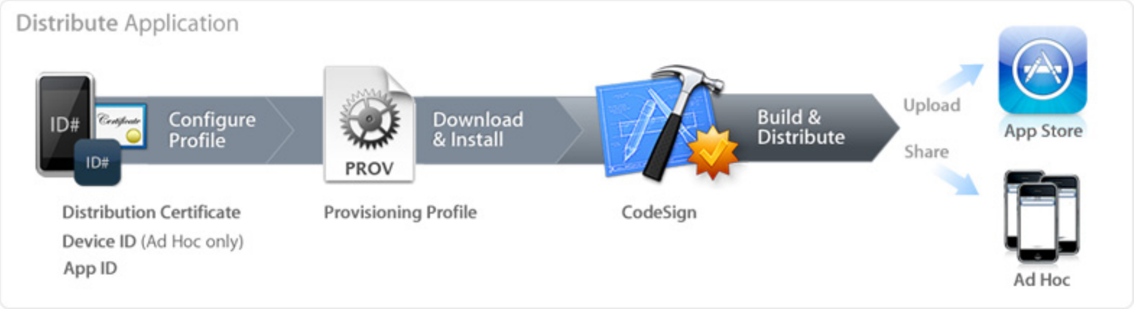
Welcome, Yuriy Donev Edit Profile Log out

Provisioning Portal : Astea Solutions AD

Go to iOS Dev Center

Prepare App **App Store** **Ad Hoc**

Distribution



The distribution area of the iOS Provisioning Portal is where you will prepare and learn how to submit your iPhone and/or iPod touch application for delivery via in-house or Ad Hoc distribution. Only Team Agents are authorized to prepare and submit applications for distribution.

For information about distributing your application on the App Store, please see the App Store tab.

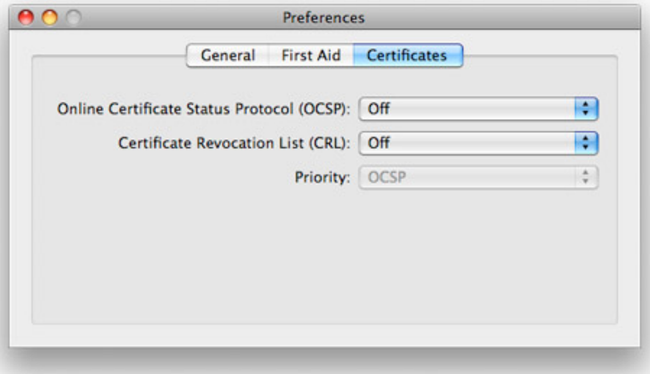
Obtaining your iOS Distribution Certificate

In order to distribute your iOS application, the Team Agent is required by Apple to create an iOS Distribution Certificate. Only the Team Agent for your team will be able to create this certificate and only this certificate will enable application submission.

Generating a Certificate Signing Request

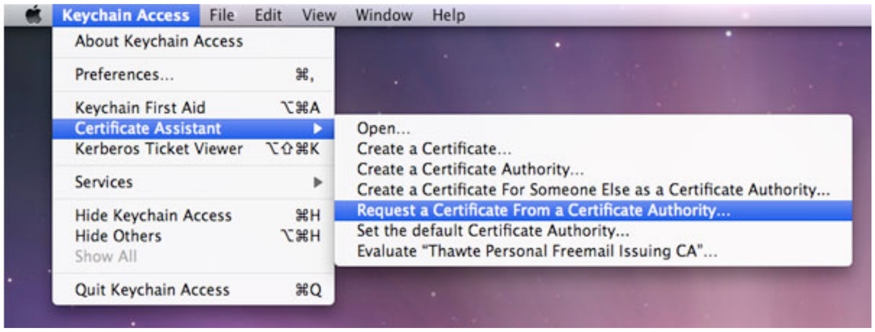
To request an iOS Distribution Certificate, you first need to generate a Certificate Signing Request (CSR) utilizing the Keychain Access application in Mac OS X Leopard. The creation of a CSR will prompt Keychain Access to simultaneously generate your public and private key pair establishing your iOS Distribution identity. Your private key is stored in the login Keychain by default and can be viewed in the Keychain Access application under the ‘Keys’ category. To generate a CSR:

1. In your Applications folder, open the Utilities folder and launch Keychain Access.
2. In the Preferences menu, set Online Certificate Status Protocol (OSCP) and Certificate Revocation List (CRL) to “Off”.

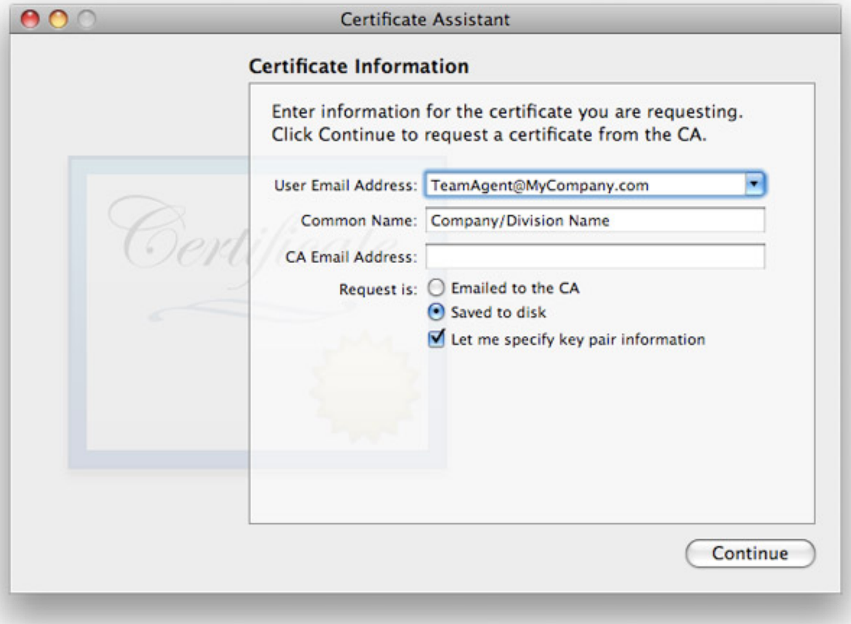


Online Certificate Status Protocol (OCSP): Off
Certificate Revocation List (CRL): Off
Priority: OCSP

3. Choose Keychain Access -> Certificate Assistant -> Request a Certificate from a Certificate Authority. Note: If you have a private key highlighted in the Keychain during this process, the resulting Certificate Request will not be accepted by the Provisioning Portal. Confirm that you are selecting “Request a Certificate From a Certificate Authority...” and not selecting “Request a Certificate From a Certificate Authority with <Private Key>…”


Keychain Access
File
Edit
View
Window
Help
About Keychain Access
Preferences... ⌘,
Keychain First Aid ⌥⌘A
Certificate Assistant
Kerberos Ticket Viewer ⌥⇧⌘K
Services
Hide Keychain Access ⌘H
Hide Others ⌥⌘H
Show All
Quit Keychain Access ⌘Q
Open...
Create a Certificate...
Create a Certificate Authority...
Create a Certificate For Someone Else as a Certificate Authority...
Request a Certificate From a Certificate Authority...
Set the default Certificate Authority...
Evaluate "Thawte Personal Freemail Issuing CA"...

4. In the User Email Address field, enter your email address. Please ensure that the email address entered matches the information that was submitted when you registered as an iOS Developer.
5. In the Common Name field enter your Company/Organization/Department name. Please ensure that the name entered matches the information that was submitted when you registered as an iOS Developer.
6. No CA Email Address is required.
7. Select the 'Saved to Disk' radio button and if present, select 'Let me specify key pair information' and click 'Continue'.

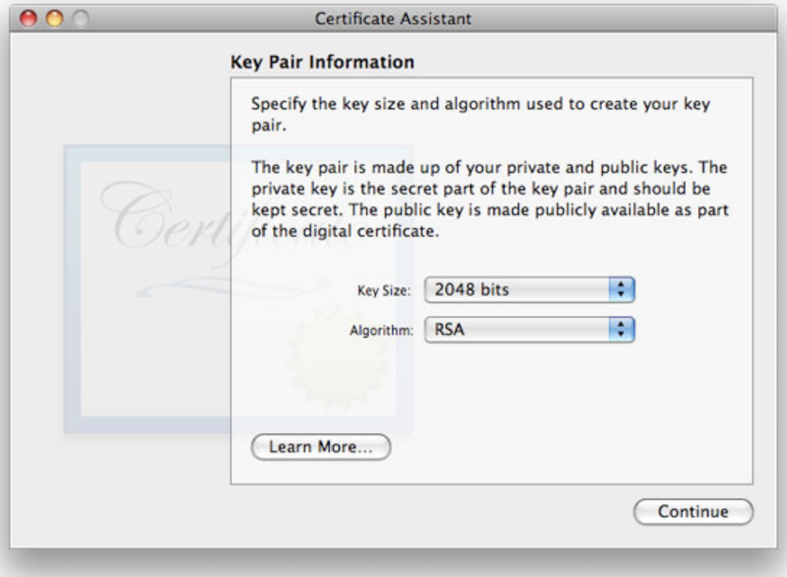

Certificate Assistant

Certificate Information

Enter information for the certificate you are requesting.
Click Continue to request a certificate from the CA.

User Email Address: TeamAgent@MyCompany.com
Common Name: Company/Division Name
CA Email Address:
Request is: ○ Emailed to the CA
● Saved to disk
☑ Let me specify key pair information

8. If 'Let me specify key pair' was selected, specify a file name and click 'Save'. In the following screen select '2048 bits' for the Key Size and 'RSA' for the Algorithm. Click 'Continue'.



Key Pair Information

Specify the key size and algorithm used to create your key pair.

The key pair is made up of your private and public keys. The private key is the secret part of the key pair and should be kept secret. The public key is made publicly available as part of the digital certificate.

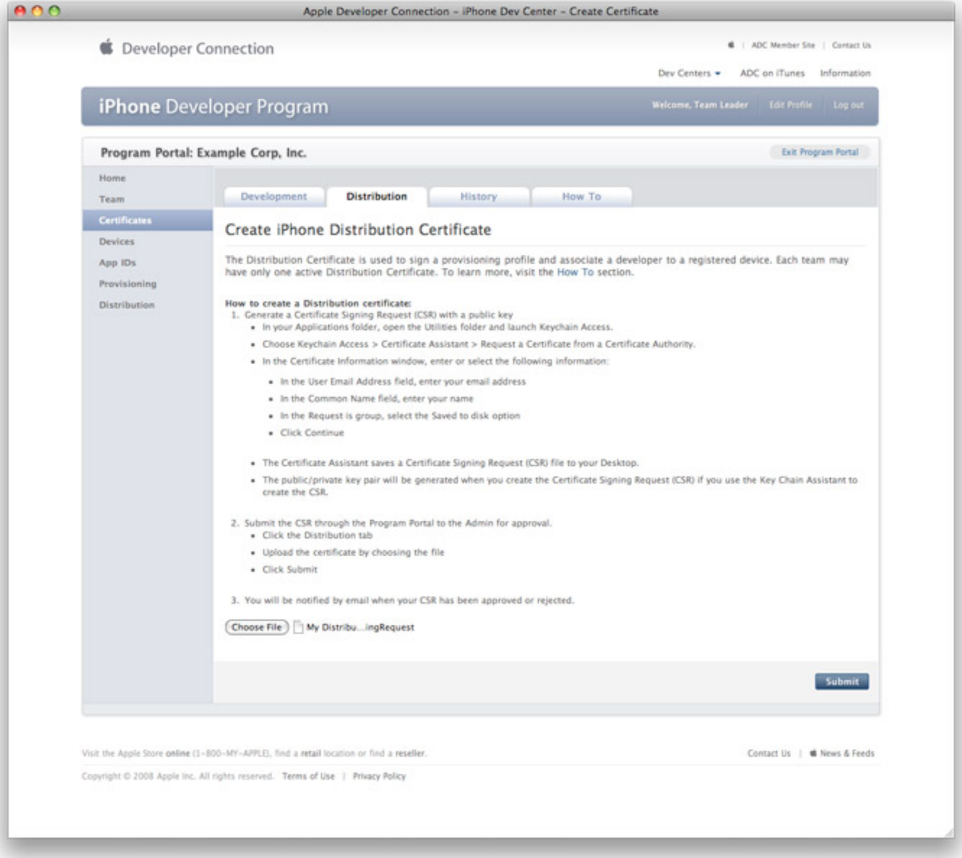
Key Size: 2048 bits

Algorithm: RSA

9. The Certificate Assistant will create a CSR file on your desktop.

Submitting a Certificate Signing Request for Approval

1. After creating a CSR, log in to the iOS Provisioning Portal and navigate to ‘Certificates’ -> ‘Distribution’ and click the ‘Add Certificate’ button.
2. Click the Upload file button, select your CSR and click ‘Submit’. If the Key Size was not set to 2048 bits during the CSR creation process, the Portal will reject the CSR.
3. Approve your iOS Distribution Certificate.



Program Portal: Example Corp, Inc.

Create iPhone Distribution Certificate

The Distribution Certificate is used to sign a provisioning profile and associate a developer to a registered device. Each team may have only one active Distribution Certificate. To learn more, visit the How To section.

How to create a Distribution certificate:

1. Generate a Certificate Signing Request (CSR) with a public key
   - In your Applications folder, open the Utilities folder and launch Keychain Access.
   - Choose Keychain Access > Certificate Assistant > Request a Certificate from a Certificate Authority.
   - In the Certificate Information window, enter or select the following information:
     - In the User Email Address field, enter your email address
     - In the Common Name field, enter your name
     - In the Request is group, select the Saved to disk option
     - Click Continue
   - The Certificate Assistant saves a Certificate Signing Request (CSR) file to your Desktop.
   - The public/private key pair will be generated when you create the Certificate Signing Request (CSR) if you use the Key Chain Assistant to create the CSR.
2. Submit the CSR through the Program Portal to the Admin for approval.
   - Click the Distribution tab
   - Upload the certificate by choosing the file
   - Click Submit
3. You will be notified by email when your CSR has been approved or rejected.

My Distribu...ingRequest

Visit the Apple Store online (1-800-MY-APPLE), find a retail location or find a reseller.

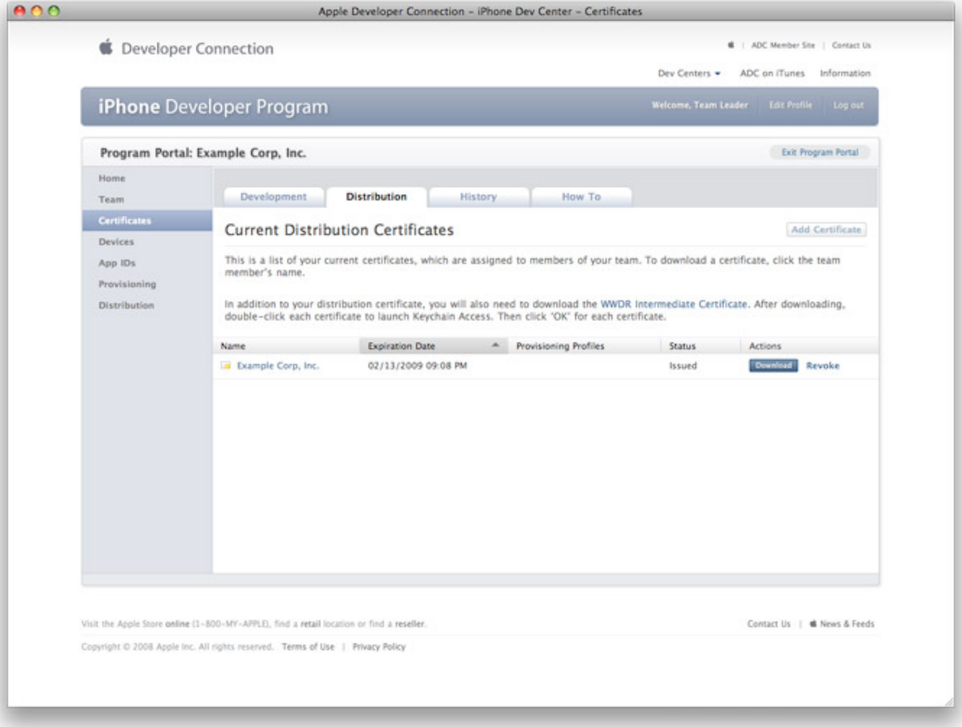
Copyright © 2008 Apple Inc. All rights reserved.

Downloading and Installing iOS Distribution Certificates

1. In the ‘Certificates’-->’Distribution’ section of the Portal, Control-Click the WWDR Intermediate Certificate link and select “Saved Linked File to Downloads” to initiate download of the certificate. After downloading, double-click the certificate to launch Keychain

Access and install.

2. In the same area of the Provisioning Portal, click on the name of the iOS Distribution Certificate to download.
3. On your local machine, double-click the downloaded .cer file to launch Keychain Access and install your certificate.



Current Distribution Certificates

This is a list of your current certificates, which are assigned to members of your team. To download a certificate, click the team member's name.

In addition to your distribution certificate, you will also need to download the WWDR Intermediate Certificate. After downloading, double-click each certificate to launch Keychain Access. Then click 'OK' for each certificate.

Name	Expiration Date	Provisioning Profiles	Status	Actions
Example Corp, Inc.	02/13/2009 09:08 PM		Issued	Download Revoke

Visit the Apple Store online (1-800-MY-APPLE), find a retail location or find a reseller.

Copyright © 2008 Apple Inc. All rights reserved.

Saving your Private Key and Transferring to Other Systems

It is critical that you save your private key somewhere safe in the event that you need to build your application on multiple Macs or decide to reinstall your system OS. Without your private key, you cannot sign binaries in Xcode and there you will be unable to upload your application to the App Store or install your application on any Apple device. When a CSR is generated, the Keychain Access application creates a private key on your login keychain. This private key is tied to your user account and cannot be reproduced if lost due to an OS reinstall. If you plan to do development and testing on multiple systems, you will need to import your private key onto all of the systems you'll be doing work on.

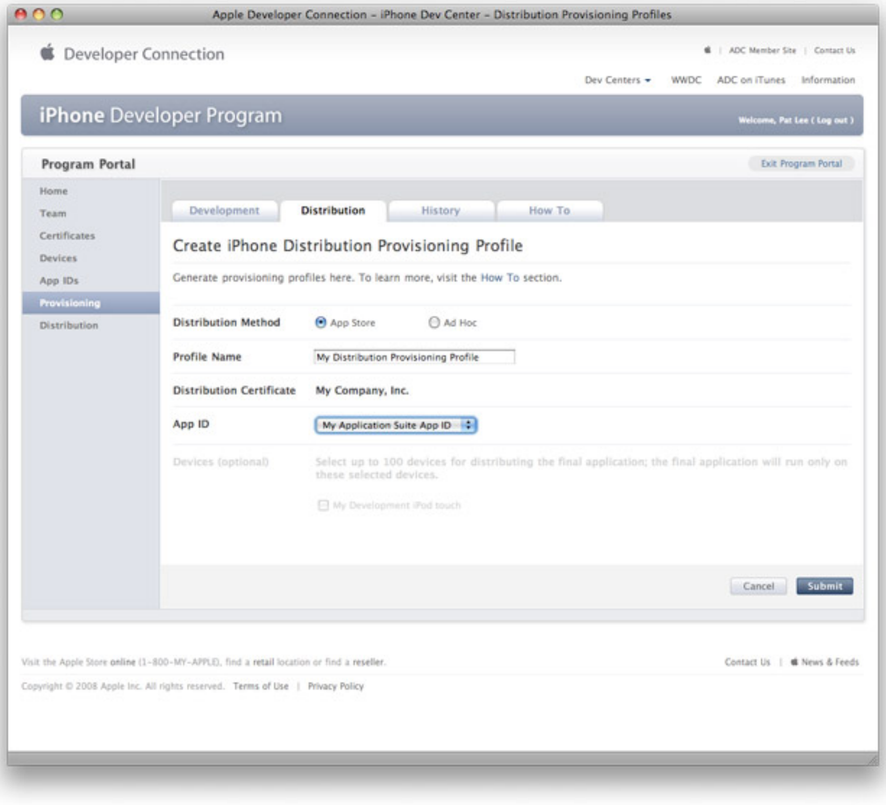
1. To export your private key and certificate for safe-keeping, open up the Keychain Access Application and select the "Keys" category.
2. Highlight the private key associated with your iOS Distribution Certificate and select "Export Items" from the 'File' menu. Save your key in the Personal Information Exchange (.p12) file format.
3. You will be prompted to create a password which will be used when you attempt to import this key on another computer.
4. You can now transfer this .p12 file between systems. Double-click on the .p12 to install on a system. You will be prompted for the password you first entered above.

Create and download your iOS Distribution Provisioning Profile for App Store Distribution

To successfully build your application with Xcode for distribution via the App Store, you first need to create and download an App Store Distribution Provisioning Profile. These are different than the Development Provisioning Profiles that were used earlier in that Apple will only accept applications if they are built with an App Store Distribution Provisioning Profile.

Note: App Store provisioning profiles do not allow for a distribution built application to be installed on an Apple device. To install your distribution ready application on a device, you must create an Ad Hoc provisioning profile.

1. Team Agents should navigate to the Provisioning section of the Provisioning Portal and select the Distribution tab.
2. Select the App Store radio button.
3. Enter the name for your Distribution Provisioning Profile.
4. Confirm your iOS Distribution Certificate has been created and is displayed.
5. Select your wild-card App ID to build all of your applications with your single Distribution Provisioning Profile.
6. Click 'Submit'.
7. Click on the name of the Distribution Provisioning Profile to download the .mobileprovision file.
8. Drag the .mobileprovision onto the Xcode or iTunes icon in the dock to install.


Apple Developer Connection – iPhone Dev Center – Distribution Provisioning Profiles
Developer Connection
ADC Member Site | Contact Us
Dev Centers
WWDC
ADC on iTunes
Information
iPhone Developer Program
Welcome, Pat Lee (Log out)
Program Portal
Exit Program Portal
Home
Team
Certificates
Devices
App IDs
Provisioning
Distribution
Development
Distribution
History
How To
Cancel
Submit
Visit the Apple Store online (1-800-MY-APPLE), find a retail location or find a reseller.
Contact Us | News & Feeds
Copyright © 2008 Apple Inc. All rights reserved. Terms of Use | Privacy Policy

Create iPhone Distribution Provisioning Profile

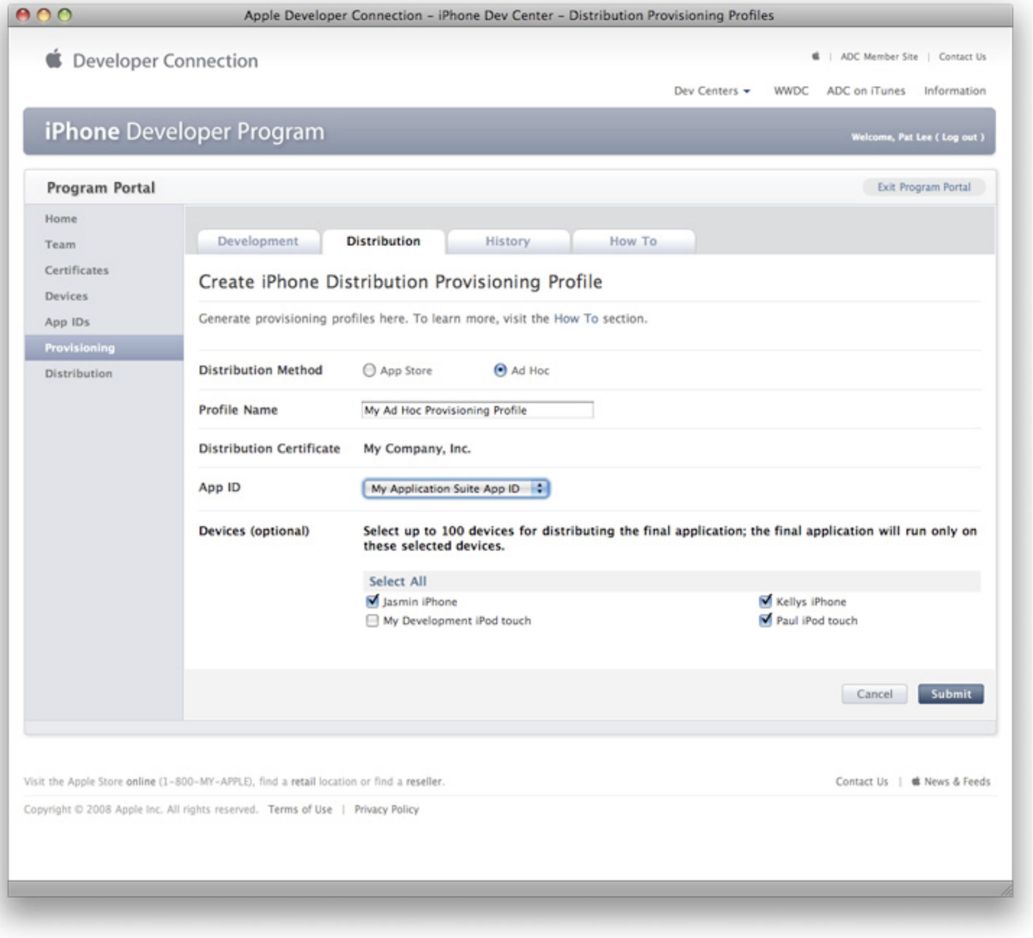
Generate provisioning profiles here. To learn more, visit the How To section.

Distribution Method	● App Store ○ Ad Hoc
Profile Name	My Distribution Provisioning Profile
Distribution Certificate	My Company, Inc.
App ID	My Application Suite App ID
Devices (optional)	Select up to 100 devices for distributing the final application; the final application will run only on these selected devices. □ My Development iPod touch

Creating and Downloading a Distribution Provisioning Profile for Ad Hoc Distribution

To successfully build your application in Xcode for Ad Hoc distribution, you will need to create and download an Ad Hoc Distribution Provisioning Profile.

1. Team Agents should navigate to the ‘Provisioning’ section of the Provisioning Portal.
2. Select the ‘Ad Hoc’ radio button.
3. Enter the name for your Ad Hoc Distribution Provisioning Profile.
4. Confirm your iOS Distribution Certificate has been created and is displayed.
5. Select the App ID for the application (or suite of applications) you wish to distribute.
6. Select up to 100 UDIDs which you wish to run your application on.
7. Click ‘Submit’.
8. Click on the name of the Distribution Provisioning Profile to download the .mobileprovision file.
9. Drag the .mobileprovision onto the Xcode or iTunes icon in the dock to install.


Apple Developer Connection – iPhone Dev Center – Distribution Provisioning Profiles
Developer Connection
ADC Member Site | Contact Us
Dev Centers
WWDC
ADC on iTunes
Information
iPhone Developer Program
Welcome, Pat Lee (Log out)
Program Portal
Exit Program Portal
Home
Team
Certificates
Devices
App IDs
Provisioning
Distribution
Development
Distribution
History
How To
Cancel
Submit
Visit the Apple Store online (1-800-MY-APPLE), find a retail location or find a reseller.
Contact Us | News & Feeds
Copyright © 2008 Apple Inc. All rights reserved. Terms of Use | Privacy Policy

Create iPhone Distribution Provisioning Profile

Generate provisioning profiles here. To learn more, visit the How To section.

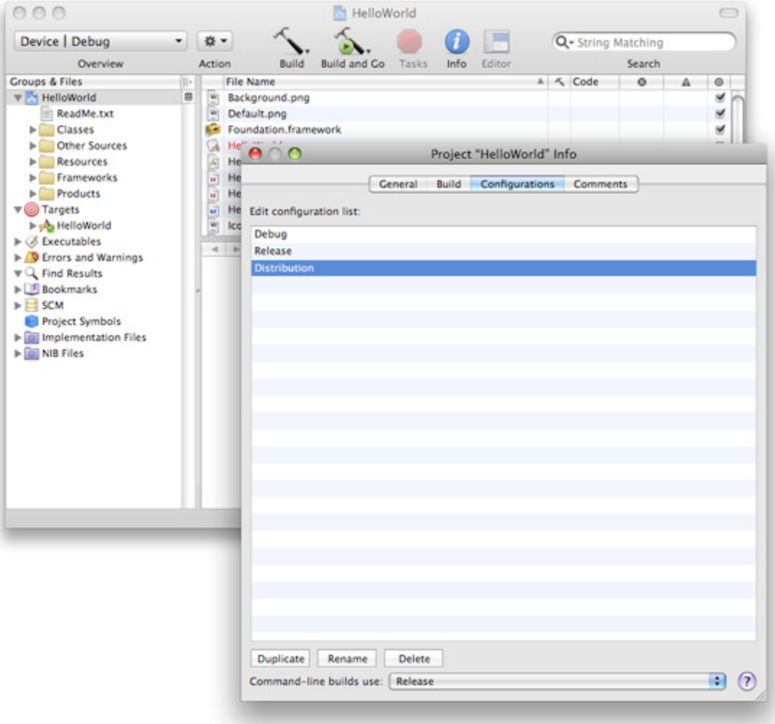
Distribution Method	○ App Store ● Ad Hoc
Profile Name	My Ad Hoc Provisioning Profile
Distribution Certificate	My Company, Inc.
App ID	My Application Suite App ID
Devices (optional)	Select up to 100 devices for distributing the final application; the final application will run only on these selected devices.

Select All

☑ Jasmin iPhone
☐ My Development iPod touch
☑ Kellys iPhone
☑ Paul iPod touch

Building your Application with Xcode for Distribution

1. Launch Xcode and open your project.
2. If you have not already done so, drag the Distribution Provisioning Profile downloaded from the Provisioning Portal onto the Xcode or iTunes icon in the dock (or, drag into '~/Library/MobileDevice/Provisioning Profiles' directory.)
3. Open the Xcode project and Duplicate the "Release" configuration in the Configurations pane of the project's Info panel. Rename this new configuration "Distribution".

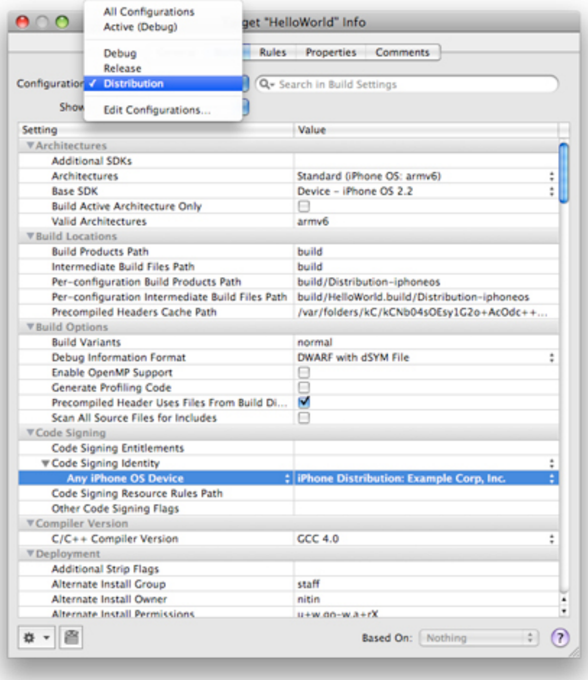


Edit configuration list:

Debug
Release
Distribution

Command-line builds use: Release

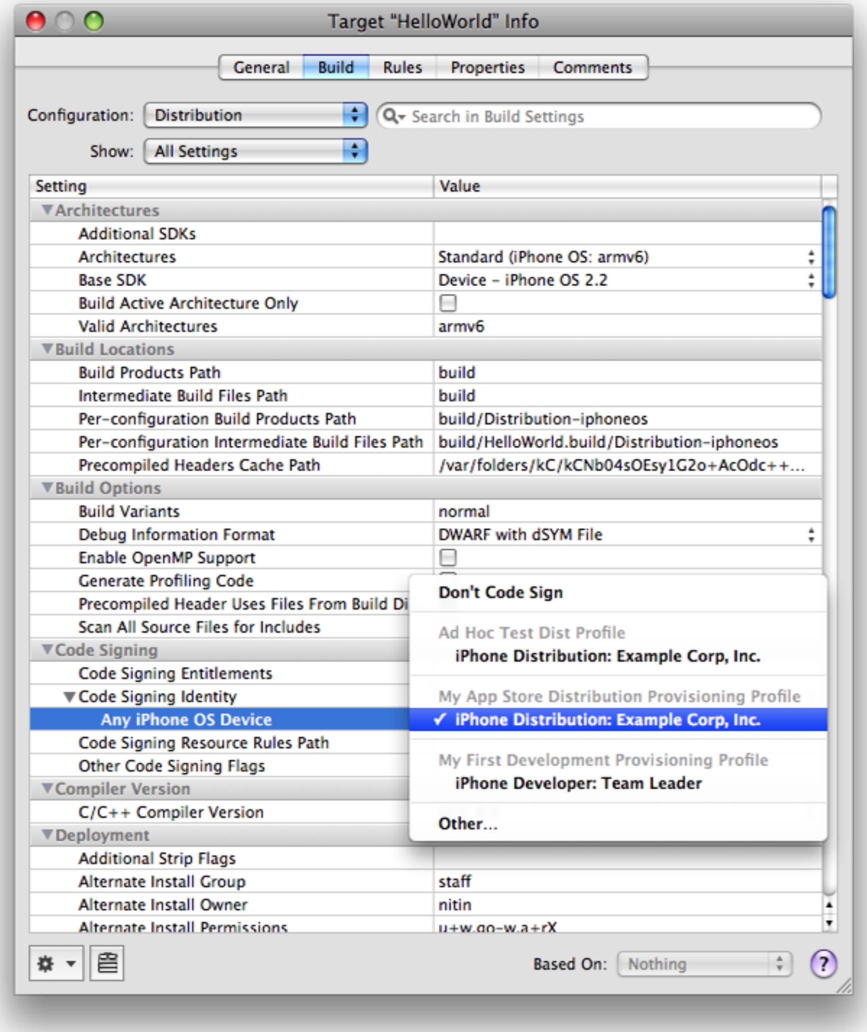
4. In the Target Info window, select the ‘Build’ tab and set the ‘Configuration’ to ‘Distribution’



Configuration: Distribution

Setting	Value
▼Architectures	
Additional SDKs	
Architectures	Standard (iPhone OS: armv6)
Base SDK	Device - iPhone OS 2.2
Build Active Architecture Only	□
Valid Architectures	armv6
▼Build Locations	
Build Products Path	build
Intermediate Build Files Path	build
Per-configuration Build Products Path	build/Distribution-iphoneos
Per-configuration Intermediate Build Files Path	build/HelloWorld.build/Distribution-iphoneos
Precompiled Headers Cache Path	/var/folders/kC/kCNb04sOEsy1G2o+AcOdc++...
▼Build Options	
Build Variants	normal
Debug Information Format	DWARF with dSYM File
Enable OpenMP Support	□
Generate Profiling Code	□
Precompiled Header Uses Files From Build Di...	☑
Scan All Source Files for Includes	□
▼Code Signing	
Code Signing Entitlements	
▼Code Signing Identity	
Any iPhone OS Device	iPhone Distribution: Example Corp, Inc.
Code Signing Resource Rules Path	
Other Code Signing Flags	
▼Compiler Version	
C/C++ Compiler Version	GCC 4.0
▼Deployment	
Additional Strip Flags	
Alternate Install Group	staff
Alternate Install Owner	nitin
Alternate Install Permissions	u+w,go-w,a+rX

5. In the Target Info window, navigate to the ‘Build’ pane. Click the ‘Any iOS Device’ pop-up menu below the ‘Code Signing Identity’ field and select the iOS Distribution Certificate/Provisioning Profile pair you wish to sign and install your code with. Your iOS Distribution certificate will be in bold with the Provisioning Profile associated with it in grey above. In the example below, ‘iOS Distribution : Example Corp, Inc.’ is the Distribution Certificate and ‘My App Store Distribution Provisioning Profile’ is the .mobileprovision file paired with it.


Target "HelloWorld" Info
General
Build
Rules
Properties
Comments
Search in Build Settings
Don't Code Sign
Ad Hoc Test Dist Profile
iPhone Distribution: Example Corp, Inc.
My App Store Distribution Provisioning Profile
✓ iPhone Distribution: Example Corp, Inc.
My First Development Provisioning Profile
iPhone Developer: Team Leader
Other...

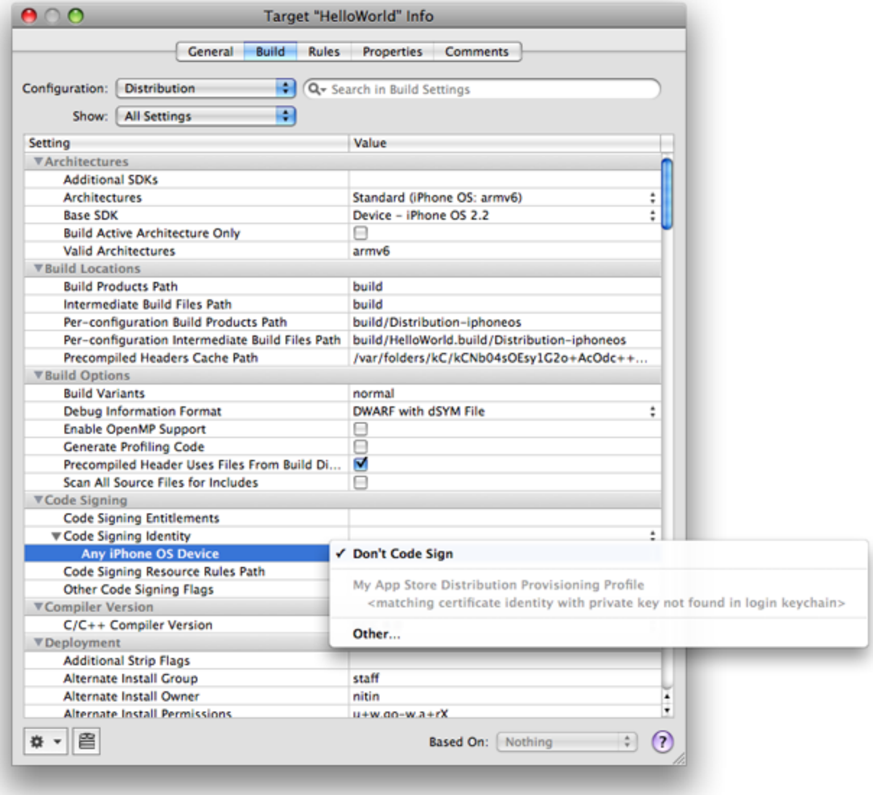
Configuration: Distribution

Show: All Settings

Setting	Value
▼Architectures	
Additional SDKs	
Architectures	Standard (iPhone OS: armv6)
Base SDK	Device - iPhone OS 2.2
Build Active Architecture Only	□
Valid Architectures	armv6
▼Build Locations	
Build Products Path	build
Intermediate Build Files Path	build
Per-configuration Build Products Path	build/Distribution-iphoneos
Per-configuration Intermediate Build Files Path	build/HelloWorld.build/Distribution-iphoneos
Precompiled Headers Cache Path	/var/folders/kC/kCNb04sOEsy1G2o+AcOdc++...
▼Build Options	
Build Variants	normal
Debug Information Format	DWARF with dSYM File
Enable OpenMP Support	□
Generate Profiling Code	
Precompiled Header Uses Files From Build Di	
Scan All Source Files for Includes	
▼Code Signing	
Code Signing Entitlements	
▼Code Signing Identity	
Any iPhone OS Device	✓ iPhone Distribution: Example Corp, Inc.
Code Signing Resource Rules Path	
Other Code Signing Flags	
▼Compiler Version	
C/C++ Compiler Version	
▼Deployment	
Additional Strip Flags	
Alternate Install Group	staff
Alternate Install Owner	nitin
Alternate Install Permissions	u+w,go-w,a+rX

Based On: Nothing

Note: If the private key for your iOS Distribution certificate is missing, you will be unable to select the iOS Distribution Certificate/Provisioning Profile pair and you will see the following. Importing the private key for your iOS Distribution certificate will correct this.

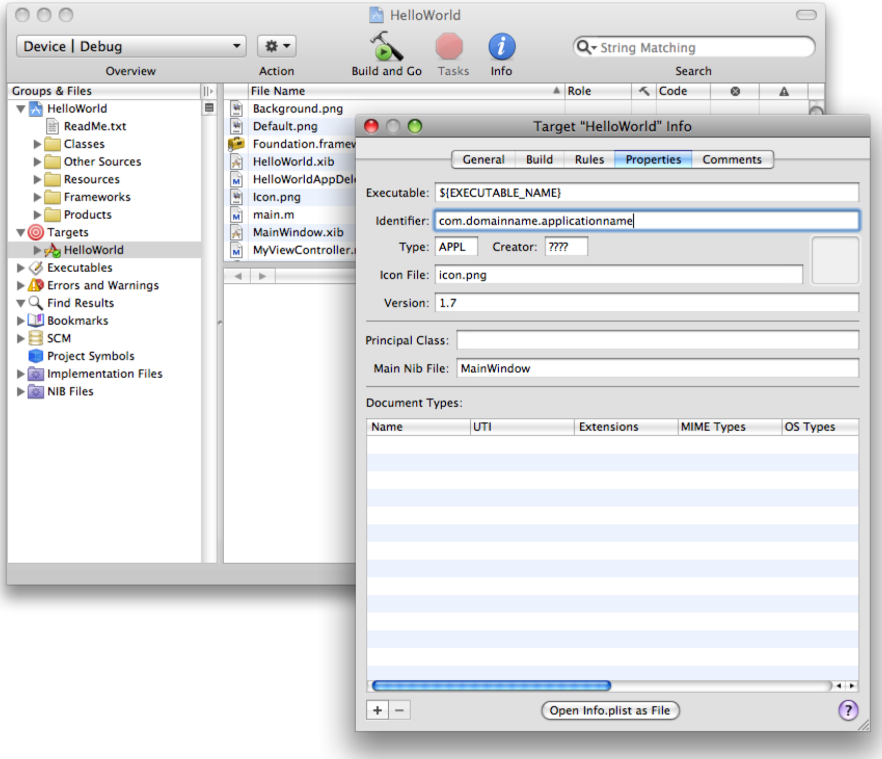

Target "HelloWorld" Info
General
Build
Rules
Properties
Comments
Search in Build Settings
Don't Code Sign
My App Store Distribution Provisioning Profile
<matching certificate identity with private key not found in login keychain>
Other...
Based On:
Nothing

Configuration: Distribution

Show: All Settings

Setting	Value
▼Architectures	
Additional SDKs	
Architectures	Standard (iPhone OS: armv6)
Base SDK	Device - iPhone OS 2.2
Build Active Architecture Only	□
Valid Architectures	armv6
▼Build Locations	
Build Products Path	build
Intermediate Build Files Path	build
Per-configuration Build Products Path	build/Distribution-iphoneos
Per-configuration Intermediate Build Files Path	build/HelloWorld.build/Distribution-iphoneos
Precompiled Headers Cache Path	/var/folders/kC/kCNb04sOEsy1G2o+AcOdc++...
▼Build Options	
Build Variants	normal
Debug Information Format	DWARF with dSYM File
Enable OpenMP Support	□
Generate Profiling Code	□
Precompiled Header Uses Files From Build Di...	☑
Scan All Source Files for Includes	□
▼Code Signing	
Code Signing Entitlements	
▼Code Signing Identity	
Any iPhone OS Device	✓ Don't Code Sign
Code Signing Resource Rules Path	
Other Code Signing Flags	
▼Compiler Version	
C/C++ Compiler Version	
▼Deployment	
Additional Strip Flags	
Alternate Install Group	staff
Alternate Install Owner	nitin
Alternate Install Permissions	u+w,go-w,a+rX

6. In the Properties Pane of the Target Info window, enter the Bundle Identifier portion of your App ID. If you have used an explicit App ID you must enter the Bundle Identifier portion of the App ID in the Identifier field. For example enter com.domainname.applicationname if your App ID is A1B2C3D4E5.com.domainname.applicationname. If you have used a wildcard asterisk character in your App ID, replace the asterisk with whatever string you choose.



Executable: ${EXECUTABLE_NAME}

Identifier: com.domainname.applicationname

Type: APPL Creator: ????

Icon File: icon.png

Version: 1.7

Principal Class:

Main Nib File: MainWindow

Here are example App IDs and what should be input into the Identifier field in Xcode.

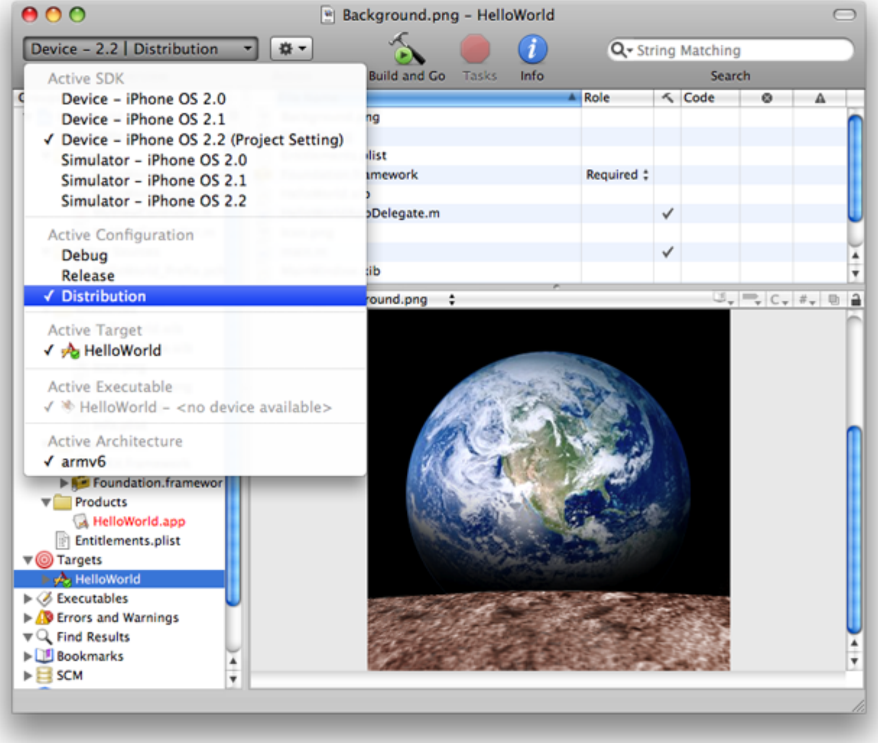
- Example App ID: *A1B2C3D4E5.com.domainname.applicationname*

Identifier to enter in Xcode: *com.domainname.applicationname*

- Example App ID: *A1B2C3D4E5.com.domainname.**
  Identifier to enter in Xcode: *com.domainname.*<name_of_application_or_suite>

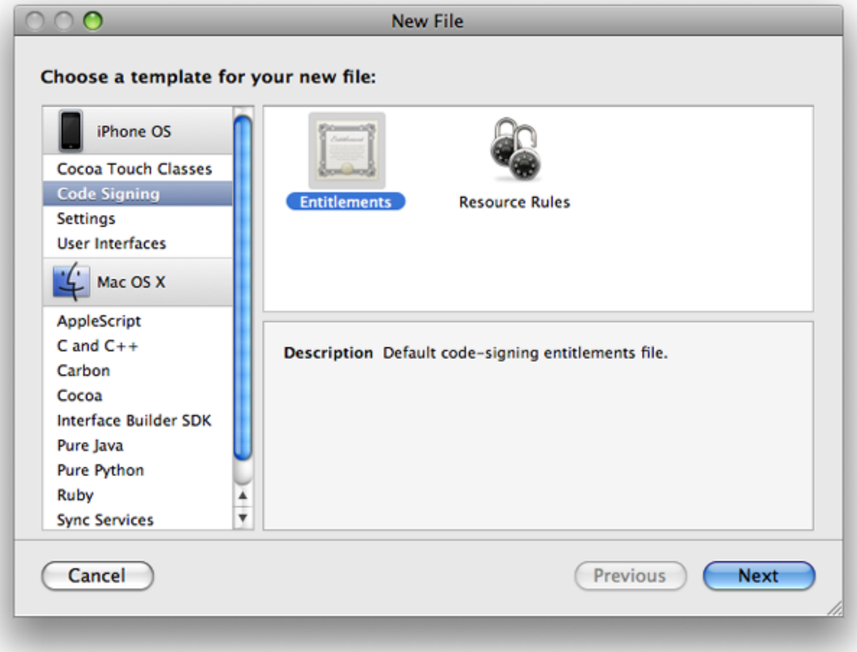
- Example App ID: *A1B2C3D4E5.**
  Identifier to enter in Xcode: <full_reverse_dns_company_and_application_or_suite_name>

7. In the project window, select the Distribution Active Configuration from the overview popup and set the Active SDK to the desired Device.



For App Store Distribution, skip to Step 12. For Ad Hoc Distribution, complete the following:

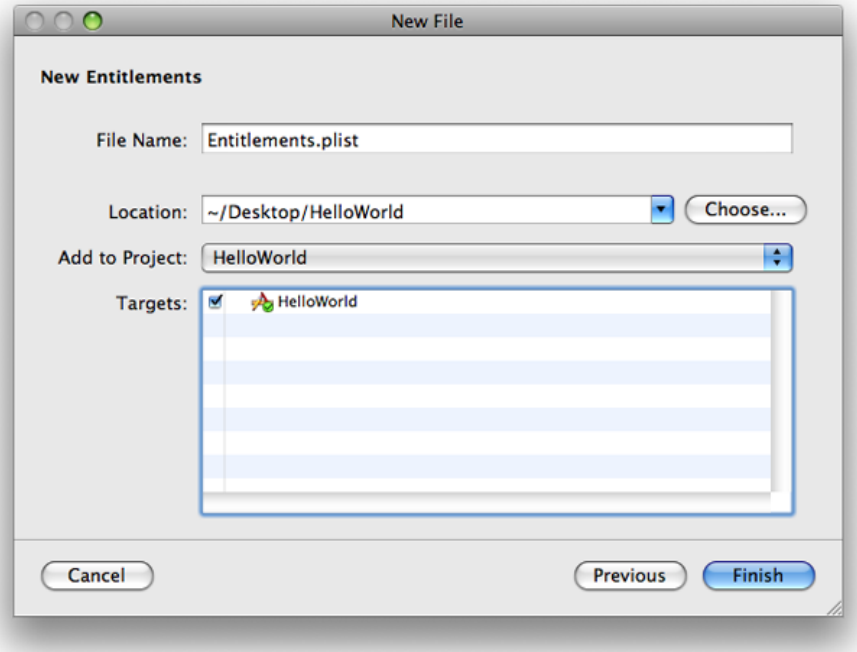
8. In the File Menu, select New File -> iPhone OS -> Code Signing -> Entitlements.



Choose a template for your new file:

Description Default code-signing entitlements file.

9. Name the file “Entitlements.plist" and click ‘Finish’. This creates a copy of the default entitlements file within the project.



New Entitlements

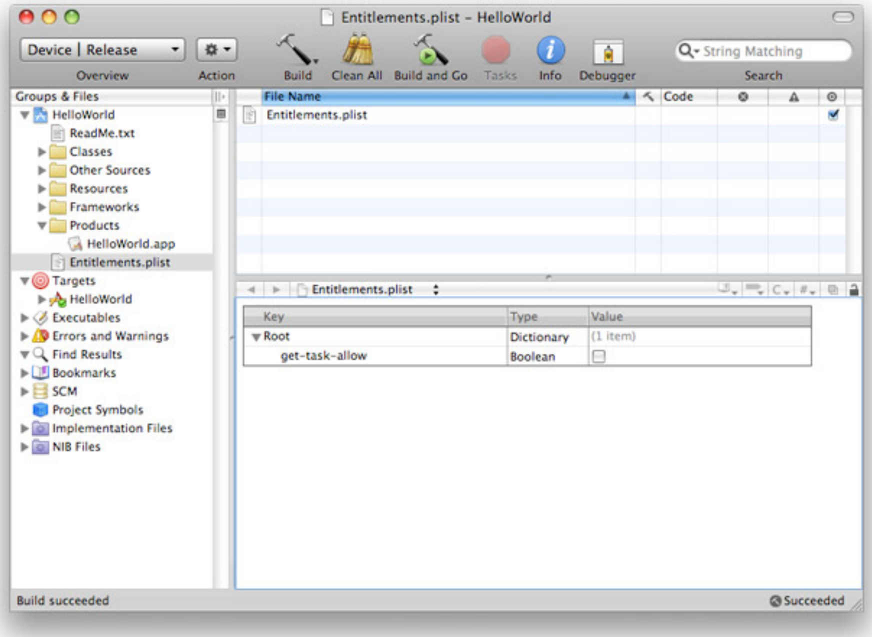
File Name: Entitlements.plist

Location: ~/Desktop/HelloWorld

Add to Project: HelloWorld

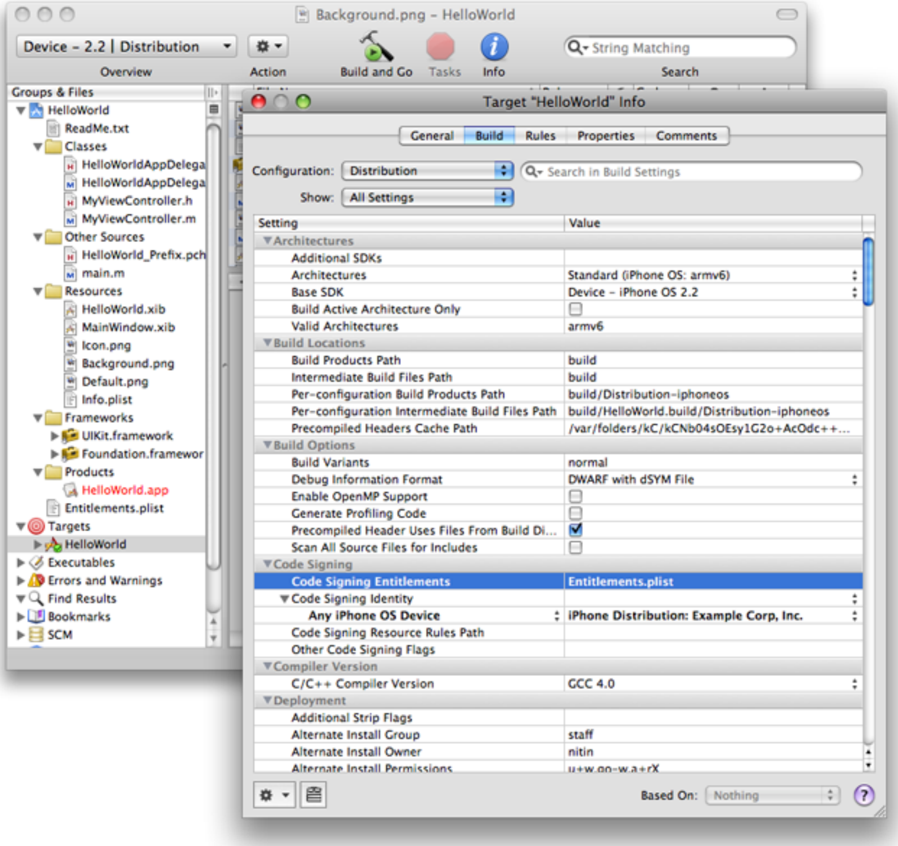
Targets: ☑ HelloWorld

10. Select the new Entitlments.plist file and uncheck the “get-task-allow” property. Save the Entitlements.plist file.



Key	Type	Value
Root	Dictionary	(1 item)
get-task-allow	Boolean	□

11. Select the Target and open the Build settings inspector. In the ‘Code Signing Entitlements’ build setting, type in the filename of the new Entitlements.plist file including the extension. There is no need to specify a path unless you have put the Entitlements.plist file somewhere other than the top level of the project.



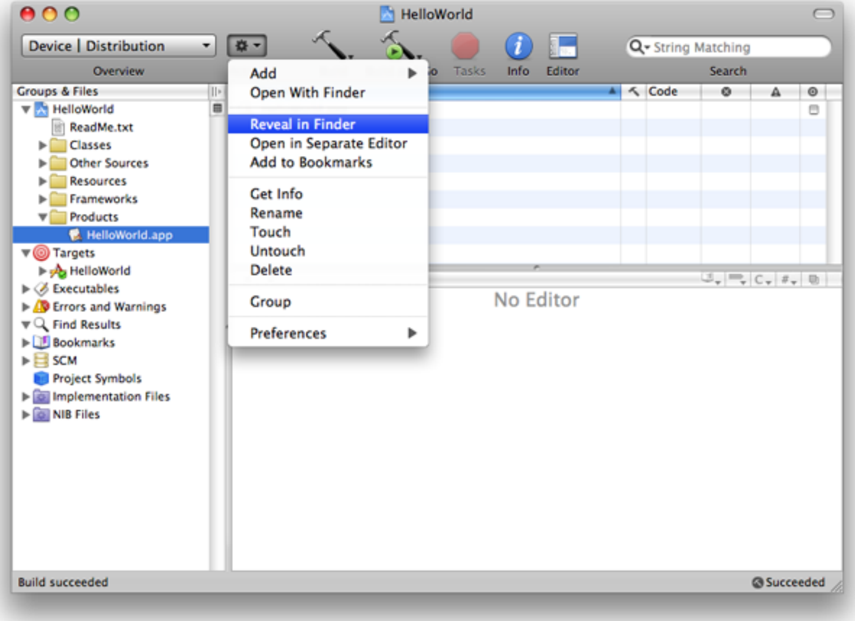
Configuration: Distribution

Show: All Settings

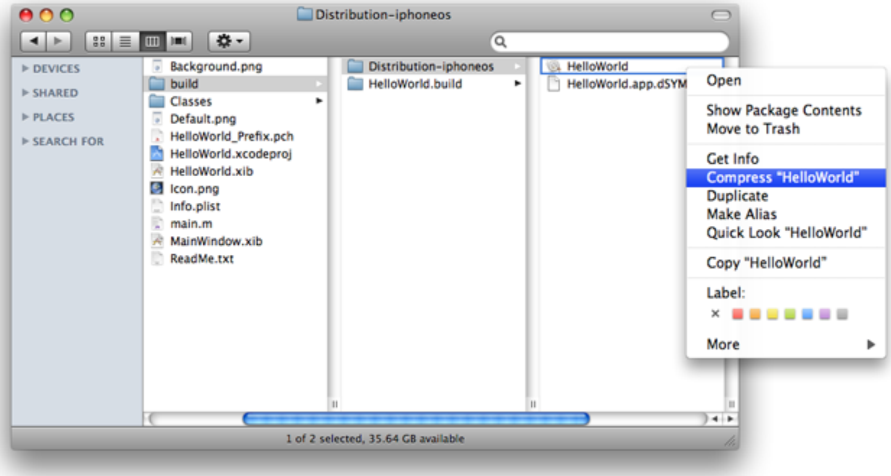
Setting	Value
Architectures	
Additional SDKs	
Architectures	Standard (iPhone OS: armv6)
Base SDK	Device – iPhone OS 2.2
Build Active Architecture Only	□
Valid Architectures	armv6
Build Locations	
Build Products Path	build
Intermediate Build Files Path	build
Per-configuration Build Products Path	build/Distribution-iphoneos
Per-configuration Intermediate Build Files Path	build/HelloWorld.build/Distribution-iphoneos
Precompiled Headers Cache Path	/var/folders/kC/kCNb04sOEsy1G2o+AcOdc++...
Build Options	
Build Variants	normal
Debug Information Format	DWARF with dSYM File
Enable OpenMP Support	□
Generate Profiling Code	□
Precompiled Header Uses Files From Build Di...	☑
Scan All Source Files for Includes	□
Code Signing	
Code Signing Entitlements	Entitlements.plist
Code Signing Identity	
Any iPhone OS Device	iPhone Distribution: Example Corp, Inc.
Code Signing Resource Rules Path	
Other Code Signing Flags	
Compiler Version	
C/C++ Compiler Version	GCC 4.0
Deployment	
Additional Strip Flags	
Alternate Install Group	staff
Alternate Install Owner	nitin
Alternate Install Permissions	u+w,go-w,a+rX

12. Click ‘Build’. (Note: Your binary must contain a flattened, square-image icon that is 57x57 pixels. This icon is displayed on the iPhone or iPod touch home screen.)

13. Highlight the app located within the "Products" sub-folder and select ‘Reveal in Finder’ from the Action popup.



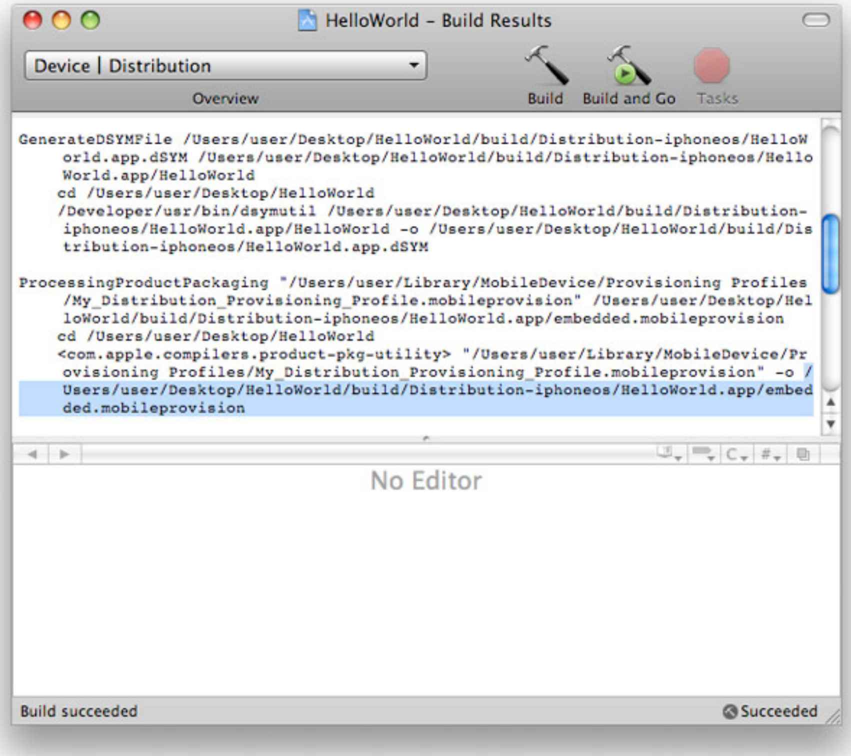
14. Use the compress option in Finder to create a .zip file containing your application. Be sure to compress only the .app file only and not the entire build folder.



Verifying a Successful Distribution Build

To confirm your build was successful, check for the following:

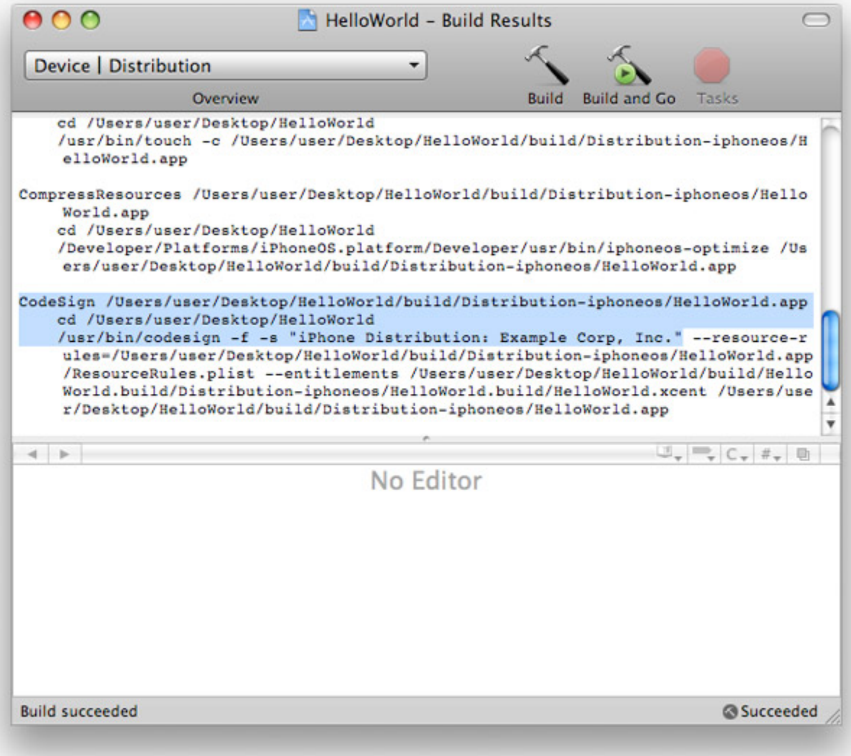
1. Open the Build Log detail view and confirm the presence of the "embedded.mobileprovision” file. This will take you to the line in the build log that shows the provisioning profile was successfully called. Ensure that the embedded.mobileproviion is located in the proper “Distribution” build directory and is not located in a “Debug” or “Release” build directory. Also, confirm that the destination path (at the very end of the build message) is the app you are building.



```
GenerateDSYMFile /Users/user/Desktop/HelloWorld/build/Distribution-iphoneos/HelloW
     orld.app.dSYM /Users/user/Desktop/HelloWorld/build/Distribution-iphoneos/Hello
     World.app/HelloWorld
    cd /Users/user/Desktop/HelloWorld
    /Developer/usr/bin/dsymutil /Users/user/Desktop/HelloWorld/build/Distribution-
     iphoneos/HelloWorld.app/HelloWorld -o /Users/user/Desktop/HelloWorld/build/Dis
     tribution-iphoneos/HelloWorld.app.dSYM

ProcessingProductPackaging "/Users/user/Library/MobileDevice/Provisioning Profiles
     /My_Distribution_Provisioning_Profile.mobileprovision" /Users/user/Desktop/Hel
     loWorld/build/Distribution-iphoneos/HelloWorld.app/embedded.mobileprovision
    cd /Users/user/Desktop/HelloWorld
    <com.apple.compilers.product-pkg-utility> "/Users/user/Library/MobileDevice/Pr
     ovisioning Profiles/My_Distribution_Provisioning_Profile.mobileprovision" -o /
     Users/user/Desktop/HelloWorld/build/Distribution-iphoneos/HelloWorld.app/embed
     ded.mobileprovision
```

2. Search for the term “CodeSign” in the Build Log detail view - this will take you to the line in the build log that confirms your application was signed by your iOS Certificate.



```
    cd /Users/user/Desktop/HelloWorld
    /usr/bin/touch -c /Users/user/Desktop/HelloWorld/build/Distribution-iphoneos/H
     elloWorld.app

CompressResources /Users/user/Desktop/HelloWorld/build/Distribution-iphoneos/Hello
     World.app
    cd /Users/user/Desktop/HelloWorld
    /Developer/Platforms/iPhoneOS.platform/Developer/usr/bin/iphoneos-optimize /Us
     ers/user/Desktop/HelloWorld/build/Distribution-iphoneos/HelloWorld.app

CodeSign /Users/user/Desktop/HelloWorld/build/Distribution-iphoneos/HelloWorld.app
    cd /Users/user/Desktop/HelloWorld
    /usr/bin/codesign -f -s "iPhone Distribution: Example Corp, Inc." --resource-r
     ules=/Users/user/Desktop/HelloWorld/build/Distribution-iphoneos/HelloWorld.app
     /ResourceRules.plist --entitlements /Users/user/Desktop/HelloWorld/build/Hello
     World.build/Distribution-iphoneos/HelloWorld.build/HelloWorld.xcent /Users/use
     r/Desktop/HelloWorld/build/Distribution-iphoneos/HelloWorld.app
```

If your project is lacking any of the above files or pointing to the wrong directory, do the following:

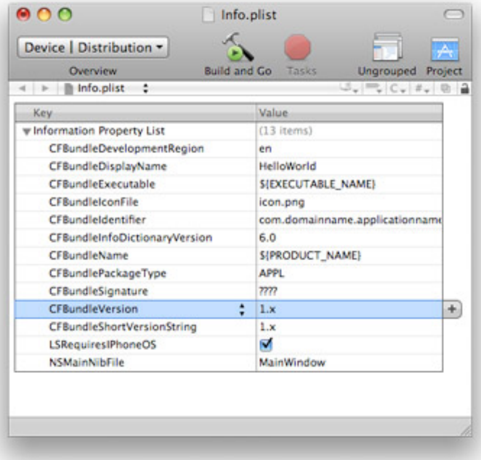
1. Select the Target and open the Build Settings Inspector. Confirm you are in the Distribution Configuration.
2. Delete the Code Signing Identity: iOS Distribution : *COMPANYNAME*
3. In the Xcode Build Menu, select Clean all Targets.

4. Delete any existing build directories in your Xcode project using Finder.
5. Re-launch Xcode and open your Project.
6. Re-enter the code-signing identity iOS Distribution : *COMPANYNAME* in the Target Build Settings Inspector.
7. Rebuild your Project.

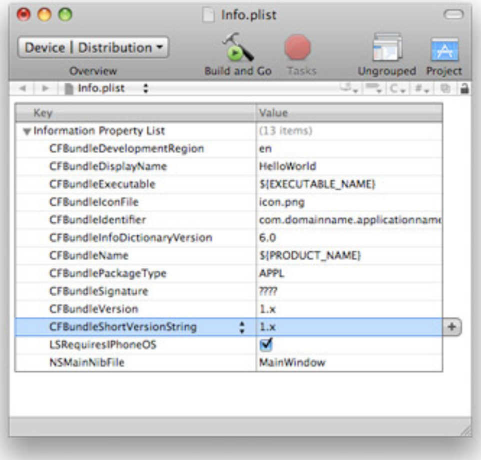
Updating your Application

The App Store uses three pieces of information in your application to identify a submission as an update to an existing application. *When you are submitting an update of your application to iTunes Connect for App Store distribution, make sure to:

1. Use the same Distribution Provisioning Profile to build each new version of your application
2. Increment the CFBundleVersion and CFBundleShortVersionString values in your project Info.plist file. Note: Version numbers must be period-delimited sequences of positive integers (1.0 to 1.1, or 2.2.1 to 2.2.2).



Key	Value
▼Information Property List	(13 items)
CFBundleDevelopmentRegion	en
CFBundleDisplayName	HelloWorld
CFBundleExecutable	${EXECUTABLE_NAME}
CFBundleIconFile	icon.png
CFBundleIdentifier	com.domainname.applicationname
CFBundleInfoDictionaryVersion	6.0
CFBundleName	${PRODUCT_NAME}
CFBundlePackageType	APPL
CFBundleSignature	????
CFBundleVersion	1.x
CFBundleShortVersionString	1.x
LSRequiresIPhoneOS	☑
NSMainNibFile	MainWindow



Key	Value
▼Information Property List	(13 items)
CFBundleDevelopmentRegion	en
CFBundleDisplayName	HelloWorld
CFBundleExecutable	${EXECUTABLE_NAME}
CFBundleIconFile	icon.png
CFBundleIdentifier	com.domainname.applicationname
CFBundleInfoDictionaryVersion	6.0
CFBundleName	${PRODUCT_NAME}
CFBundlePackageType	APPL
CFBundleSignature	????
CFBundleVersion	1.x
CFBundleShortVersionString	1.x
LSRequiresIPhoneOS	☑
NSMainNibFile	MainWindow