Couchbase Mobile on iOS — Tutorial One

Abstract

For queries regarding the documentation, please contact editors@couchbase.com.

Couchbase

Table of Contents

Using Couchbase Mobile an application allows you to store data in a local instance of Couchbase Mobile. Doing so, gains your application the power of Couchbase views and easy to work with JSON document structure. Also, since replication is a built in feature of Couchbase your mobile application can be configured to sync back to a server in the cloud, or to other devices on the network.

In order to get started you need to setup a new project and include the Couchbase iOS Framework in the project. The latest Couchbase Mobile for iOS Framework can be downloaded from the couchbase.com website. In order to keep the project organized, it is helpful to create a "Frameworks" directory in the root directory of the Xcode project. These instructions assume you have created a "Frameworks" directory in the root of your Xcode project, and have copied the downloaded Couchbase.framework there. If your building multiple iOS apps you may prefer to store the framework in a shared location. If you didn't put it in the Frameworks subdirectory of your project, you'll want to edit the path used in the rsync command below in step 5.

Wherever you put the downloaded Couchbase.framework, here's how to add it to a project:

- Open your Xcode project.
- Drag the `Couchbase.framework` into the Frameworks section of the file list in your project window. In the sheet that opens, make sure there's a checkbox next to your app's target.
- You will need to link some libraries with your project: Go to the target's Build Phases, open Link Binary With Libraries, click the + button, and add `libstdc++.dylib`, `Security.framework` and `libz.dylib` from the iOS section.
- Go to the target's `Build Phases` and add a new `Run Script` phase.
- Paste the following into the script content of the new phase.

Important

This assumes you put the framework in a `Frameworks` subfolder. If you put it elsewhere, update the path in the 2nd argument to **rsync** accordingly.

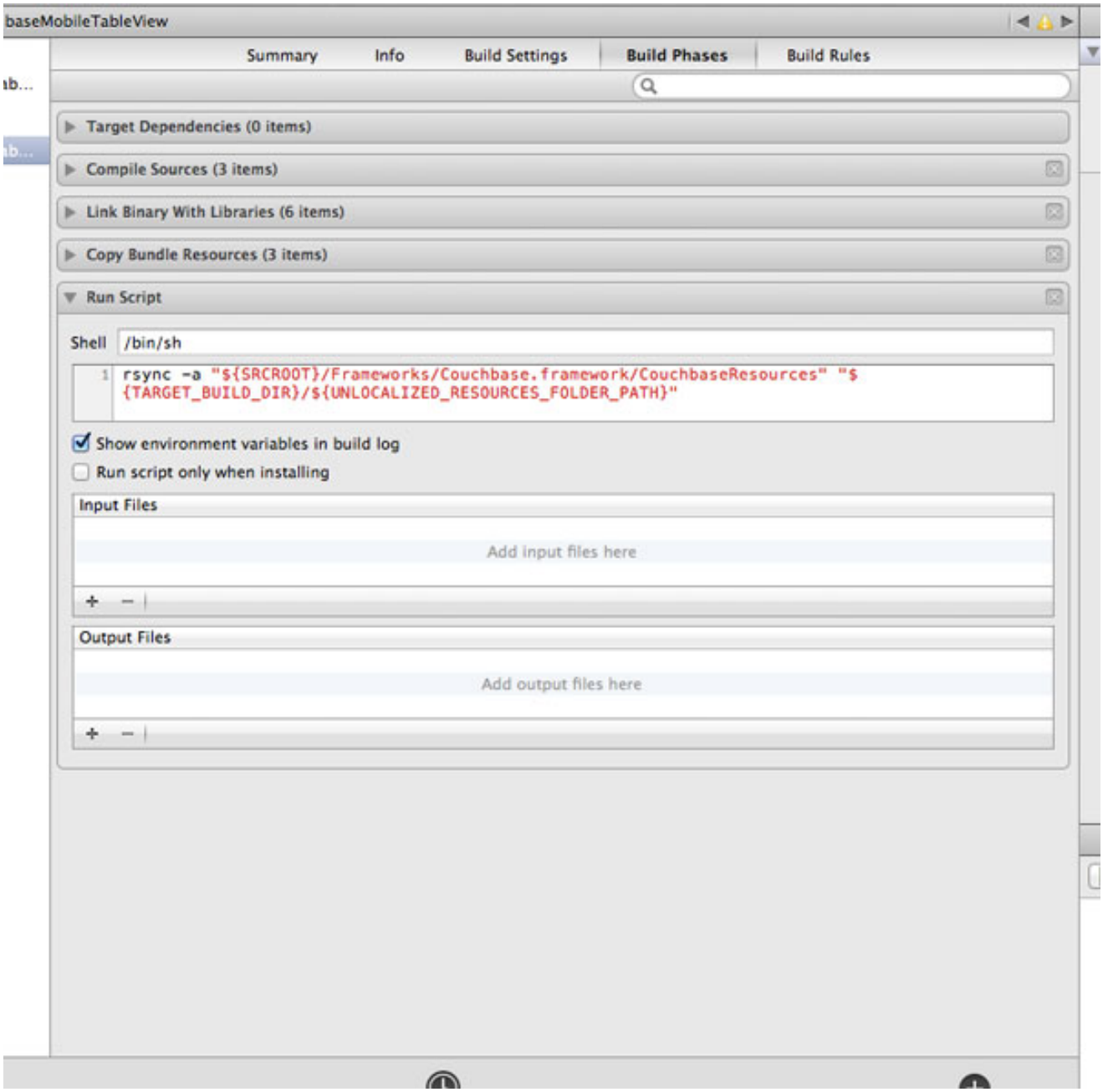
Important

The **rsync** command below is a single long line. Do not put a newline in the middle!

```
# The 'CouchbaseResources' subfolder of the framework contains
# resources needed at runtime. Copy it into the app bundle:

rsync -a "${SRCROOT}/Frameworks/Couchbase.framework/CouchbaseResources" \
     "${TARGET_BUILD_DIR}/${UNLOCALIZED_RESOURCES_FOLDER_PATH}"
```

Figure 1. iOS Couchbase Build Phase Run Script



Now that both the Couchbase framework as well as the needed frameworks from Apple are included in the project, you can add to your application code what is need to start Couchbase Mobile at launch and confirm it started properly. See CouchbaseMobile.h for the interface, and see EmptyAppDelegate.m for an example of how to call it. It involves added a couple of functions to your AppDelegate.

In your MyAppDelegate.h file, add the declaration that you implement the Couchbase protocol, by importing the header file and adding "CouchbaseDelegate" inside the brackets:

```
#import <Couchbase/CouchbaseMobile.h>

@interface MyAppDelegate : UIResponder <UIApplicationDelegate, CouchbaseDelegate>
```

You need to start the Couchbase Mobile server. This is normally done along with the launch of the application. In MyAppDelegate.m, add this code to your didFinishLaunchingWithOptions callback:

```
CouchbaseMobile* cb = [[CouchbaseMobile alloc] init];
cb.delegate = self;
NSAssert([cb start], @"Couchbase didn't start! Error = %@", cb.error);
```

Now that the MyAppDelegate class is set as the callback for Couchbase there are a couple of functions that you need to add. First, a callback for when the server has sucessfully started. Second, you need a callback if the Couchbase server fails to start.

```
-(void)couchbaseMobile:(CouchbaseMobile*)couchbase didStart:(NSURL*)serverURL {
    NSLog(@"Couchbase is Ready, go! %@", serverURL);
}

-(void)couchbaseMobile:(CouchbaseMobile*)couchbase failedToStart:(NSError*)error {
    NSAssert(NO, @"Couchbase failed to initialize: %@", error);
}
```

When Couchbase Mobile launches it will call didStart with its local serverURL. That url can be used to interact with Couchbase. For now just log the url. If you are running your application in the simulator, another way to confirm that Couchbase Mobile is running properly is to visit the server url in a web browser. You should see a JSON welcome message like:

```
{"couchdb":"Welcome","version":"1.0.1-emonk-ios"}
```

Now you have Couchbase Mobile running on iOS! Most of the example apps use CouchCocoa to interact with Couchbase, but there are plenty of options for how to use Couchbase, with people using everything from raw REST connections, to Core Data interfaces, to WebView interfaces. Couchbase can fit any iOS application.

Licenses

This documentation and associated software is subject to the following licenses.

Documentation License

content, products, and services. Couchbase Inc. and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.