



Document v1.1 - March 23rd, 2015



Preface:

This is an update to the original Offline Map Kit. Most of the changes are transparent to the end user/developer. However, here is what actually happened:

With the advent of a Mandatory 64bit update required on all new apps submitted to the app store, and the fact that the route-me library had not been updated in a few years, I figured I'd better start looking at



Smug's Offline Map Kit for iOS - Setup

another SDK soon. I had a MapBox account, but was always afraid to use it because I shun paying for services, if I can help it. Well, I **had** to take a look at it.

I can't say how pleasantly surprised I was. Not only was the code updated and 64bit ready, it was easier to implement than the old Route-Me SDK, resulting in a re-write of this manual, due to the elimination of so many old requirements. My old code meshed seamlessly with the new SDK, and had the added benefit of better annotation, so I was able to improve the appearance of the Callout Views. Yay.

The best part is, if you're using offline maps (which is why we're here, right?) then you don't even need a MapBox API Key. Pinch me, I'm dreaming.

All of these changes were to the iOS side of the mapkit. There is also a MapBox SDK for Android, and you can believe I'll be testing it soon, so we won't have to create separate databases for each platform. One method for both platforms is a particular goal of mine. I hope that in the future it can come to pass.

Please read the instructions 'first' before performing any steps. Get clarification on any questions you may have. Once you are confident you understand, then proceed with caution.

We assume before we begin that you have created a BT v3 project for iOS, and have included the 'Smug Offline Map Kit' plugin. You should have assigned it as a menu choice, or as a tab choice. The settings are almost identical to the 'BT_screen_map', so if you have specific settings, the control panel should accomodate you similarly.

Thanks for listening!

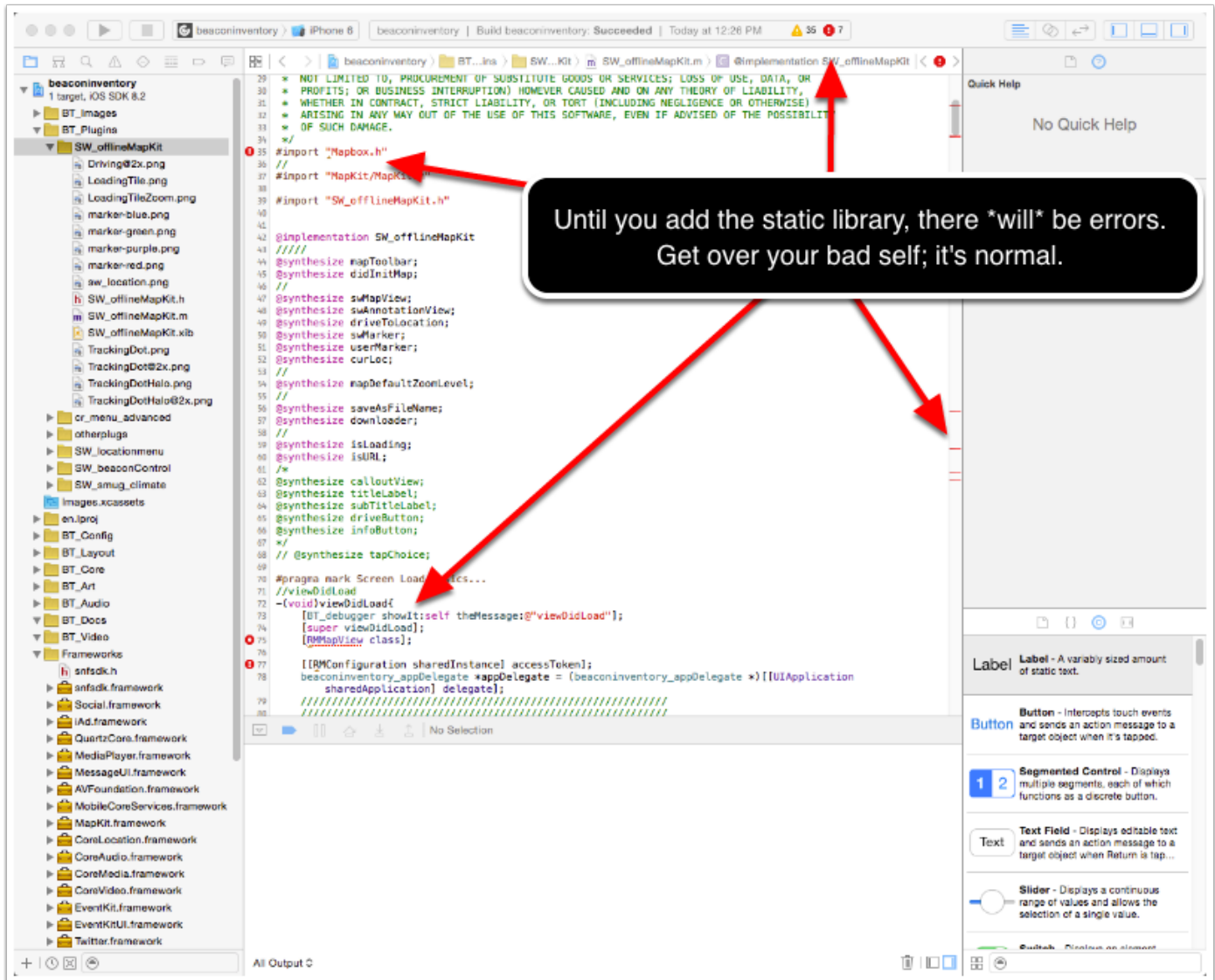
Cheers!

-- Smug



Create a BT Project; include the offlineMapKit as a plugin

Create a BT Project, and as one of your screens, select 'Smugs Offline Map Kit'.



Copy the needed resources into your BT Project

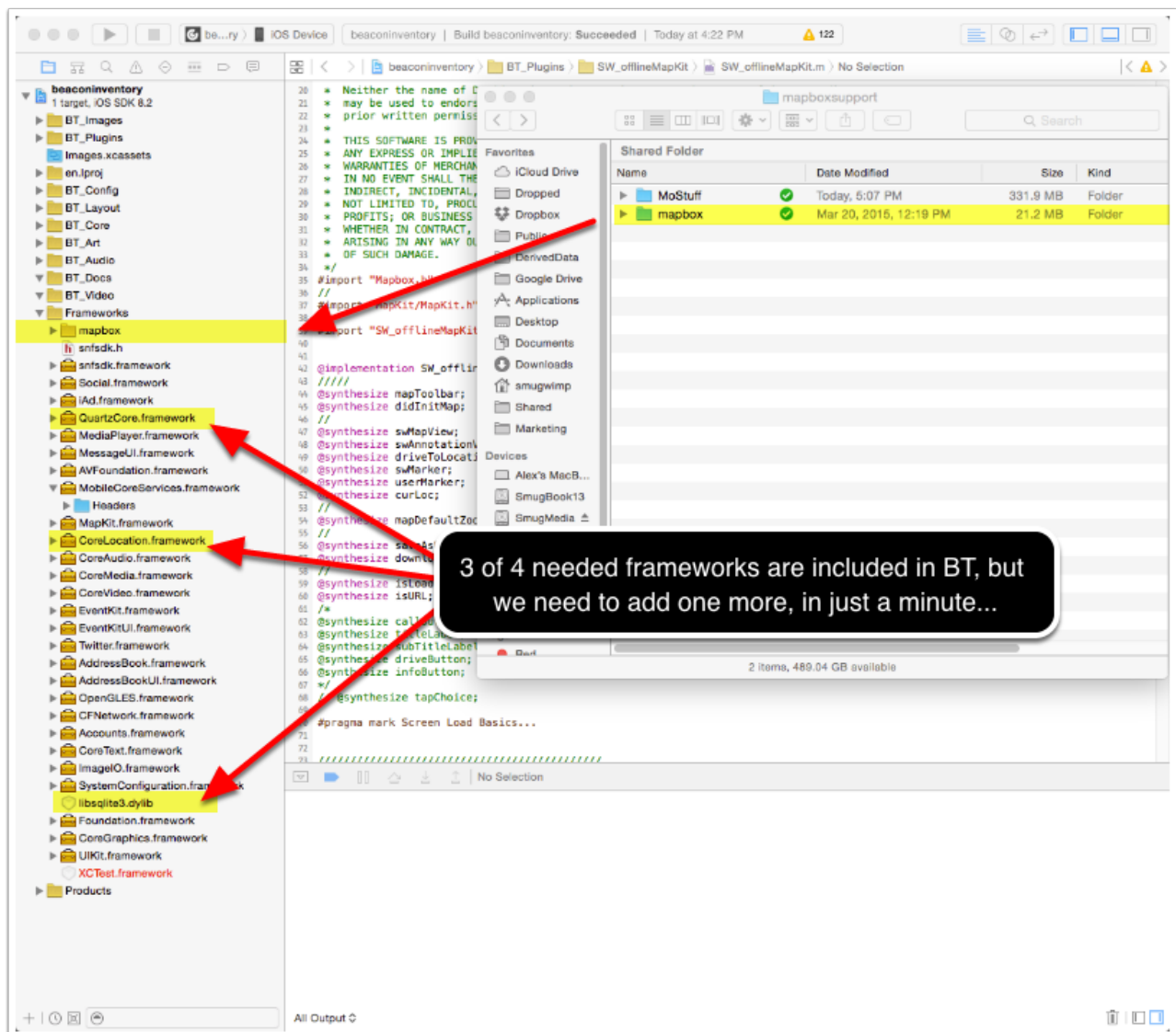
First things first. Get a copy of the latest Mapbox SDK. You want the latest 'static library' at <http://mapbox-ios-sdk.s3.amazonaws.com/index.html>



Smug's Offline Map Kit for iOS - Setup

At the time of this writing it was v1.6.0; or you can get the same copy as I used in this tutorial here:
http://www.marianasgps.com/sharedfiles/sw_olmk.zip

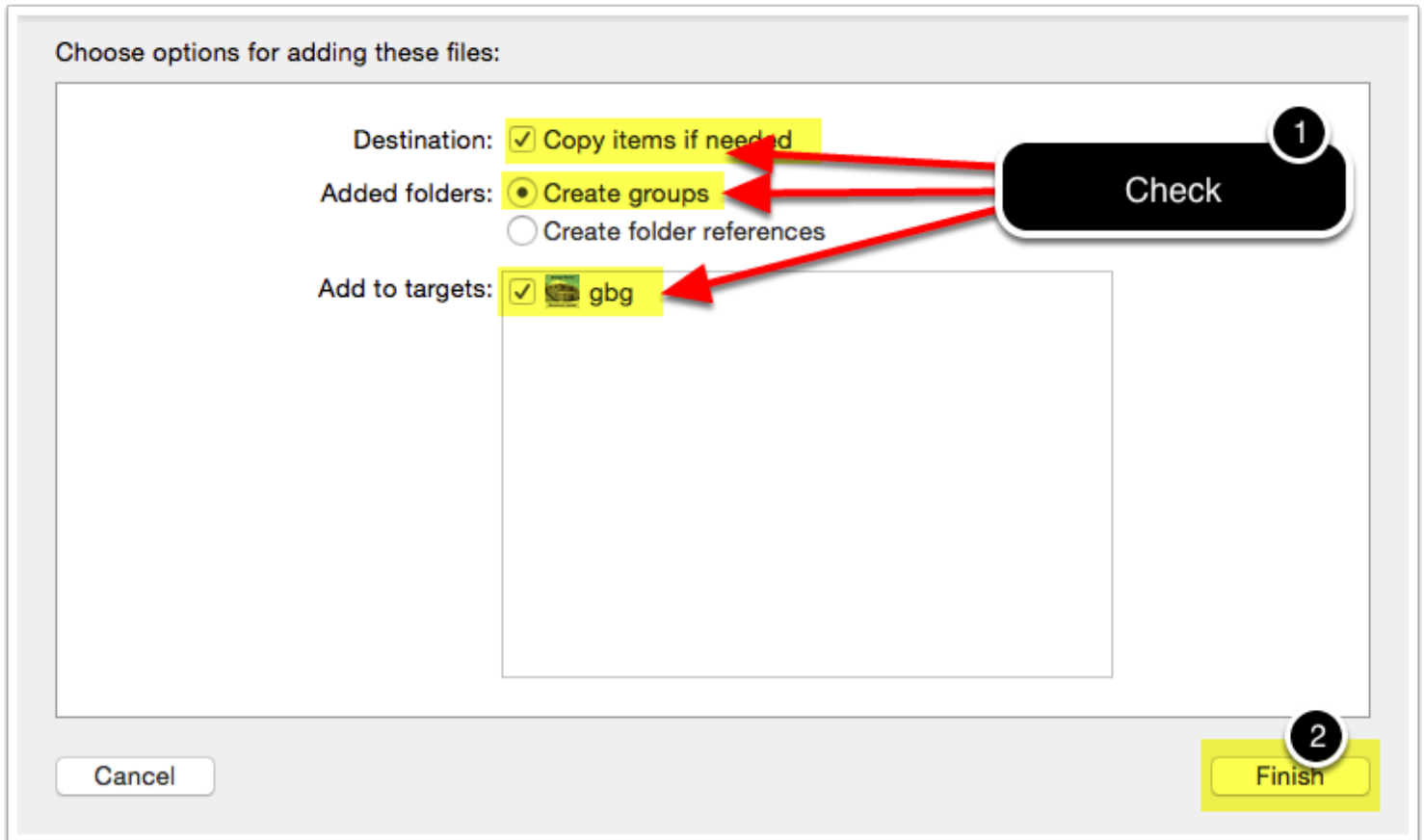
After you have downloaded the Static Library compressed file, Unzip the package, rename the directory to something 'simple' (like "mapbox") and copy the directory (including contents) into your project under 'Frameworks'. When it asks if you want to add the project, ensure the 'copy' box is selected, and allow the SDK to be copied into your project directory.





"Add" the Static Library to your Project

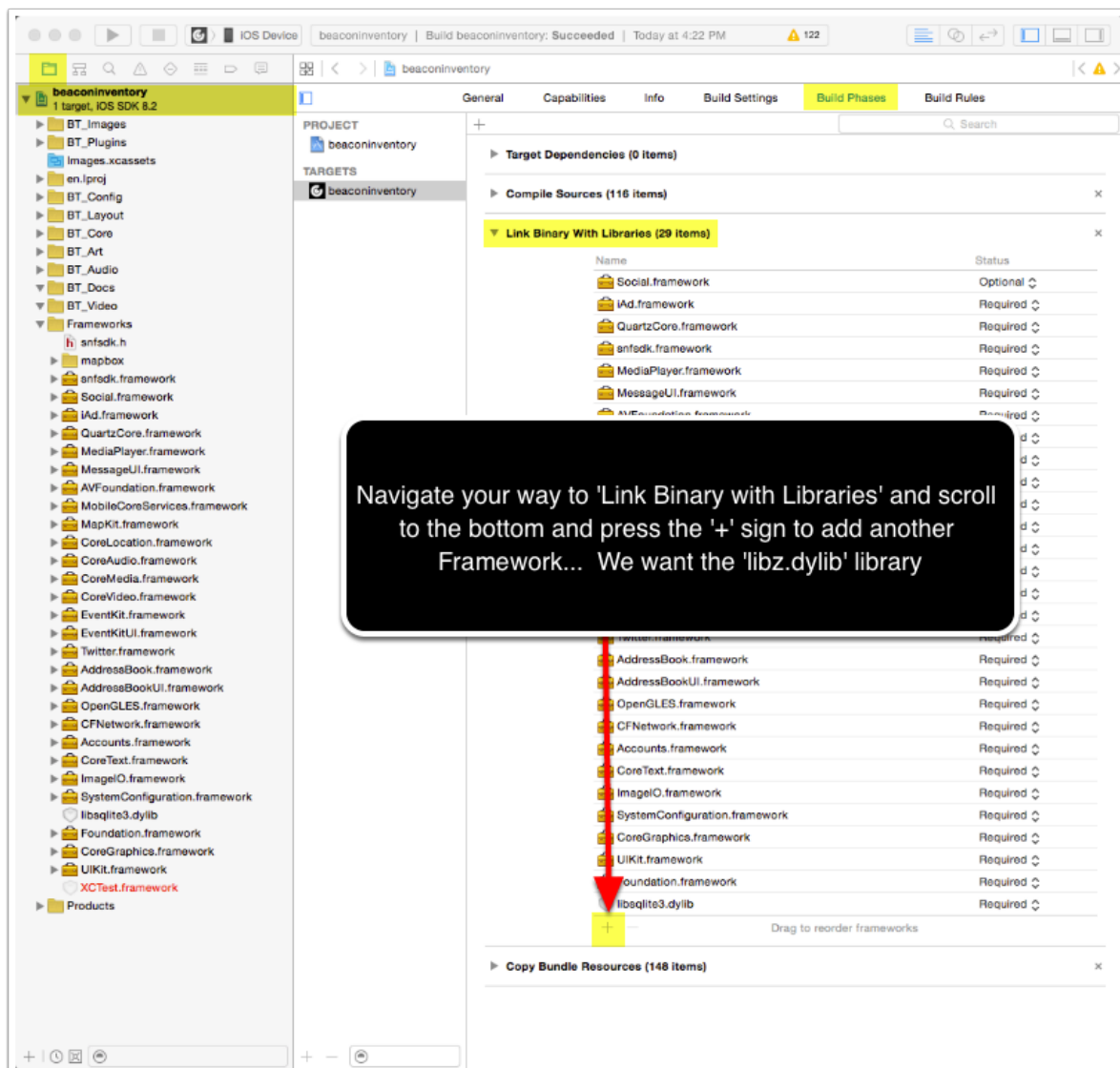
1) Ensure you add the files to the project 'while it is running'. A dialog box should appear. Ensure that 'copy items' is checked, as well as 'create groups' and 'add to targets'.





Add one more library: libz.dylib

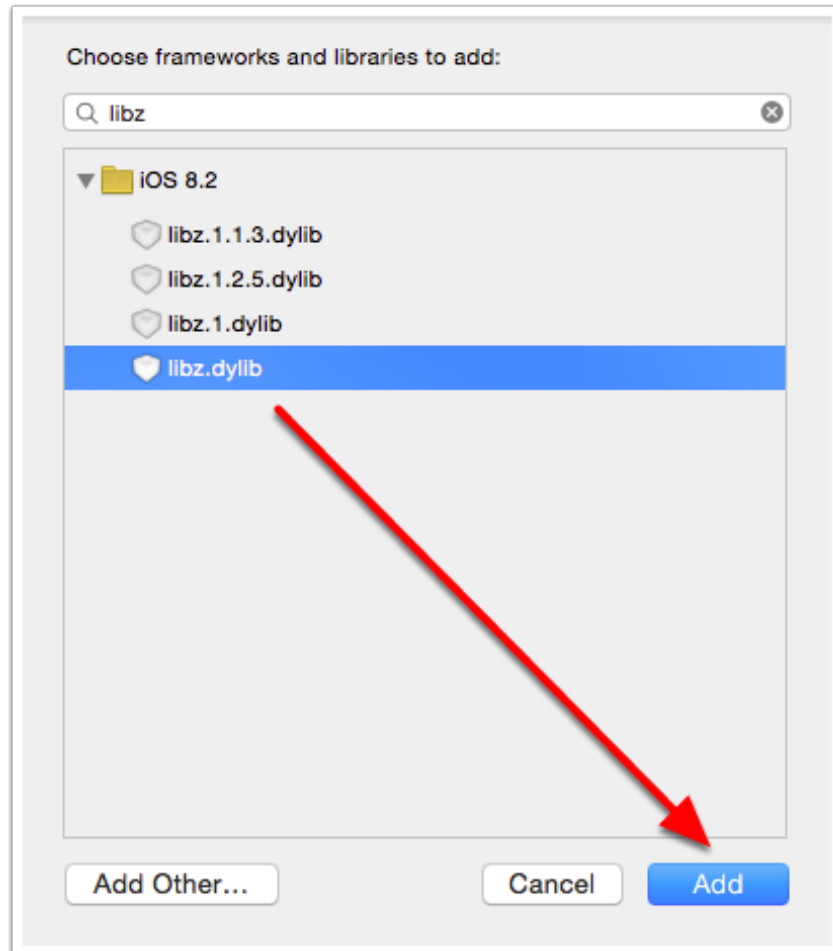
4 iOS SDK's are required, 3 of them are already in your BT Project. You need to add one more; the libz.dylib.





Select 'libz.dylib'

There are a lot of choices. To make things easier, type 'libz' in the search bar, and select 'libz.dylib' from the choices...





Ensure a successful addition

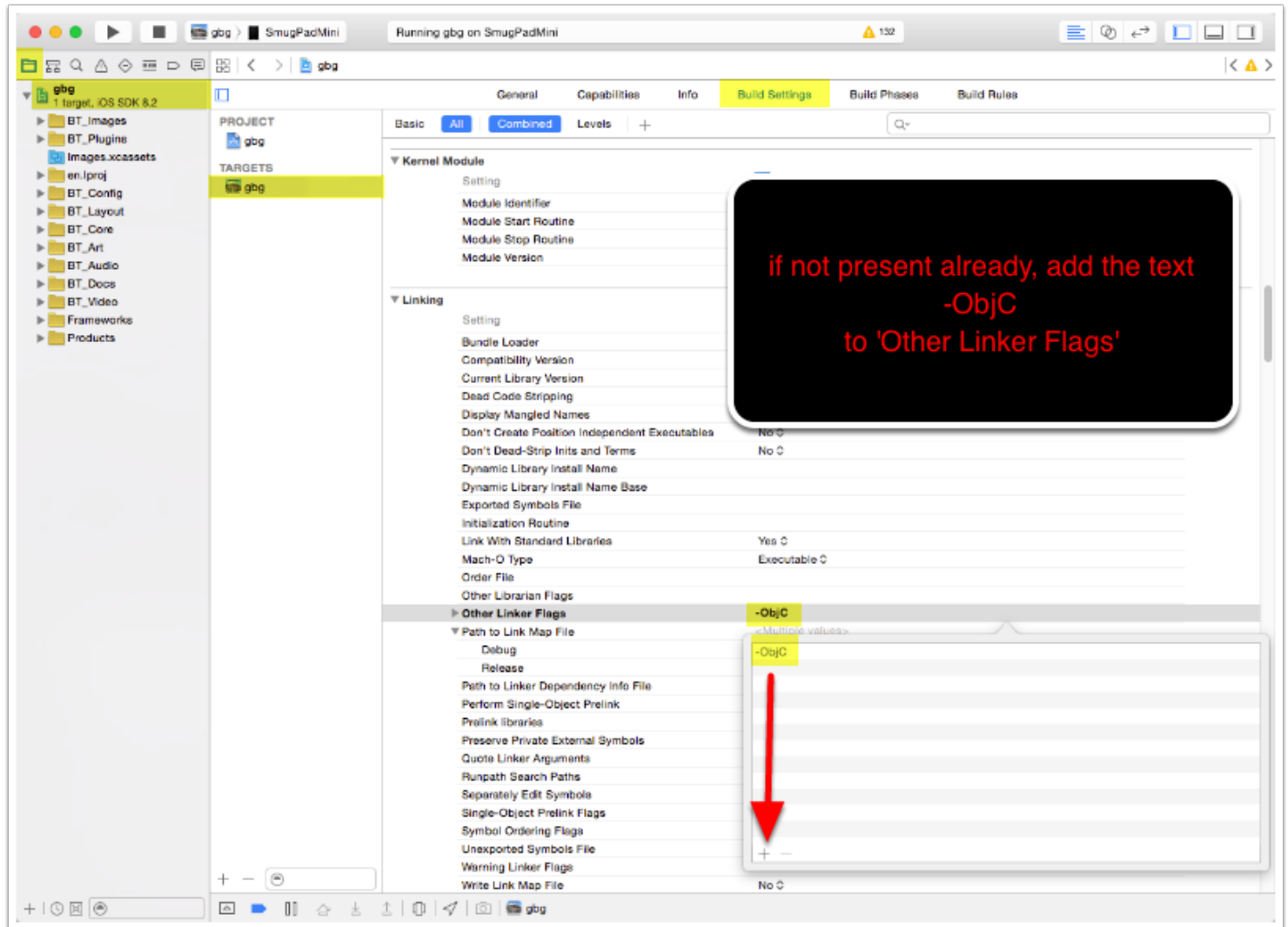
Check to make sure the library is in the right spot, and move on to the next step.

The screenshot shows the Xcode project settings for a project named 'beaconinventory'. The left sidebar shows the project structure with folders like 'BT_Images', 'BT_Plugins', 'en.lproj', 'BT_Config', 'BT_Layout', 'BT_Core', 'BT_Art', 'BT_Audio', 'BT_Docs', 'BT_Video', 'Frameworks', and 'libz.dylib'. The 'Frameworks' folder is expanded, showing 'libz.dylib' selected. The right pane shows the 'General' tab with a search bar and a list of dependencies. The 'Link Binary With Libraries' section is expanded, showing a list of libraries with their status.

Name	Status
libz.dylib	Req
Social.framework	Opti
iAd.framework	Req
QuartzCore.framework	Req
snfsdk.framework	Req
MediaPlayer.framework	Req
MessageUI.framework	Req



Add -ObjC as a linker flag

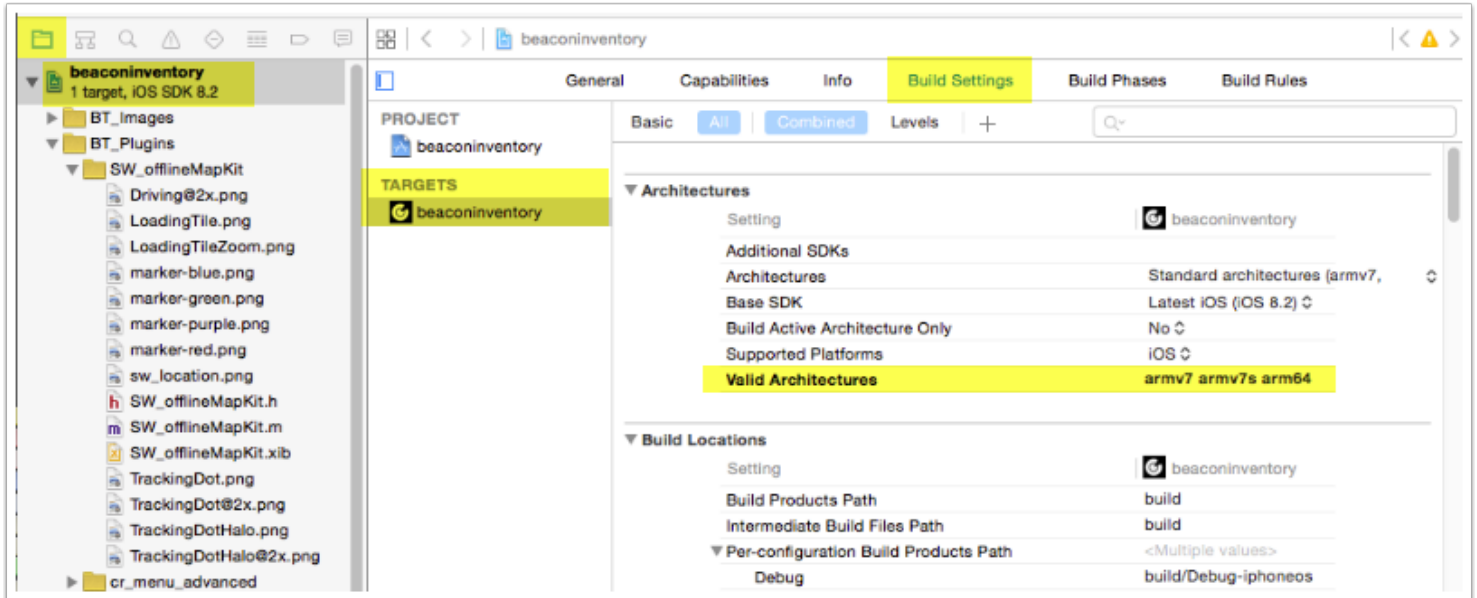


Once the files are 'black' (recognized in the project) you can quit. We're moving to 'your' project now.



Be sure to include 64bit, if it's not already configured in your project

If it's not already there, add 'arm64' into your 'Valid Architectures' to ensure 64 bit compatibility.





Aaaaaaand you're done.

Now that was way freaking easy. Ask anyone who used an older version how much fun it was to integrate route-me into iOS. What a Headache. But not now. We're done. On to the next item.






Some Control Panel Notes...

Child Items in the Smug Offline Map Kit are entered 'exactly' the same as in the BT_location_map plugin. It accepts Control Panel ChildItems, Database driven or online childItem Files, or bundled childItem files. Your method is our solution.

Map Locations

Create locations to show on the map. When a location is tapped a call-out window will open showing the title on the first line and the sub-title on the second line. For each location, determine what happens when the call-out window is tapped.

Location Title	Tapping this location...	refresh
 George Flores Historical Center	Shows driving directions	delete show on map

1 - 1 of 1

Title

Sub-Title (2nd line in call-out window)

Enter an address as a subtitle to automatically populate the latitude and longitude.

Latitude / Longitude (36.615, -121.904)

Callout Window Tap Action

Do nothing, don't show an icon

Load Screen With Nickname [Select](#)

[Find the latitude / longitude of the sub-title](#)
[Show another latitude and longitude tool](#)

add

Pin Color

Default Pin

Transition Type

Default transi

Only applies if tap loads another screen.
Does not apply to driving directions.



More Control Panel Notes...

Offline Map File

Enter your database name here. Do not include the extension. I will add that programmatically.
"MyDatabase.db" = **BAD**. "MyDatabase" = **GOOD**

Offline Map Filename:

Do not use an extension. I'll take care of it.

Screen Data Source

This plugin allows data from 3 sources;

1. Control Panel childItems. Basically it means you enter val
2. dataURL that points to a childItem file on the net.
3. A 'local' childItem file bundled with the app.

If you use a local file, do not include the letters 'http' in the

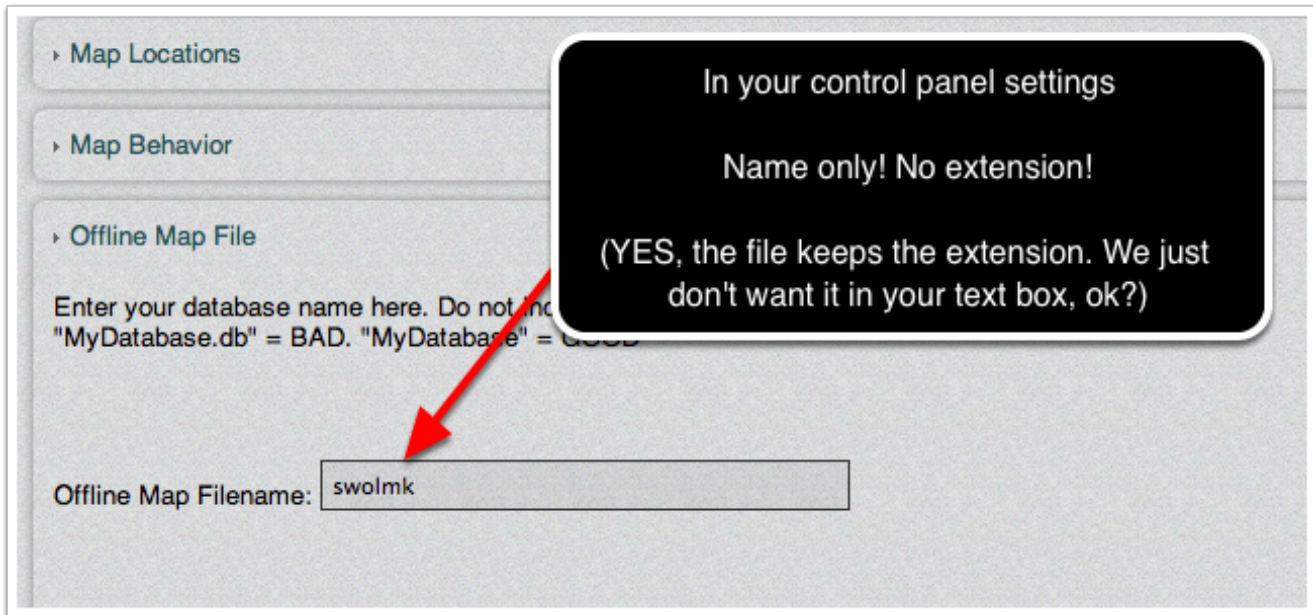
dataURL = "http://yourserver.com/childItem.txt"
localFile = "childItem.txt"

Data URL

Whether you want to use a bundled childItem file, an online childItem file, online Database Query, or control panel childItems, we've got you covered!



Now you need a map.



I spent a little time putting together a different tutorial that encompasses creating the 'right' kind of map for use with the Smug Offline Map Kit, and you can get a copy from [this link](#), if you don't already have a copy.

But I need to emphasize this. In your control panel, you will provide only the 'name' of the file, not the 'extension'. The extension of the file is detected via iOS Code, and can not should not will not be changed. It 'will' be 'mbtiles'. It is different for Android ('.sqlite') but that's a different topic.

If you already have a map in mbtiles format you wish to use, great. If not, follow the tutorial to create one and create one.

For testing purposes I have created a global map that should work for everyone not living in the north or south pole. It contains a database for both android and iOS. It is about 30MB in size, and can be [downloaded from my server](#). It is only levels 0 - 6, and each 'platform' database is about 22MB each uncompressed. This should give you a slight indication about how large a map database can be if you include too much detail data.



A word about database filesize...

Name	Date Modified	Size	Kind
▶ archived_atlases	Today 2:20 PM	15.9 MB	Folder
mgpsGU.mbtiles	Apr 15, 2014 3:51 PM	15 MB	MBTil...p 1
mgpsGU.sqlitedb	Apr 20, 2014 10:10 PM	14.8 MB	SQLite...ur
swolmk_test.mbtiles	Today 1:45 PM	6.9 MB	MBTil...p 1
swolmk_test.sqlitedb	Apr 20, 2014 10:26 AM	7.6 MB	SQLite...ur
swolmk.mbtiles	Today 2:07 PM	22.3 MB	MBTil...p 1
swolmk.sqlitedb	Today 2:07 PM	22.4 MB	SQLite...ur

Be mindful of the levels you include in your database, as mentioned in the 'mobac' tutorial.

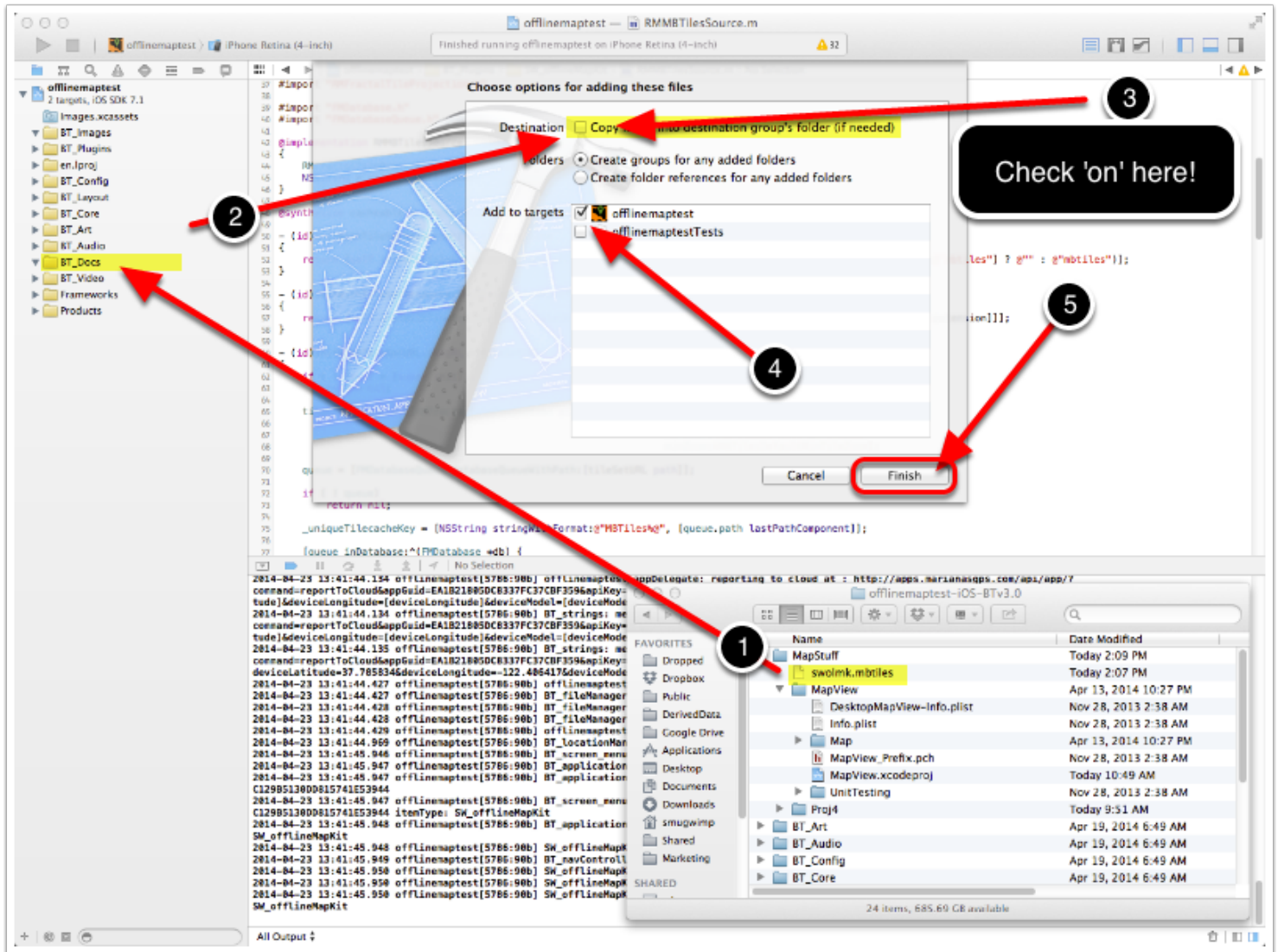
Consider that your database will be included in your bundle. However, it will not be 'read' from your bundle. What actually happens is your database gets copied to the cache, and read from the cached copy. Therefore, a 5MB database now becomes 10MB of space. And larger if more detail is included.

It's up to you what 'the balance' should be between detail and filesize. All I'm saying is 'this is going to happen, be aware of it'. And remember, most stores have a 50MB limit on app filesize.

Your atlas filesize will be added to the overall size of your app. As of this writing, you can not update the database remotely. Perhaps in the future.



Add your map to the project



It's just like any other resource you add to a BT Project.

- 1) Select and drag your database file from your source directory.
- 2) Drop it into the BT_Docs directory of your project.
- 3) When the dialog appears, Make sure "Copy items into destination group folder..." **is** checked.
- 4) Make sure your target reference **is** checked.

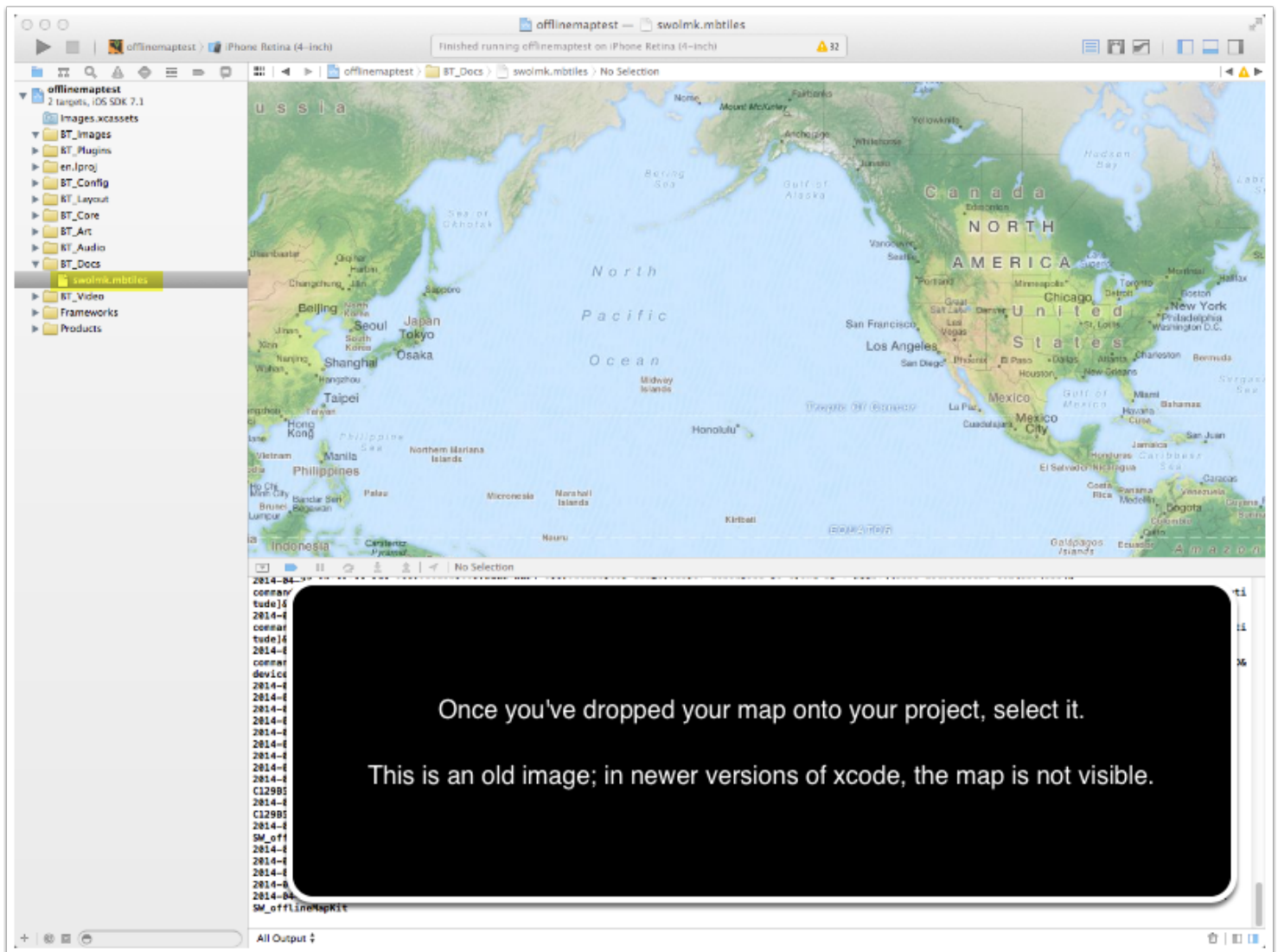


Smug's Offline Map Kit for iOS - Setup

5) If all seems good, press 'Finish'

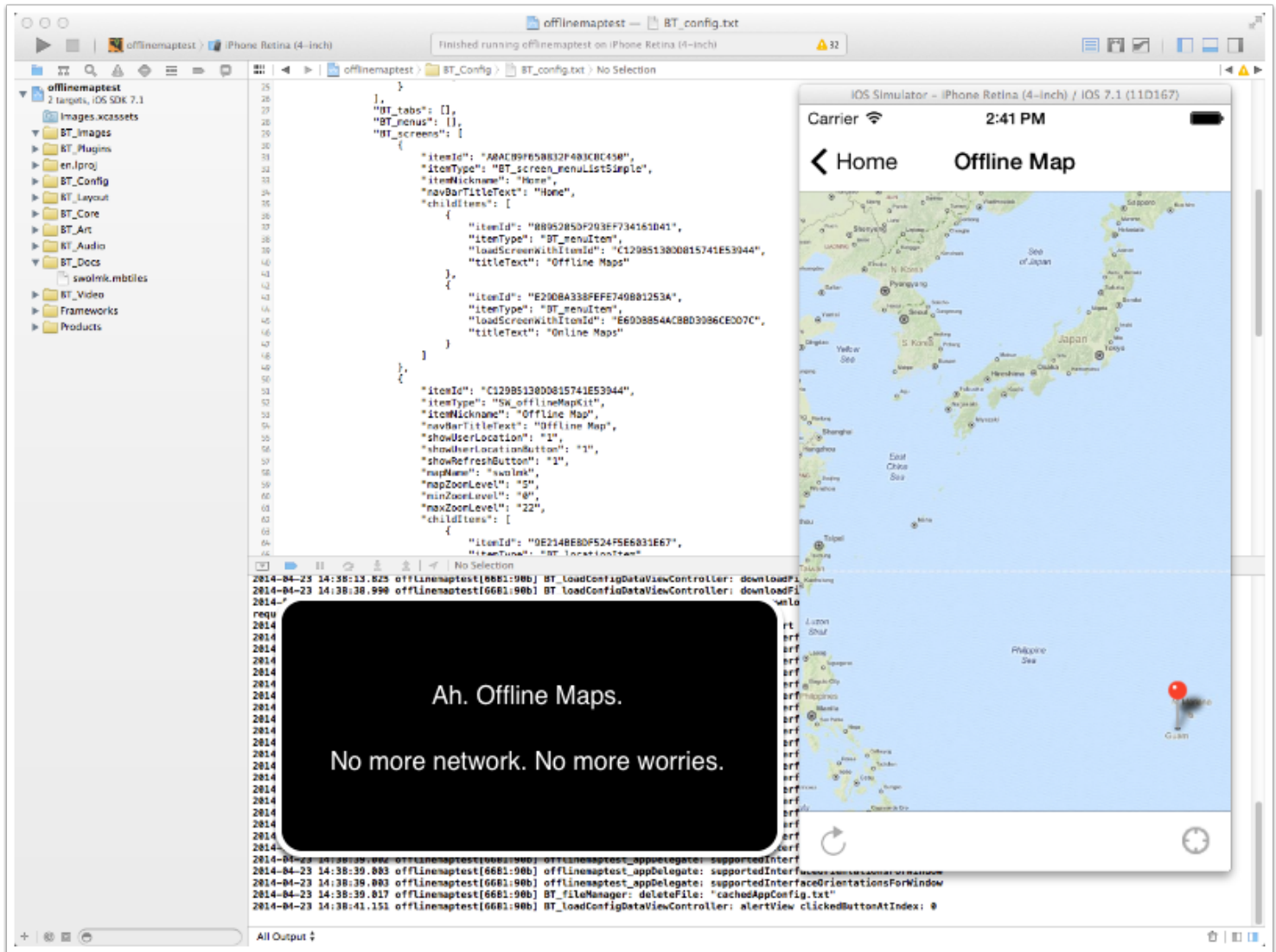
Since our previous version kept the map database within the 'mapstuff' directory, there was no need to copy it; but now that there is no mapstuff directory, you need to copy the database into the bundle to ensure it gets added with everything else. Make sure the 'copy' check box is checked!

Map resource resides in BT_Docs





Compile and Go!



From this point, you should be able to compile your project and display your map. Ensure you've made the correct entries in any menu or tab settings.

If things don't compile, and show errors, take steps back to see if you accurately completed each previous step.

If you're certain that you've accurately completed each step, try repeating the tutorial using the maps I've created. If they work and your maps do not, then it indicates a corrupt or incorrectly configured database. Try the mobac tutorial again, and specifically ask for 'mbtiles' format for your database.



If neither database works, it could be an issue with the code. Please try and recreate the problem, and document your steps and notify the developer (me).

For the most part, things should work just fine.

Android is so simple it's not funny. Create the database, drop it in your project, and go.

I tried to document this procedure as detailed as possible to ensure a high accuracy rate. If you see a missed or incorrect step, or have any suggestions on how I could improve this (or any) document, please let me know at your convenience.