Jaspersoft Mobile SDK Install Guide

I. Installation via Linking as Static Library

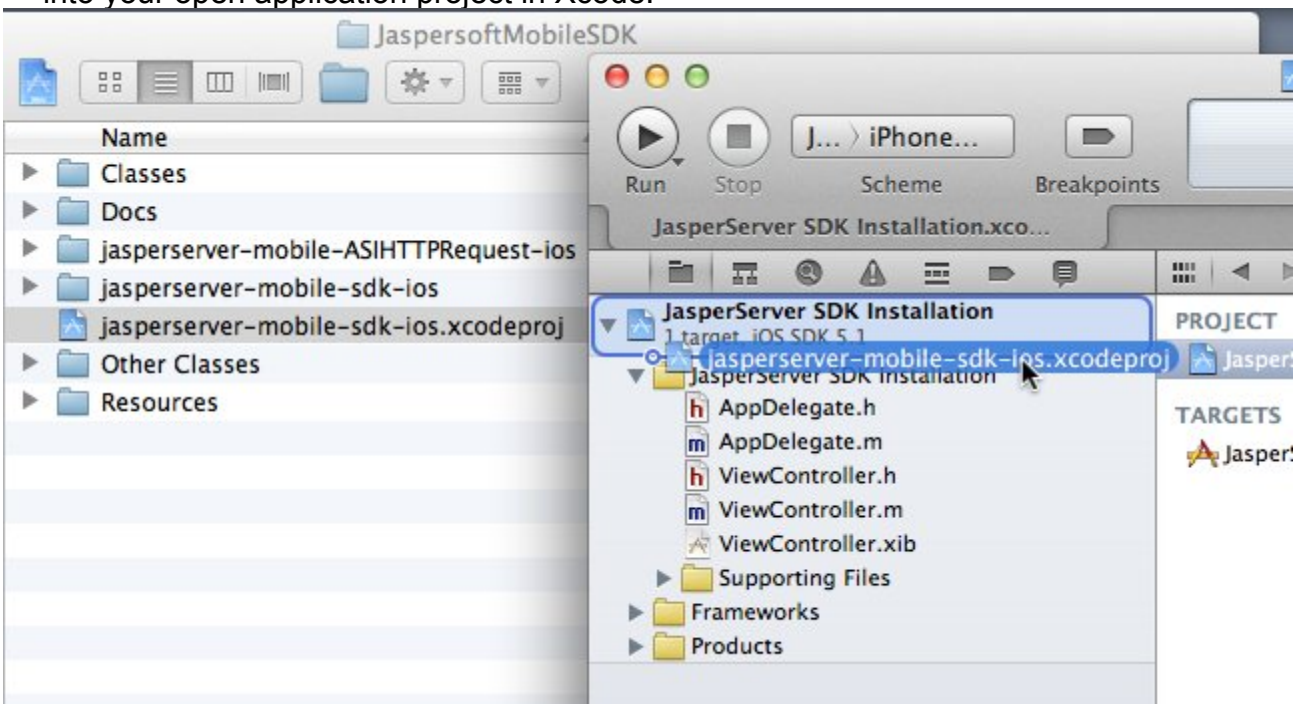
Static library **linking** is the recommended way to install the Jaspersoft Mobile SDK. Using this approach, you will get the following benefits:

- you can easily update/replace SDK
- solve classes name collision (when different classes have same name)
- ARC independent

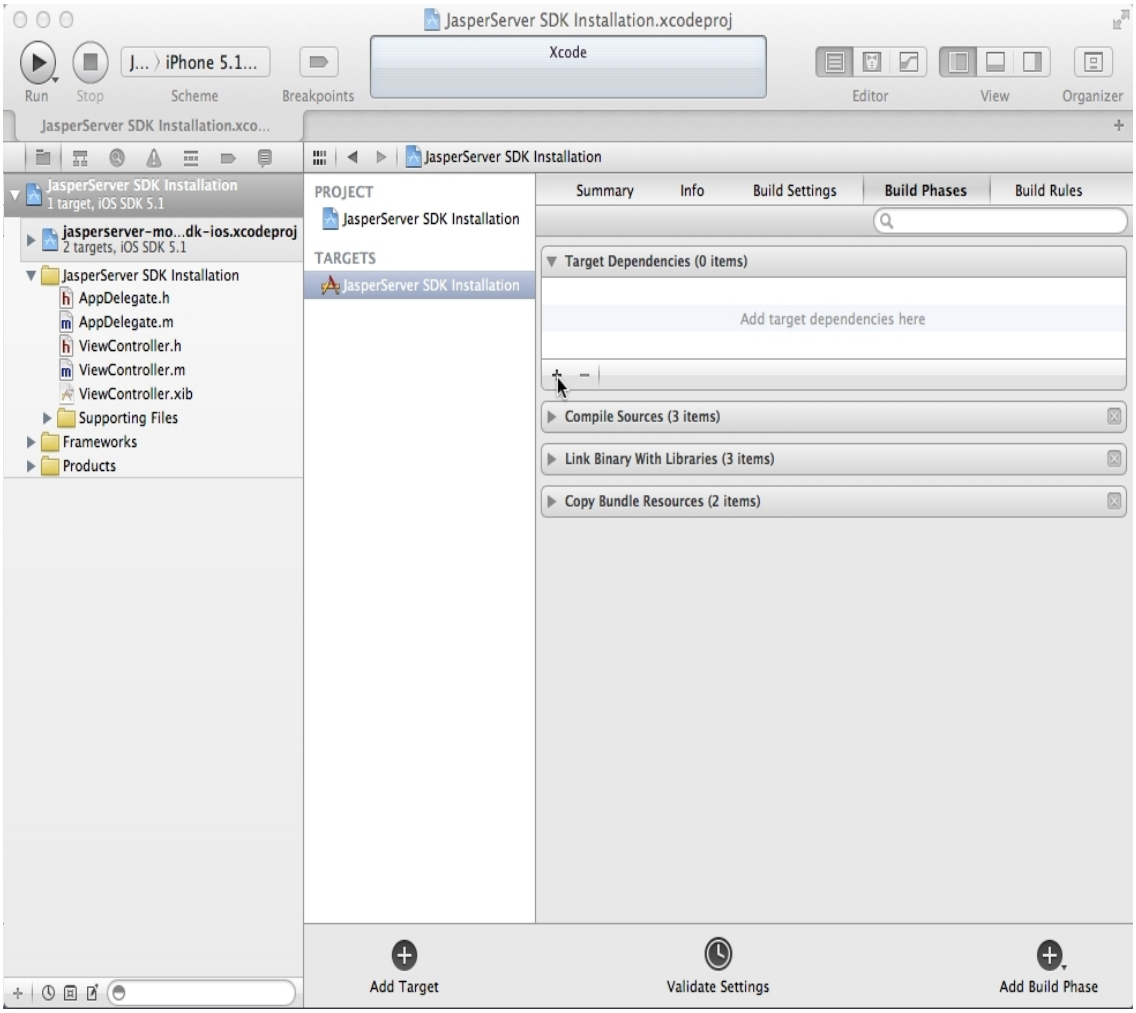
There are a few more install steps (compared to just adding files directly, covered in Section II), but the benefits outweigh the costs, and we will walk you through all of the steps below.

To install via **linking**:

1. Add a cross-project reference by dragging the **jasperserver-mobile-sdk-ios.xcodeproj** file into your open application project in Xcode:

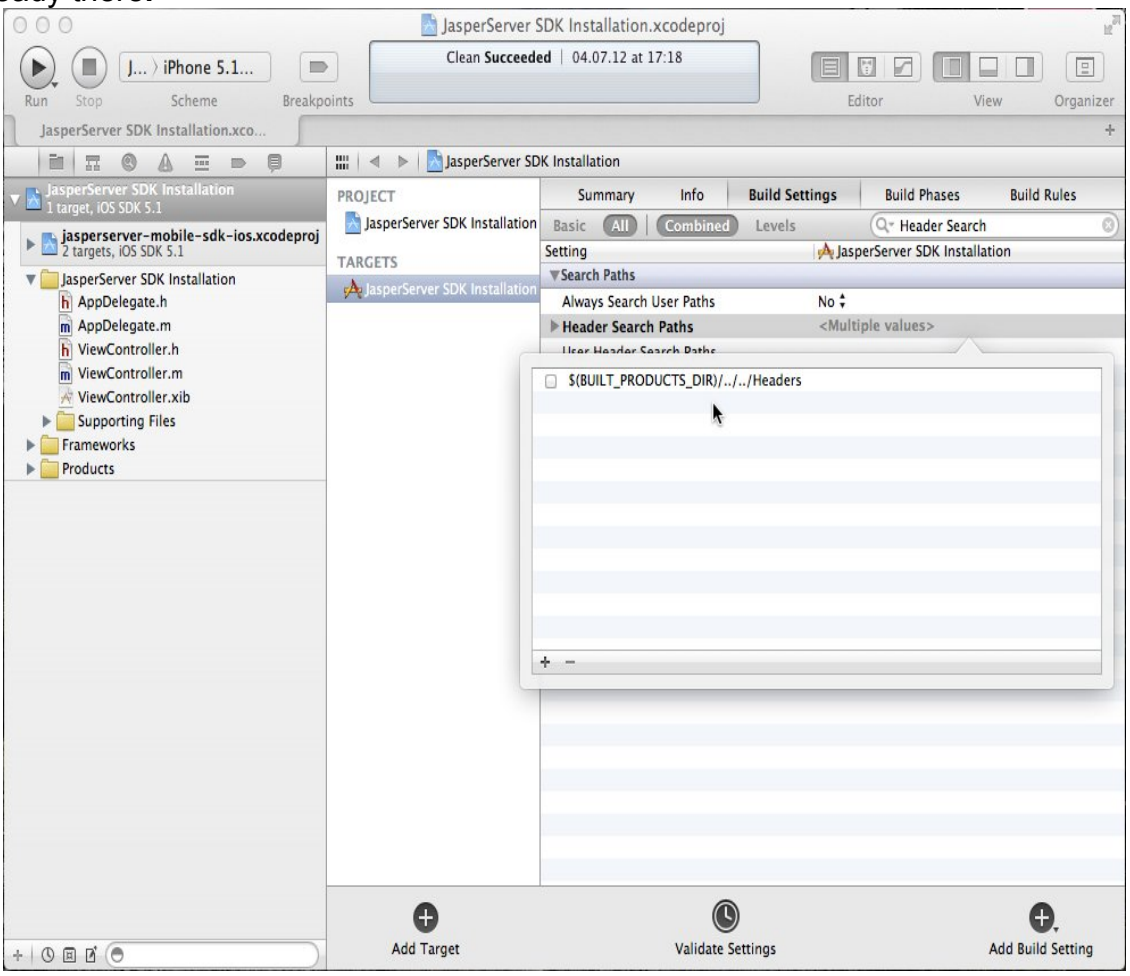


2. Open target settings editor -> **Build Phases** for the target you want to link library into
3. In the tab **Target Dependencies** tab add dependency on the **jasperserver-mobile-sdk-ios** aggregate target.
   Note: If you don't have the **ASIHTTPRequest** library included in your project, you don't need to do anything (it will be added automatically because of the dependency with the SDK). If you already have it in your project, you can exclude it (see instructions in Section III).

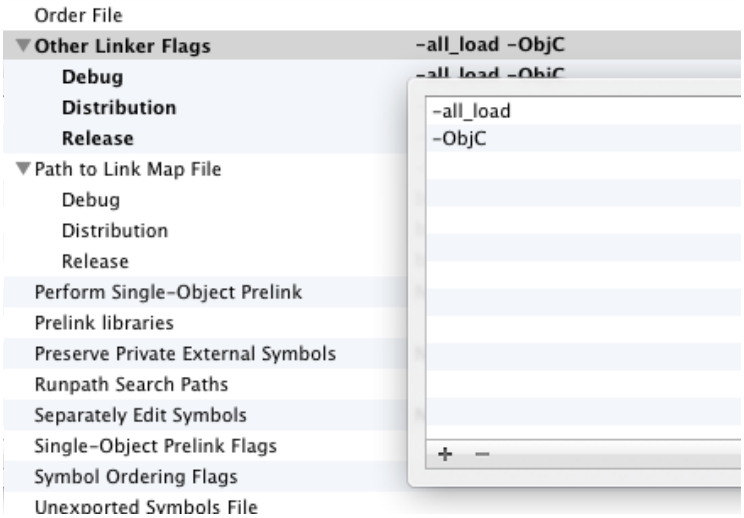


4. In the tab **Link Binary With Libraries** link against next frameworks and libraries:
    - **jasperserver-mobile-sdk-ios.a** (don't worry that it shows up in red)
    - **CFNetwork.framework**
    - **SystemConfiguration.framework**
    - **MobileCoreServices.framework**
    - **QuartzCore.framework**
    - **libxml2.dylib**
    - **libz.dylib**

5. Open **Build Settings -> Search Paths -> Header Search Paths** and double-click on the value column in the top level. Add **"$(BUILT_PRODUCTS_DIR)/../../Headers",** if it's not already there**.**



6. Open **Build Settings -> Linking -> Other Linker Flags,** and add "**-ObjC**" and "**-all_load**", if they are not already there (you can copy/paste the following string, just make sure it gets properly separated into 2 flags: -objC -all_load)**.** This is used for loading all JS Controller classes which extend from UITableViewController / UIViewController. This makes it possible to set any of JS Controller classes as a **Custom Class** in a **.xib** file.



7. You have now completed installation of the **Jaspersoft Mobile SDK for iOS** into your application.

8. To verify the installation, open up your App Delegate and add an import of the **JSClient** header:
   #import <jasperserver-mobile-sdk-ios/JSClient.h>
   Note: If you are building the JasperMobile app for iOS, it's already there.
9. Build your project and verify build output. Project should build without any issues. Congratulations!

II. Installation by Adding Source

If you don't want to **link** the JS Mobile SDK you can install it as source. That basically means you are copying the SDK classes (and all of the dependent libraries) directly into your Xcode project. This provides some benefits:

- easy SDK library modification
- install is a little bit easier
- simpler **#import**

But there are also some drawbacks

- possible class name collisions (as Objective-C has no namespace)
- problems with ARC code (SDK was written for iOS version earlier than 4.2, pre-ARC
- you still need to have **ASIHTTPRequest** library
- future problems with updating SDK (you need to replace all classes again).

To install the SDK as source:

1. Copy all files from **Classes** directory to your Xcode project
2. SDK require **ASIHTTPRequest** library which locates in **Other Classes** directory. So you can copy all classes from directory or install **ASIHTTPRequest** manually
3. Open target settings editor -> **Build Phases** for the target you want to link library into
4. In the tab **Link Binary With Libraries** link against next frameworks and libraries:
   - **CFNetwork.framework**
   - **SystemConfiguration.framework**
   - **MobileCoreServices.framework**
   - **QuartzCore.framework**
   - **libxml2.dylib**
   - **libz.dylib**
5. Open **Build Settings** and add **"${SDK_DIR}/usr/include/libxml2"** to the **Header Search Path** section
6. Use **#import "JSClient.h"** in your source files (you don't need to do this if you're building the JasperMobile app for iOS, as the import is already there).
7. Build the project to verify installation is successful.

III. Disable Inclusion of ASIHTTPRequest library

If you chose the **linking** install approach, but you have already installed the **ASIHTTPRequest** library, you can disable including it for the JS Mobile SDK (this will reduce the size of the **jasperserver-mobile-sdk-ios.a** file when you build).

To disable including **ASIHTTPRequest** library for SDK:

1. Open **jasperserver-mobile-sdk-ios** target settings
2. Go to **Build Phases** tab

3. In section **Target Dependencies** remove **jasperserver-mobile-ASIHTTPRequest-ios** target
4. In section **Link Binary WIth Libraries** remove **jasperserver-mobile-ASIHTTPRequest-ios.a** library file

This will build the SDK without including the **ASIHTTPRequest** directly into your library file.