Submitting Your App to iTunes Connect

v1.0

By GoNorthWest

30 December 2011

Congratulations! You've completed designing your app, tested it in the simulators and on a real device, and are ready to submit it to the App Store. Excellent! There are basically two processes you need to go through in order to get your app into iTunes, and ultimately to your customers. First, you have to add your application to iTunes via the Apple Member Center.

Adding Your App to iTunes

1. Log into your Apple Developer Account via Member Center.
2. Click on **Submit and manage your apps on the App Store** under the iTunes Connect heading.
3. Sign into iTunes Connect (no idea why you have to log in twice!)
4. Click **Manage Your Applications.**
5. In the upper-left corner, click **Add New App.**
6. Provide the following info on this screen, then hit **Continue**.
    a. **App Name**: This is the name that will show up in iTunes. It's what your customers will see. Examples are "AZ Fishing Spots" or "OVFPS" (those are my apps!)
    b. **SKU Number:** This can be absolutely anything at all. It's just a way for you to keep track of your apps if you want to use a SKU. I use my company name_APPxx, where xx is the number app that I've built.
    c. **Bundle ID:** An identifier used by iOS and Mac OS X to recognize any future updates to your app. Your Bundle ID must be registered with Apple and unique to your app. Bundle IDs are app-type specific (either iOS or Mac OS X). You should have created a Bundle ID earlier in your certificate generation process. I use a wildcard ID. This field should be populated with Bundle IDs that you created, so pick the appropriate one.
    d. **Bundle ID Suffix:** When you upload your binary, this Bundle ID will not work unless it matches the one in your info.plist.
7. Provide your **Pricing and Availability** information. Hit **Continue.**
8. Fill in the **Metadata** in this page.

a. **Version Number:** Has to match the version number you specify in info.plist, and which can be configured in Xcode.

b. **Description:** This is what your customers will read when they are deciding if they should download your app or not. Make it good, and try to capture them early on.

c. **Primary Category:** Select one.

d. **Secondary Category:** Select one.

e. **Keywords:** These are words that people might use when searching for an app that does what yours does. Comma separated.

f. **Copyright:** Enter any copyright info.

g. **Contact Email Address:** An email address that app users can contact for problems.

h. **Support URL:** Do not forget to put this, or your app will get rejected. And make sure the URL is live, otherwise it'll get rejected also.

i. **App URL:** If you want to.

j. **Privacy Policy URL:** If you have one.

k. **Review Notes:** Anything you want to tell the reviewers? This info will not show up in iTunes when you app is published.

l. **Rating:** Provide the appropriate rating information.

m. **EULA:** If you have one, provide it here.

n. **Large 512x512 Icon:** A large version of your app icon that will be used on the App Store. It must be at least 72 DPI and a minimum of 512x512 pixels (it cannot be scaled up). It must be flat artwork without rounded corners. When I create my Buzztouch project, I actually start by creating a 512x512 image that I use for my app icon, which gets uploaded to Buzztouch. Then I use the same image for this part of iTunes. This should be a high quality .jpeg, .jpg, .tif, .tiff, or .png file.

o. **iPhone and iPod Touch Screenshots:** iPhone and iPod touch Screenshots must be .jpeg, .jpg, .tif, .tiff, or .png file that is 960x640, 960x600, 640x960, 640x920, 480x320, 480x300, 320x480 or 320x460 pixels, at least 72 DPI, and in the RGB color space.

p. **iPad Screenshots:** iPad Screenshots must be .jpeg, .jpg, .tif, .tiff, or .png file that is 1024x768, 1024x748, 768x1024 or 768x1004 pixels, at least 72 DPI, and in the RGB color space.

9. **Hit Save** in order to save all this information.

10. You will now be taken to a **Manage Apps** page, and it lists all the apps you currently have in the store, or waiting to be uploaded, or whatever. Click on the icon for the app you want to upload.

11. Under **Versions**, click **View Details** for the version that you want to upload. At the moment it should say **Prepare to Upload.**

12. In the upper-right and lower-right corners of this screen, there is a **Ready to Upload Binary** button. Click this if your binary has been compiled in Xcode, and you believe it is ready to hit the store.

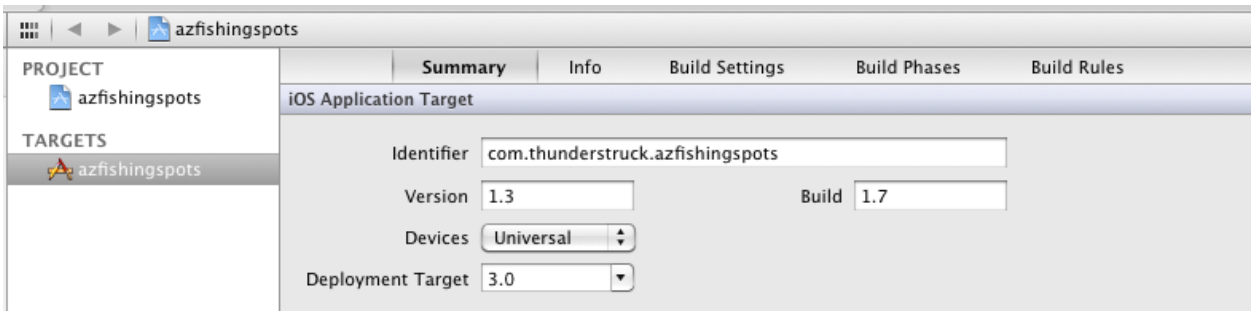
13. Answer a couple of questions. After that, your status should turn to **Waiting For Upload.** Now it's time to head back to Xcode and upload the actual package. You're almost done!

What we've done to this point is tell the iTunes store about our application, and give it all the metadata that it requested. We've also told iTunes that we're ready to upload our binary. There are two ways to do this: via Application Loader, which comes installed when you install your SDK, or via Xocde and Organizer. Personally, I prefer to use Organizer, because it's integrated into Xcode, and I don't have to launch another application. So, the rest of this tutorial will be based on Xcode and Organizer.

Compiling and Submitting to ITunes

1. Open up Xcode and load your app if you haven't already.

2. Now highlight your Target from within this screen.



3. Make sure you hit the **Summary** tab, and ensure that you have the correct identifier, version, build, devices and deployment target. The information in **Identifier** and **Version** need to match what you inputted in the first step of this document.

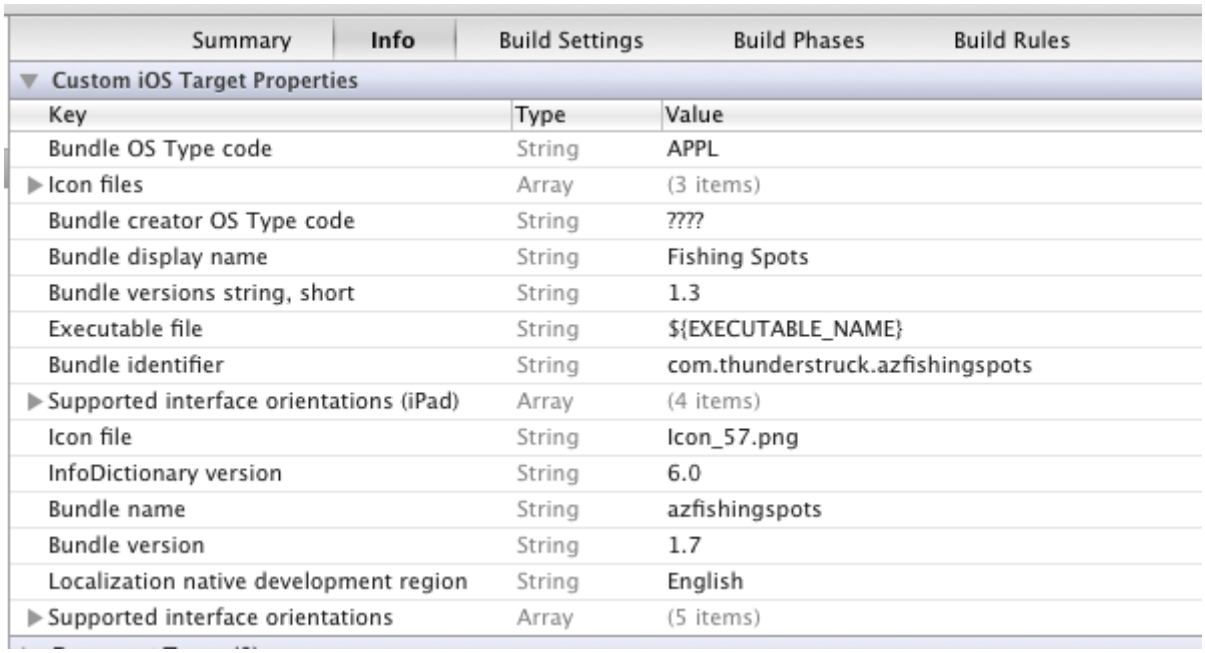
    a. **Identifier**: Should end in the same name that you have in your **Bundle ID Suffix**. What comes before that is up to you, but should basically be com.<company_name>

    b. **Version**: This also needs to match what you put in iTunes, and should reflect the true version of your app. Initial version is usually v1.0, with subsequent iterations a dot increase.

c. **Devices**: Specify **Universal** if you want to run on both iPhone/iTouch and iPad. Otherwise, pick what you want it to run on.

d. **Deployment Target**: This should be the minimum iOS level you want your app to run on. I choose iOS 3 so that I can have the greatest number of devices my app will work on. You may want to restrict your app to particular iOS levels based on the features of your app. Totally your choice!

4. Now click on the **Info** tab, and fill in the information.



Summary	Info	Build Settings	Build Phases	Build Rules
▼ Custom iOS Target Properties				
Key	Type	Value		
Bundle OS Type code	String	APPL		
▶ Icon files	Array	(3 items)		
Bundle creator OS Type code	String	????		
Bundle display name	String	Fishing Spots		
Bundle versions string, short	String	1.3		
Executable file	String	${EXECUTABLE_NAME}		
Bundle identifier	String	com.thunderstruck.azfishingspots		
▶ Supported interface orientations (iPad)	Array	(4 items)		
Icon file	String	Icon_57.png		
InfoDictionary version	String	6.0		
Bundle name	String	azfishingspots		
Bundle version	String	1.7		
Localization native development region	String	English		
▶ Supported interface orientations	Array	(5 items)		

5. Ensure that you have the following fields filled out correctly. This information gets written to your BT_config.plist file.

a. **Bundle Display Name**: This is the text that gets written underneath the app icon on the device. Make it too long, and it might look odd. I think the max length is around 16 characters.

b. **Bundle Version String, short**: Should be already configured and the same as the earlier step.

c. **Bundle Identifier**: Should be the same as the identifier in the previous step.

d. **Bundle Name**: This is basically the name of your app package. It is NOT the display name, or what gets shown in iTunes. It should match the Bundle ID Suffix that you specified earlier in the process.

6. Now it's time to click on **Build Settings** and fill in a few bit of information. This is where things get tricky, and if you mess things up here, you won't be able to validate your app, and you won't be able to submit it to the store.

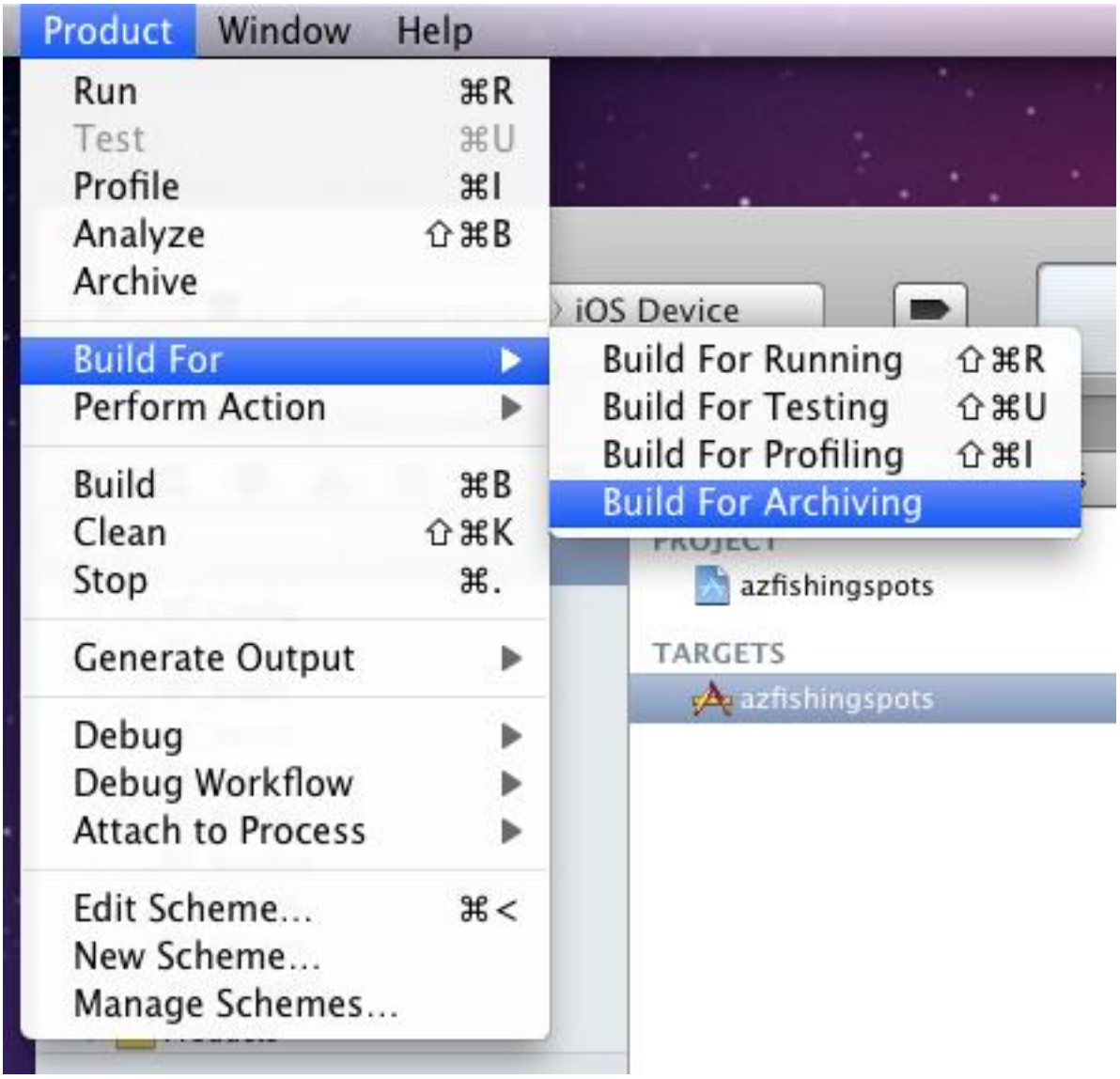
    a. **Architecture**. The first area to make sure it's good is the Architecture settings. What you see below should be what you need, unless you're doing something special.

    b. **Code Signing.** This section will be your nemesis if you don't get it right. Basically, this is where you indicate which Development and Distribution Certificates that you use in your app. If you get it right…smooth sailing! Get it wrong…you won't be able to successfully build your app. If you get it right, it'll say "(currently matches <something>)" and all will be well!

    c. **Deployment.** Only two things that need to be set here, and they are near the bottom. Specify the **Targeted Device Family** and the **iOS Deployment Target**. Easy peasy!

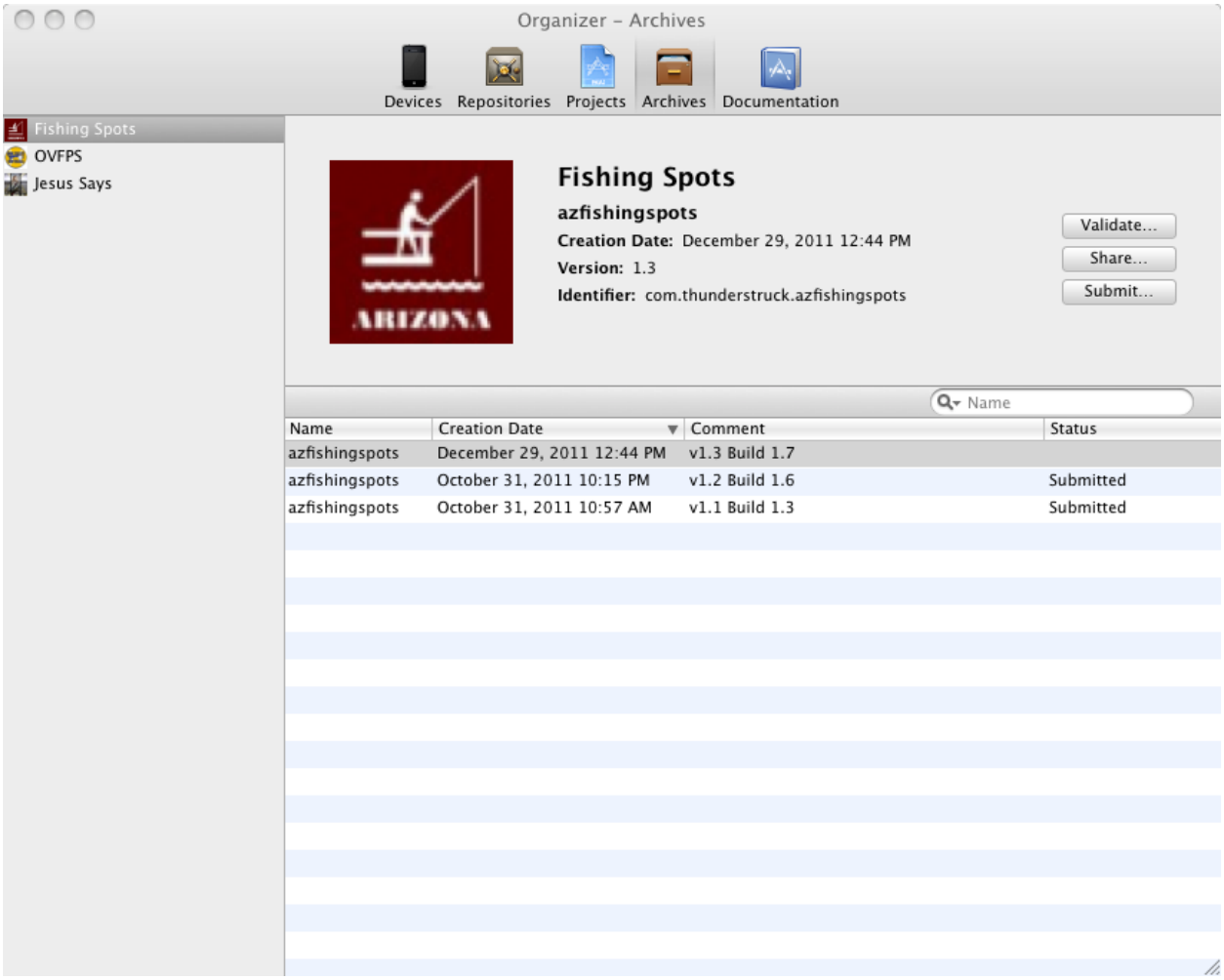
7. Believe it or not, you're done messing with the settings. As long as both the **Project** and **Targets** have the same settings, and they are consistent with what you specified in iTunes Connect, then you should be good!

8. Time to **Build For Archiving.** This will create a zipped up package that can then be Archived, which will add the application archive to Organizer, and enable you to submit it to Apple.



9. After Xcode has finished building and archiving your app…hopefully without errors, then it's time to actually Archive your app and have it sent to Organizer. Go to **Product…Archive** and let it rip. It will package everything up, and send it over to Organizer. I think we're actually doing a couple steps twice here, but the bottom line is that it works, and I've been able to submit my app after doing these steps, so I'm not going to mess with what works!

10. If Organizer has not automatically opened, go to **Window…Organizer** in Xcode and bring it up that way. Click on the Archives tab, and you should see something similar to below.

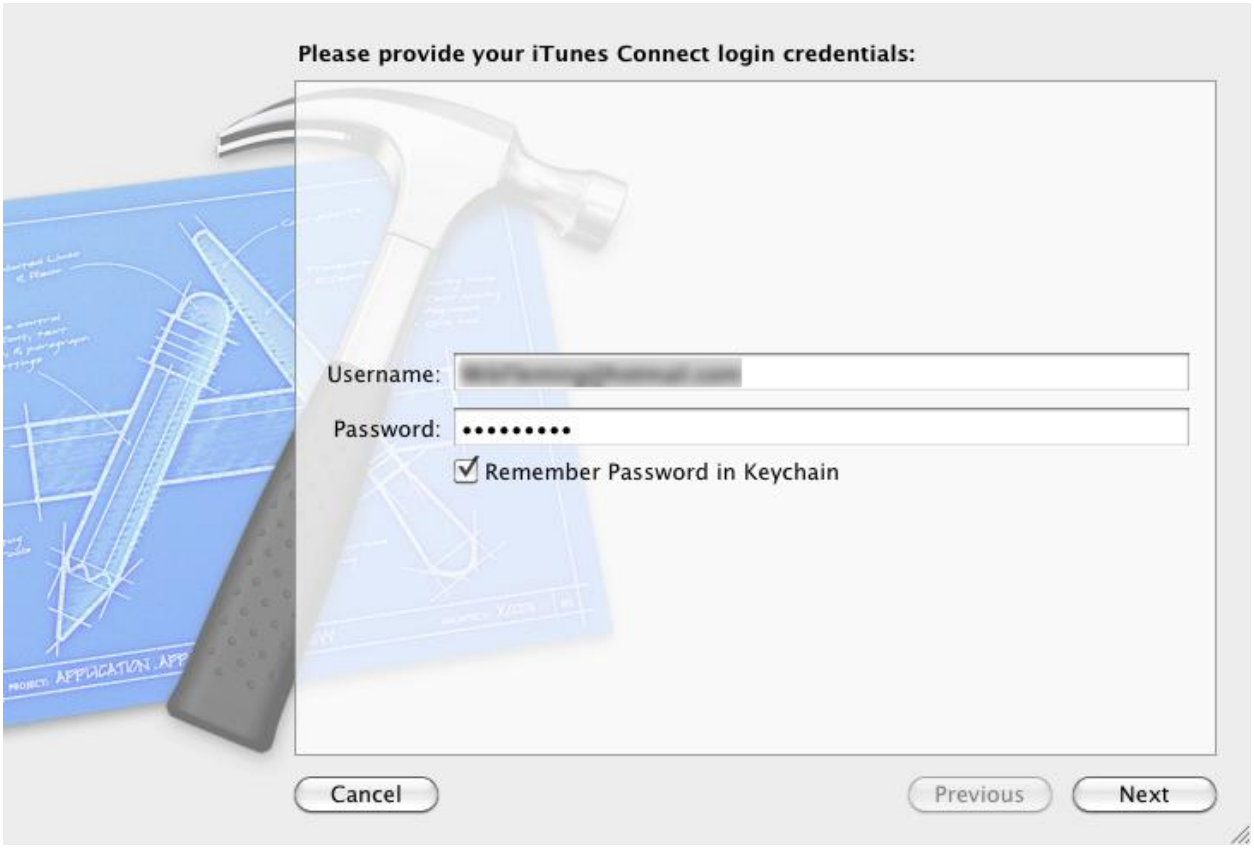


11. At this point, there are basically two things you can do once you highlight the build that you are working with: you can **Validate**, and check to make sure that everything is good before you actually submit; or, you can **Submit** and go through the actual submission process, which includes validating your app. I generally go with a Validate then Submit, just to make sure there are no issues. I hate being disappointed by rejection during the Submit process for one reason or another!

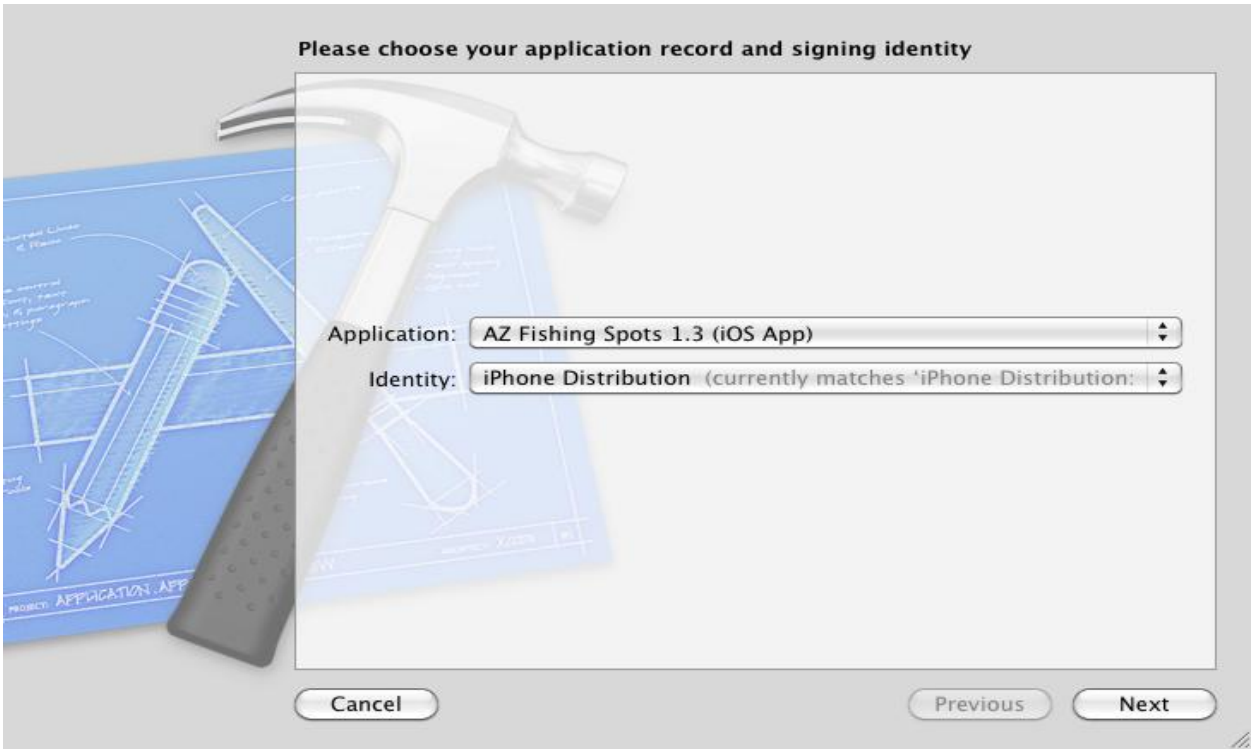
    a. **Validate:** This will validate your app and make sure everything, including certificates, is good to go.

        i. **Highlight** the package you are interested in.

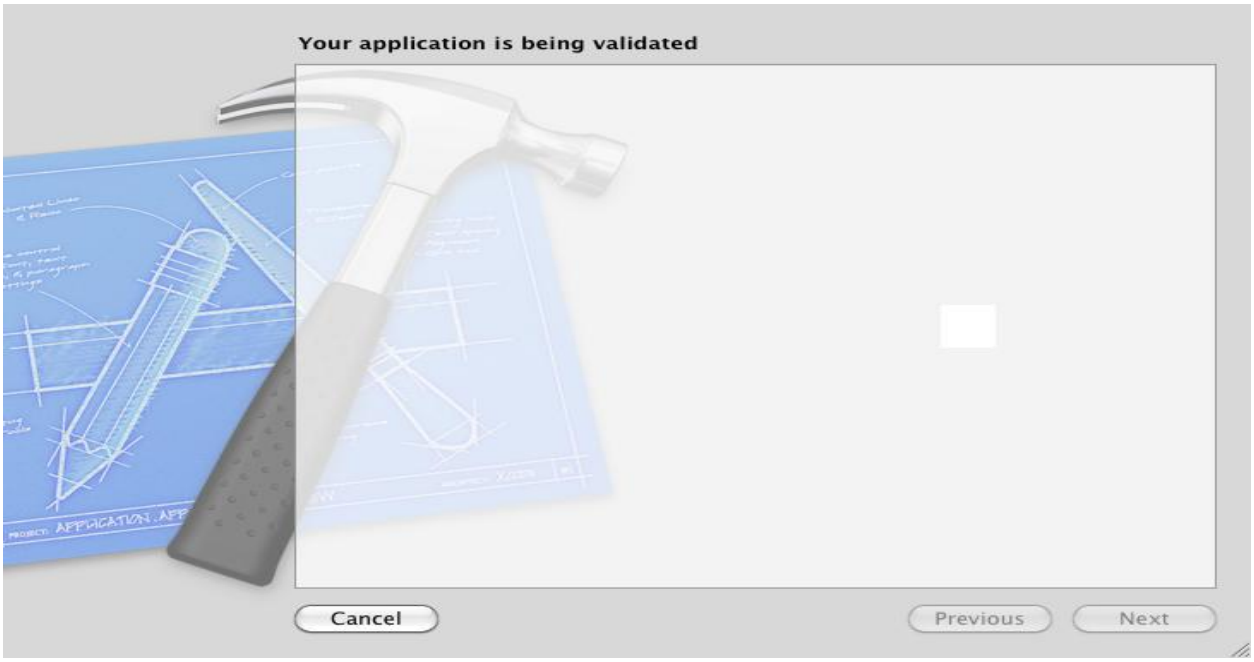
        ii. Hit the **Validate** button. The following screen will be displayed.



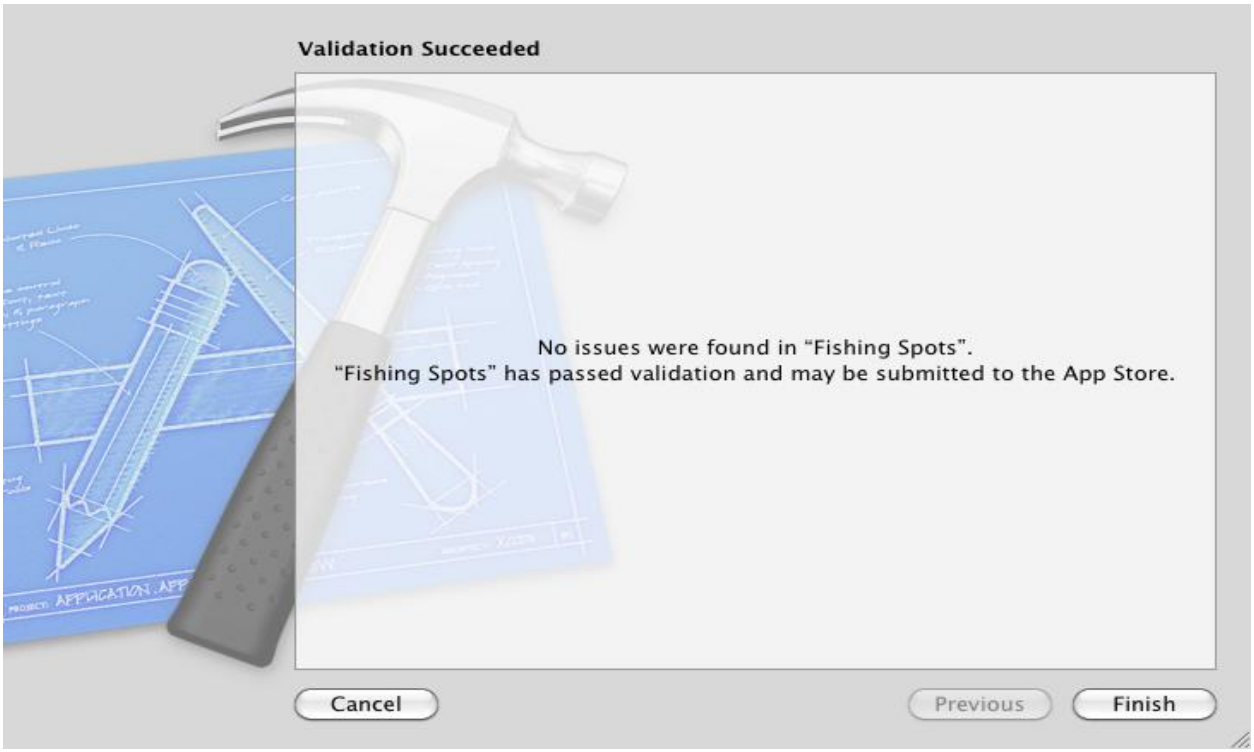
iii. Hit **Next**, and provide the requested information.



iv. Hit **Next**, and wait while your app package gets validated.



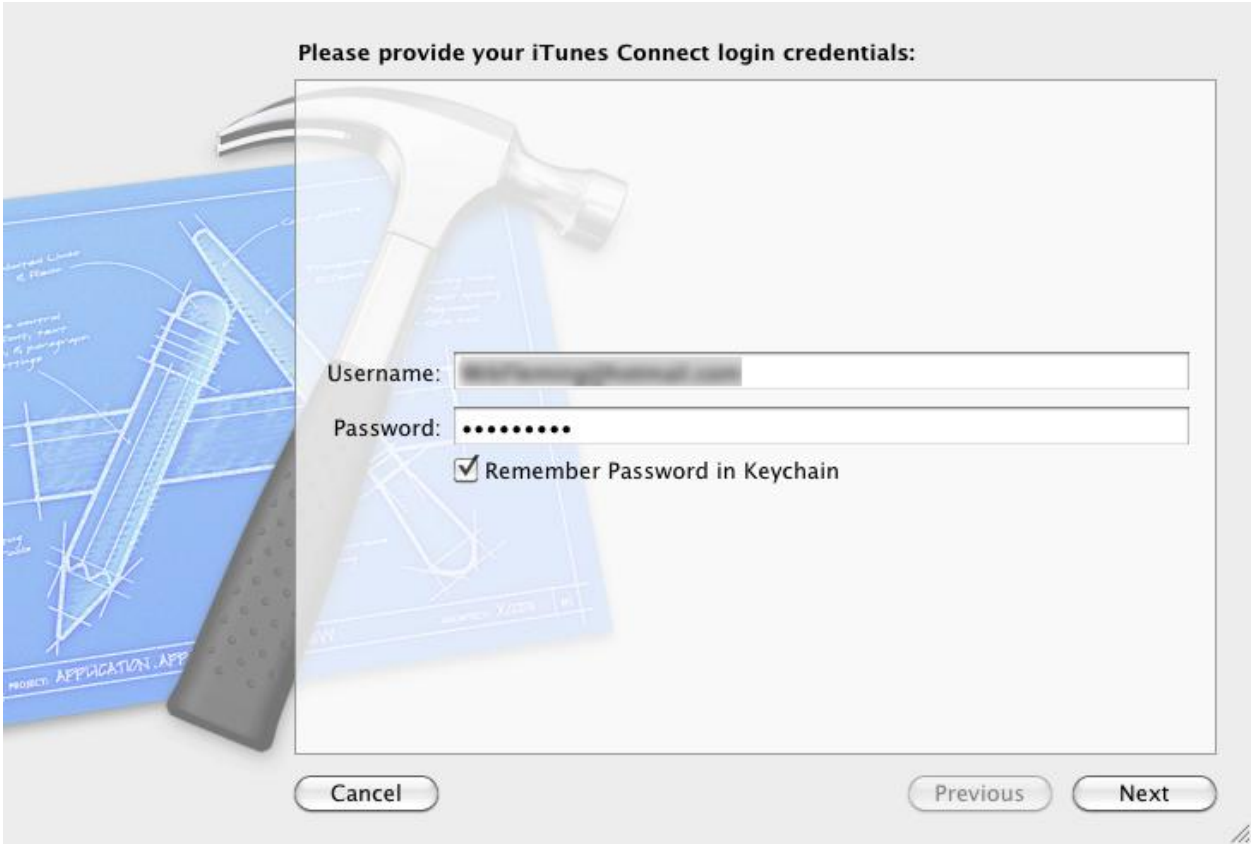
v. Pretty soon you'll see the results of your validation exercise! It'll either have worked, and you can submit to iTunes Connect, or it'll have failed, and a reason will be presented. Either way, you can now hit **Finish**.



b. **Submit.** This will actually submit your app to iTunes Connect, and get it in the queue for review. This will ONLY work if you have your app in a "Ready to Upload" state.

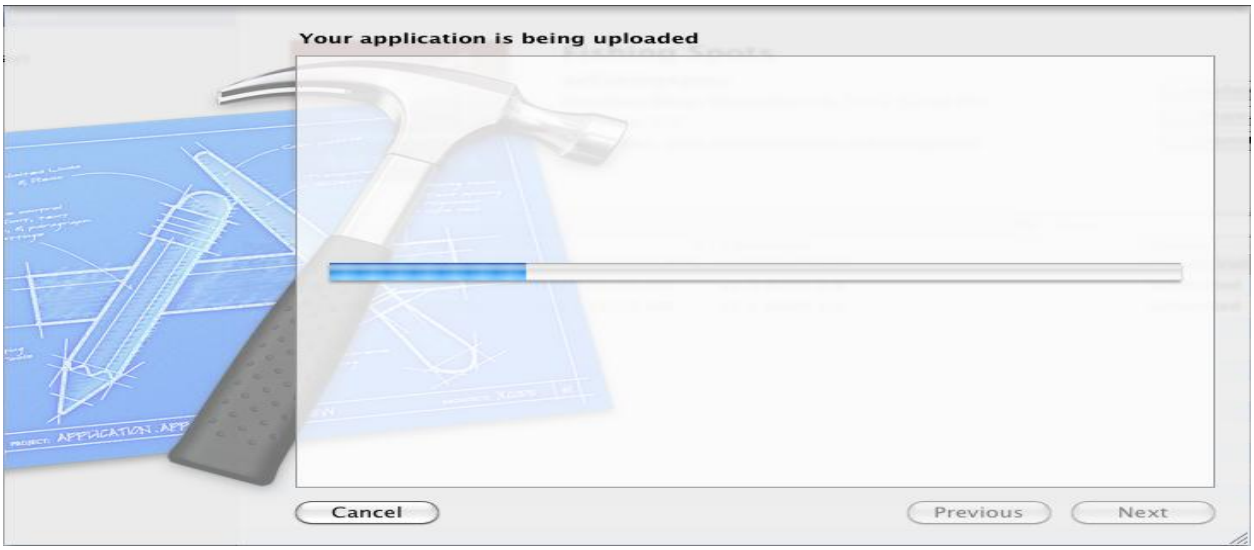
   i. **Highlight** the package you are interested in.

   ii. Hit the **Submit** button. The following screen will be displayed.

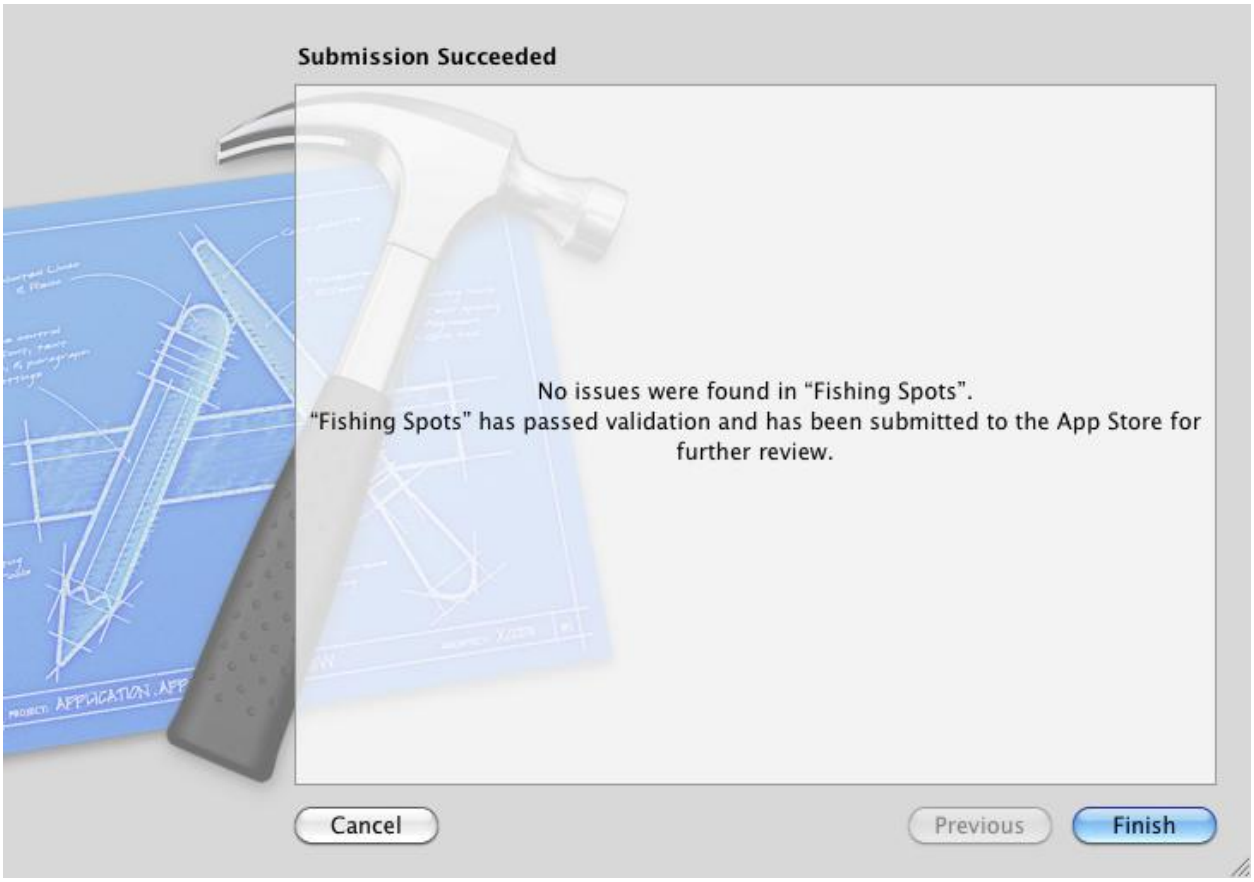


   iii. Hit **Next**, and provide the requested information.

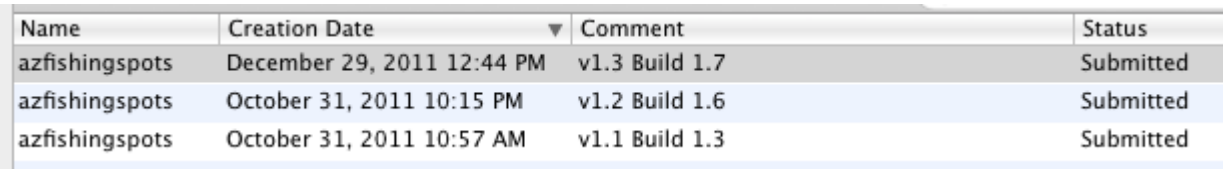
iv. Wait while your app is uploaded to iTunes Connect. Grab a cup of coffee, or maybe send me a SoBe Fruit Punch…I love that stuff!



v. Revel in the success of your app being accepted by iTunes Connect, and being placed in "Waiting For Review" status! Hit **Finish.**



12. If you return to **Organizer**, you should now see that your app has been submitted.



Name	Creation Date	Comment	Status
azfishingspots	December 29, 2011 12:44 PM	v1.3 Build 1.7	Submitted
azfishingspots	October 31, 2011 10:15 PM	v1.2 Build 1.6	Submitted
azfishingspots	October 31, 2011 10:57 AM	v1.1 Build 1.3	Submitted

13. At this point, it's just a waiting game. You'll receive emails when your app is under review, and when it either passes or fails review. If all your ducks are in a row, the process shouldn't take more than 6 – 10 business days (they don't appear to work weekends or holidays at Apple).

So, there you have it! That's pretty much the steps necessary for submitting your app to iTunes Connect, and ultimately having it end up in the App Store for millions of people to purchase or use. Pretty sweet, huh?! As you can tell, this is a fairly detailed process, and it's entirely possible I missed a step, or left out a crucial bit of information, so please don't hesitate to send in corrections.

Good luck with all your app development endeavors!

Comments? Post them in the forum or email me at MrkFleming@gmail.com.

Revision Log			
v1.0	12/30/11	Initial release of document.	GoNorthWest