

Creating Great Mobile Libraries

Daniel Tull

What is a library?

- * A collection of code to do a particular task
 - * Better to do one thing well
- * Often stand-alone, sometimes has dependencies

Doesn't have to be big



<http://novelideasmanly.blogspot.co.uk/2007/03/library-afloat-sets-sail.html>

Fit for purpose



<http://www.eifl.net/ghana-library-board-mobile-library-service>

Make it stand out



<http://en.wikipedia.org/wiki/File:NSPublicMobileLibrary.jpg>

Wait for Apple to replace it



<http://www.hdelectriccompany.com/latest-news/hd-electric-demo-van-becomes-bookmobile.htm>

Wait for Apple to replace it



It's aerodynamic

A little bit shiny

It runs on
electric!

<http://www.hdelectriccompany.com/latest-news/hd-electric-demo-van-becomes-bookmobile.htm>

Version Control

- * For this I will assume git
- * Bring in libraries with git submodules, svn externals etc

Drag & Drop Files

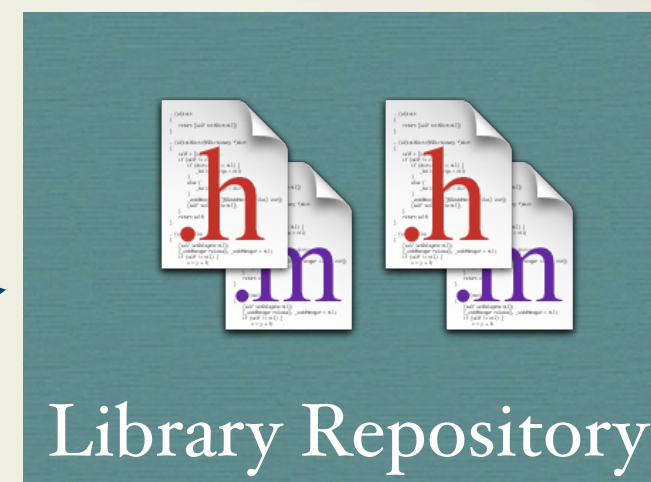
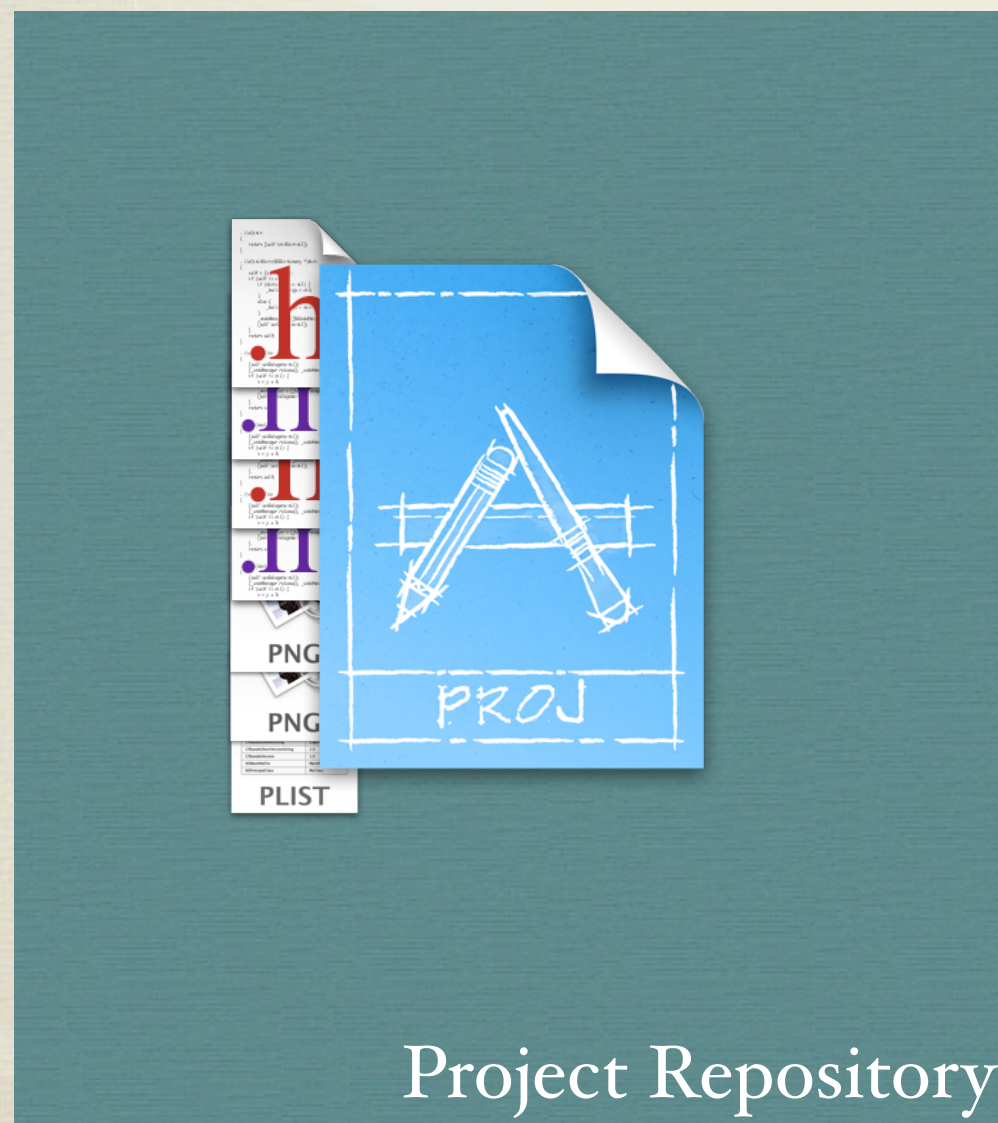
Drag & Drop Files

Create a repository and add files

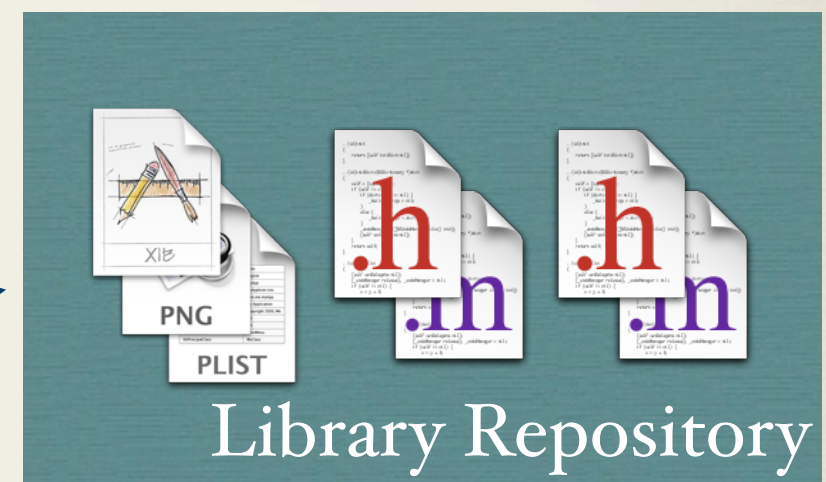
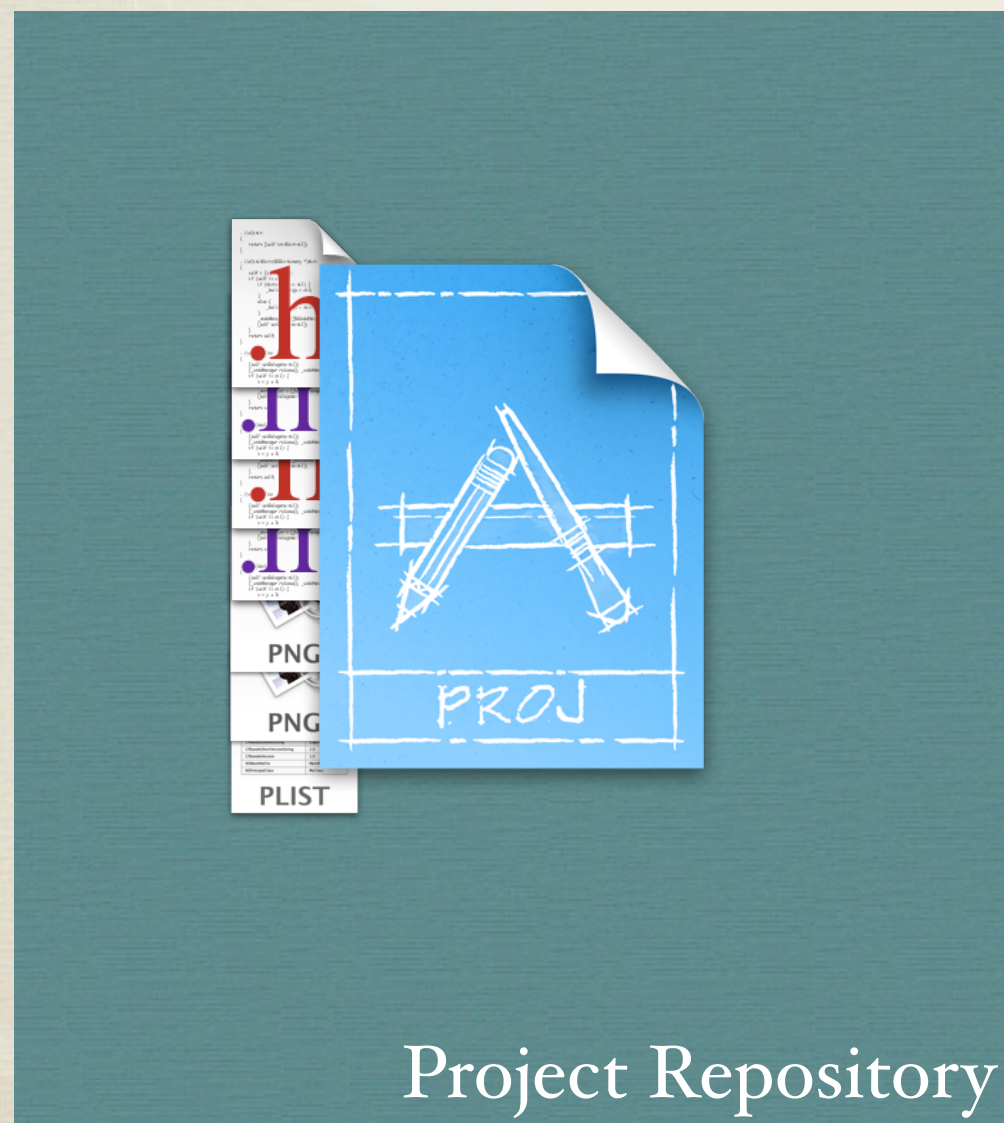
Add submodule reference to library repository

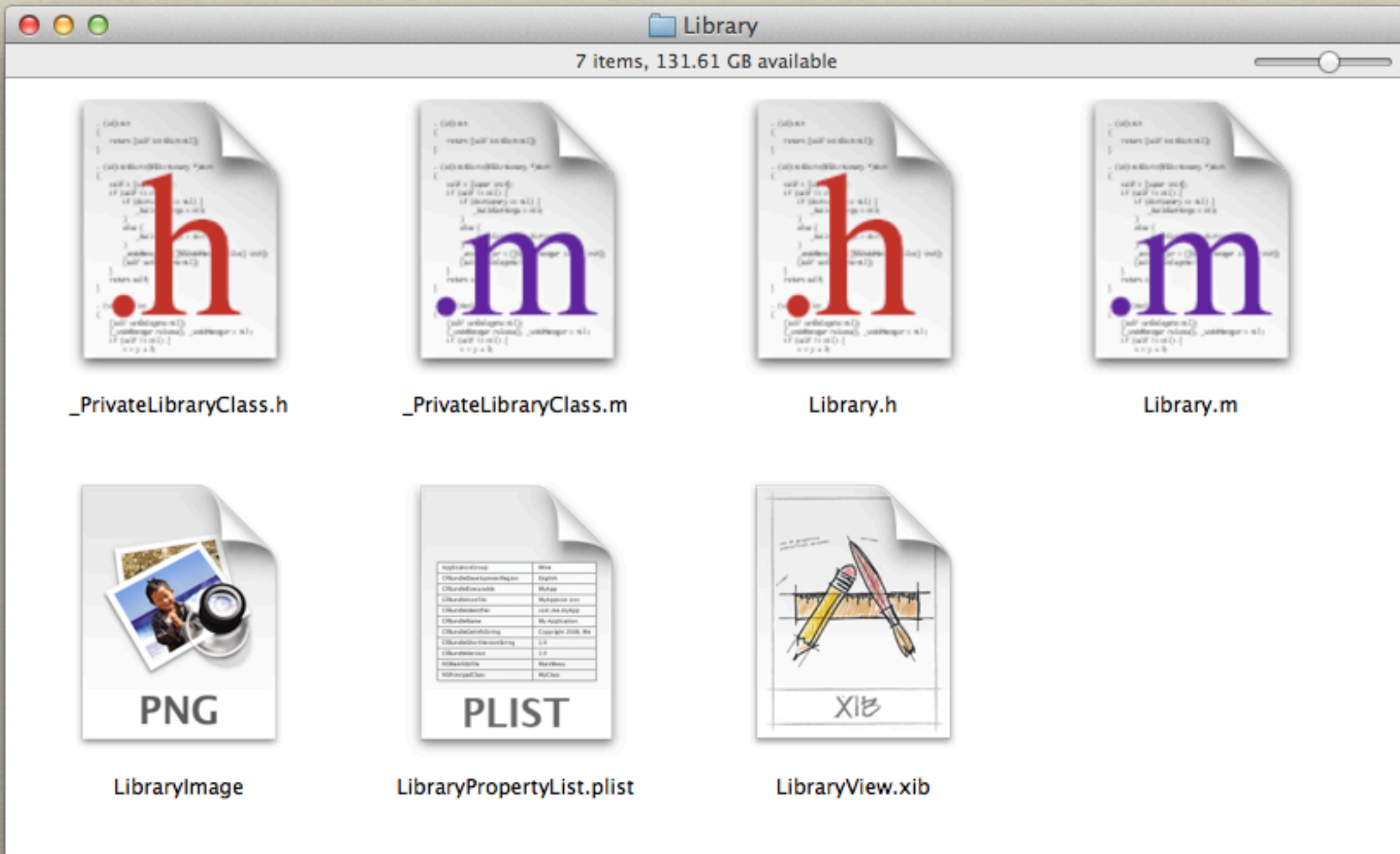
Referencing projects drag the required files in

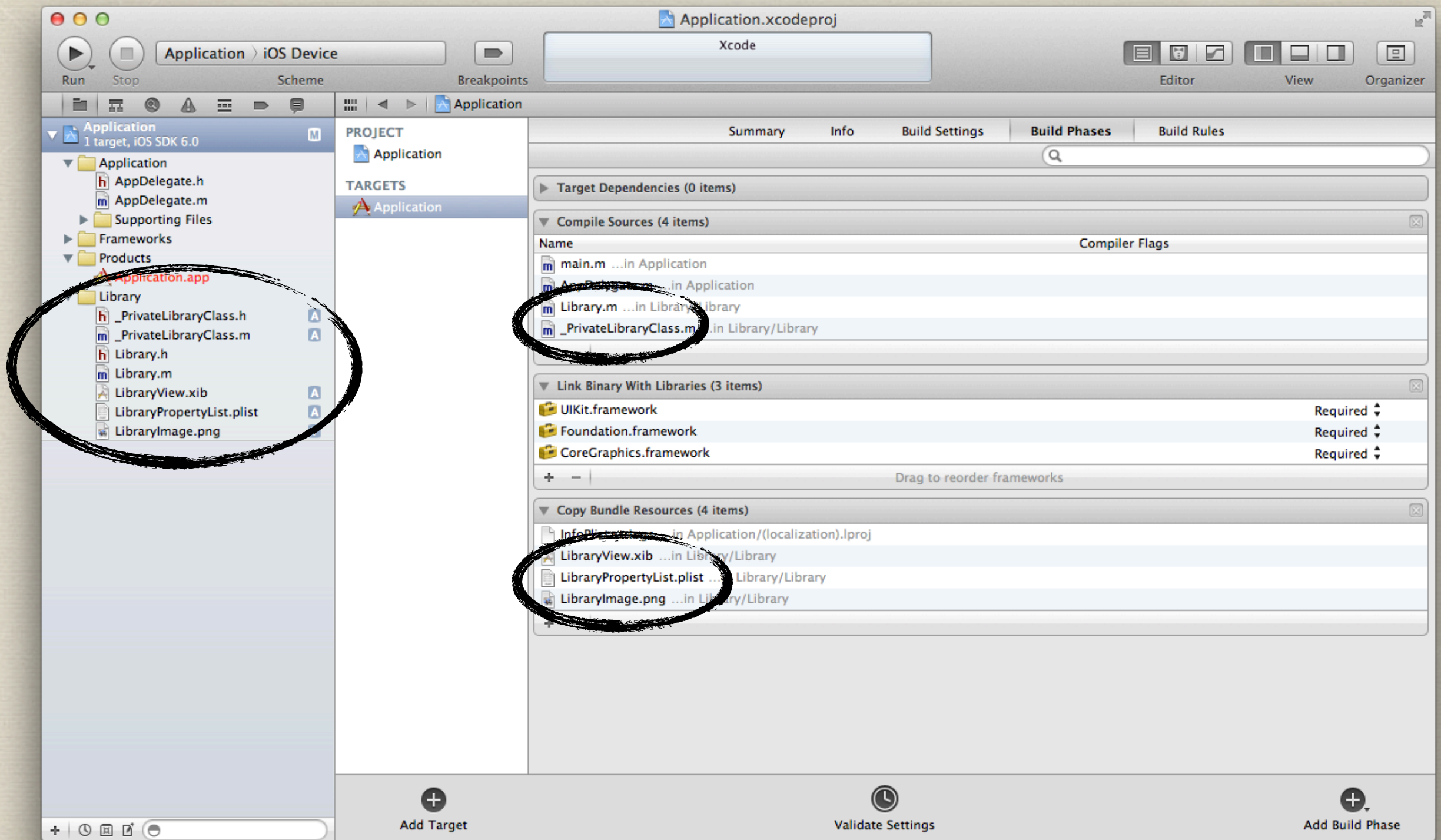
Drag & Drop Files



Drag & Drop Files







Drag & Drop Files

- ✓ Really simple to create
- ✓ Easy to drop in for the user

Drag & Drop Files

- ✗ Need to know whether it's written for ARC
- ✗ Need to know about file changes
- ✗ Library unit tests not run
- ✗ Warnings show up in app build
- ✗ Users can see and use private library classes

Static Libraries

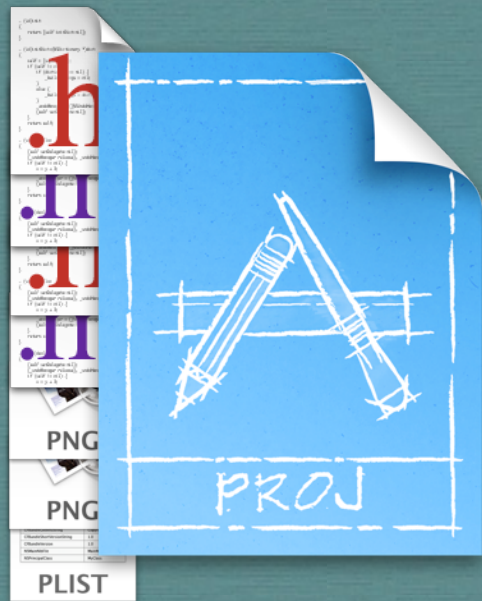
Static Libraries

Create a new static library project

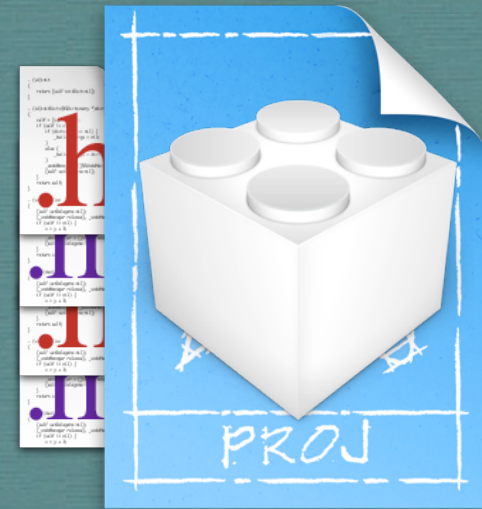
Add classes to the static library target

Link the library into the app project

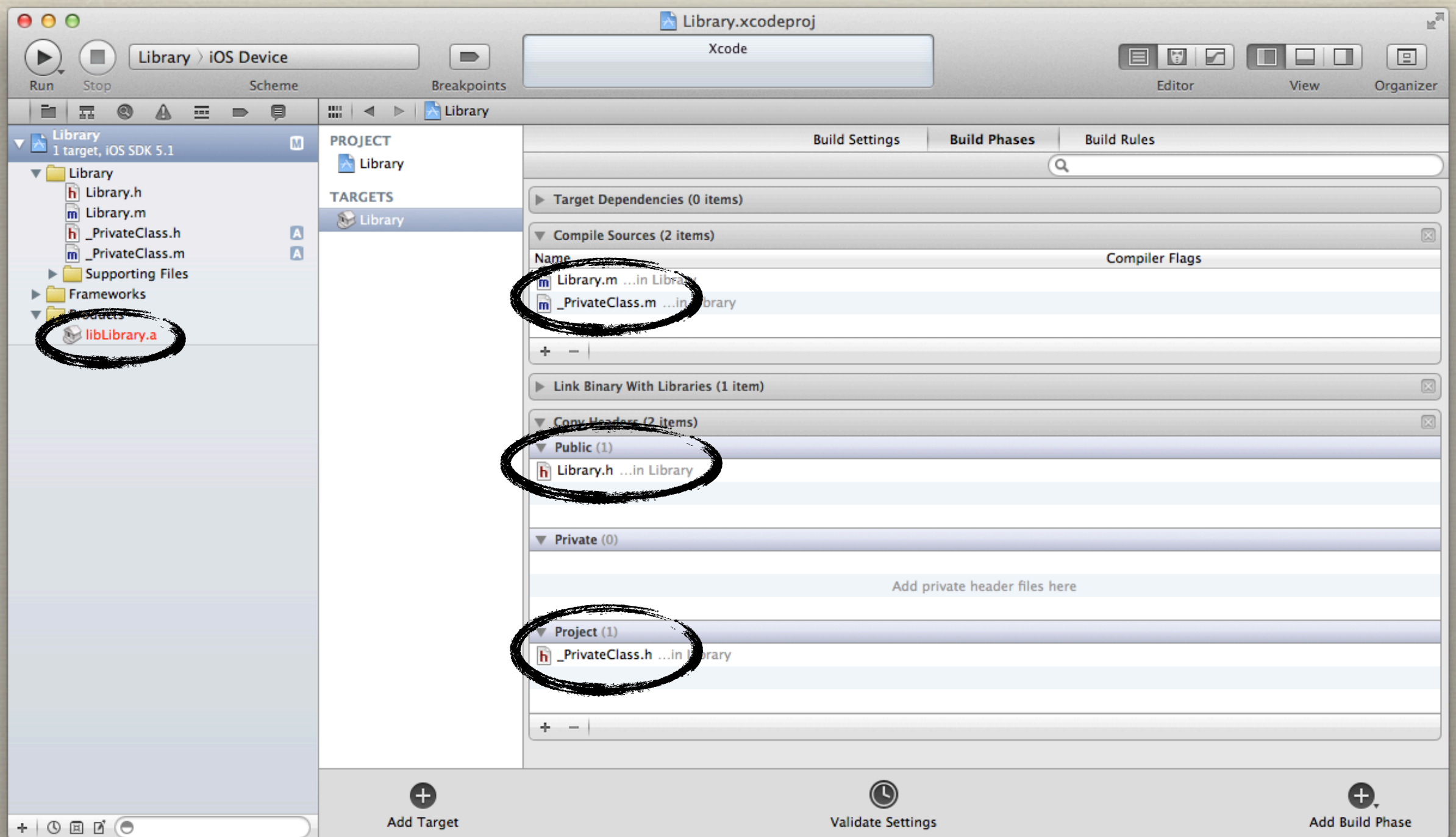
Static Libraries

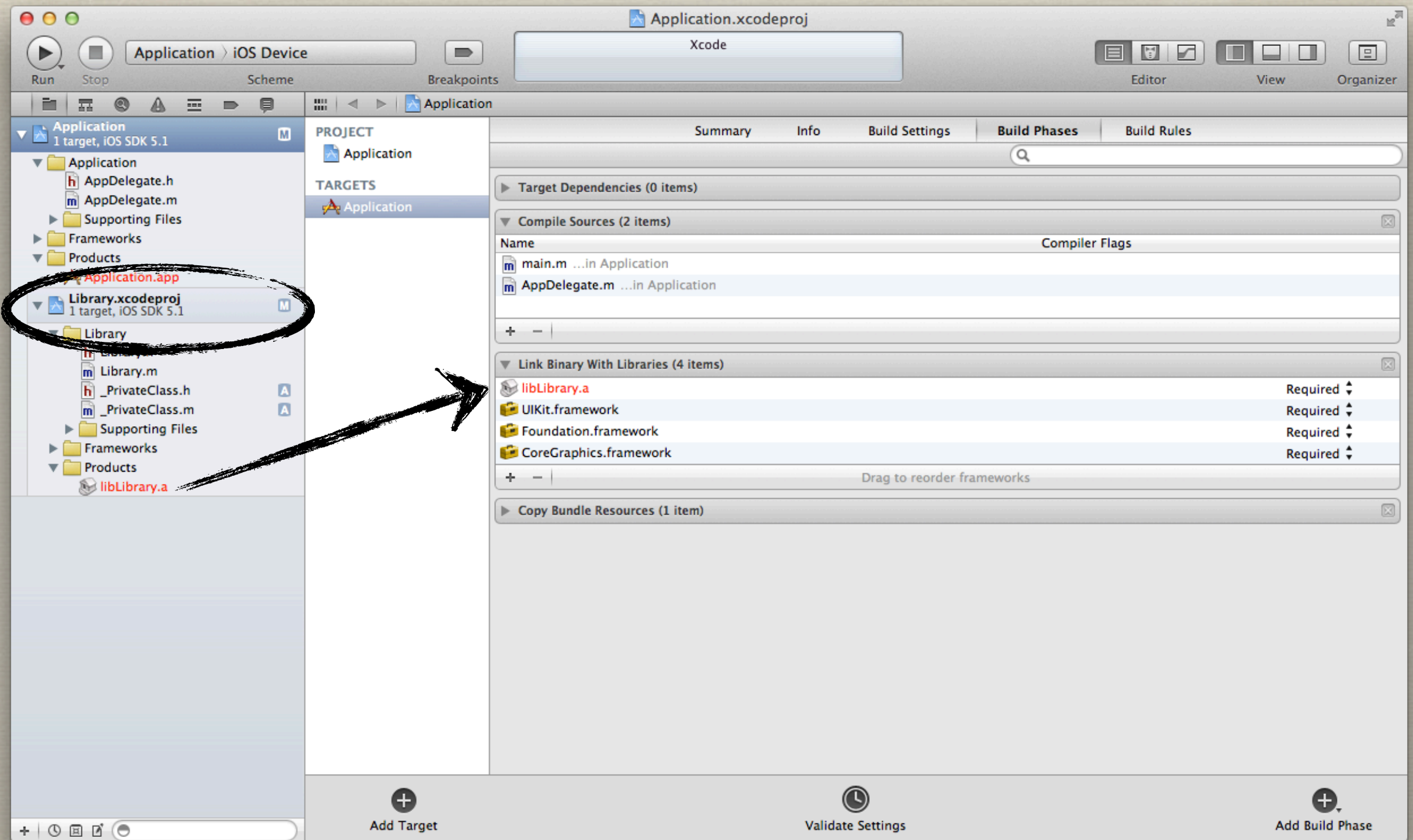


Project Repository

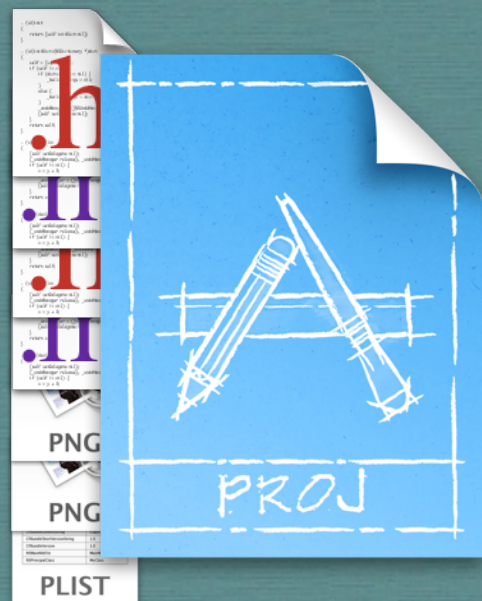


Library Repository

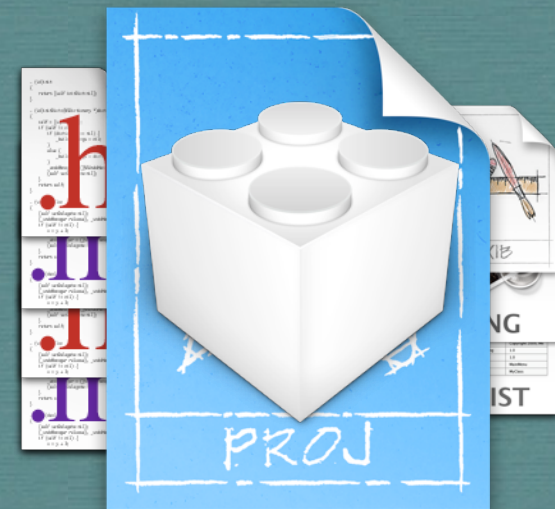




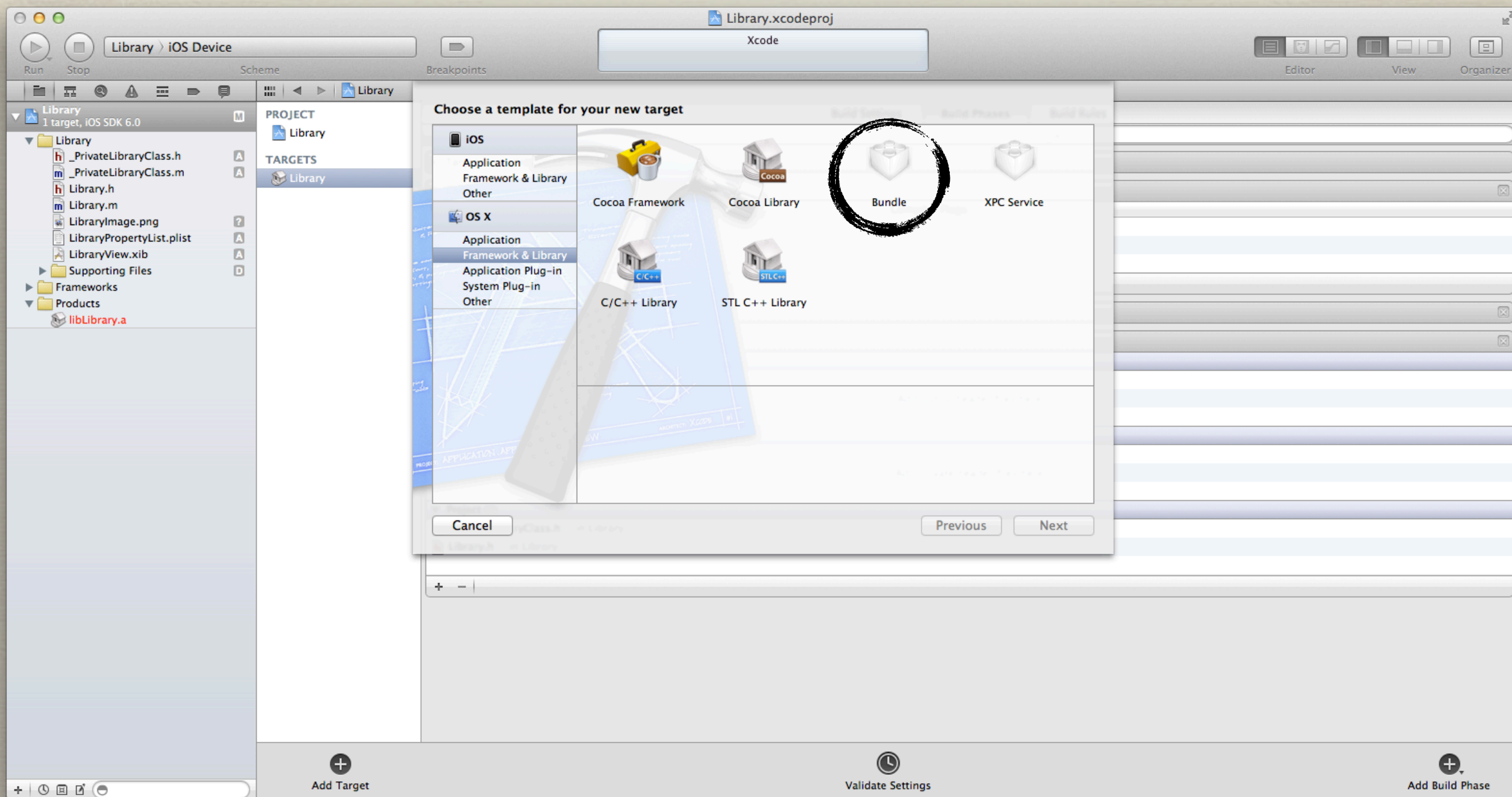
Static Libraries

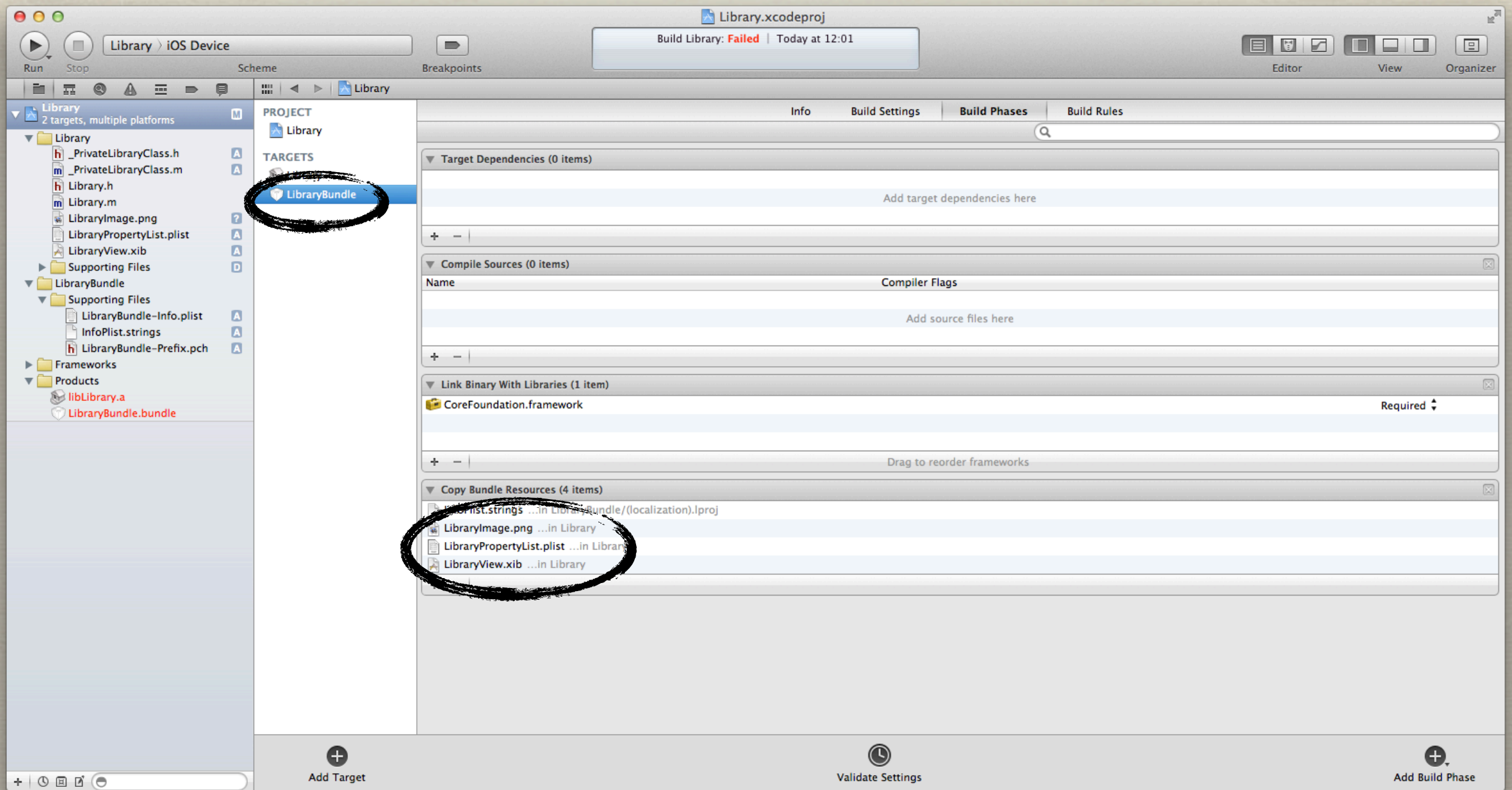


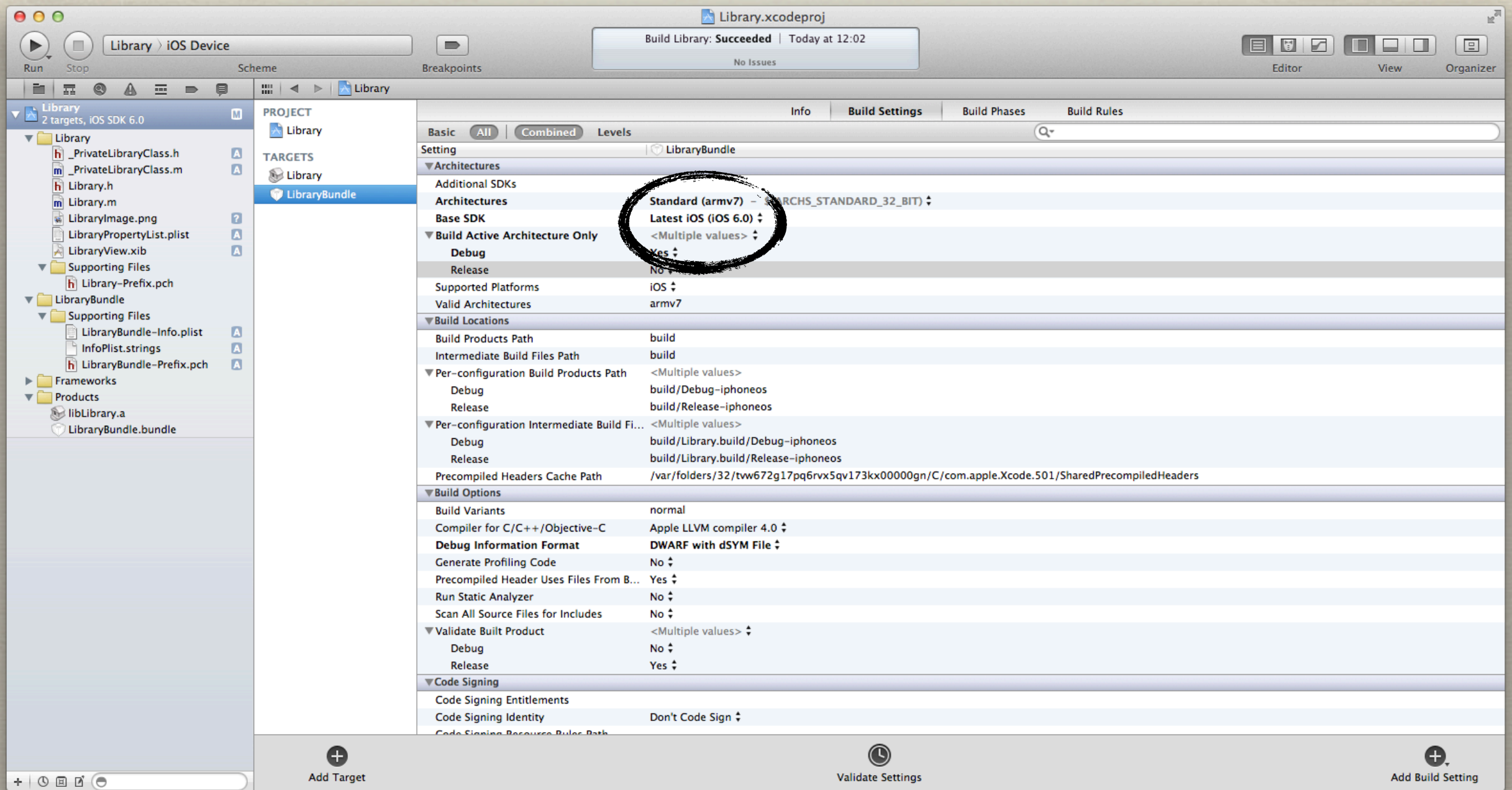
Project Repository

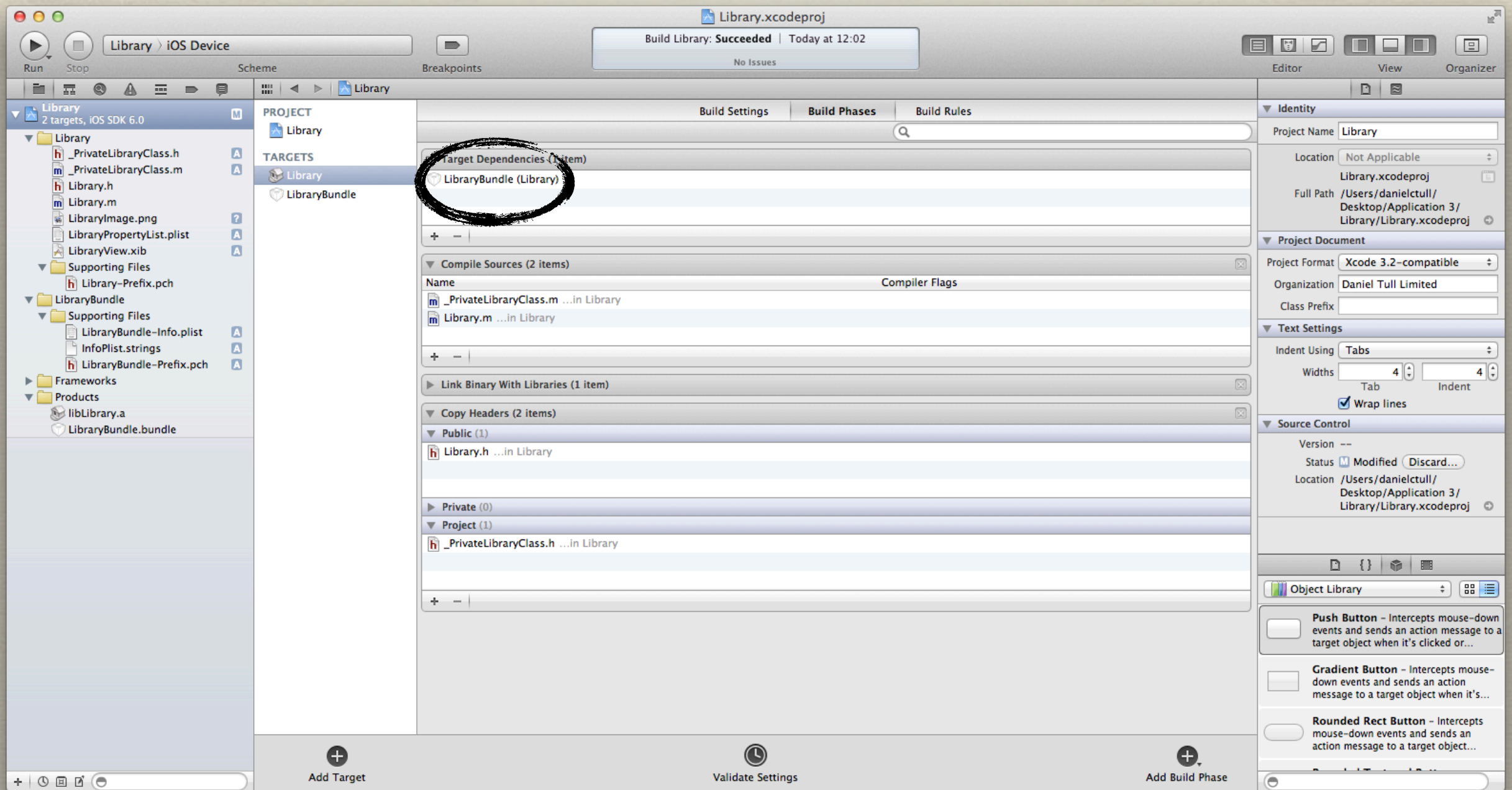


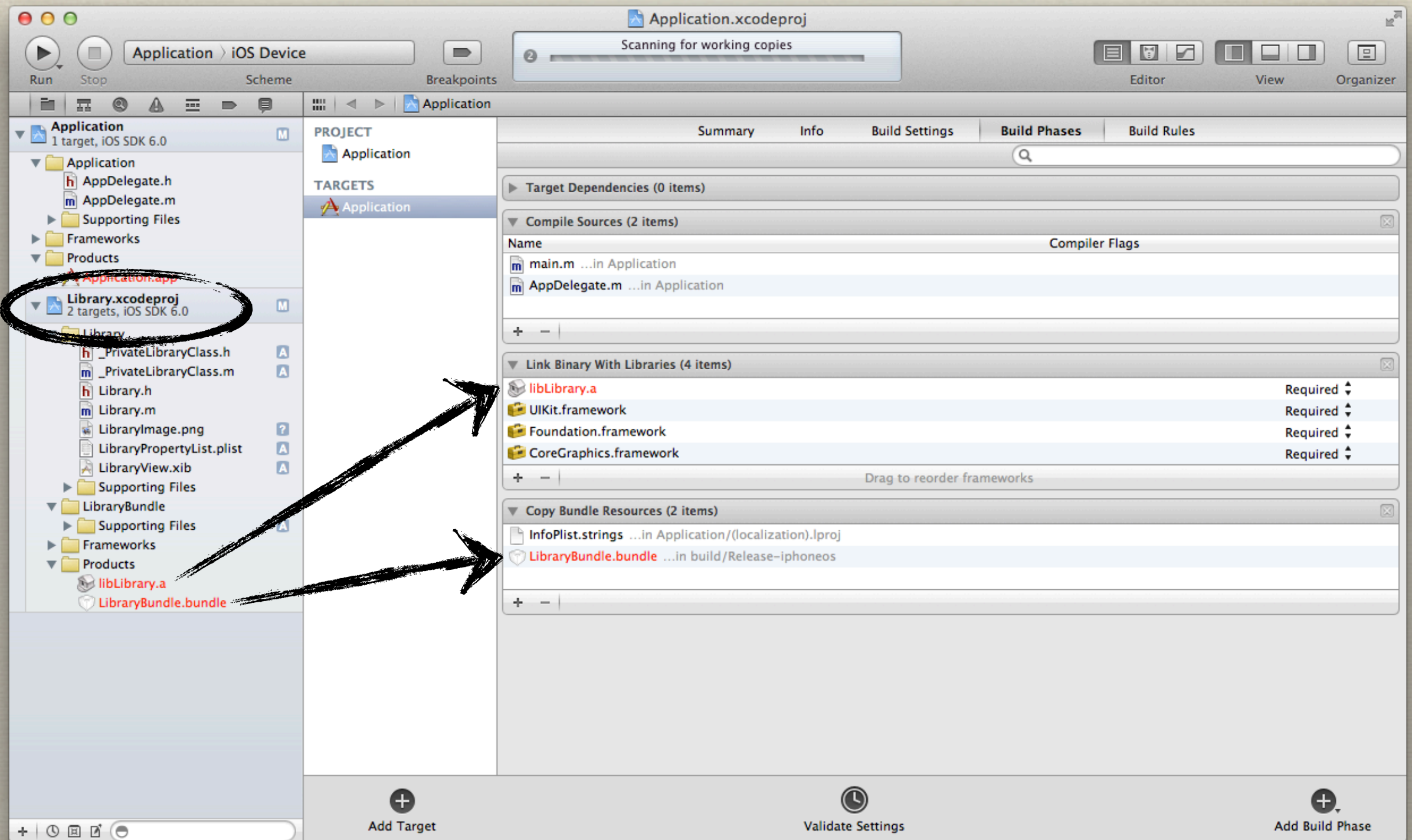
Library Repository











Static Libraries

```
+ (NSBundle *)bundle {  
  
    NSFileManager *fm = [NSFileManager new];  
    NSURL *mainBundleURL = [[NSBundle mainBundle] bundleURL];  
    NSDirectoryEnumerator *enumerator = [fm enumeratorAtURL:mainBundleURL  
                                         includingPropertiesForKeys:nil  
                                         options:NSDirectoryEnumerationSkipsHiddenFiles  
                                         errorHandler:NULL];  
  
    for (NSURL *URL in enumerator)  
        if ([[URL lastPathComponent] isEqualToString:@"LibraryBundle.bundle"])  
            return [NSBundle bundleWithURL:URL];  
  
    return nil;  
}
```


Static Libraries

- ✓ New files will be pulled in
- ✓ Guaranteed to work with ARC **and** MRR
- ✓ Library unit tests are run when you build the app
- ✓ Warnings are contained to library target
- ✓ Private classes are hidden

Static Libraries

- ✗ A little overhead to set up
- ✗ Recursive dependencies can be a little tricky

CocoaPods

CocoaPods

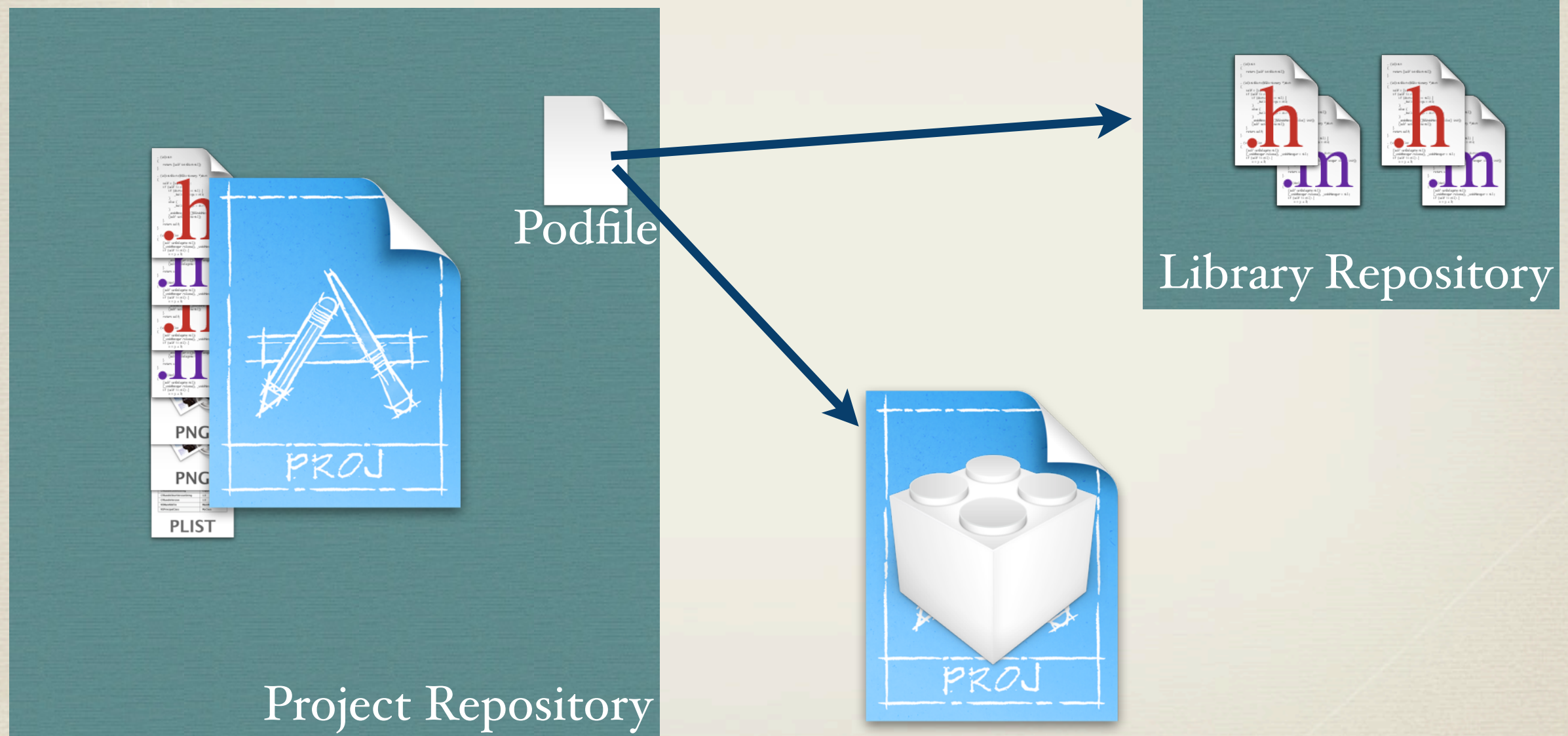
Install CocoaPods

Create a Podfile to specify the libraries

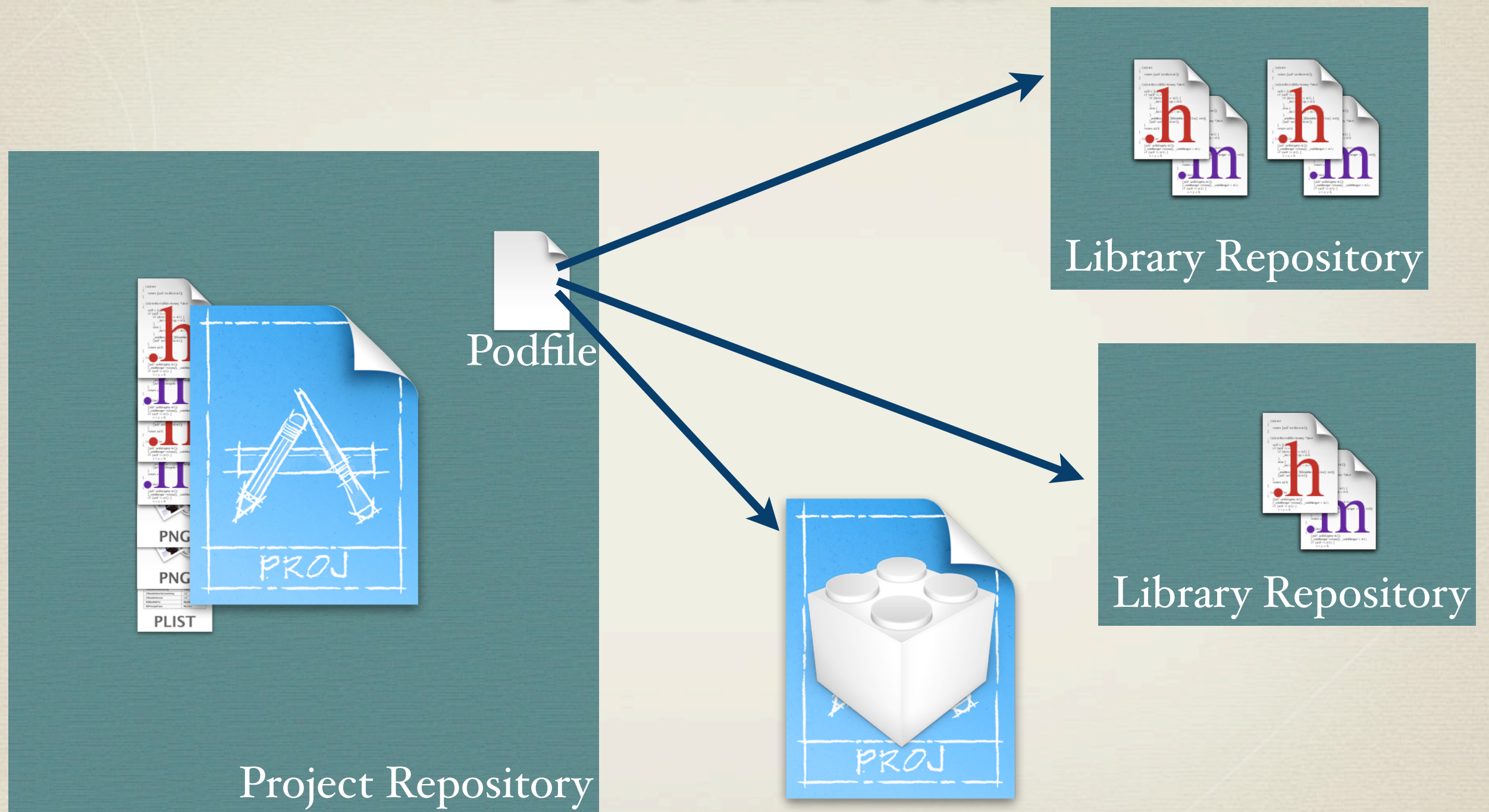
Run cocoapods

Use the created workspace instead of your app project

CocoaPods



CocoaPods



CocoaPods

- ✓ Independent of version control
 - ✓ Link to mercurial repositories from git
- ✓ Handles dependencies
- ✓ Warnings are contained to CocoaPods static library target

CocoaPods

- ✗ Complex to setup
- ✗ Requires knowledge of Ruby
- ✗ All members need CocoaPods to build and run app
- ✗ Unit tests likely not brought in with library code

	Drag and Drop	Static Library	CocoaPods
Contained warnings	✗	✓	✓
Build with unit tests	✗	✓	✗
Build upon clone	✓	✓	✗
VCS independent	✗	✗	✓
Dependencies	✗	✗	✓
Hidden classes	✗	✓	✗
File handling	✗	✓	✓
Resources	✓	✗ (✓)	✓

github.com/danielctull/DCTConnectionController/tree/develop

github.com/danielctull/DCTImageSizing

github.com/danielctull/DCTCoreDataStack

github.com/danielctull/DCTNetworkActivityIndicatorController

github.com/danielctull/DCTTextFieldValidator



Daniel Tull

@danielctull

danieltull.co.uk