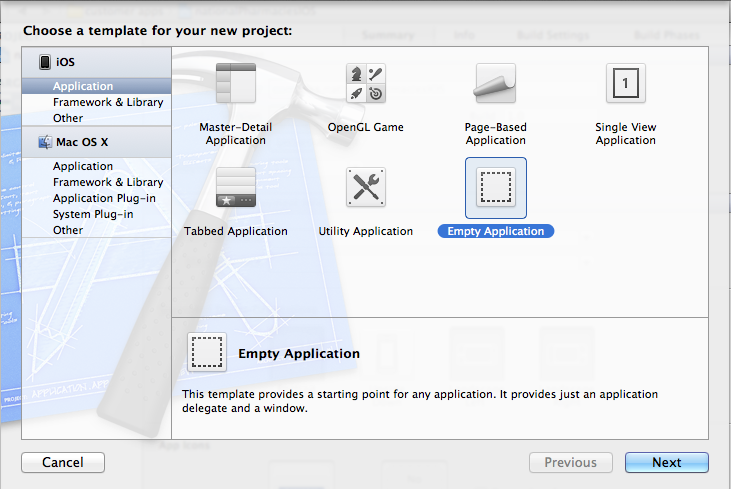
**New Project Setup for a nomadPOS based app**

Retail app

Created on: June 19, 2012 Author: Julieta Delfouneso

Last amended on: Oct 5, 2012 Amended by: Julieta Delfouneso

1. Create a new project in Xcode.
   1. Load Xcode.
   2. Click on **File**, **New**, **Project…**
   3. In the **Choose a template for your new project:** screen, select Application under iOS from the list on the left hand side.
   4. Select **Empty Application** on the right hand side and click on **Next**.

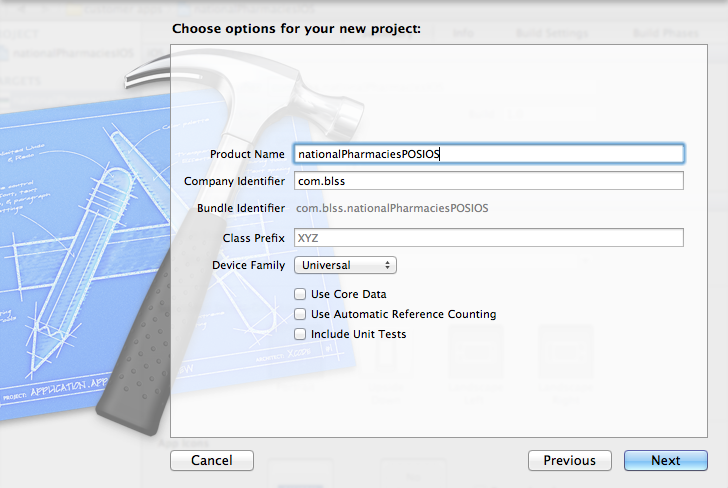


* 1. On the **Choose options for your new project:** screen, fill in the following data and click on **Next.**

Product Name: <clientName>POSIOS

Company Identifier: com.blss

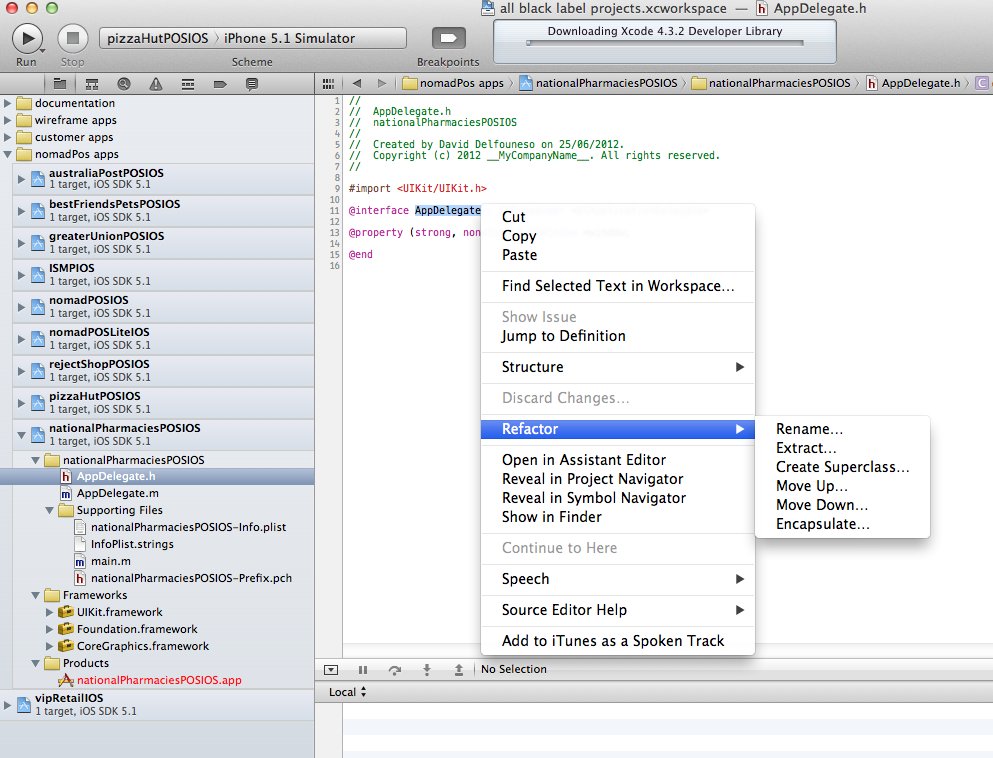
Device Family: Universal



* 1. Save the project in its own folder called “<**clientName>POSIOS**”. Customer app folder name has an “IOS” suffix while Retail app folder name has a “POSIOS” suffix.
  2. Make sure that **Create local git repository for this project** is **NOT** checked.
  3. Group should be “nomadPOS apps”. Click on **Create**.

create app.tiff

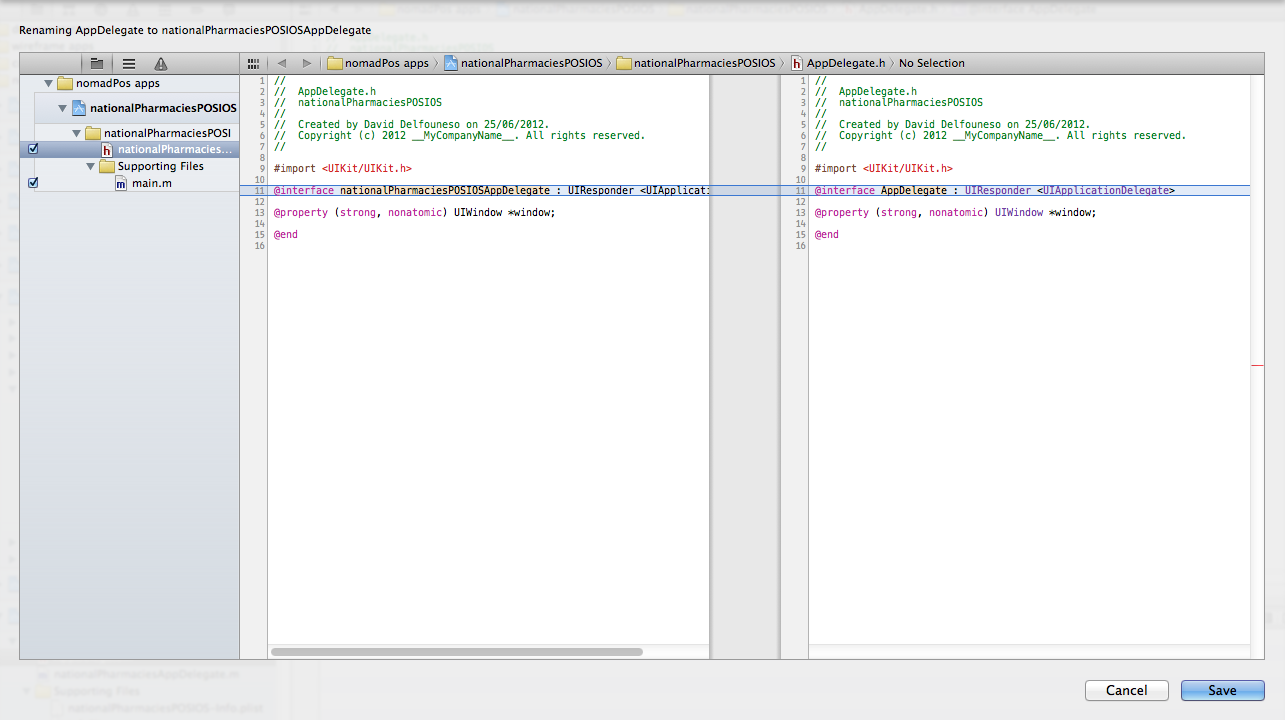
1. Edit the file **<clientName>POSIOSAppDelegate.h**
   1. Under the newly created project select AppDelegate.h. This will open the file on the right hand side.
   2. On the right hand side highlight the interface name i.e. highlight the word “AppDelegate”
   3. Right click the highlighted interface name, select **Refactor** from the menu and then click on **Rename.**



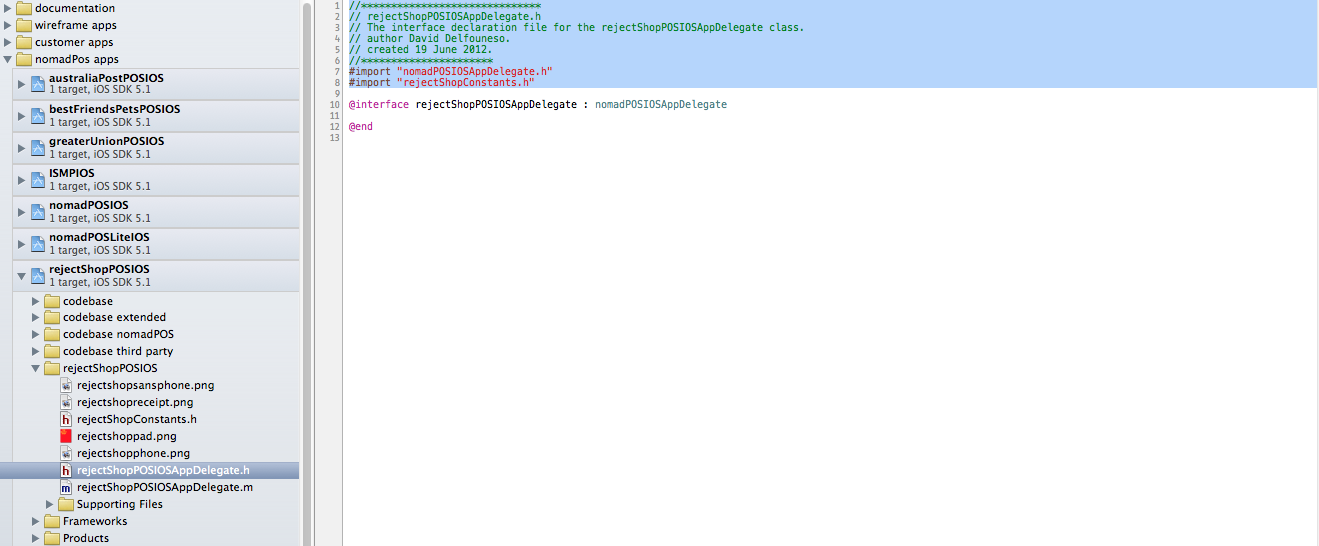
* 1. Rename AppDelegate to: **<clientName>AppDelegate**
  2. Make sure that the **Rename related files** box is checked and click on **Preview**.

1.tiff

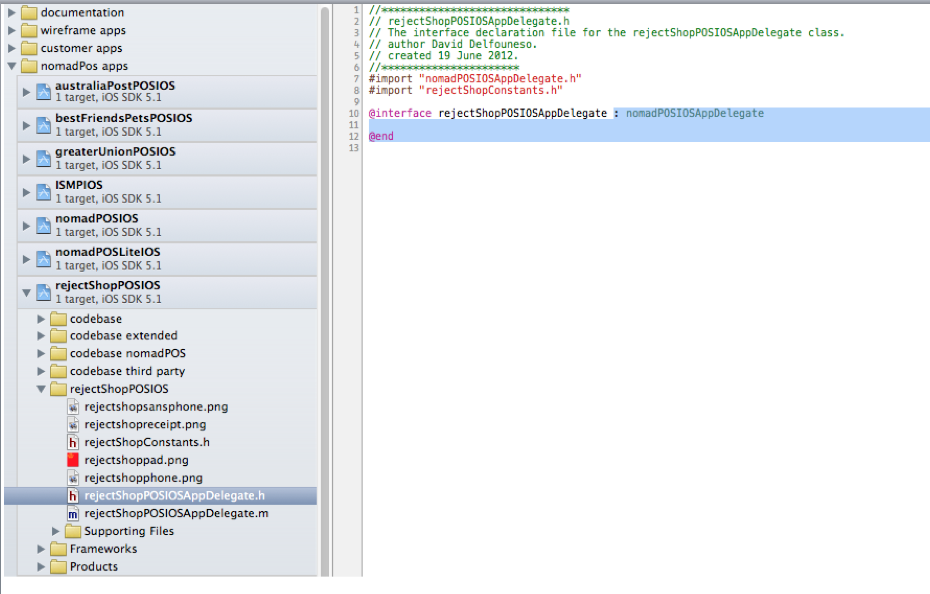
* 1. On the **Renaming AppDelegate to <clientName>AppDelegate** screen, Click on **Save**.



* 1. Copy comments and import declarations. To do this:
* Open an existing working app e.g. Reject Shop.
* Open the **rejectShopPOSIOSAppDelegate.h** file
* Copy from the top of the file up to, but not including, the Interface declaration.



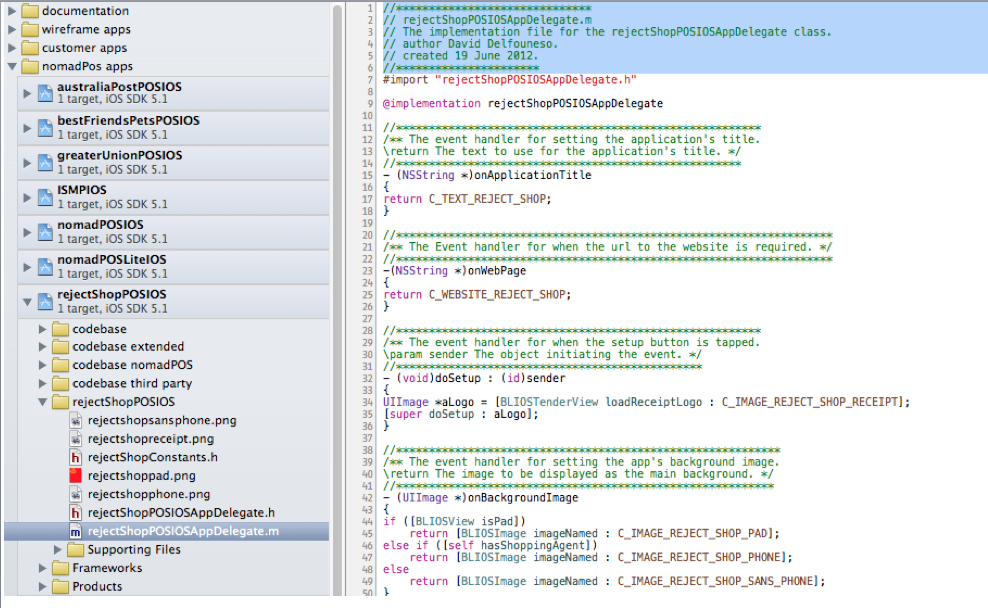
* 1. Open the new project’s AppDelegate.h file.
  2. Replace everything in the file from the top up to but not including the Interface declaration.
  3. Replace the previous client’s name with the new client’s name.
  4. Change the author.
  5. Change the date.
  6. Make sure that the line of asterisks at the top is aligned with the first line of the comments.
  7. Make sure that the line of asterisks at the bottom is aligned with the last line of the comments.
  8. Copy the body of the Interface declaration code. To do this:
* Go back to the opened existing working app e.g. Reject Shop
* Go back to the **rejectShopPOSIOSAppDelegate.h** file
* At the Interface declaration, highlight from “**:**” up to the end of the declaration and copy.



* 1. Replace the corresponding texts in the new project’s AppDelegate file.
  2. The new project’s **<clientName>AppDelegate.h** file should look similar to the image below.



1. Edit the file **<clientName>POSIOSAppDelegate.m**
   1. Go back to the opened existing working app e.g. Reject Shop
   2. Open the **rejectShopPOSIOSAppDelegate.m** file.
   3. Copy the first set of comments right at the top of the file.



* 1. Go back to the new project’s Implementation file i.e. **<clientName>POSIOSAppDelegate.m**
  2. Paste the copied comments into the Implementation file.
  3. Change the client name.
  4. Change the date.
  5. Change the author.
  6. Make sure that the line of asterisks at the top is aligned with the first line of the comments.
  7. Make sure that the line of asterisks at the bottom is aligned with the last line of the comments.
  8. Go back to the opened existing working app e.g. Reject Shop
  9. Go back to the **rejectShopPOSIOSAppDelegate.m** file
  10. Select everything from the line below the “**@implementation**”
  11. Copy the selected lines.
  12. Go back to the new project’s Implementation file i.e. **<clientName>POSIOSAppDelegate.m**
  13. Replace everything from the line below “@implementation” to the end with the copied lines from the other app.
  14. Replace all occurrences of the previous client’s names with the new client’s name using the search utility.

1. Add Units
   1. Right click on the new project’s name and click on “Add files to ” <projectName>

2.tiff

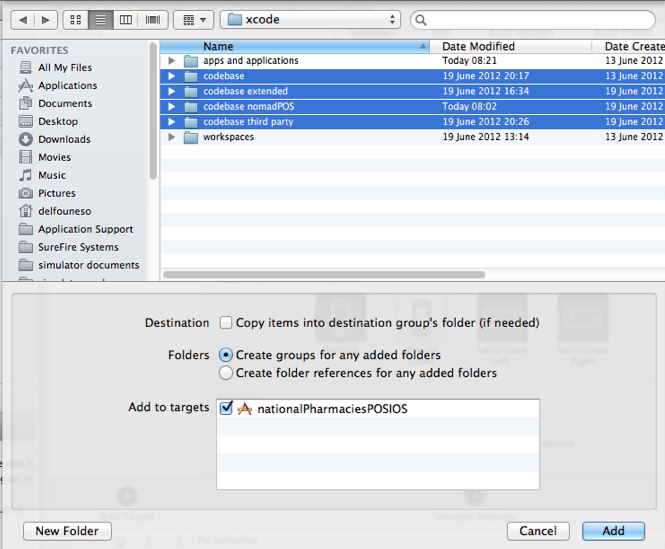
* 1. Locate the codebase folders listed below, select them and then click on **add**.

- codebase

- codebase extended

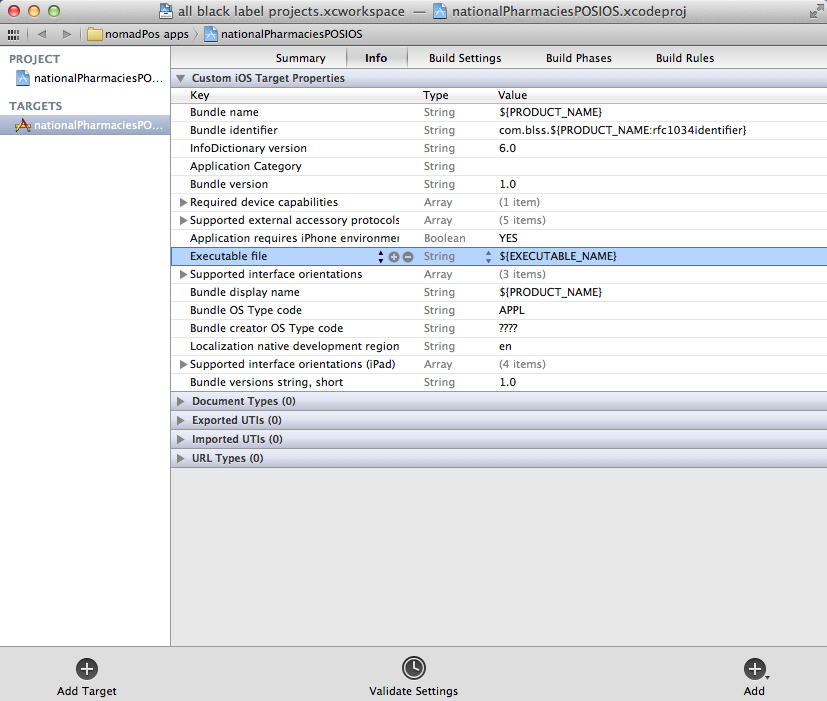
- codebase nomadPOS

- codebase third party

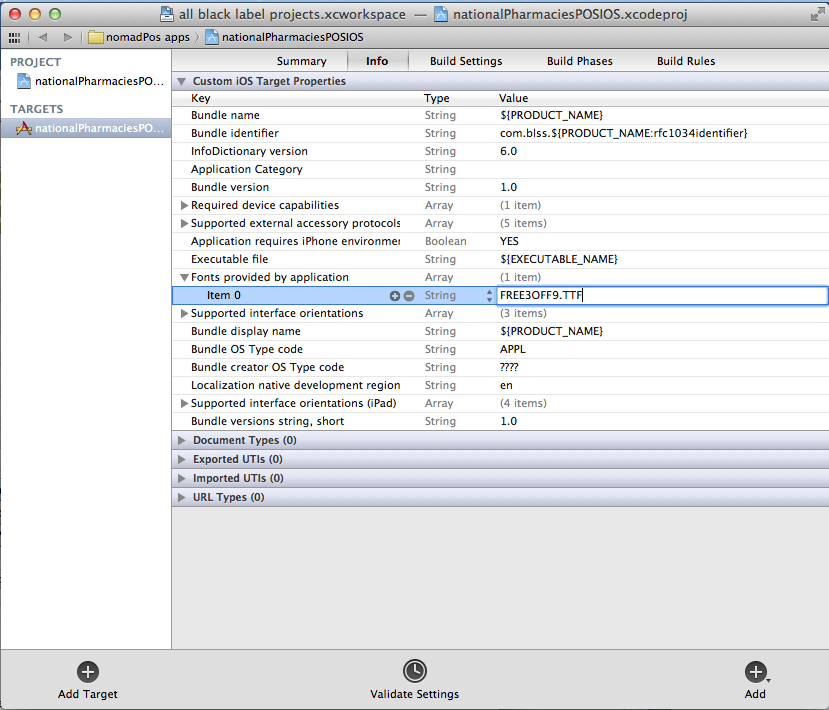


1. Add Fonts
   1. Click on the new project name to view its settings.
   2. On the project settings view, under **Targets**, select the project name.
   3. Click on the **Info** tab.
   4. Add a new **key** under **Custom iOS Target Properties**. To do this:

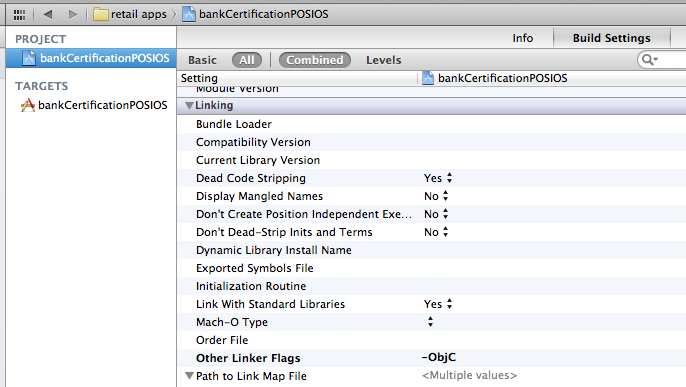
* highlight any of the existing key, e.g. **Executable file**
* click on the “**+**” sign just to the right side of the word **Executable file**. This will create a new entry just below the **key** you highlighted.



* 1. Set the value of the new **key** to “Fonts provided by Application” selecting it from the list.
  2. Expand the disclosure triangle for **Fonts provided by application** to reveal **Item 0** just below it.
  3. Set the value of **Item 0** to “FREE3OFF9.TTF” (This is a barcode font)



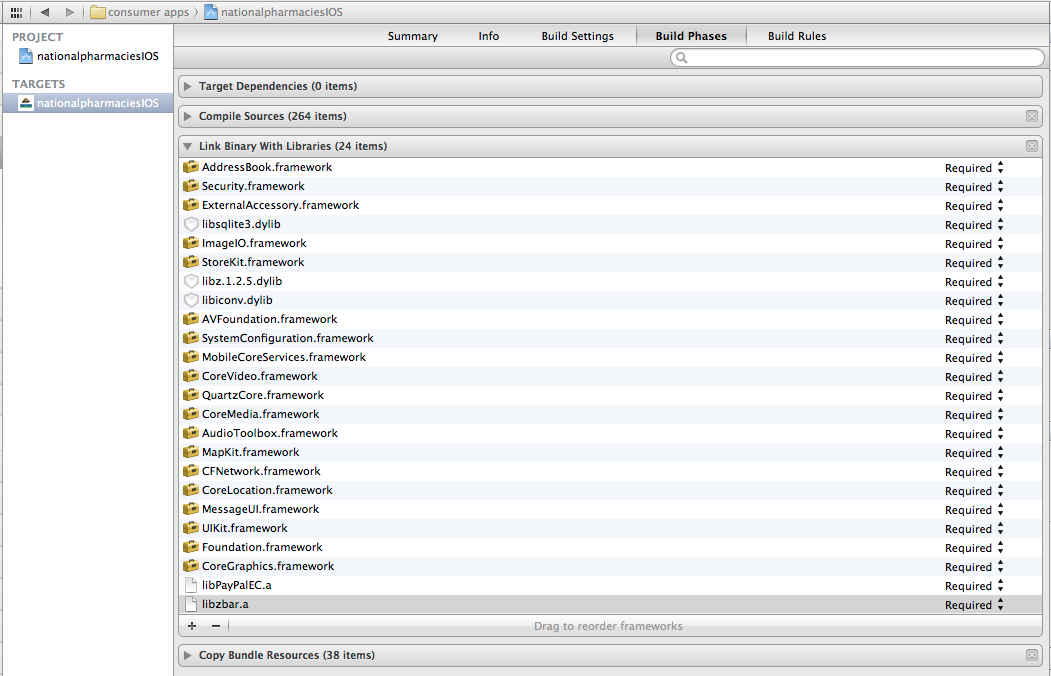
1. Set Linking properties
   1. Still in **Targets**, **Build Settings**, expand **Linking** and set **Other Linker Flags**  to “–ObjC”



1. Add Frameworks.
   1. Click on the new project name to view its settings.
   2. On the project settings view, under **Targets**, select the project name.
   3. Click on the **Build Phases** tab.
   4. Expand the disclosure triangle for the **Link Binary with Libraries.**
   5. Working from the bottom, add the following libraries. To add, click on the “**+**” sign at the bottom of the **Link Binary with Libraries** section. (NOT the “**+**” for “Add target” or “Add Build Phase” )
   6. In the **Choose frameworks and libraries to add:** window, type in the framework name e.g. MessageUI.framework, select and then click on Add.

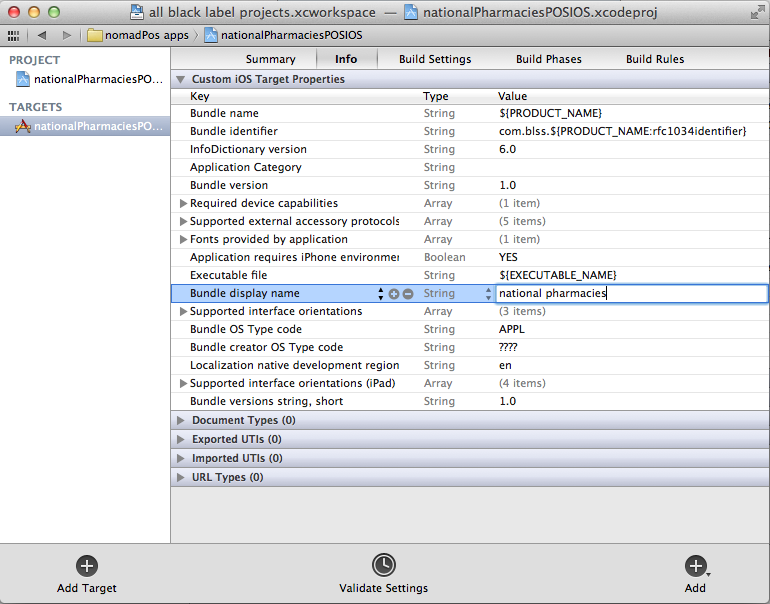
6.tiff

* 1. Repeat the above procedure to add the rest of the missing frameworks in the following order: (there should be 24 items in total)
* libPayPalEC.a
* libzbar.a
* CoreGraphics.framework
* Foundation.framework
* UIKit.framework
* MessageUI.framework
* CoreLocation.framework
* CFNetwork.framework
* MapKit.framework
* AudioToolbox.framework
* CoreMedia.framework
* MobileCoreServices.framework
* SystemConfiguration.framework
* AVFoundation.framework
* Libiconv.dylib
* Libz.1.2.5.dylib
* Libsqlite3.dylib
* ExternalAccessory.framework
* StoreKit.framework
* ImageIO.framework
* CoreVideo.framework
* QuartzCore.framework
* Security.framework
* AddressBook.framework

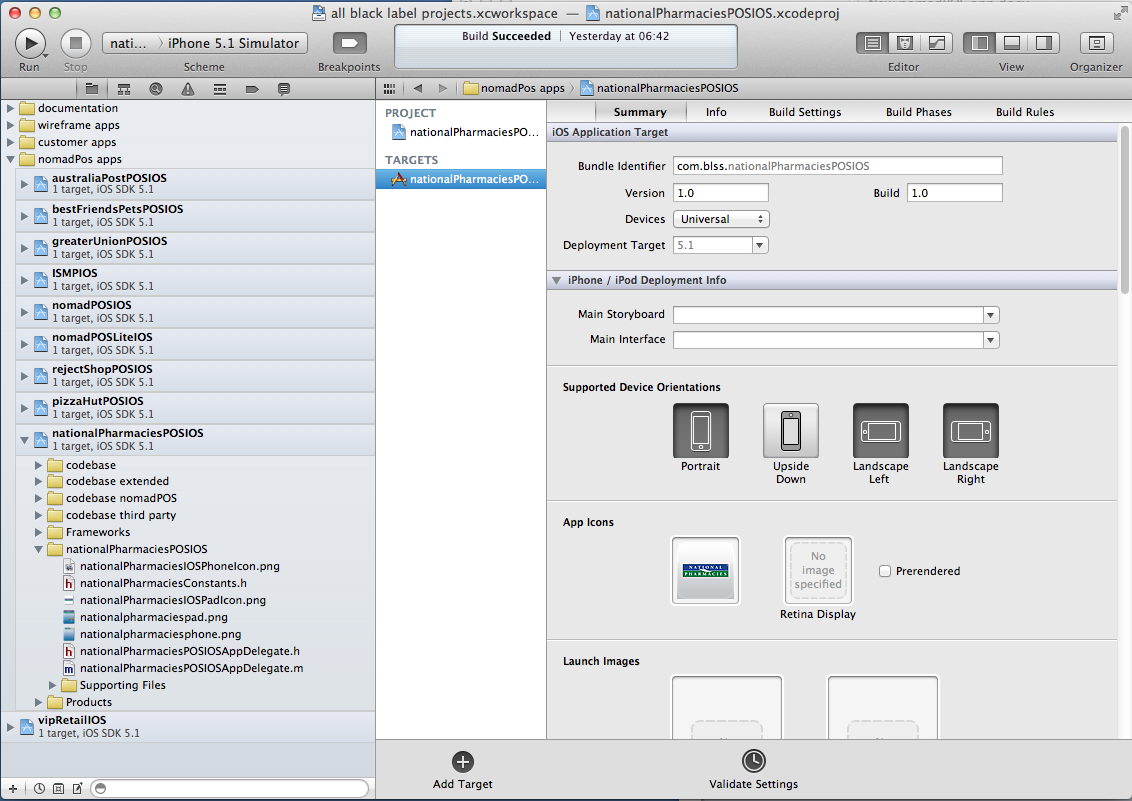


* 1. Select all the frameworks that were just added to the project and move into the Frameworks group.

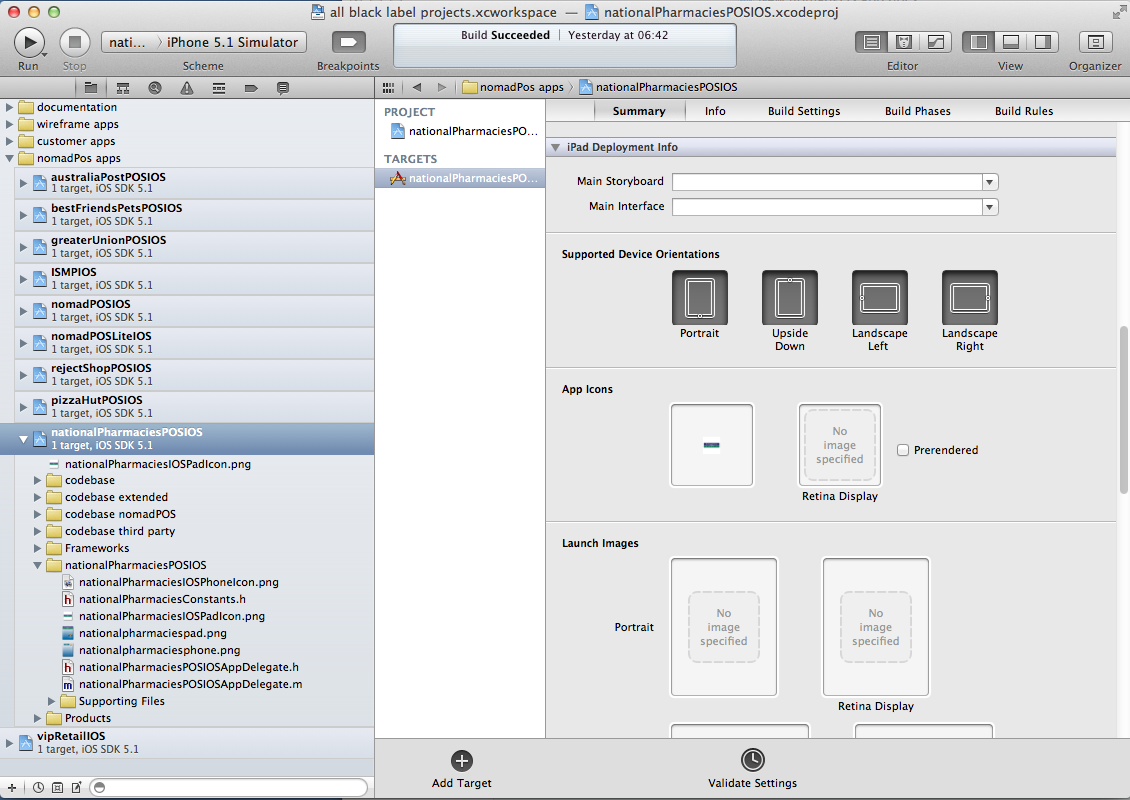
1. Setup the app’s desktop icon settings.
   1. Click on the new project name to view its settings.
   2. On the project settings view, under **Targets**, select the project name.
   3. Click on the **Info** tab.
   4. Change the value of **Bundle display name** to the Client’s name.



* 1. Click on the **Summary** tab.
  2. Set the app desktop icon graphics for the iPhone.
     + Locate your desktop icon graphics for the iPhone,
     + click and hold,
     + and then drag it into the the **App Icons** section under **iPhone/iPod Deployment Info**.
     + if necessary, you can tick the check box called **Prerendered** to remove the gloss effect of the graphics.

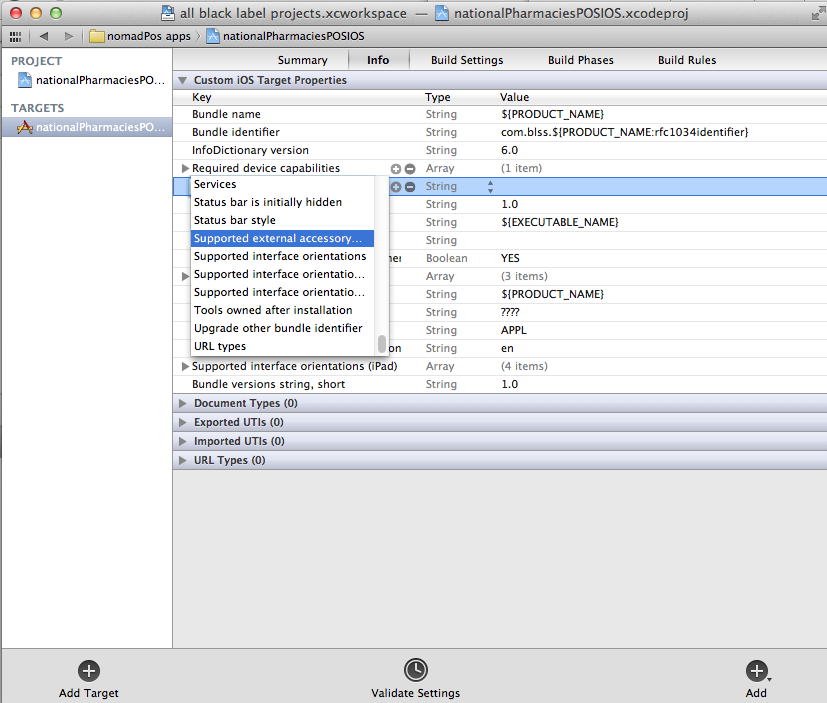


* 1. Set the app desktop icon graphics for the iPad.
     + Locate your desktop icon graphics for the iPad,
     + click and hold,
     + and then drag it into the the **App Icons** section under **iPad Deployment Info**.
     + if necessary, you can tick the check box called **Prerendered** to remove the gloss effect of the graphics.



1. Setup the Build Configuration.
   1. Click on the new project name to view its settings.
   2. On the project settings view, under **Targets**, select the project name.
   3. Click on the **Info** tab.
   4. Expand the disclosure triangle for **Custom iOS Target Properties.**
   5. Create a new **key** under **Custom iOS Target Properties**. To do this:

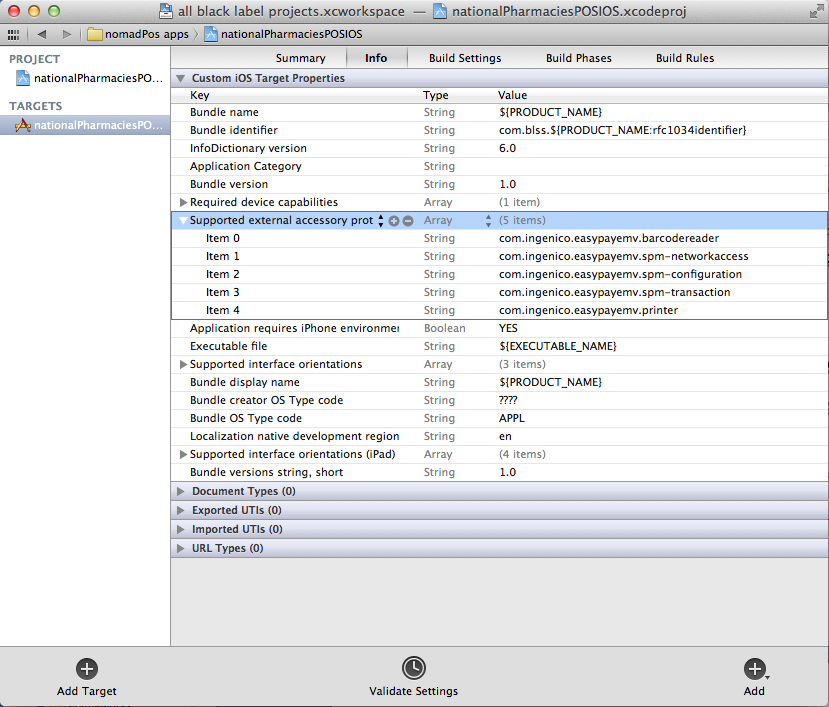
* select **Required device capabilitites** but do not expand. (you can create a key anywhere but we will create it below **Required device capabilities** for consistency)
* click the “**+”** sign just to the right of **Required device capabilities.** This will add a new entry just below.
* when the list comes up, select **Supported external accessory protocol**



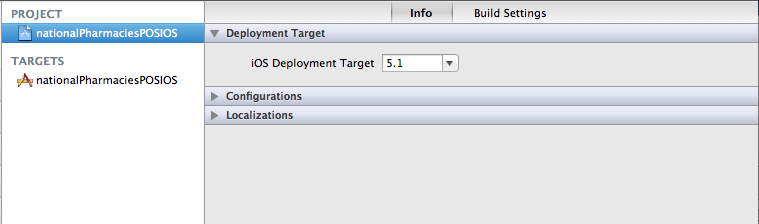
* 1. Click on the disclosure triangle to expand the newly created **Supported external accessory protocol** and select **Item 0**.
  2. Change the value of **Item 0** to “com.ingenico.easypayemv.barcodereader”
  3. Select **Item 0**  and click the “**+**” sign just to the right side of it to create a new entry.

Screen Shot 2012-06-28 at 09.06.03.png

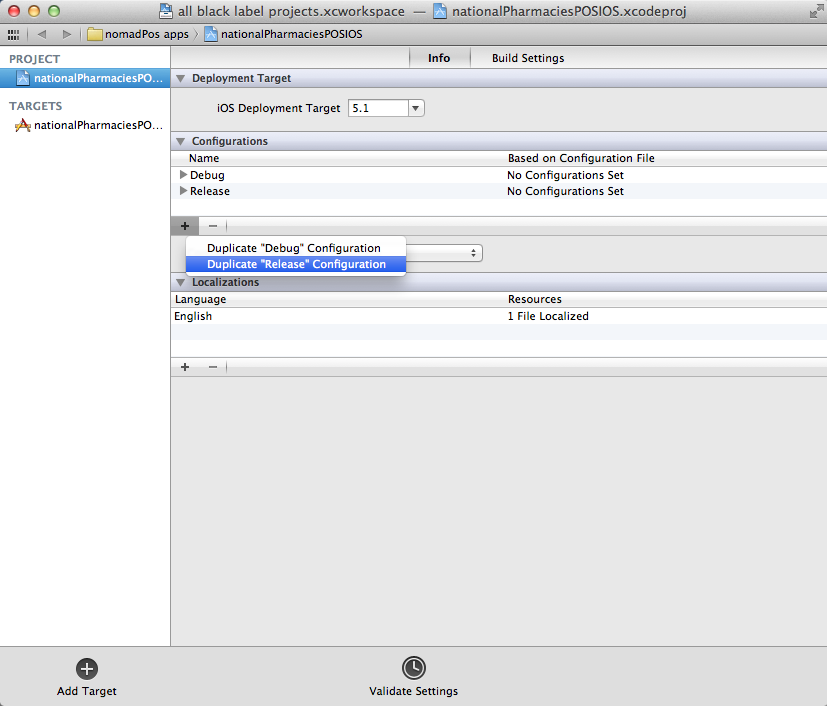
* 1. **Item 1** will be created just below **Item 0**. Change the value of **Item 1** to “com.ingenico.easypayemv.spm-networkaccess”
  2. Similarly create Item 2 with the value of “com.ingenico.easypayemv.spm-configuration”
  3. Create Item 3 with the value of “com.ingenico.easypayemv.spm-transaction”
  4. Create Item 4 with the value of “com.ingenico.easypayemv.printer”



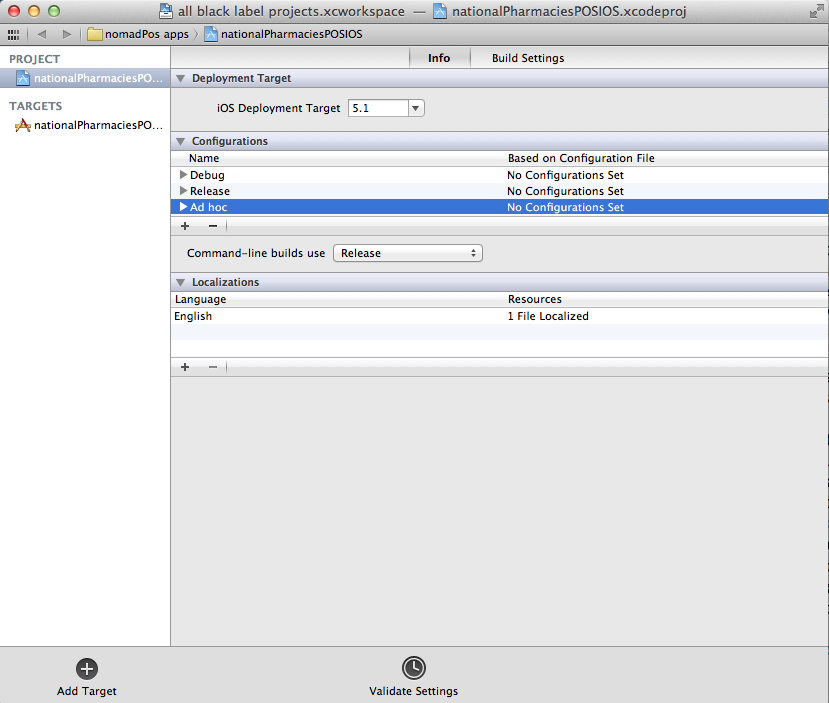
* 1. Select the new project name under **PROJECT**.



* 1. Click on the **Info** tab.
  2. Expand **Configurations** by clicking on the disclosure triangle.
  3. Click on the “**+**” sign at the bottom of the **Configurations** section and select “Duplicate Release Configuration” from the list. (NOT the “+” sign for “Add Target”)

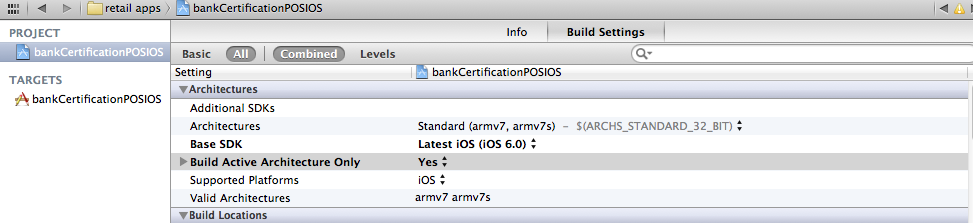


* 1. Rename the newly created configuration to “**Ad hoc**”

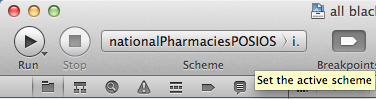


* 1. Click on the **Build settings** tab.

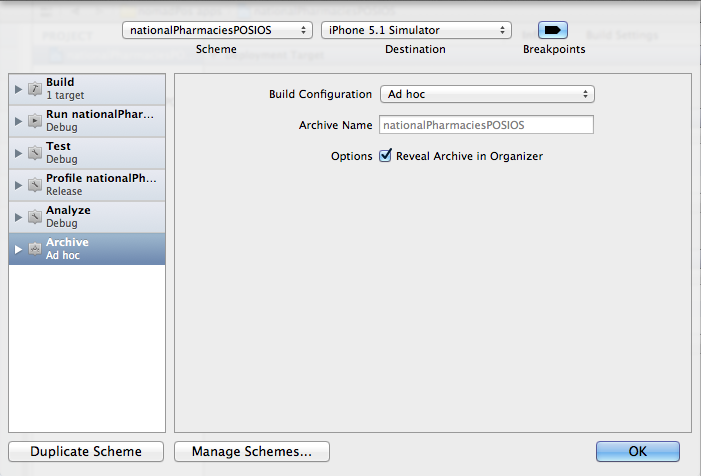
Expand **Architectures**, and set **Build Active Architecture Only** to “Yes”



* 1. Set the Active scheme to the new project name.

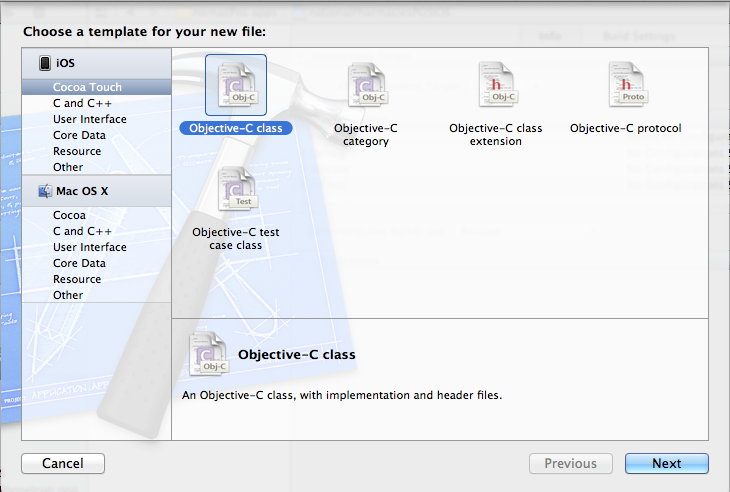


* 1. From the Xcode menu, click on **Product** and select **Edit Scheme …**
  2. Select **Archive** from the left hand side.
  3. On the right hand side, change **Build Configuration** to “Ad hoc” by selecting it from the list.



1. Create the constant file.
   1. Right click on the project name group (e.g. nationalPharmaciesPOSIOS) and select **New File …**
   2. In the **Choose a template for your new file:** screen,

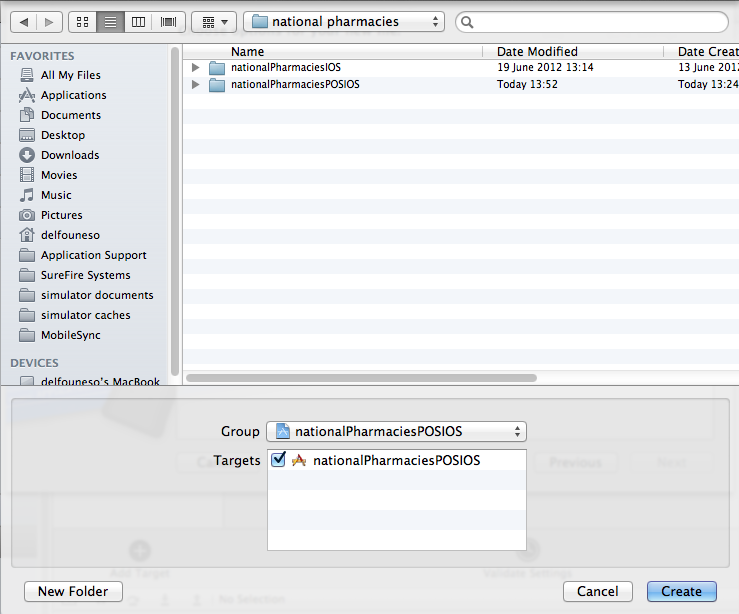
* on the left hand side, under **iOS** select **Cocoa Touch**.
* on the right hand side, select **Objective-C** class
* click **Next**.



* 1. In the **Choose options for your new file:** screen, type in the name of your class using the naming convention **<clientName>Constants** and click on **Next**.

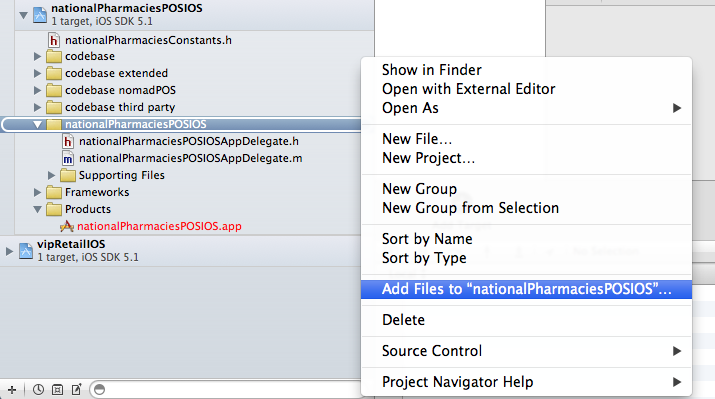


* 1. Save the constant file in the project folder. Click on **Create**.

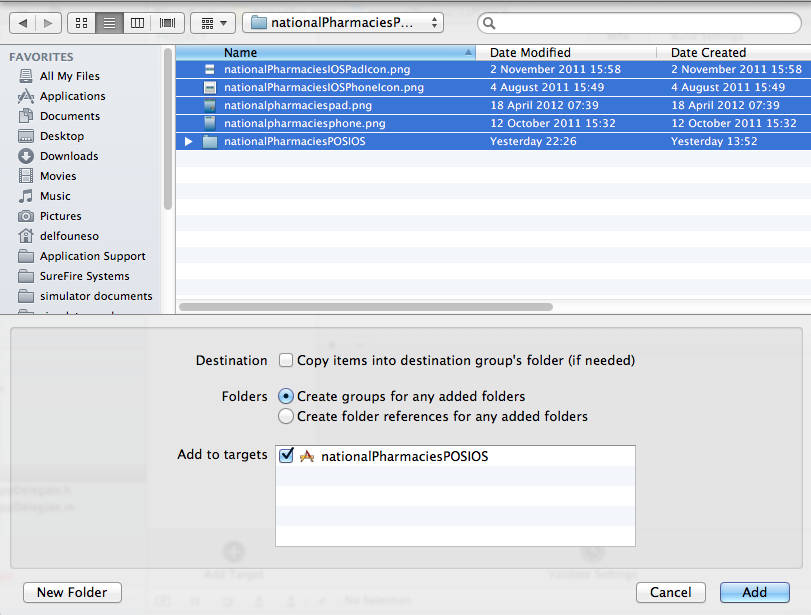


* 1. As a result of the above action two files are created, a header file and an implementation file.
  2. Delete the implementation file “**<clientName>Constants.m**”. When the delete confirmation alert appears click on the **Move to trash** button.
  3. Copy the content of the constants file of a previous app and replace everything in the new project’s constants file.
  4. In the new project’s new constants file, change the client name, creation date and author with the new client details. **ALL** constant values must be in lower case.

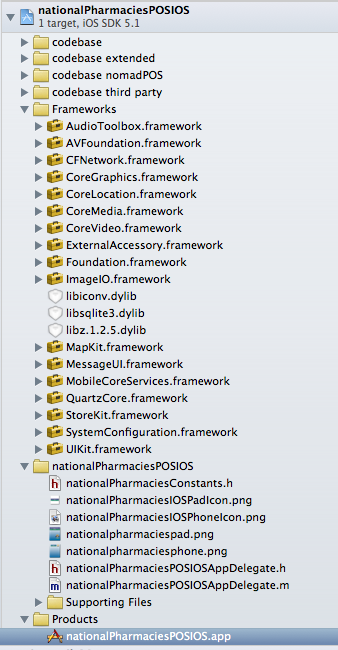
1. Add client’s graphics
   1. Create the client’s graphics and save the files in the same location as the project file.
   2. In Xcode, add the graphics files to the project group.



* 1. Locate the graphics files, select them and click on **Add**.



* 1. The application has now been created. Below is the list of files for the project.



1. Deploy the app
   1. Create the app certificate
   2. Create the code signing identities.
   3. Deploy the ad hoc app.

