

Linking Libraries

Not every app uses all the native capabilities, and including the code to support all those features would impact the binary size... But we still want to support adding these features whenever you need them.

With that in mind we exposed many of these features as independent static libraries.

For most of the libs it will be as quick as dragging two files, sometimes a third step will be necessary, but no more than that.

NOTE

All the libraries we ship with React Native live in the `Libraries` folder in the root of the repository. Some of them are pure JavaScript, and you only need to `require` it. Other libraries also rely on some native code, in that case you'll have to add these files to your app, otherwise the app will throw an error as soon as you try to use the library.

Here are the few steps to link your libraries that contain native code

Automatic linking

Install a library with native dependencies:

```
npm install <library-with-native-dependencies> --save
```

INFO

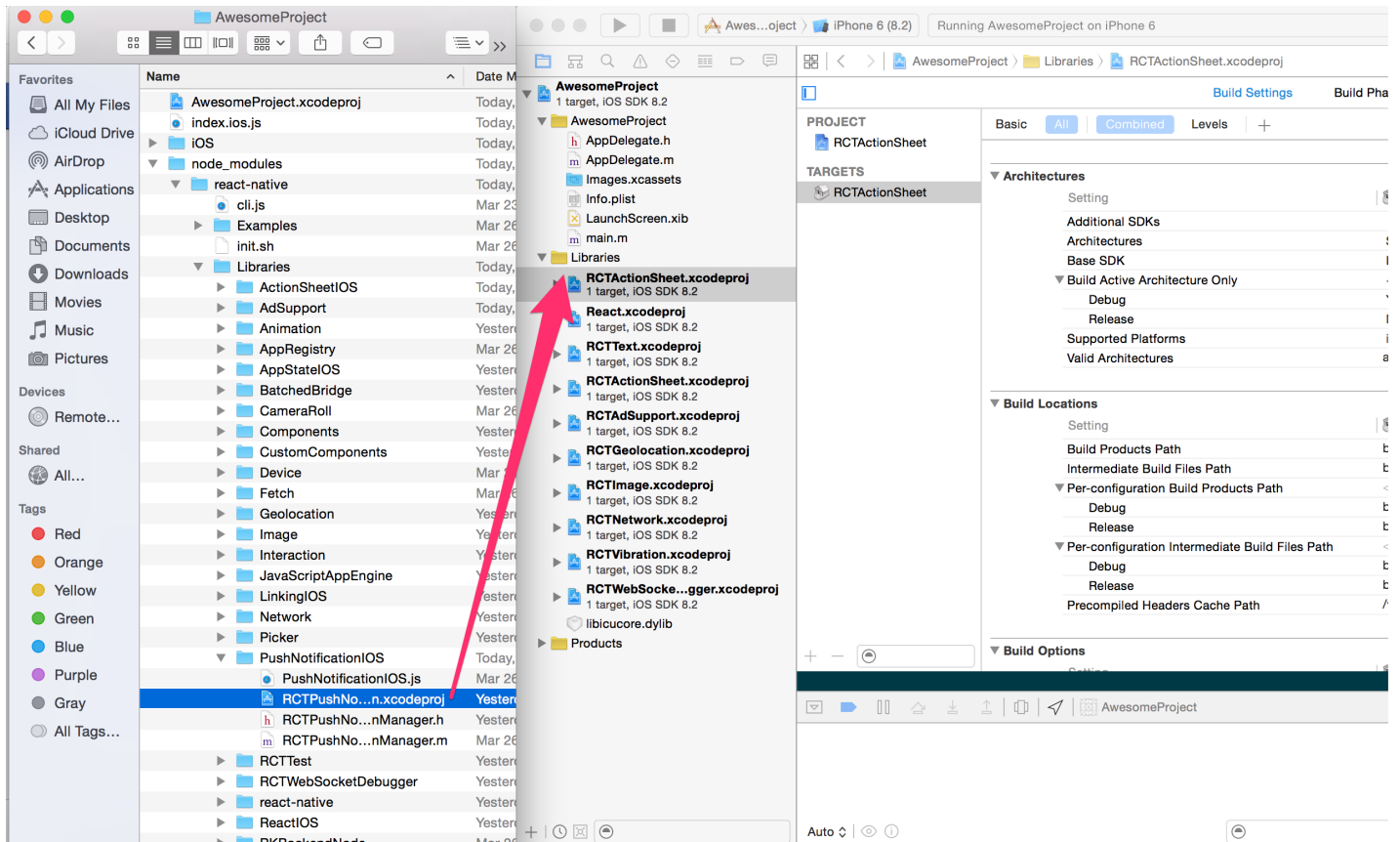
`--save` or `--save-dev` flag is very important for this step. React Native will link your libs based on `dependencies` and `devDependencies` in your `package.json` file.

That's it! Next time you build your app the native code will be linked thanks to the autolinking mechanism.

Manual linking

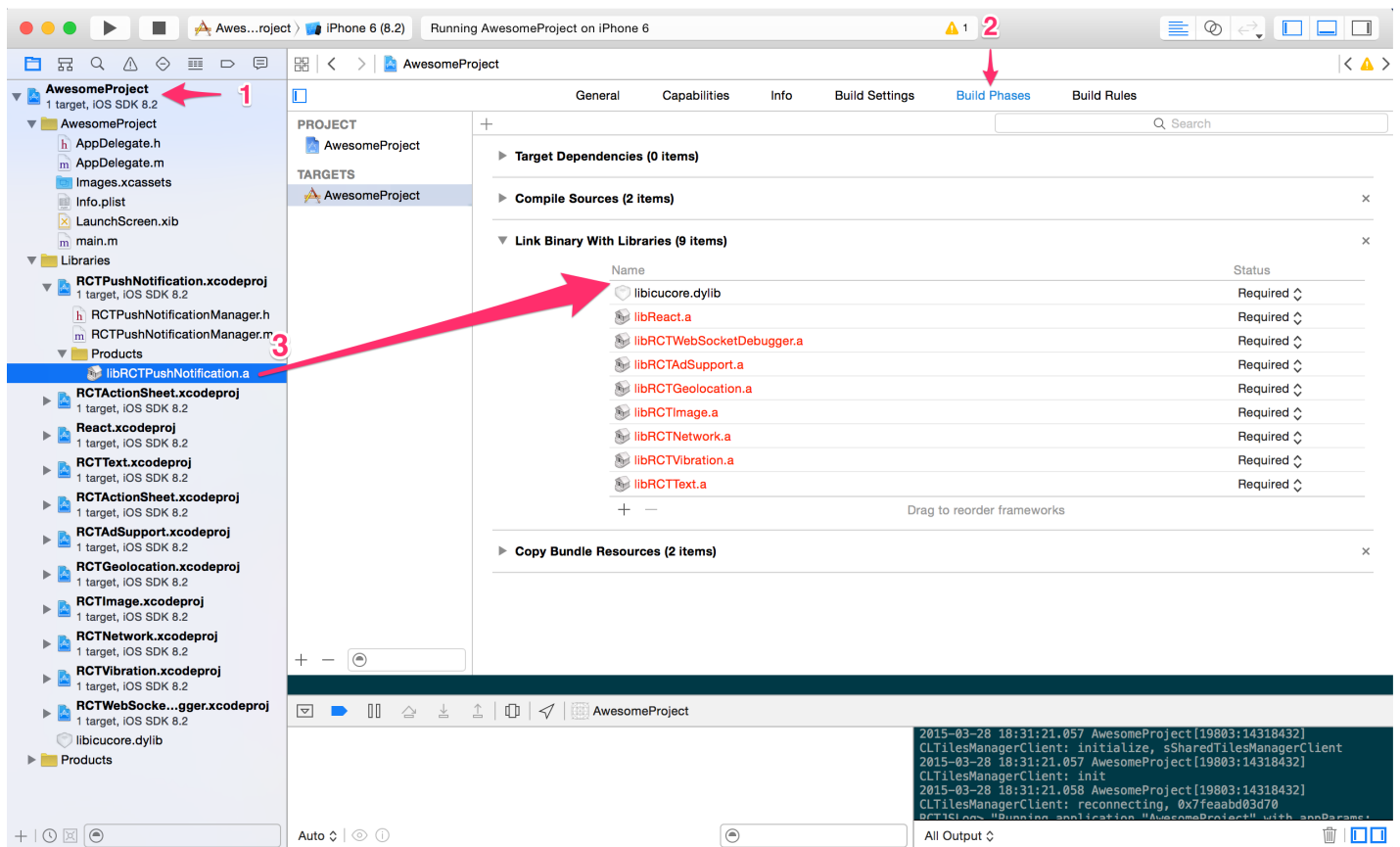
Step 1

If the library has native code, there must be an `.xcodproj` file inside its folder. Drag this file to your project on Xcode (usually under the `Libraries` group on Xcode);



Step 2

Click on your main project file (the one that represents the `.xcodproj`) select `Build Phases` and drag the static library from the `Products` folder inside the `Library` you are importing to `Link Binary With Libraries`



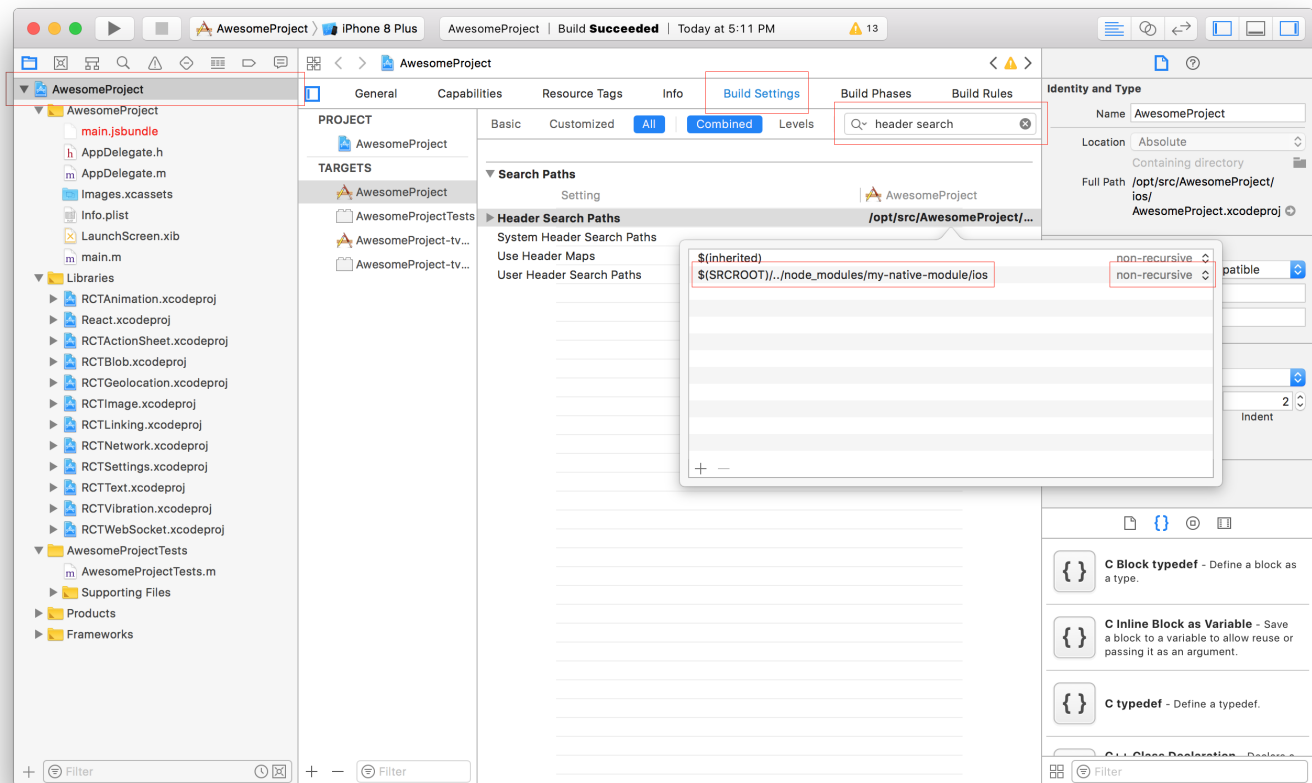
Step 3

Not every library will need this step, what you need to consider is:


Do I need to know the contents of the library at compile time?

What that means is, are you using this library on the native side or only in JavaScript? If you are only using it in JavaScript, you are good to go!

If you do need to call it from native, then we need to know the library's headers. To achieve that you have to go to your project's file, select **Build Settings** and search for **Header Search Paths**. There you should include the path to your library. (This documentation used to recommend using `recursive`, but this is no longer recommended, as it can cause subtle build failures, especially with CocoaPods.)



Is this page useful?  

 Edit this page

Last updated on **Jun 21, 2023**