

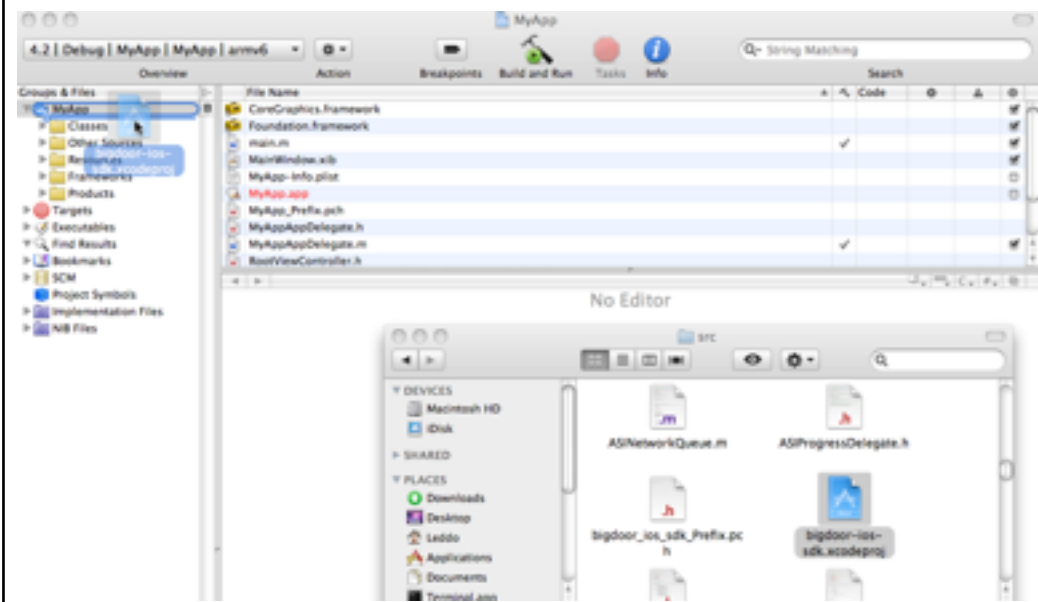
Adding BigDoor SDK as a framework

The BigDoor IOS SDK is compiled as a static library, and the easiest way to add it to your project is to use Xcode's "dependent project" facilities. The steps required to add the library can be a bit tricky, but if you follow these instructions you should be up and running in minutes.

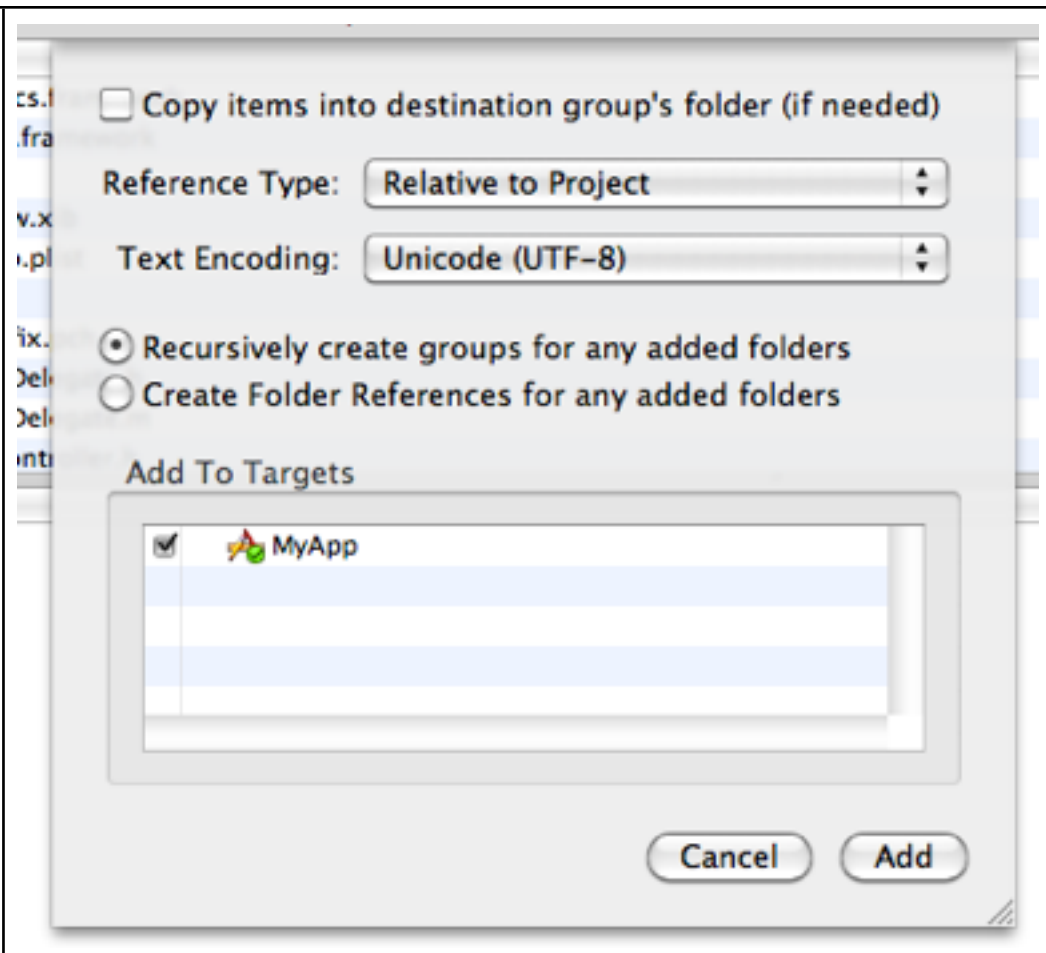
Clone the Bigdoor-ioskit bitbucket repository: `hg clone https://leddo@bitbucket.org/leddo/bigdoor-ioskit`. Make sure you store the repository in a permanent place because Xcode will need to reference the files every time you compile your project.

Locate the "bigdoor-ioskit.xcodeproj" file under "src".

Drag bigdoor-ioskit.xcodeproj and drop it onto the root of your Xcode project's "Groups and Files" sidebar of the project you want to include it in



A dialog will appear — make sure “Copy items” is unchecked and “Reference Type” is “Relative to Project” before clicking “Add”

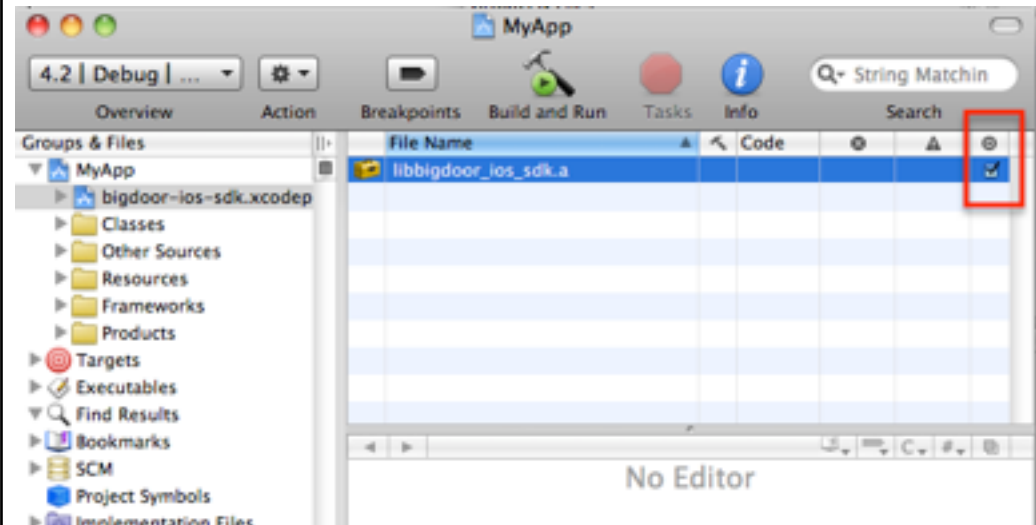


Now you need to link the bigdoor-ioskit static library to your project.

Click the “bigdoor-ioskit.xcodeproj” item that has just been added to the sidebar.

Under the “Details” table on the right, you will see a single item: libbigdoor_ios_sdk.a.

Check the checkbox on the far right of libbigdoor_ios_sdk.a

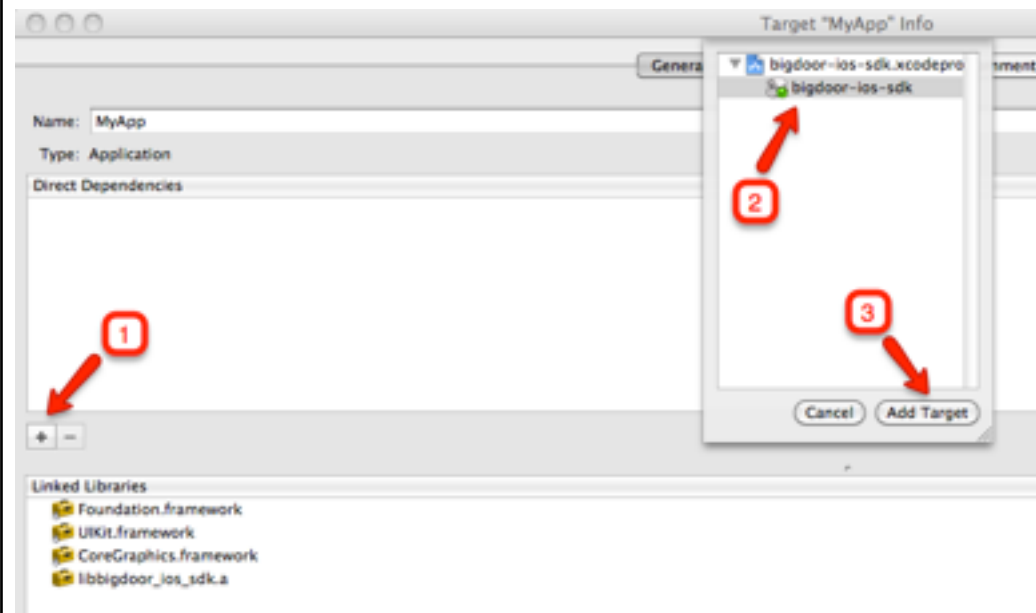


Add bigdoor-ioskit as a dependency of your project, so Xcode compiles it whenever you compile your project.

Expand the “Targets” section of the sidebar and double-click your application’s target.

Under the “General” tab you will see a “Direct Dependencies” section.

Click the “+” button (step 1), select “bigdoor-ioskit” (step 2), and click “Add Target” (step 3)



Add additional frameworks

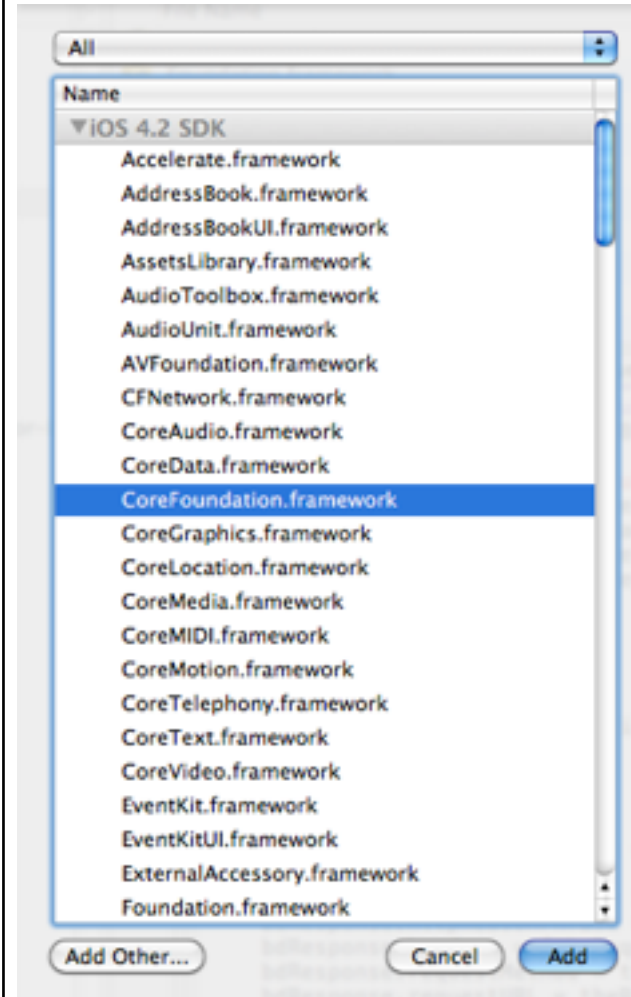
The following frameworks are also required (if you have not included them already).

- CoreFoundation.framework
- CFNetwork.framework
- SystemConfiguration.framework
- MobileCoreServices.framework
- libxml2.2.7.3.dylib (or greater)
- libz.1.1.3.dylib (or greater)

To add a framework, select the “frameworks” folder of your project on the left sidebar. Right click and choose the option to “Add Additional Frameworks”.

Find the required framework, select it, then click the “Add” button.

Repeat for the remaining frameworks.



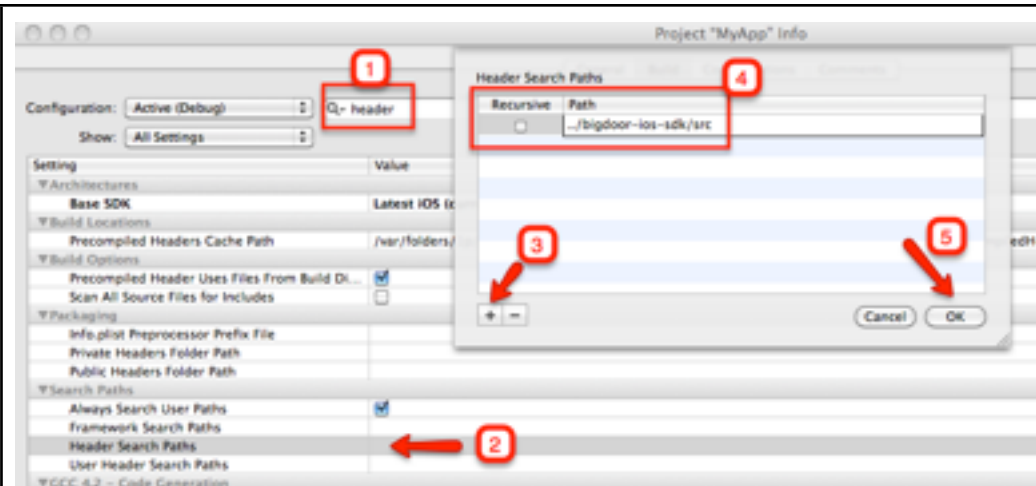
Finally, we need to tell your project where to find the bigdoor-ioskit headers.

Open your “Project Settings” and go to the “Build” tab.

Type in “header” in the quick search box (step 1) and look for “Header Search Paths” and double-click it (step 2).

Click the “+” button to add a new path (step 3), and then type in the **RELATIVE** path to the location of the bigdoor-ioskit/src folder (step 4).

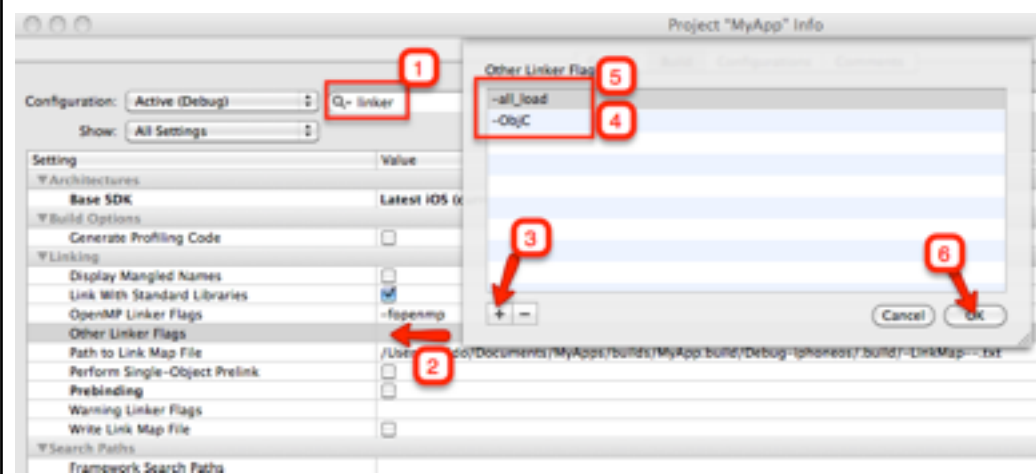
Click the OK button to add the path (step 5)



Then type in “linker” in the quick search bar (step 1) and double click on the “Other Linker Flags” under the “Linker” section.

Click the “+” button to add a new flag and enter “-ObjC” (step 4) and repeat for “-all_load” (step 5).

Click the OK button (step 6) to add list of flags



You should now be able to build your app without error.

Usage

In Your application, load the headers as follows:

```
#import "BigDoor.h"
```

The bigdoor-ioskit library uses a singleton pattern. You reference the bigdoor library by calling the sharedBigDoor method. eg:

```
[BigDoor sharedBigDoor]
```

To initialize the library, you need to call the “initWithApplicationKey: Secret: Host:” the very first time using the values provided to you:

```
BigDoor *bd = [[BigDoor sharedBigDoor]
               initWithApplicationKey:@"<insert api key here>"
               Secret:@"<insert secret here>"
               Host:@"http://api.bigdoor.com"];
```

There are some good articles on stackoverflow on the best practices to adopt when storing Authentication Keys in iphone apps. One of the preferred methods is to use your own server as a proxy. <http://stackoverflow.com/questions/4123806/best-practices-for-securing-api-credentials-as-part-of-an-iphone-app>

From then on, you can use the published methods to access the data.

eg:

```
[bd getWithObjectPath:@"/end_user" params:nil delegate:self];
```

and ensure you implement the BigDoorDelegate methods:

- (void)requestDidReturnResponse:(BigDoorResponse *)bdResponse
- (void)requestDidError:(BigDoorResponse *)bdResponse

Example calls:

1) Add Currency to User

```
BigDoor *bd = [BigDoor sharedBigDoor];

NSMutableDictionary *env = [NSMutableDictionary dictionaryWithObjectsAndKeys:
                           amount.text, @"amount",
                           nil];

[bd postWithObjectPath:[NSString stringWithFormat:@"/named_transaction_group/%@/execute/%@", transactionID,
endUserID]
        params:nil
        envelope:env
        delegate:self];
```

2) Update User Profile information

```
NSString *endUserLogin = @"user1";

NSMutableDictionary *env = [NSMutableDictionary dictionaryWithObjectsAndKeys:
    @"test", @"first_name",
    @"user", @"last_name",
    @"user1", @"display_name",
    nil];

BigDoor *bd = [BigDoor sharedBigDoor];

[bd putWithObjectPath:[NSString stringWithFormat:@"/end_user/%@/profile/publisher", endUserLogin]
    params:nil
    envelope:env
    delegate:self];
```