBeacon isn't working correctly, the first thing to do is make sure you have the UUID set correctly, and that the Major and Minor values are either set correctly or set to -1. If you've checked that, next check your phone: is

Bluetooth turned on? Finally, try restarting your iPhone - sometimes if you're testing and going in and out of test beacons a lot it can confuse your phone, so restarting it is likely to clear the problem. If you're still having problems, please email us at hello@beacondo.com.

Configuring Xcode

Once you've tested your app in Beacondo Viewer and have followed the instructions under "Testing with Xcode", you're almost ready to ship your app to Apple for distribution on the App Store.

Before you do that, there are a few admin tasks you'll want to get done, and we're going to cover them now. These tasks are:

- Creating your own app icon.
- Creating your own splash screen for when the app launches.
- Setting your app name that should appear on the home screen.
- If you're using iBeacons, what message should appear when the user comes near one of your beacons and the app isn't running.
- If you're using Beacondo Premium or Enterprise, you'll need to enter your license key.
- Miscellaneous settings, including support orientations.

The Beacondo SDK provides a number of different types of projects, but the one you'll want to use is either called `BeacondoApp` (if you intend to use iBeacons) or `BeacondoApp-No-iBeacons` (if you don't intend to use iBeacons) – the others are examples demonstrating how you can customize the Beacondo project to do more if you have iOS developers who can help.

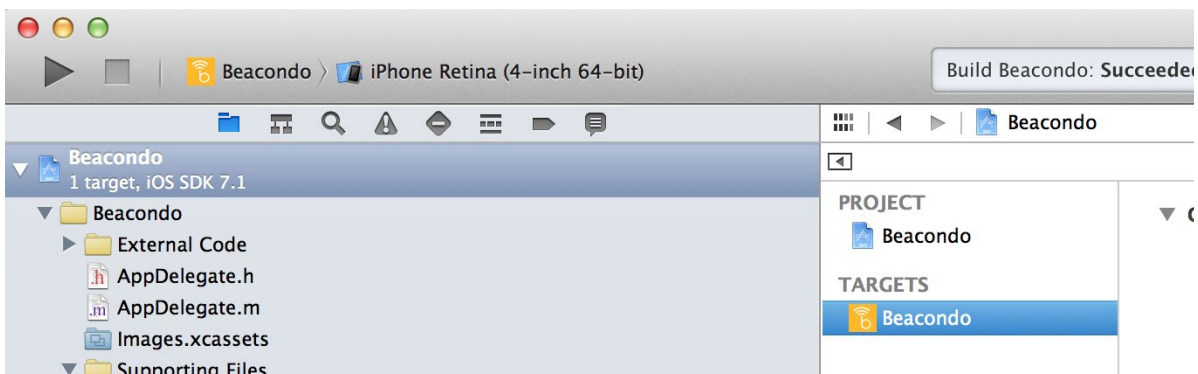
Creating an icon and splash screen

You need to produce three special assets, then copy them into Xcode. The first is the icon to appear on the iPhone home screen, the second is a splash screen for iPhone 4 users, and the third is a splash screen for iPhone 5 users. We've provided example files for you, and the best way to create each of these assets is to open them in your graphics editor of choice and draw over them – this means you'll be sure to get the size exactly correct.

Once you've created these three files, look in Xcode for the file `Images.xcassets`, and select it. In the middle of the screen you'll see two options: `AppIcon` and `LaunchImage`. Choose `AppIcon`, then drag your application icon to the area marked "iPhone App iOS 7 60pt" – our example app icon will be there, so you can just drag it onto there. Now click `LaunchImage`, and drag your iPhone 4 screen to the slot marked "2x" and your iPhone 5 screen to the slot marked "R4" – that means "retina 4-inch," which is what Apple calls the iPhone 5-sized device family.

Setting your app name

Apps need a name on the iPhone home screen. Apple requires it to be too similar to your app's name in the App Store so that users aren't confused. To set the app name, select the first entry in the top-left of the project view. The name of this will vary depending on which example project you chose, but in the picture below it's where it says "Beacondo" then "1 target, iOS SDK 7.1". When you choose that, in the middle of the screen you'll see "Project Beacondo" and "Targets Beacondo" (on the right of the picture below). Choose the "Beacondo" listed under Targets. This will reveal tabs to the right: General, Capabilities, Info, and so on.



To set your app name, click on the Info tab, look for the row marked "Bundle display name" and change it to be whatever you wish to call your app. Please note that if the name is too longer, your iPhone will shorten it so that it fits on one line, and replace the end with "...", which most people try to avoid. *Please note: don't change "Bundle name" – you're looking to change "Bundle display name", which is quite different!*

Configuring your iBeacon message and adding your license key

Also with the Info tab is where you can configure what message should be shown to the user if the iPhone picks up one of your beacons while the app isn't running, and also where you should add your Beacondo license key if you've upgraded to Premium or Enterprise.

First, your iBeacon message: this is in the same location where you set the Bundle Display Name in the previous section, but this time you're looking for the line `BeacondoAlertMessage`. By default it reads, "Swipe to unlock your phone and launch our app!" but you can change that to any text you want. When the user is first asked for location permission, the text in "Privacy - Location Usage Description" is used –again, you can change that to something specific to your app.

Second, your Beacondo license key: to set this, look for the lines `BeacondoLicenseName` and `BeacondoLicenseKey`, which are set to `YOUR LICENSE NAME HERE` and `YOUR LICENSE KEY HERE` by default respectively. You should change these to the license name and key you were provided when you upgraded your license.

Miscellaneous settings, including device orientation

In the previous two sections we've been working inside the Info tab, but now you need to change to the General tab. This provides three options you're likely to want to change:

- Bundle identifier: this should match whatever you have configured on iTunes Connect.
- Version: this is 1.0 by default, but if you want to put out app updates you should make sure you increase this to match whatever you listed on iTunes Connect.
- Device orientation: Portrait, Landscape Left and Landscape right are enabled by default, which is the standard for iPhone. To change the supported orientations just check or uncheck these boxes.

Submitting your app to Apple

By this point you should have built your Beacondo app, configured and tested in Xcode, and followed Apple's instructions to set your app up in iTunes Connect. All that remains is to make a final build of your app and submit to Apple, at which point there's usually a wait of about seven days before Apple approve your app for distribution to users.

The first step in the process is to log in to iTunes Connect, select your app, then add some review notes for your app. You'll find this option towards the bottom of the top right of the app detail screen – that's the one where you can edit its description and screenshots. These review notes are optional, but if you don't fill them in correctly Apple may reject your app and ask you a barrage of questions, so it's best to avoid that delay and just tell them what they want to know.

The most important thing to tell Apple is about your usage of iBeacons. The basic Beacondo app ships with two versions: BeacondoApp and BeacondoApp-No-iBeacons. If you are not using iBeacons in your app, you should use the No-iBeacons version of the project, because that does not include the code related to iBeacons. The reason for this is because of these review notes: if you have used the iBeacon-enabled version of the library, Apple will ask you how you're using iBeacons. This applies even if you haven't configured Beacondo to look for any beacons – the fact that the code still exists in Beacondo is enough for Apple to reject your app unless you explain why you're using them.

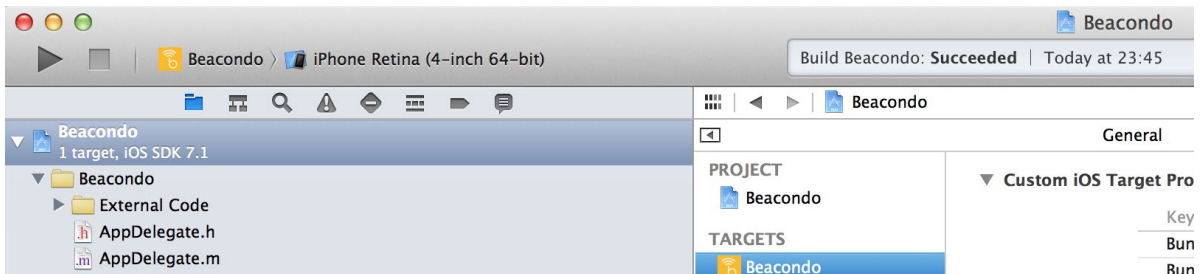
So: if you're not using iBeacons, make sure you've used the No-iBeacons version of the project. If you have used iBeacons, you should have used the normal BeacondoApp project, and you'll need to add the following to your review notes: "This app uses iBeacons to detect the users location in physical locations, and respond with messages." If you want to be specific about example locations (e.g. "Starbucks coffee shops") that always helps, and Apple usually like to have an example iBeacon UUID to work with so they can be sure it's all correct and working as you intend.

It's not fun to jump through these hoops, but Apple does focus very strongly on the security of user's information and a smooth app user experience, so they do like to be very sure that apps work as advertised.

Still in iTunes Connect, click the button marked “Ready to Upload Binary.” You’ll be asked:


- Whether you have any encryption in your app: the answer is YES.
- Whether your app qualifies for encryption exemptions: the answer is YES.
- Whether you are using third-party content in your app: if you’re using all your own content, the answer is NO. If you’re using content the answer is YES, and you’ll also need to select YES that you have a license for that content.

When you click Save, your app will enter the state “Waiting for upload.” You should now transfer to Xcode, and make an archived build of your app. This is a little different to the test builds you’ve done so far, because it tells Xcode to produce something that’s as efficient as possible rather than something that can easily be tested. It also wraps it up in a distributable form that will be validated against Apple’s guidelines then shipped off to Apple for a full review. To kick things off, look in the toolbar to the right of the play and stop buttons – you’ll see the name of your project (“Beacondo”, “BeacondoApp” or one of the other example project names) and to the right of that something like “iPhone Retina (4-inch 64-bit).” You can see this in the screenshot below:



Click the “iPhone Retina (4-inch 64-bit) button and a pop menu of alternatives will appear. Scroll to the top of that list and choose “iOS Device” as shown at the top of the screenshot below:





When that's done, go to the Product menu and choose Archive to produce the final build of your app. When that finishes a new window will appear with Validate and Distribute buttons on the right-hand side. You should click Validate and follow the on-screen instructions, then, when it comes back all clear click Distribute and follow the on-screen instructions.

When this process finishes, your app will have been sent to Apple for review, so you now just need to sit back and wait – they always wait at least a week before even looking at apps, so if you need to reject and resubmit your app you have plenty of time.

Tips

Tips

Getting your image sizes right

If you're placing icons into tab bars and navigation bars, you will want to make sure you get the right image sizes. Beacondo Designer comes with a template app that gives you a selection of icons all sized correctly for use, so the best thing to do is create a template and use our example icons as the basis for your own. As a rule, icons should be no larger than 56x56.

As a reminder, the iPhone 4 family has a screen size of 640x960, and the iPhone 5 family has a screen size of 640x1136. All the layout items in Beacondo screens are the full width of the iPhone, which means you can rely on them being 640 wide by whatever height you set.

Working with lots of iBeacons

If you intend to use lots of iBeacons, the recommended method is to give them all the same UUID and distinguish between them using major and minor values. The Basic and Premium Beacondo licenses let you track up to 20 beacons this way, and the Enterprise license lets you track as many as you like – just set unique major and minor values as needed, and you can identify individual stores or even particular departments inside individual stores.

Working with barcodes and QR codes

If your app is for retail, it's easy to see that barcodes are very useful – just enter the EAN for each product into your pages as you make them, then add a button somewhere that lets users scan barcodes. This will allow them to go up to any product in your store and scan its barcode to see more information in your app.

If you aren't building an app for retail, you can use QR codes to hijack the EAN system to show information in much the same way. Whenever Beacondo finds a code from a barcode or QR code, it immediately looks for pages that have an identical EAN set. This means that you can put anything in that QR code, and as long as it matches the EAN you typed in Beacondo will match it. For example, if you're creating an app for a museum, you could create a QR code containing the text "welcome" and set your welcome page to have the EAN "welcome" so that it is shown when the user scans that QR code.

Working with iTunes Connect

Although it can feel quite complicated at first, Apple does produce comprehensive documentation for iTunes Connect to help you get started. And, best of all, you only need to do it once – app updates much easier! There are two things you ought to keep in mind:

- Make sure you state clearly what features you use in your app review notes. If you are using iBeacons, you need to say so otherwise Apple might reject your app.
- Make sure the icon you put on iTunes Connect is similar if not identical to the icon you put into Xcode. Apple does not want users to be confused by two very different icons.