



iOS Developer Program

Standard Program User Guide for iOS 4

Version 2.7

Welcome	1
Getting Started	
Development	2
Setting up your iOS Development Team (if applicable)	
<i>Team Member Roles and Responsibilities</i>	2
<i>Adding Team Admins and Team Members</i>	3
Obtaining your iOS Development Certificate	
<i>Generating a Certificate Signing Request</i>	5
<i>Submitting a Certificate Signing Request for Approval</i>	8
<i>Approving Certificate Signing Requests</i>	11
<i>Downloading and Installing Development Certificates</i>	12
<i>Saving your Private Key and Transferring to other Systems</i>	13
<i>Managing Certificate Expiration</i>	13
Assigning Apple Devices to your Team	
<i>Locating a Unique Device ID with Xcode</i>	14
<i>Locating a Unique Device ID with iTunes 7.7 or later</i>	15
<i>Locating a Unique Device ID with iPhone Configuration Utility</i>	16
<i>Adding Individual Devices</i>	17
<i>Bulk Upload of Devices</i>	18
<i>Removing Devices from your Development Team</i>	20
Installing Pre-release versions of iOS	
Creating your App ID	
<i>Wildcard App IDs</i>	23
<i>Explicit App IDs</i>	24
<i>Generating an App ID</i>	24
<i>Configuring your App ID for Apple Push Notification service</i>	28
<i>Configuring your App ID for In App Purchase and Game Center</i>	31
<i>Enabling Keychain Data Sharing with your App ID</i>	32
Creating and Downloading Development Provisioning Profiles	
<i>Creating a Development Provisioning Profile</i>	35
<i>Installing a Development Provisioning Profile</i>	37
Building and Installing your Development Application	
Distribution	42

Preparing your Application for Distribution	
<i>Obtaining your iOS Distribution Certificate</i>	42
<i>Generating a Certificate Signing Request</i>	42
<i>Submitting a Certificate Signing Request for Approval</i>	45
<i>Downloading and Installing iOS Distribution Certificates</i>	48
<i>Saving your Private Key and Transferring to Other Systems</i>	48
<i>Managing Certificate Expiration</i>	49
<i>Creating and Downloading a Distribution Provisioning Profile for App Store Distribution</i>	50
<i>Creating and Downloading a Distribution Provisioning Profile for Ad Hoc Distribution</i>	51
<i>Building your Application with Xcode for Distribution</i>	52
Submitting your application via iTunes Connect	
<i>Distribution Terms & Conditions</i>	61
<i>Gathering your Application Metadata</i>	61
<i>Preparing your Application Artwork</i>	62
<i>Uploading your Application</i>	63
<i>Guidelines for Submitting Updates to Your iPhone Application</i>	63
<i>Updating your Application</i>	63
Distributing your application with Ad Hoc Distribution	
Glossary	65
Revision History	66
<i>Version 2.7 - September 1, 2010</i>	66
<i>Version 2.6 - March 4, 2010</i>	66
<i>Version 2.5 - September 9, 2009</i>	66

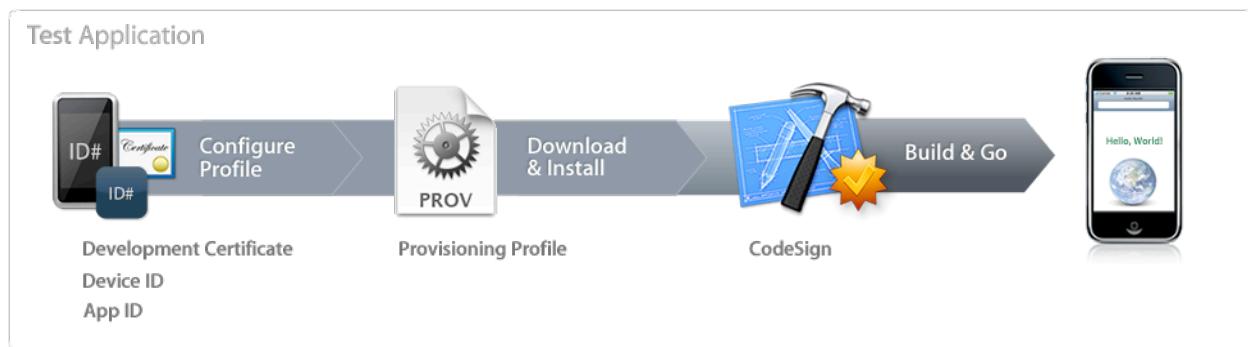
Welcome

Getting Started

Welcome to the iOS Developer Program, your destination for all the resources you need to develop, test and distribute your iOS applications. The iOS Dev Center and Program Portal are designed to walk you through each step necessary to get your application from code to customer.

In order to install and test your application on an Apple device, you'll need to complete the following tasks in the Program Portal:

- Set up your Development Team (for those enrolled as a Company)
- Request and Authorize iOS Development Certificates
- Designate Apple Devices for your Development Team
- Create unique App IDs for your Applications
- Create and Download a Development Provisioning Profile



When you are ready to distribute your application, your Team Agent will need to complete the following additional tasks in the Portal:

- Create and Download an iOS Distribution Certificate
- Create and Download a Distribution Provisioning Profile
- Build your application with Xcode
- Submit your application for Distribution



The user guide contains detailed information regarding each of the above steps.

Development

The following topics will inform you about the required steps for installing your code on an Apple device (iPhone, iPad or iPod touch) for development and debugging purposes.

Setting up your iOS Development Team (if applicable)

If you have enrolled your company in the iOS Developer Program, you first need to set up your iOS Development Team in the 'Team' section of the Program Portal.

If you have enrolled as an Individual in the iOS Developer Program, you are designated the Team Agent and therefore have access to all Program Portal capabilities. You can not add any additional team members.

Team Member Roles and Responsibilities

A development team consists of individuals with the following roles:

- Team Agent

The Team Agent (the original enrollee accepted into the iOS Developer Program) can invite Team Admins and Team Members, approve digital certificate requests, enter devices into the Program Portal, create App IDs, obtain an iOS Distribution Certificate, and create Development and Distribution Provisioning Profiles. In addition, the Team Agent can enable App IDs for Apple Push Notification service and In App Purchase and create SSL Certificates used to connect to the Apple Push Notification service. The Team Agent is the primary contact for the Development Team and is responsible for accepting all iOS Developer Program agreements.

- Team Admin

Team Admins can invite new Team Admins and Team Members, approve digital certificate requests, enter devices into the Program Portal, and create Development Provisioning Profiles. All teams with multiple members require a minimum of two Team Admins (including the Team Agent). There is no maximum for the number of Team Admins that can be on a team with multiple members.

- Team Member

Team Members can request digital certificates and download Development Provisioning Profiles. All three types of members have the ability to test applications on Apple devices.

Adding Team Admins and Team Members

Team Agents and Team Admins can add new Team Members, who can have either an Admin or Member role, by navigating to the 'People' tab in the Apple Developer [Member Center](#). Navigate to the 'Invitations' section and click on the 'Invite Person' button to invite new people to join your team. You will be asked to provide the name, email address, and Program Role you wish to assign to your new member. After hitting 'Send Invitation', the Member Center will generate and deliver an email to the invitee(s). Team Admins will be notified via a separate email when an invitation has been accepted.

The screenshot shows the 'Member Center – Apple Developer' interface. The top navigation bar includes links for 'People', 'Programs & Add-ons', and 'Your Account'. The user is logged in as 'Hi, Nitin Mishra | Log out'. On the left, a sidebar lists 'All People', 'iPhone Developers', 'Mac Developers', and 'Invitations', with 'Invitations' currently selected. The main content area is titled 'Invite People to Your Team' and contains a sub-section 'Invite a Person to Your Team' with a placeholder for sending an invitation. Below this is a 'Recent Invitations' table:

Name	Email	Date Sent	Status	Details
Business Time	nitin+1@apple.com	Jul 23, 2009	Accepted	Details

At the bottom of the page, there is a footer with copyright information: 'Copyright © 2010 Apple Inc. All rights reserved.' followed by links to 'Terms of Use' and 'Privacy Policy'.

The screenshot shows the Apple Developer Member Center interface. The top navigation bar includes links for 'Member Center - Apple Developer', 'Developer', 'People', 'Programs & Add-ons', and 'Your Account'. A sidebar on the left lists 'Organization: Development Company' and categories: 'All People', 'iPhone Developers', 'Mac Developers', and 'Invitations' (which is currently selected). The main content area is titled 'Invite a Person to Your Team'. It contains two sections: 'Contact Information' (with fields for First Name, Last Name, and Email Address) and 'Access and Roles' (under the 'iPhone Developer Program'). In the 'Access and Roles' section, there are three radio button options: 'No Access', 'Member', and 'Admin' (which is selected). Below these options is a link to 'iPhone Developer Program Roles Overview' and an icon of two iPhone devices. At the bottom right of the dialog are 'Cancel' and 'Send Invitation' buttons. The footer of the page includes copyright information: 'Copyright © 2010 Apple Inc. All rights reserved.' and links to 'Terms of Use' and 'Privacy Policy'.

If you need to designate a new Team Agent, change your name or change your address, please contact [Apple Developer Support](#).

Obtaining your iOS Development Certificate



In the 'Certificates' section of the iOS Developer Program Portal, you can request individual iOS Development Certificates. All iOS applications must be signed by a valid certificate before they can run on an Apple device. In order to sign applications for testing purposes, Team Members need an iOS Development Certificate.

A digital identity is an electronic means of identification consisting of a secret "private key" and a shared "public key". This private key allows Xcode to sign your iOS application binary.

The digital certificates you request and download are electronic documents that associate your digital identity with other information, including your name, email address, or business. An iOS Development Certificate is restricted to application development only and is valid for a limited amount of time. The Apple Certification Authority can also invalidate ("revoke") a certificate before it expires.

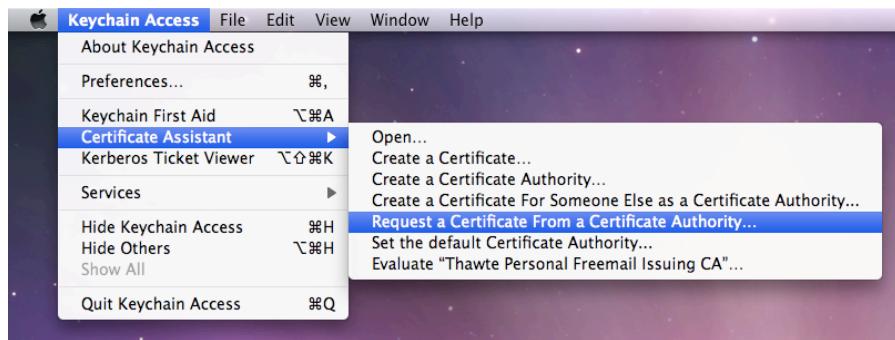
Generating a Certificate Signing Request

To request an iOS Development 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 iPhone Developer 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 (OCSP) and Certificate Revocation List (CRL) to "Off".



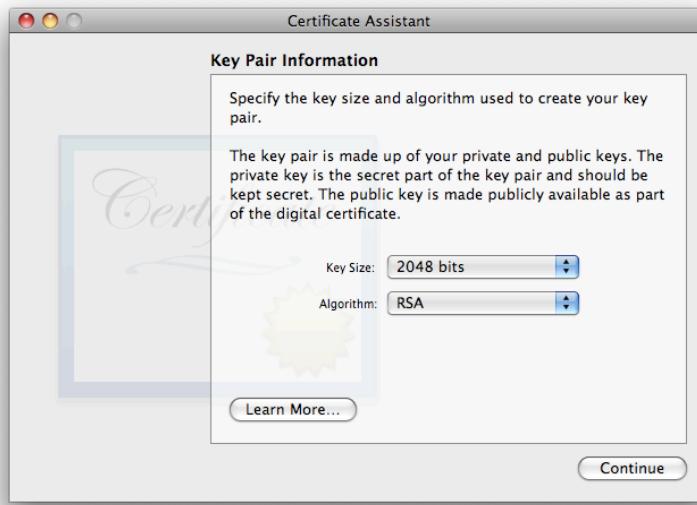
3. Choose Keychain Access -> Certificate Assistant -> Request a Certificate from a Certificate Authority. Note: If you have a noncompliant private key highlighted in the Keychain during this process, the resulting Certificate Request will not be accepted by the Program Portal. Confirm that you are selecting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



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 a unique name to identify for your iOS Developer Private Key in the Keychain Access application..
6. No CA (Certificate Authority) Email Address is required. The 'Required' message will be removed after completing the following step.
7. Select the 'Saved to Disk' radio button and if prompted, select 'Let me specify key pair information' and click 'Continue'.



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'.



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

Submitting a Certificate Signing Request for Approval

- After creating a CSR, log in to the iOS Developer Program Portal and navigate to 'Certificates' -> 'Development' and click 'Request Certificate'

The screenshot shows the 'Development' tab selected in the 'Certificates' section of the iPhone Developer Program portal. In the 'Current Development Certificates' section, there is a message: 'You currently do not have a valid certificate' with a 'Request Certificate' button. Below it, a note says: '*If you do not have the WWDR intermediate certificate installed, [click here to download now.](#)' In the 'Team Signing Requests (0)' section, there is a message: 'Your team currently does not have any active signing requests' with 'Reject Selected' and 'Approve Selected' buttons.

2. Click the 'Choose File' button on the next page, then 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.

The screenshot shows the 'Create iPhone Development Certificate' page in the Apple Developer Connection. The left sidebar has 'Certificates' selected. The main content area has tabs for 'Development' (selected), 'Distribution', 'History', and 'How To'. The 'How to create a development certificate:' section lists three steps:

- 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.
- Submit the CSR through the Program Portal to the Admin for approval.
 - Click the Development tab
 - Upload the certificate by choosing the file
 - Click Submit
- You will be notified by email when your CSR has been approved or rejected.

A 'Choose File' button is shown with the text 'no file selected'. A 'Submit' button is at the bottom right. At the bottom of the page, there are links for the Apple Online Store, Mailing Lists, RSS Feeds, Terms of Use, and Privacy Policy.

- Upon submission, Team Admins will be notified via email of the certificate request.

The screenshot shows the 'iPhone Developer Program' interface in the Apple Developer Connection. The left sidebar has a 'Certificates' section selected. The main content area is titled 'Current Development Certificates'. A message box says 'The Certificate Request has been submitted for approval.' Below it, a table lists a single certificate entry:

Name	Expiration Date	Provisioning Profiles	Status	Actions
Team Member			Pending Approval	

*If you do not have the WWDR intermediate certificate installed, click here to download.

- Once your CSR is approved or rejected by a Team Admin, you will be notified via email of the change in your certificate status.

Approving Certificate Signing Requests

Team Agents and Team Admins have the authority and responsibility to approve or reject all iOS Development Certificate requests. In order to approve/reject Team Members' requests, all Team Admins should first submit their own CSR for approval.

1. After submitting a CSR for approval, Team Admins will be directed to the 'Development' tab of the 'Certificates' section. Here, CSRs can be approved or rejected by selecting the checkbox next to each request and clicking on the desired action at the bottom of the page.
2. Once a CSR is approved or rejected, the requesting Team Member is notified via email of the change in their certificate status. Each iOS Development Certificate is available to both the Team Member who submitted the CSR for approval and to the Team Admin(s).

The screenshot shows the 'Development' tab of the 'Certificates' section in the iPhone Dev Center. The left sidebar has 'Certificates' selected. The main area shows 'Current Development Certificates' with one entry:

Name	Provisioning Profiles	Expiration Date	Status	Action
Team Leader		02 Sep 2010	Issued	Download Revoke

A note below says: '*If you do not have the WWDR intermediate certificate installed, [click here to download now](#).'

Below this is the 'Team Signing Requests (1)' section:

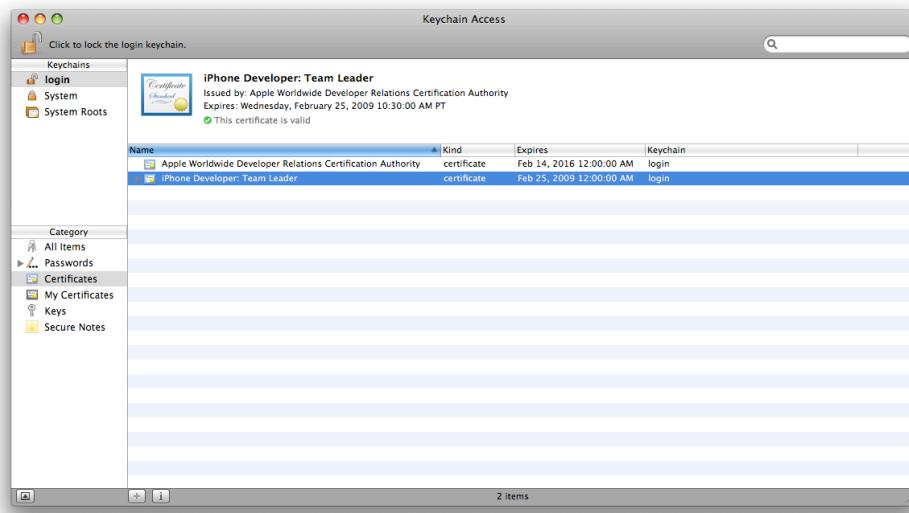
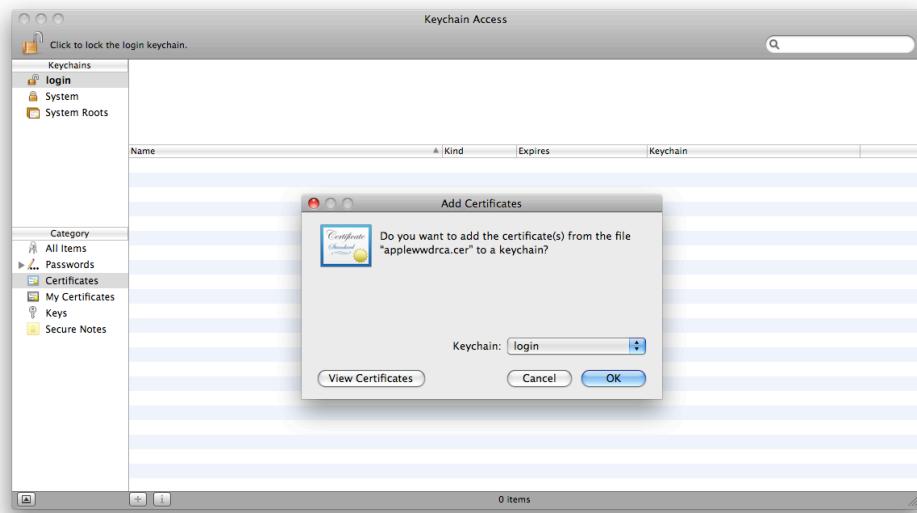
Signing Requests		Active Certificates	
Name	Email	Status	Date Submitted
<input checked="" type="checkbox"/> Team Member	TeamMember@example.com	Pending Approval	02 Sep 2009

At the bottom of this section are 'Approve Selected' and 'Reject Selected' buttons.

At the bottom of the page, there are links for the Apple Online Store, Apple Retail Store, reseller, Mailing Lists, RSS Feeds, Terms of Use, and Privacy Policy.

Downloading and Installing Development Certificates

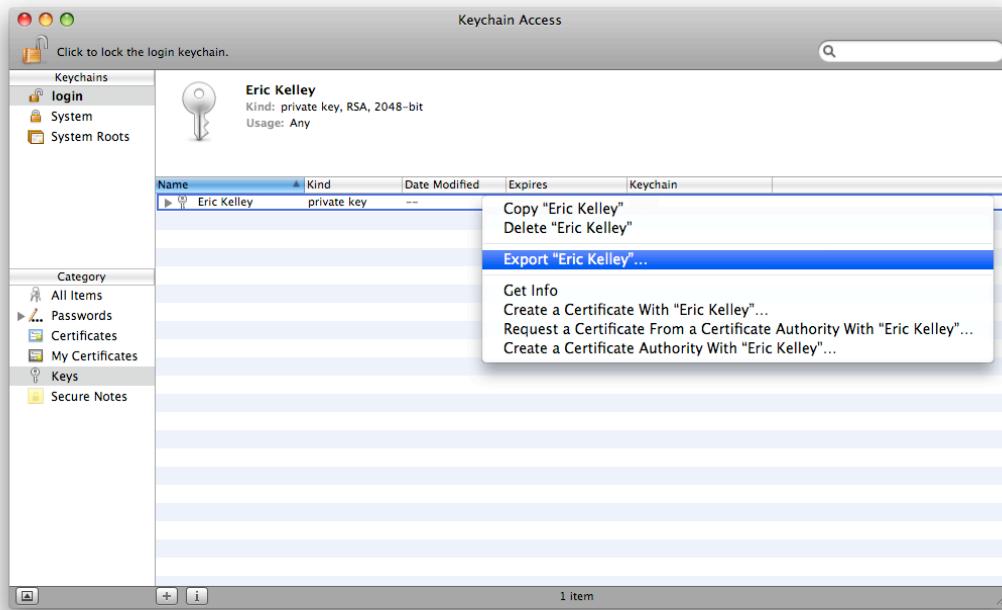
- Upon CSR approval, Team Members and Team Admins can download their certificates via the 'Certificates' section of the Program Portal. Click 'Download' next to the certificate name to download your iOS Development Certificate to your local machine.
- On your local machine, double-click the downloaded .cer file to launch Keychain Access and install your certificate.
- Team Members can only download their own iOS Development Certificates. Team Admins have the authority to download the public certificates of all of their Team Members. Apple never receives the private key for a CSR. The private keys are not available to anyone except the original key pair creator and are stored in the login keychain of that Team Member.



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 develop on multiple computers or decide to reinstall your system OS. **Without your private key, you will be unable to sign binaries in Xcode and test 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 and for enabling development on multiple systems, open up the Keychain Access Application and select the 'Keys' category.
2. Control-Click on the private key associated with your iOS Development Certificate and click 'Export Items' in the menu. The private key is identified by the iPhone Developer: <First Name> <Last Name> public certificate that is paired with it.
3. Save your key in the Personal Information Exchange (.p12) file format.
4. You will be prompted to create a password which is used when you attempt to import this key on another computer.
5. You can now transfer this .p12 file between systems. Double-click on the .p12 to install it on a system. You will be prompted for the password you entered in Step 4.



Managing Certificate Expiration

iOS Development Certificates are valid for one year from date of issue. After your iOS Development Certificate expires, you will no longer be able sign and install iOS applications on your devices. To continue development, you can revoke your iPhone Developer Certificate in the Apple Developer Member Center or iPhone Provisioning Portal. After your certificate has been revoked, you can create a new Certificate Signing Request and obtain a new iOS Development Certificate. **In the iPhone Developer Standard Program, revoking iOS Development Certificates will NOT affect applications already running on your device. However, when your iOS Development Certificate expires, your application will no longer run on your device.**

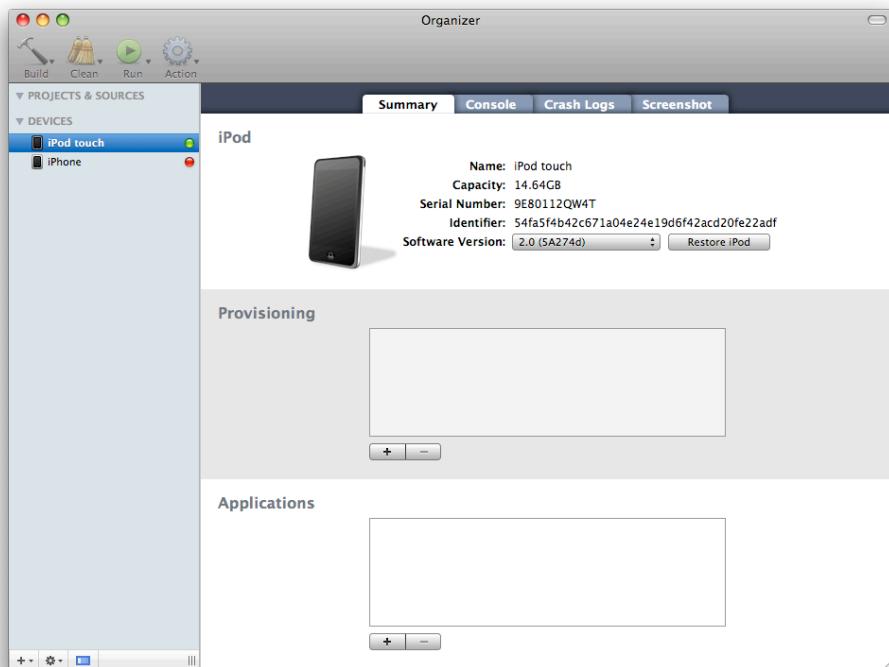
Assigning Apple Devices to your Team



The Devices section of the iOS Developer Program Portal allows you to enter the Apple devices that you will be using for your iOS development. In order to debug your iOS application on an Apple device, a Team Agent or Team Admin must first enter the Unique Device Identifier (UDID) for each iPhone, iPad and iPod touch into the Program Portal. The UDID is a 40 character string that is tied to a single device, similar to a serial number. These UDIDs are included in the provisioning profiles created later. You can input up to 100 devices for your development team. A development team can add up to 100 devices to their team in a given membership year. Removing devices does not restore the device slot.

Locating a Unique Device ID with Xcode

To find out your device's UDID, simply connect your device to your Mac and open Xcode. In Xcode, navigate to the 'Window' drop down menu and select 'Organizer'. The 40 character string in the Identifier field is your device's UDID. Team Members should send this string to their Team Admins for input into the Program Portal.



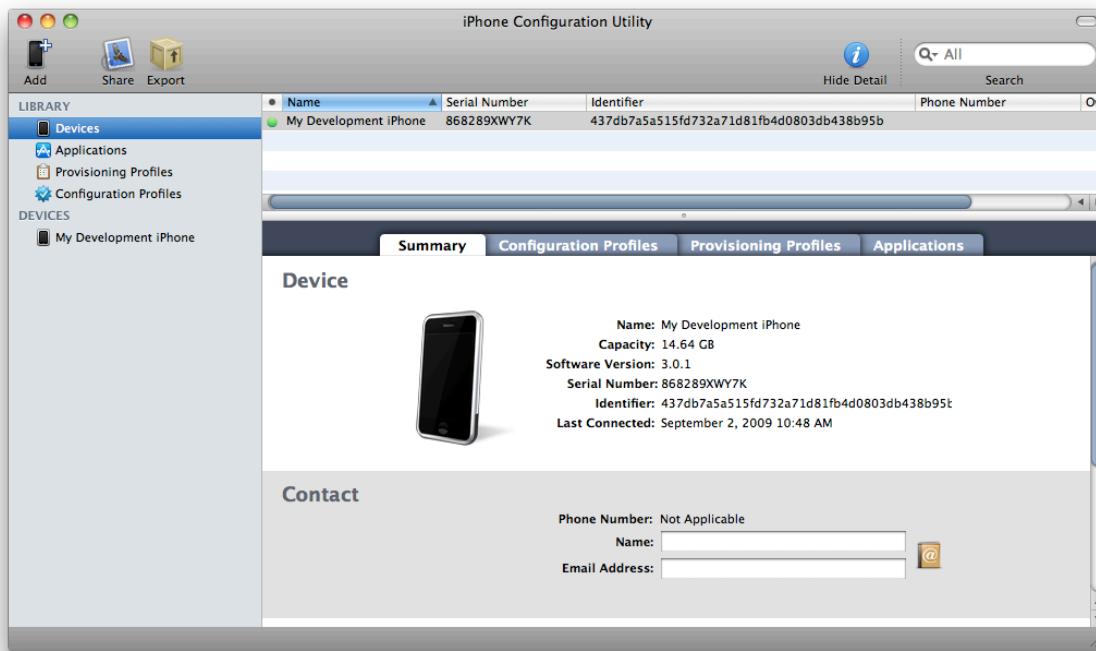
Locating a Unique Device ID with iTunes 7.7 or later

Alternatively, you can also find out your device's UDID using iTunes 7.7 or later. To do this, connect your device to your Mac and launch iTunes. In iTunes, select your device in the 'Devices' section and navigate to the Summary tab. Click on the Serial Number label to reveal the Identifier field and the 40 character UDID. Press Command+C to copy the UDID to your clipboard.



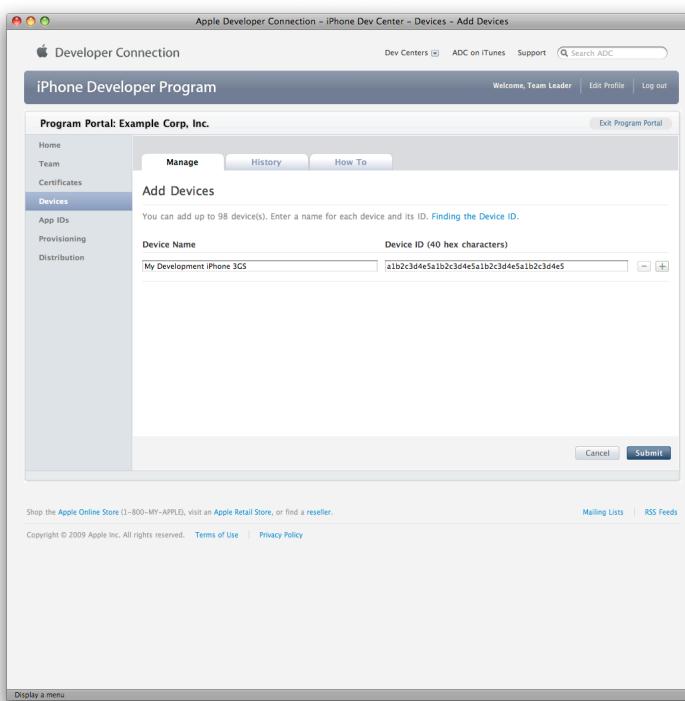
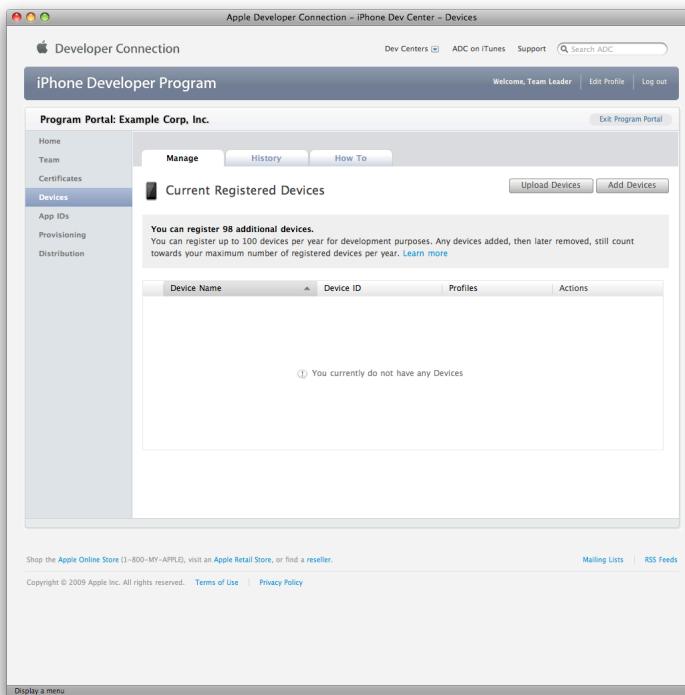
Locating a Unique Device ID with iPhone Configuration Utility

In iPhone Configuration Utility, navigate to your device in the 'Library' or 'Devices' section of the application and copy the 40 character string in the Identifier field on the 'Summary' tab. After locating your device's UDID, Team Members should send this string to their Team Admins for input into the Program Portal.



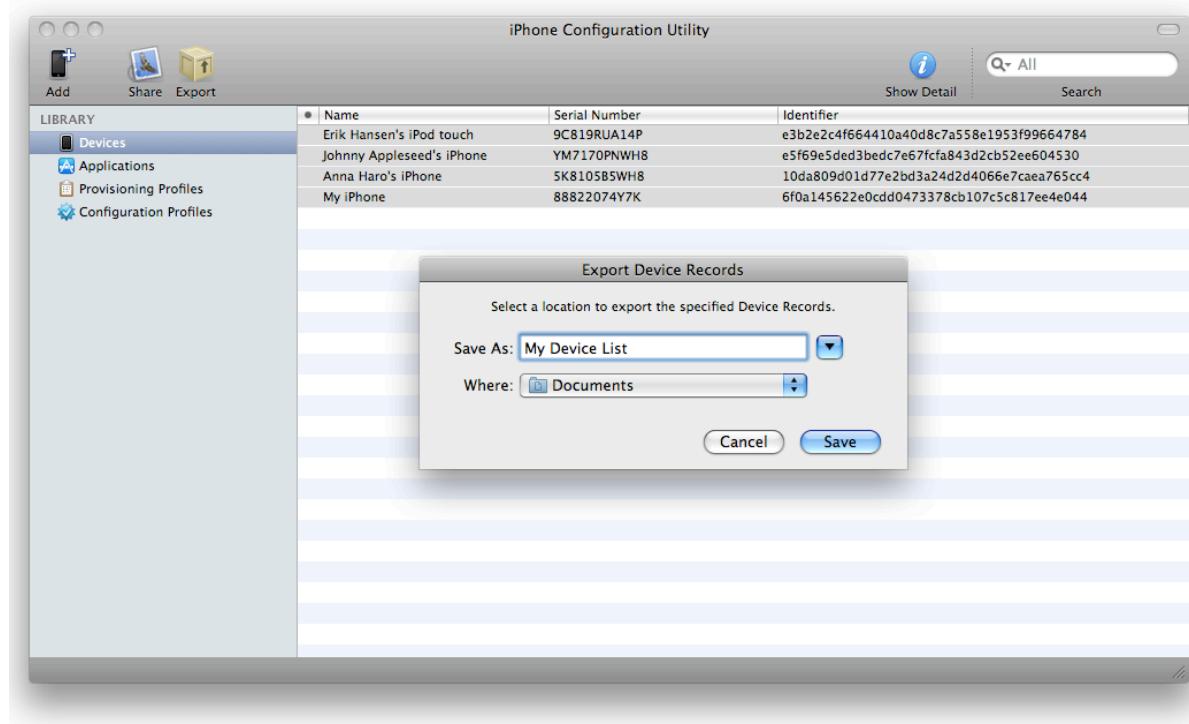
Adding Individual Devices

Upon receiving a UDID, Team Admins should navigate to the ‘Devices’ section of the Program Portal and click ‘Add Device’. Admins will enter a Device Name as well as the UDID and click ‘Submit’.

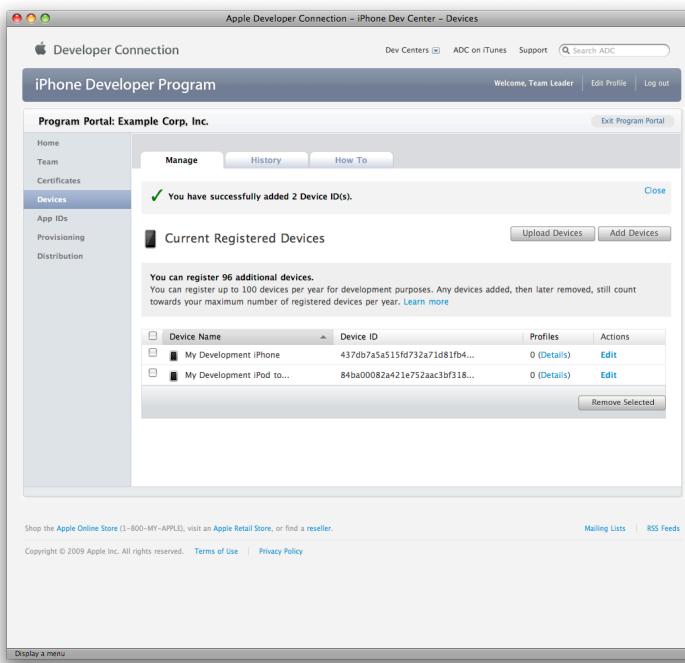
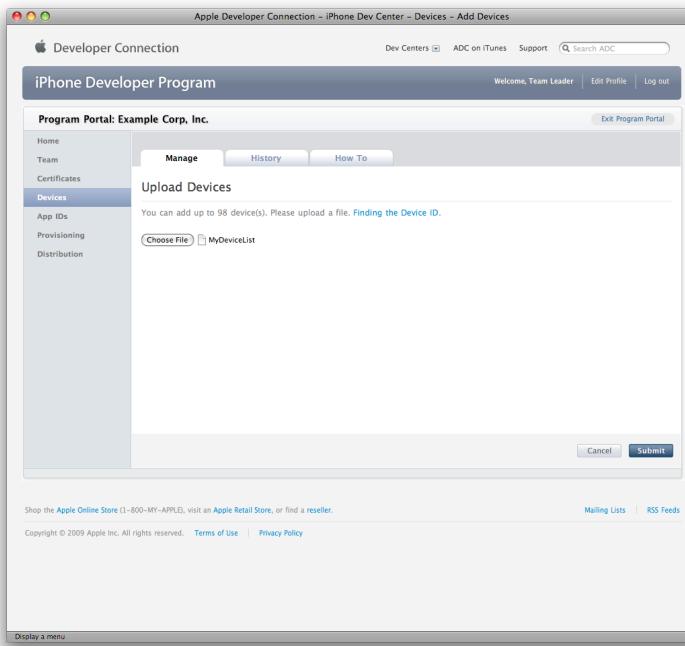


Bulk Upload of Devices

Alternatively, Team Admins can gather multiple UDIDs in one single export once by creating a .deviceids file generated by the [iPhone Configuration Utility](#) or uploading a tab delimited, .txt file. To do this, connect each device you wish to register to your Mac. iPhone Configuration Utility will save the information for each device in the 'Library' section under 'Devices'. Within the iPhone Configuration Utility, select the devices you wish to upload and click the Export button. This will create a .deviceids file that can be uploaded to the Program Portal.



Upon generating a deviceid file, Team Admins should navigate to the 'Devices' section of the Program Portal and click 'Upload Device'. Select your deviceid file and click 'Submit'.



If you wish to upload a tab-delimited .txt file containing your devices, follow the above steps but choose your .txt file instead of a .deviceid file. Your file should contain one Device ID and the Device Name in each row. (Note: The first row in your .txt file will be ignored as it should contain headers only).

Removing Devices from your Development Team

To remove a device from your development team, navigate to the 'Devices' area of the Program Portal and click the 'Remove' link to the right of the desired device. Removing a device from your development team will invalidate all provisioning profiles containing that device. Affected provisioning profiles will need to be modified in order to be downloaded again. Removing devices does not restore the device slot. Therefore, if you add a device, your device count will be reduced from 100 available devices to 99 available devices. Removing that device does not increase the available count to 100 again.

Resetting your Device List

At the start of a new membership year, Team Agents and Team Admins can remove devices and restore the available device count for their development team to 100 devices. When Team Agents or Admins first log into the Program Portal at the start of a new membership year, they will be presented with the option to remove devices and restore the device count for those removed devices. Devices can be removed and their device count restored until the first new device is added. Therefore, it is important to remove all devices you are no longer using for development prior to adding any new devices.

Installing Pre-release versions of iOS

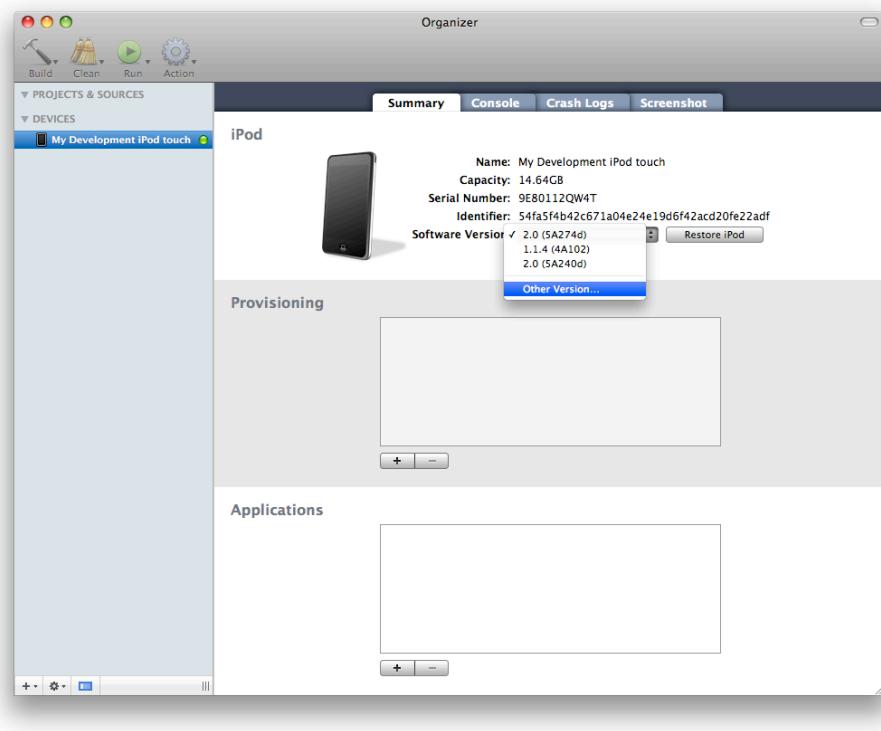
To run your code on an Apple device, you will need to install iOS onto each development device and "restore" each device to a development state. Note: Once a pre-release version of iOS is installed on the device you cannot restore the device to an earlier version of iOS. The device may only be used for development and testing purposes until that version of iOS is publicly released. Please DO NOT install the iOS before registering device UDIDs, as installation on non-registered devices will render them inoperable. **The public release version of iOS should be installed using iTunes.**

To download pre-release versions of iOS (if available):

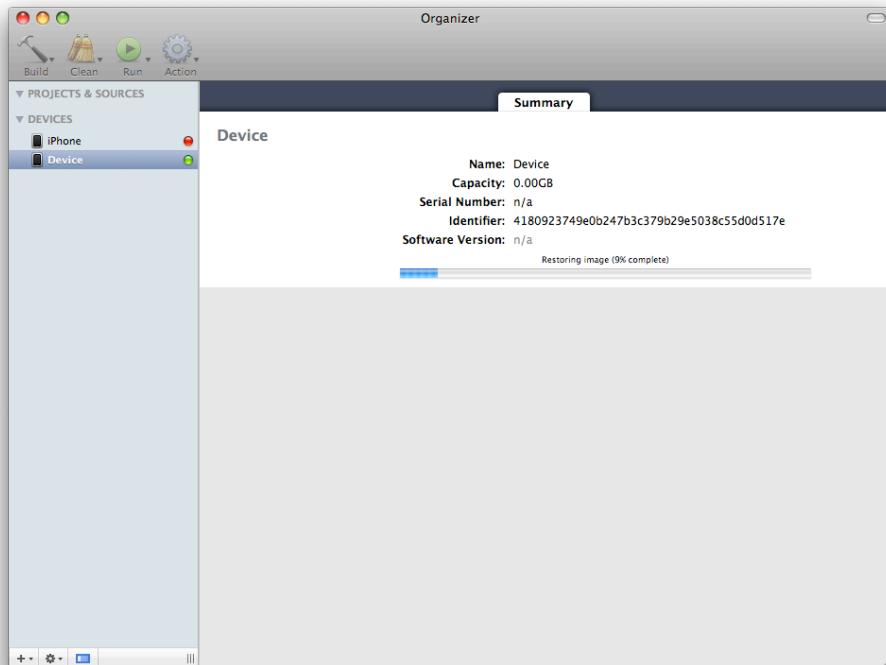
1. Download the iOS Disk Image (.dmg) from the iOS Dev Center for the Apple device you are using.
2. Connect your device to your Mac, close iTunes and launch Xcode.
3. Once the device is detected by Xcode, select 'Use for Development' when prompted.



4. In Xcode, Navigate to 'Window'->'Organizer'. There you should see your device and current software version.
5. In the 'Software Version' drop-down menu, select 'Other Version' and navigate in Finder to the iOS .ipsw file downloaded from the iOS Dev Center. Click 'Restore iPod'.



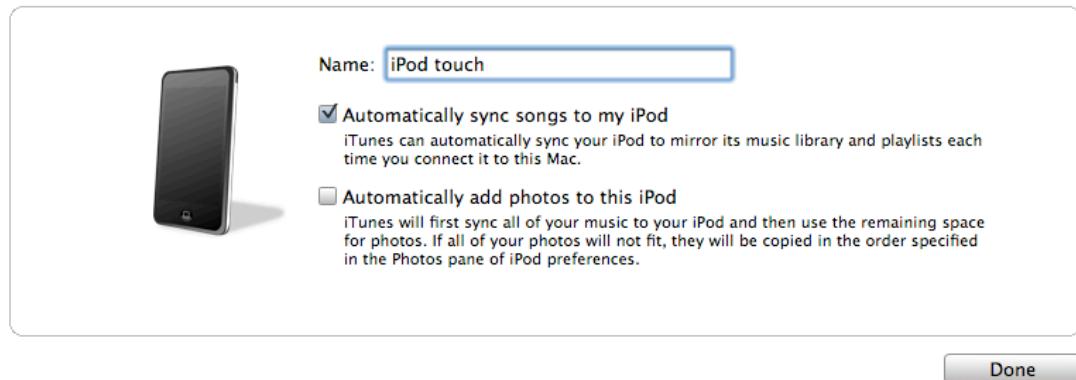
6. In Xcode, the restore proceeds. This process should NOT be interrupted; it will take approximately 5 minutes. Close any prompts that iTunes may present during the restore process.



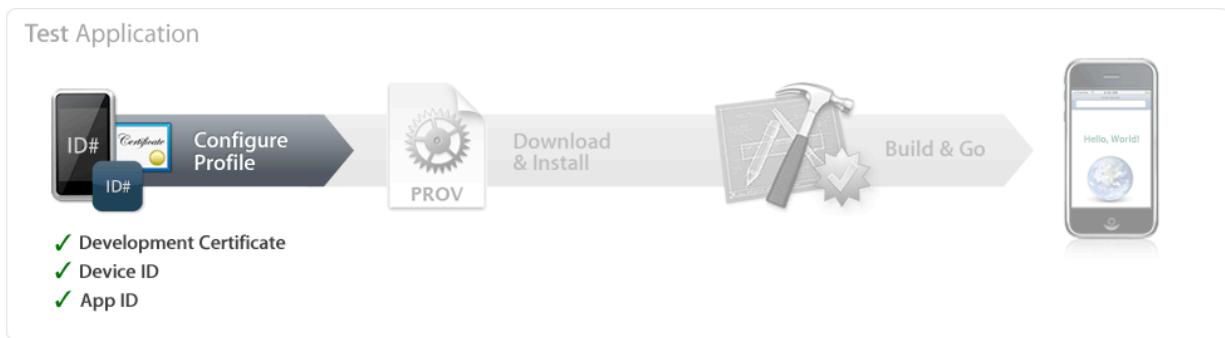
7. After the device reboots, Xcode again displays the 'New Device Detected' dialog. Click 'Use for Development'.

8. To complete the iOS installation, launch iTunes and set up the device.

Set Up Your iPod



Creating your App ID



An App ID is a unique identifier that designates your application and is part of your Provisioning Profile. App IDs enable users to take advantage of iOS features such as Apple Push Notification, In App Purchase, Game Center and Keychain Data sharing between applications.

Each App ID consists of a universally unique 10 character "Bundle Seed ID" prefix generated by Apple and a "Bundle Identifier" suffix that is entered by a Team Agent or Team Admin in the Program Portal.

Bundle Seed ID <i>(generated by Apple)</i>	Bundle Identifier <i>(determined by Developer)</i>
A1B2C3D4E5	. com.domain.app

The Bundle Seed ID prefix is the portion of your App ID that allows you to share keychain data between various applications. Any App IDs that have the same Bundle Seed ID prefix can be used for sharing keychain data (such as usernames and passwords) between applications.

The Bundle Identifier portion of your App ID can contain the number 0-9, letters A-Z, a period (.) and/or a hyphen (-). The recommended practice is to use a reverse-domain name style string for the "Bundle Identifier" portion of the App ID. The string you enter in the Identifier portion of your App ID must be exactly the same as the string you enter as your CF Bundle Identifier in your Xcode project. A specific Bundle Identifier can only be entered once. If your chosen Bundle Identifier is unavailable, you will be prompted to input a different string.

There are two types of App IDs that can be created. The first type are called Wildcard App IDs which allow you to use the same App ID to build and install all of your applications. The second type of App ID is called an Explicit App ID. This type of App ID restricts a provisioning profile to only allow the installation of one application. Explicit App IDs are required to enable iOS features such as In App Purchase and Apple Push Notification service in your applications.

Wildcard App IDs

Wildcard App IDs are helpful when getting started with iOS development because a single Wildcard App ID can be used to build and install all of your applications, including Sample Code provided in the iOS Reference Library. The Provisioning Assistant in the Program Portal generates a Wildcard App ID for you. Wildcard App IDs are created by entering an asterisk in the Identifier portion of your App ID. The wildcard asterisk character must be the last character in the App ID string. Example App IDs for this situation could be:

Bundle Seed ID <i>(generated by Apple)</i> A1B2C3D4E5	Bundle Identifier <i>(determined by Developer)</i> * or	Bundle Seed ID <i>(generated by Apple)</i> A1B2C3D4E5	Bundle Identifier <i>(determined by Developer)</i> com.domain.*
--	--	--	---

In these wildcard situations, you'll simply replace the asterisk with whatever string you wish in the CF Bundle Identifier field in Xcode. In the first example above, the CF Bundle Identifier in Xcode can be any string you wish. In the second example above, the CF Bundle Identifier in Xcode must start with 'com.domain.' and the asterisk can be replaced with a string of your choosing.

Explicit App IDs

When are you ready to start incorporating iOS features into your application, such as push notification, In App Purchase and Game Center, you will need to create explicit App IDs for your application. Explicit App IDs can only be used to build and install a single application so a new Explicit App ID and separate Provisioning Profile will be required for each application you are developing.

Explicit App IDs are created by entering a specific string in the Identifier portion of your App ID. The Bundle Identifier portion of your App ID can contain the number 0-9, letters A-Z, a period (.) and/or a hyphen (-). The recommended practice is to use a reverse-domain name style string for the Bundle Identifier portion of the App ID. An example of an Explicit App ID is:

Bundle Seed ID <i>(generated by Apple)</i>	Bundle Identifier <i>(determined by Developer)</i>
A1B2C3D4E5	. com.domain.app

Remember that whatever string you put into the Identifier field of an explicit App ID must be entered as your CF Bunde Identifier in your Xcode project Info.plist file.

Generating an App ID

1. Team Agents or Team Admins should navigate to the 'App ID' section of the Program Portal.
2. Click 'New App ID'.
3. Enter a Description for your App ID. This is a name for easy reference and identification within the Program Portal.
4. In the Bundle Seed ID drop down box, you can either generate a new Bundle Seed ID or select a previously created one. If you want your application to share data with another application, make sure you select the same Bundle Seed ID that you used when you created the App ID for your other application.
5. Enter a Bundle Identifier in the free-form text field. This can be either the wildcard asterisk character, an explicit string or a combination of both. This Bundle Identifier will need to match whatever CF Bundle Identifier you use for your application in Xcode. Note: The Bundle Seed ID does not need to be entered into Xcode. If you don't plan on utilizing iOS 3.0 features such as In App Purchase or Apple Push Notification service, you may use a wildcard character in the Bundle Identifier (e.g. *com.domainname.** or ***).
6. Click 'Submit'. At this time, the 10 character Bundle Seed ID is generated (unless you have chosen a previously generated Bundle Seed ID) and concatenated with the Bundle Identifier you entered. This resulting string is your App ID.
7. Generate a new App ID for each application using In App Purchase or Apple Push Notification. If you don't plan on utilizing iOS 3.0 features such as In App Purchase or Apple Push Notification service, you may use a wildcard character in the Bundle Identifier (e.g. *com.domainname.** or ***) and replace the asterisk in Xcode with whatever CF Bundle Identifier you choose. The wildcard App ID can be used to build all of your applications.

Example of Wildcard App ID Creation

The screenshot shows the "Add App IDs" page in the iPhone Dev Center. The left sidebar has "App IDs" selected. The main area is titled "Create App ID" and contains fields for "Description" (containing "My Wildcard App ID"), "Bundle Seed ID (App ID Prefix)" (with a "Generate New" button), and "Bundle Identifier (App ID Suffix)" (containing "*"). At the bottom right are "Cancel" and "Submit" buttons.

Apple Developer Connection – iPhone Dev Center – Add App IDs

Developer Connection

iPhone Developer Program

Welcome, Team Leader | Edit Profile | Log out

Program Portal: Example Corp, Inc.

Home Team Certificates Devices **App IDs** Provisioning Distribution

Manage How To

Create App ID

Description

Enter a common name or description of your App ID using alphanumeric characters. The description you specify will be used throughout the Program Portal to identify this App ID.

My Wildcard App ID You cannot use special characters as @, &, *, * in your description.

Bundle Seed ID (App ID Prefix)

Generate a new or select an existing Bundle Seed ID for your App ID.

Generate New If you are creating a suite of applications that will share the same Keychain access, use the same bundle Seed ID for each of your application's App IDs.

Bundle Identifier (App ID Suffix)

Enter a unique identifier for your App ID. The recommended practice is to use a reverse-domain name style string for the Bundle Identifier portion of the App ID.

*

Example: com.domainname.appname

Cancel Submit

Shop the [Apple Online Store](#) (1-800-MY-APPLE), visit an [Apple Retail Store](#), or find a [reseller](#). Mailing Lists | RSS Feeds

Copyright © 2009 Apple Inc. All rights reserved. [Terms of Use](#) | [Privacy Policy](#)

Example of Explicit App ID Creation

The screenshot shows the "Add App IDs" section of the iPhone Dev Center. The left sidebar has "App IDs" selected. The main area is titled "Create App ID" and contains three input fields: "Description" (containing "My First Application"), "Bundle Seed ID (App ID Prefix)" (containing "Generate New"), and "Bundle Identifier (App ID Suffix)" (containing "com.domainname.appname"). Below these fields are "Cancel" and "Submit" buttons. At the bottom, there are links for the Apple Online Store, Mailing Lists, RSS Feeds, Terms of Use, and Privacy Policy.

Example of App ID Creation that can share keychain data from another App ID

The screenshot shows the "Add App IDs" page in the iPhone Developer Program. The left sidebar is titled "Program Portal: Example Corp, Inc." and includes links for Home, Team, Certificates, Devices, App IDs (which is selected), Provisioning, and Distribution. The main content area has tabs for "Manage" and "How To". A sub-section titled "Create App ID" is displayed. It contains fields for "Description" (My Second Data Sharing Application), "Bundle Seed ID (App ID Prefix)" (VLT2AAENAP), and "Bundle Identifier (App ID Suffix)" (com.domainname.secondappname). Below these fields are "Cancel" and "Submit" buttons. At the bottom of the page, there are links for the Apple Online Store, Apple Retail Store, and reseller, along with Mailing Lists and RSS Feeds.

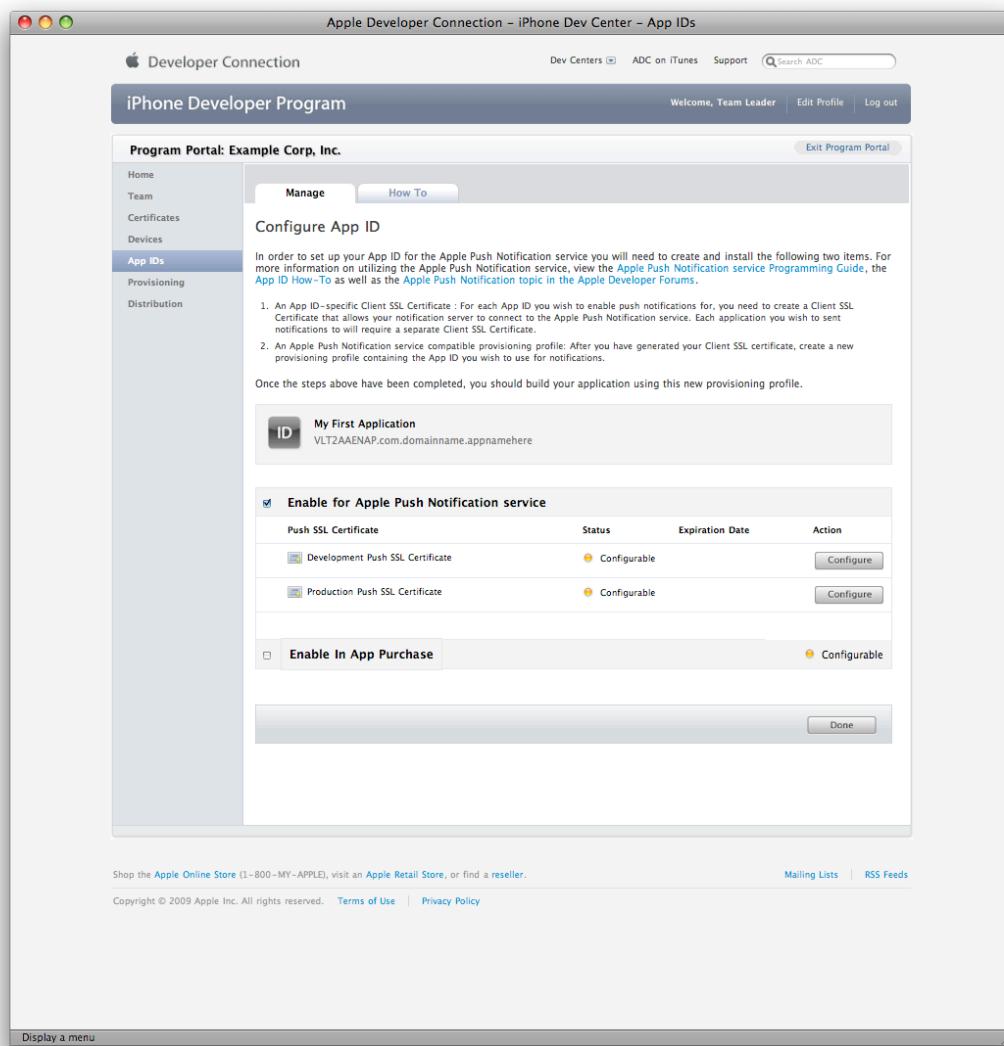
Configuring your App ID for Apple Push Notification service

In order to utilize the Apple Push Notification service, you need to enable your App ID for use with the service and then create a new Provisioning Profile containing your newly enabled App ID. Enabling an App ID for APNs allows you to create and download Client Authentication SSL Certificates for your application that you will use to establish connectivity between your notification server and the Apple Push Notification service. Because the Client SSL Certificates are issued to an application, only explicit App IDs can be registered to use the service. This allows only the developer who owns the application to send notifications to it. After you have enabled an App ID to use APNs, you will be able to create a Provisioning Profile that contains an entitlement in it allowing your application to connect to the Apple Push Notification service. To enable an App ID to use APNs:

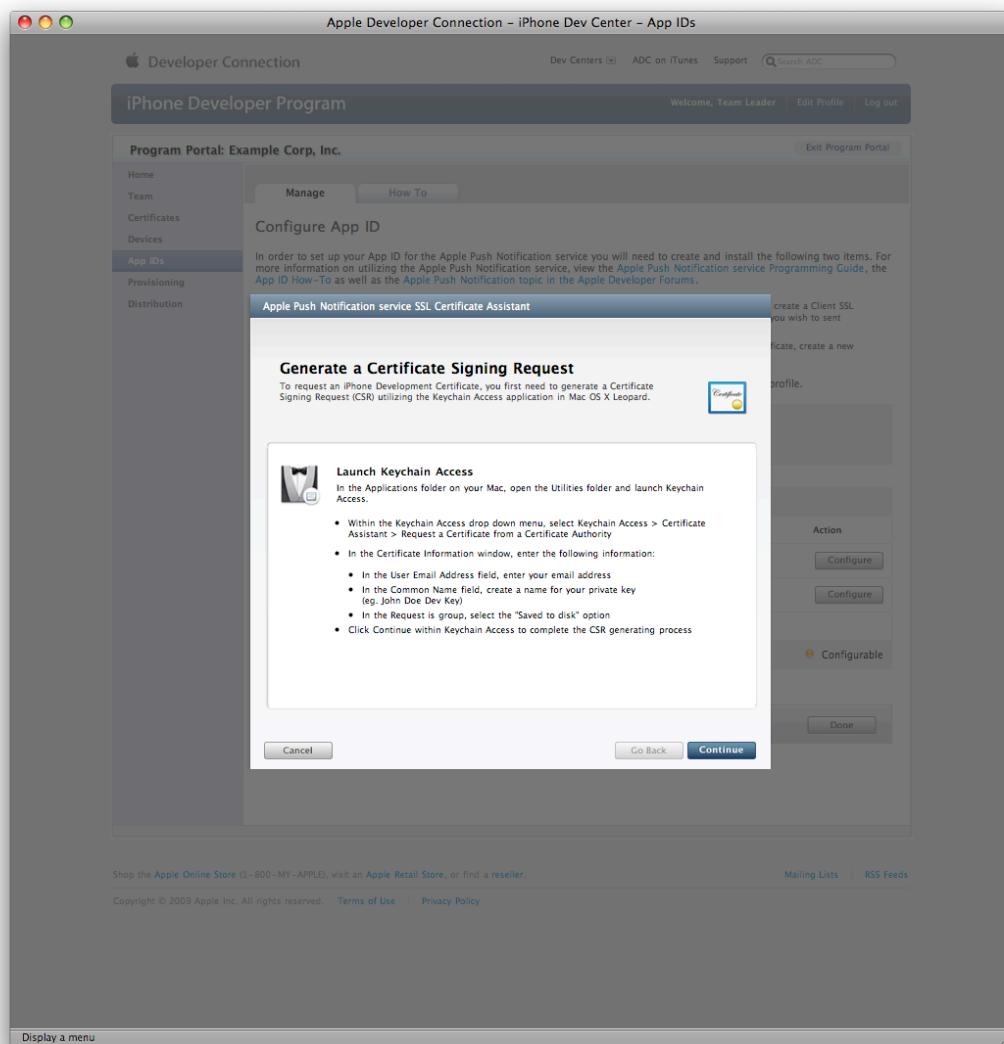
1. Navigate to the App ID section of the Program Portal. Any App IDs you have that are eligible for APNs will have a yellow indicator light indicating it is Configurable for the Development (or Sandbox) Push Notification service and the Production Apple Push Notification service. Wildcard and duplicate App IDs will be Unavailable for APNs. Click on the 'Configure' button in the Action column next to the App ID which you wish to enable for push notifications.

Description	Apple Push Notification service	In App Purchase	Action
VLT2AAENAP.* My Wildcard App ID	<input checked="" type="radio"/> Unavailable	<input checked="" type="radio"/> Unavailable	Details
VLT2AAENAP.com.domainname.appnamehe... My First Application	<input checked="" type="radio"/> Configurable for Development <input checked="" type="radio"/> Configurable for Production	<input checked="" type="radio"/> Configurable	Configure

2. In the Configure App ID screen, click on the checkbox next to "Enable for Apple Push Notification service" to enable the App ID. After the App ID is enabled, you can click on the Configure button next to the Development or Production SSL Certificate to create the Client Authentication SSL Certificate for the environment you wish to connect your notification server to.



3. Clicking the Configure Button will launch the Apple Push Notification service SSL Certificate Assistant which will guide you through the steps to create your SSL Certificate. The private key and public SSL certificate that is generated should both be installed on your notification server.



4. After you have enabled an App ID for APNs, make sure to generate a new provisioning profile containing the newly enabled App ID. Your new provisioning profile will now have the necessary entitlement to receive notifications in your application.

Configuring your App ID for In App Purchase and Game Center

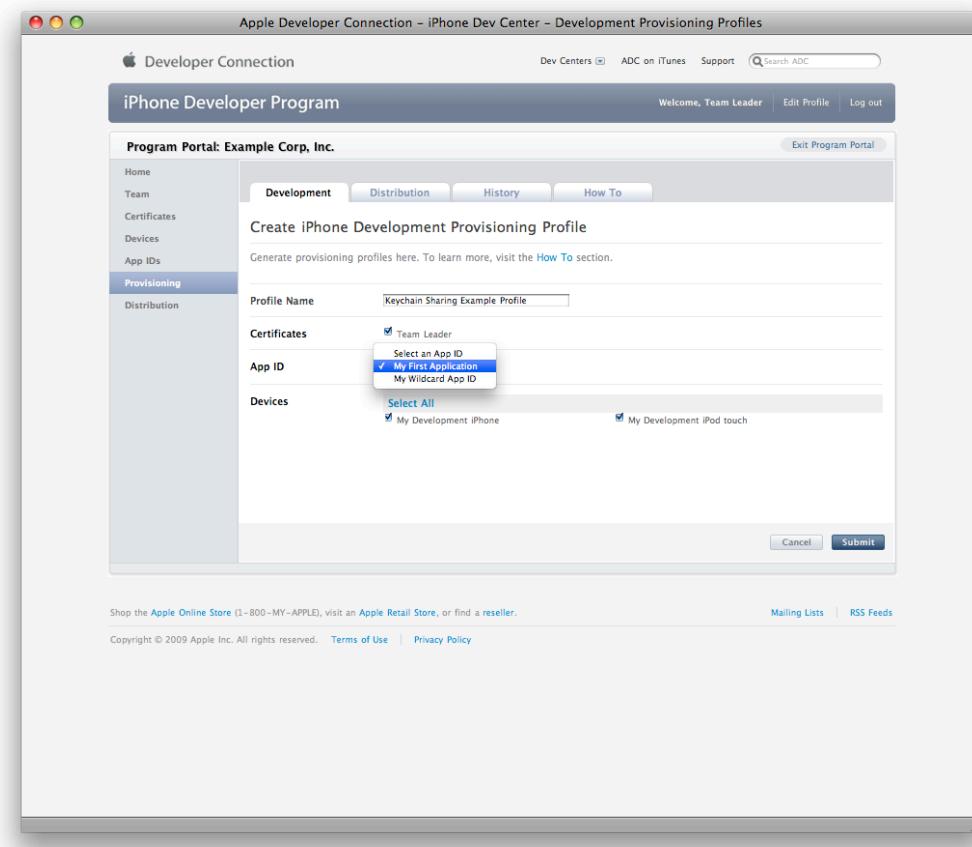
In app purchases are offerings that your customers can purchase from within your application. If you wish to offer in app purchases for your applications, you will need to have a Paid Applications contract in effect with Apple. All in app purchases will need to be created and managed within [iTunes Connect](#) as well as tested in a sandbox environment to ensure functionality before being sent to Apple for review.

In order to test In App Purchases and Game Center in a sandbox environment, you must register an Explicit App ID for your application in the Program Portal and then set up test accounts and In App Purchases within [iTunes Connect](#). Registering an Explicit App ID for In App Purchase and/or Game Center registers your App ID (and CF Bundle Identifier) in iTunes Connect. This prevents other developers from submitting applications to iTunes Connect with the CF Bundle Identifier you are using for development work. Because In App Purchases are tied to specific applications and CF Bundle Identifiers, only explicit App IDs can be registered to use the service. This allows only the developer who owns the application to set up In App Purchases for it. Explicit App IDs are automatically registered for In App Purchase. After you have registered your Explicit App ID for In App Purchase, review the In App Purchases section of the [iTunes Connect User Guide](#) for instructions on how to set up the required Test User Accounts and actual In App Purchases.

Enabling Keychain Data Sharing with your App ID

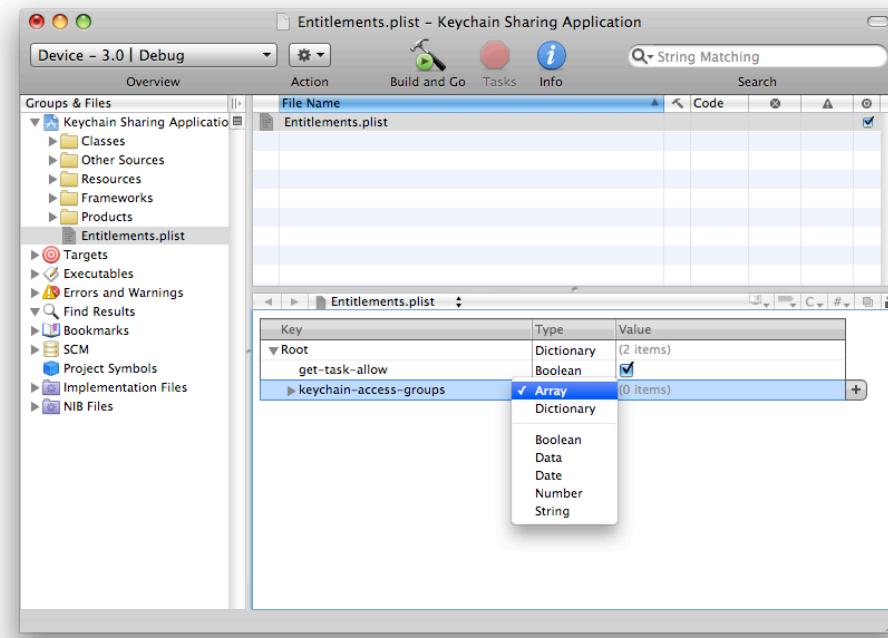
Applications that are built with a provisioning profile containing the same Bundle Seed ID portion of the App ID can share data via the system keychain.

1. Create a provisioning profile or profiles that contain an App ID with the same Bundle Seed ID prefix.

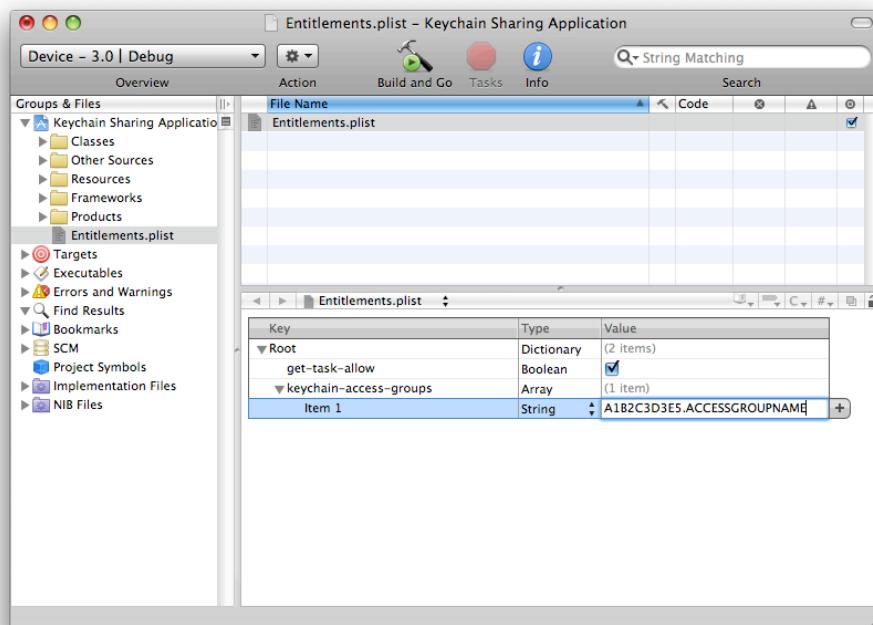


2. In your Xcode project, create a new Entitlements.plist file (as outlined in Step 8 of the Building your Application for Distribution section of this guide).

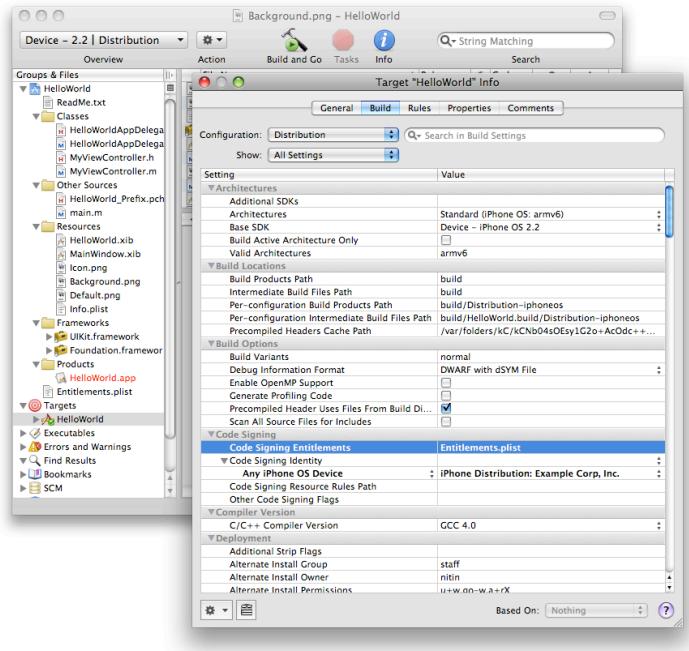
3. In your Entitlements.plist file, add a new entitlement called 'keychain-access-groups' and set it's value to 'Array'.



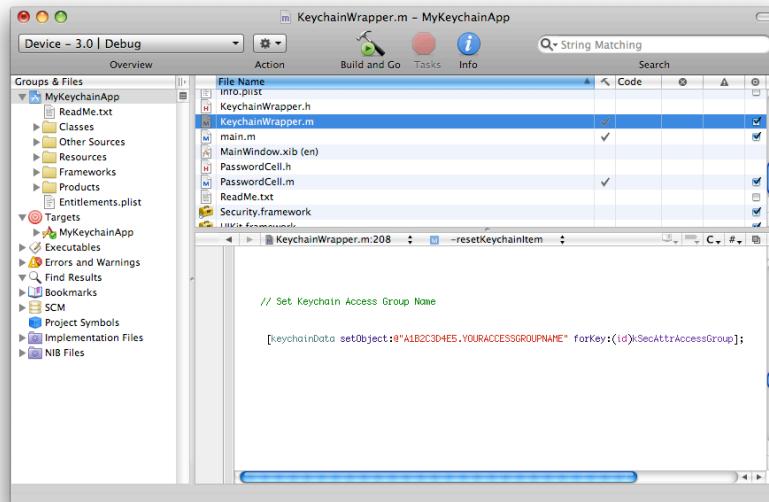
4. Add a new item in the array and enter your Bundle Seed ID prefix followed by a period and a name for your access group. The name for your access group can be any string you wish but should be the same for all projects you wish to share keychain data with.



5. Next, add your Entitlements.plist file to the Code Signing Entitlements field of your Target Info window.



6. Now that you have configured your access group, you can define it in your code using the `kSecAttrAccessGroup` symbol. Your keychain access group should be added to all of your projects you wish to share data between.



7. View the [Keychain Services Programming Guide](#) and [Keychain Services Reference](#) for more information on how to share keychain data between your applications.

Creating and Downloading Development Provisioning Profiles



A provisioning profile is a collection of digital entities that uniquely ties developers and devices to an authorized iPhone Development Team and enables a device to be used for testing. A Development Provisioning Profile must be installed on each device on which you wish to run your application code. Each Development Provisioning Profile will contain a set of iOS Development Certificates, Unique Device Identifiers and an App ID and is valid for 90 days.

Devices specified within the provisioning profile can be used for testing only by those individuals whose iOS Development Certificates are included in the profile. A single device can contain multiple provisioning profiles.

Creating a Development Provisioning Profile

1. In the 'Provisioning' section of the Portal, Team Admins should click 'Add' on the Development tab.
2. Enter a name for the provisioning profile.
3. Specify which devices will be associated with the provisioning profile. You must specify a device in order for that device to utilize the provisioning profile. If a device's UDID is not included in the provisioning profile the profile and your application cannot be installed on that device.
4. Specify which iOS Development Certificates will be associated with the provisioning profile. You must specify an iOS Development Certificate in order for the application code signed with that same certificate to run on the device.
5. Specify a single App ID for the Development Provisioning Profile. Each Development Provisioning Profile can specify only ONE App ID, therefore, if you have multiple applications using Apple Push Notification service or In App Purchase , you will need to create a separate Development Provisioning Profile for each of those applications. If you are installing several applications that are not utilizing iOS features such as Apple Push Notification service, Game Center or In App Purchase, you can use an App ID containing the wildcard asterisk character to build all of your applications.
6. Click 'Submit' to generate your Development Provisioning Profile.

Note: If you have recently enabled an App ID for Apple Push Notification service, make sure to create a new provisioning profile containing that App ID. Provisioning Profiles created before an App ID was enabled for APNs will not work for testing APNs.

Apple Developer Connection - iPhone Dev Center - Development Provisioning Profiles

Developer Connection

iPhone Developer Program

Welcome, Team Leader | Edit Profile | Log out | Exit Program Portal

Program Portal: Example Corp, Inc.

Home | Team | Certificates | Devices | App IDs | **Provisioning** | Distribution

Development | Distribution | History | How To

Create iPhone Development Provisioning Profile

Generate provisioning profiles here. To learn more, visit the [How To](#) section.

Profile Name: My First Provisioning Profile

Certificates: Team Leader

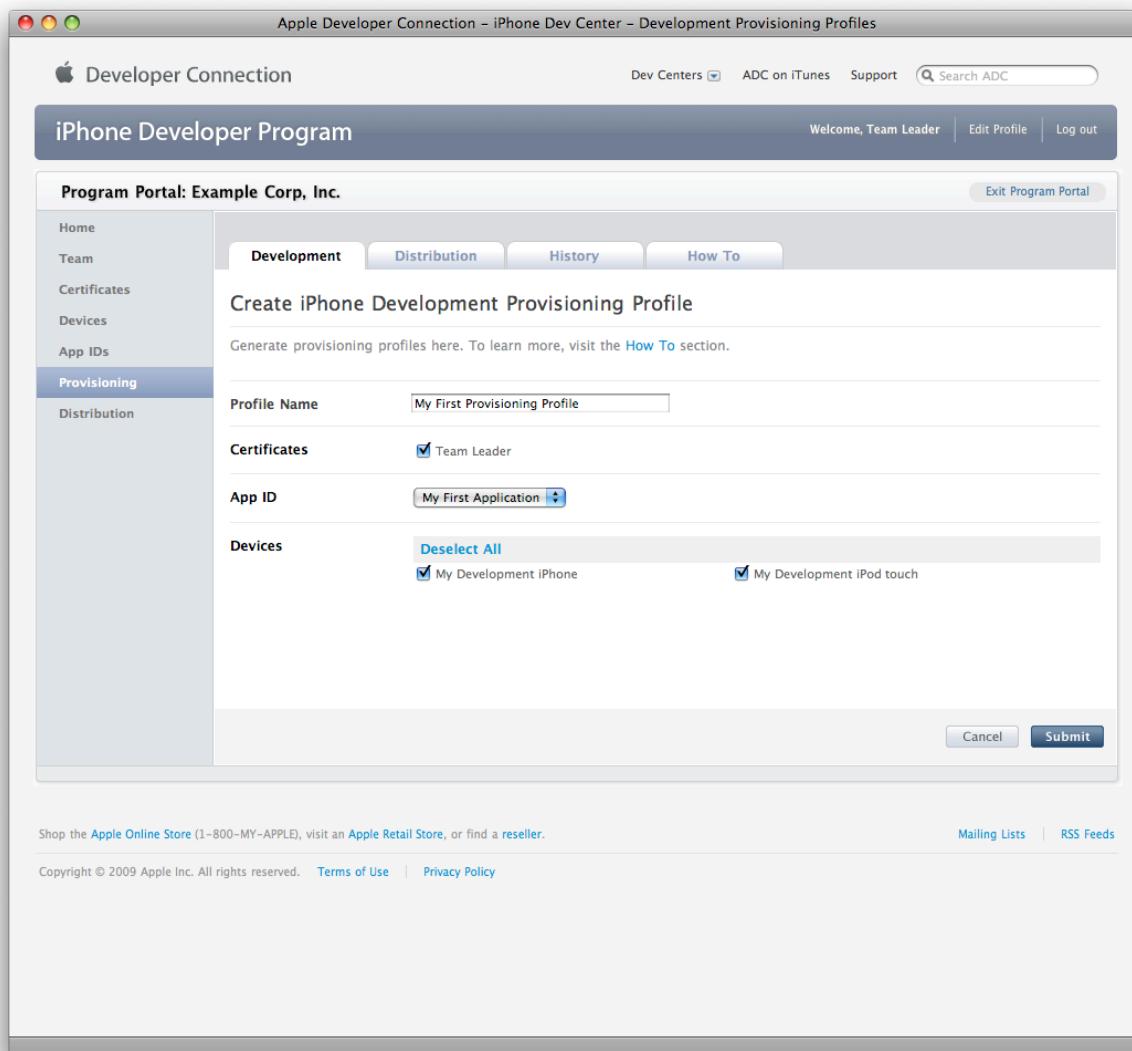
App ID: My First Application

Devices: Deselect All
 My Development iPhone My Development iPod touch

Cancel **Submit**

Shop the Apple Online Store (1-800-MY-APPLE), visit an [Apple Retail Store](#), or find a [reseller](#). [Mailing Lists](#) | [RSS Feeds](#)

Copyright © 2009 Apple Inc. All rights reserved. [Terms of Use](#) | [Privacy Policy](#)

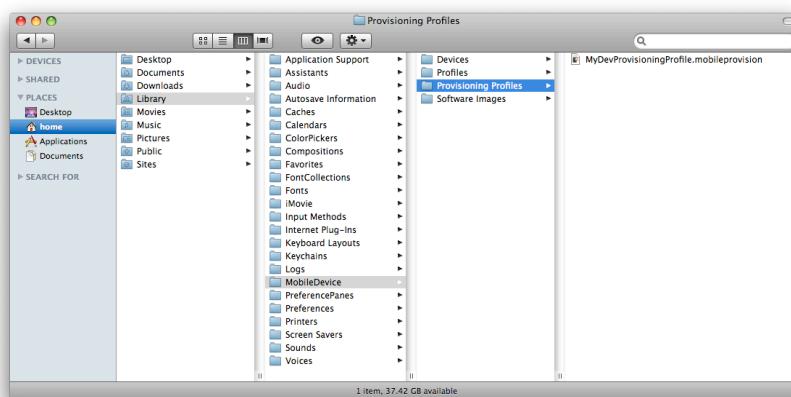
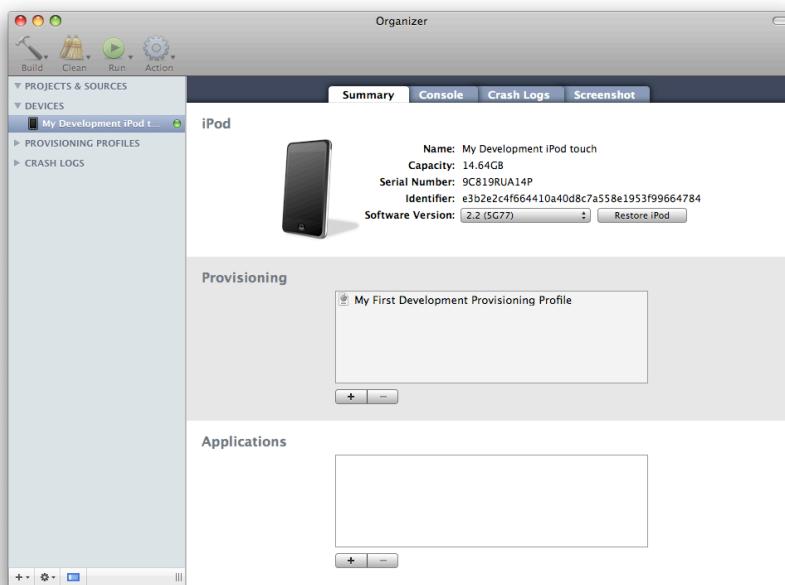


Installing a Development Provisioning Profile

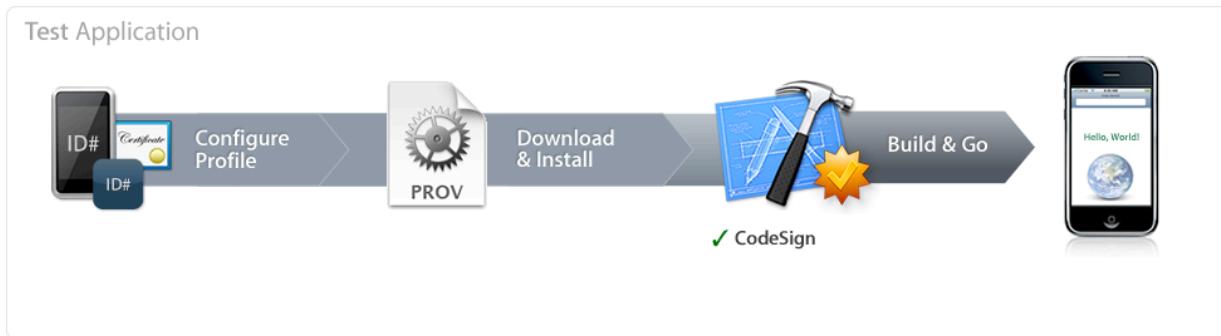
All Team Agents, Admins and Members can download a Development Provisioning Profile from the 'Provisioning' section of the Portal after it has been created. *Only those developers whose Apple device IDs and iOS Development Certificates are included in the provisioning profile will be able to install and test applications on their device.*

1. In the 'Provisioning' section of the Program Portal, click the download button next to the desired provisioning profile.
2. Drag the downloaded file onto the Xcode, iTunes or iPhone Configuration Utility application icon in the dock to install the provisioning profile on your Mac. Alternatively, you can copy the file to '~/Library/MobileDevice/Provisioning Profiles'. If the directory does not exist yet, you will need to create it.
3. Connect your device to your Mac and launch Xcode.
4. Navigate to the 'Organizer' window in Xcode and select your device. Click on the '+' sign in the Provisioning section of the Organizer window to navigate to your Provisioning Profile. Select your Provisioning Profile to install it on your device.

Note: It is important to verify that the Provisioning Profile is installed on both your device and your Mac to successfully build and install your applications.

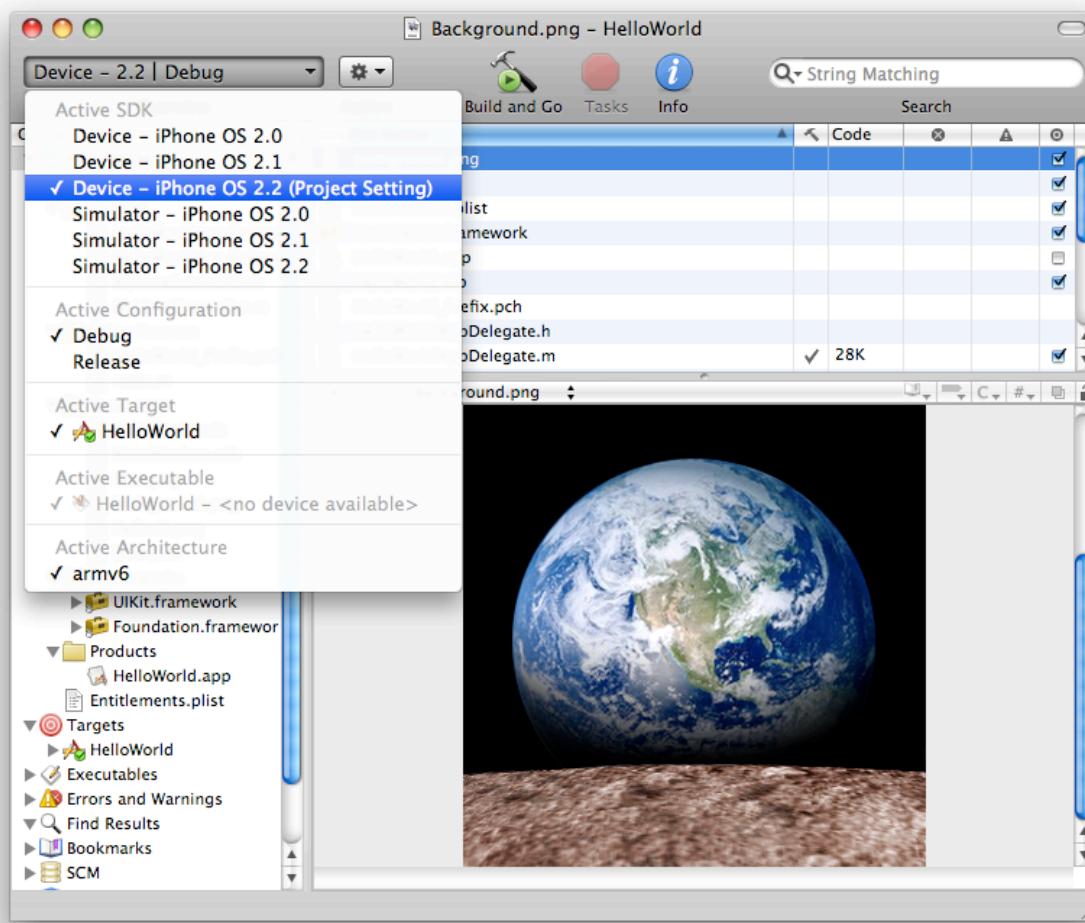


Building and Installing your Development Application

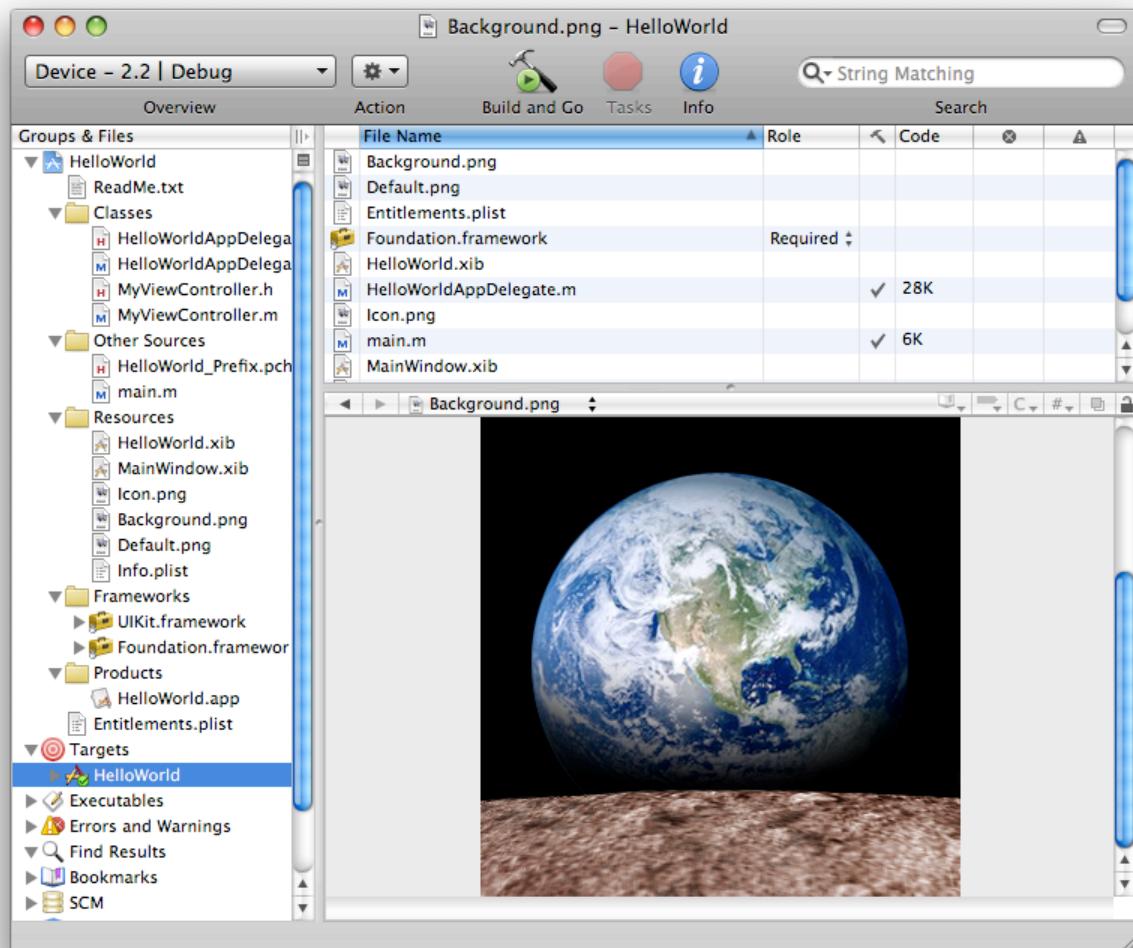


Now that you have an approved iOS Development Certificate, an assigned Apple device and a properly installed Development Provisioning Profile, Xcode can now build your application and install it on your development device. If you have a single iOS Development Certificate and iPhone Development Provisioning Profile, you don't need to change any settings in Xcode to start running your applications. To compile and install your code:

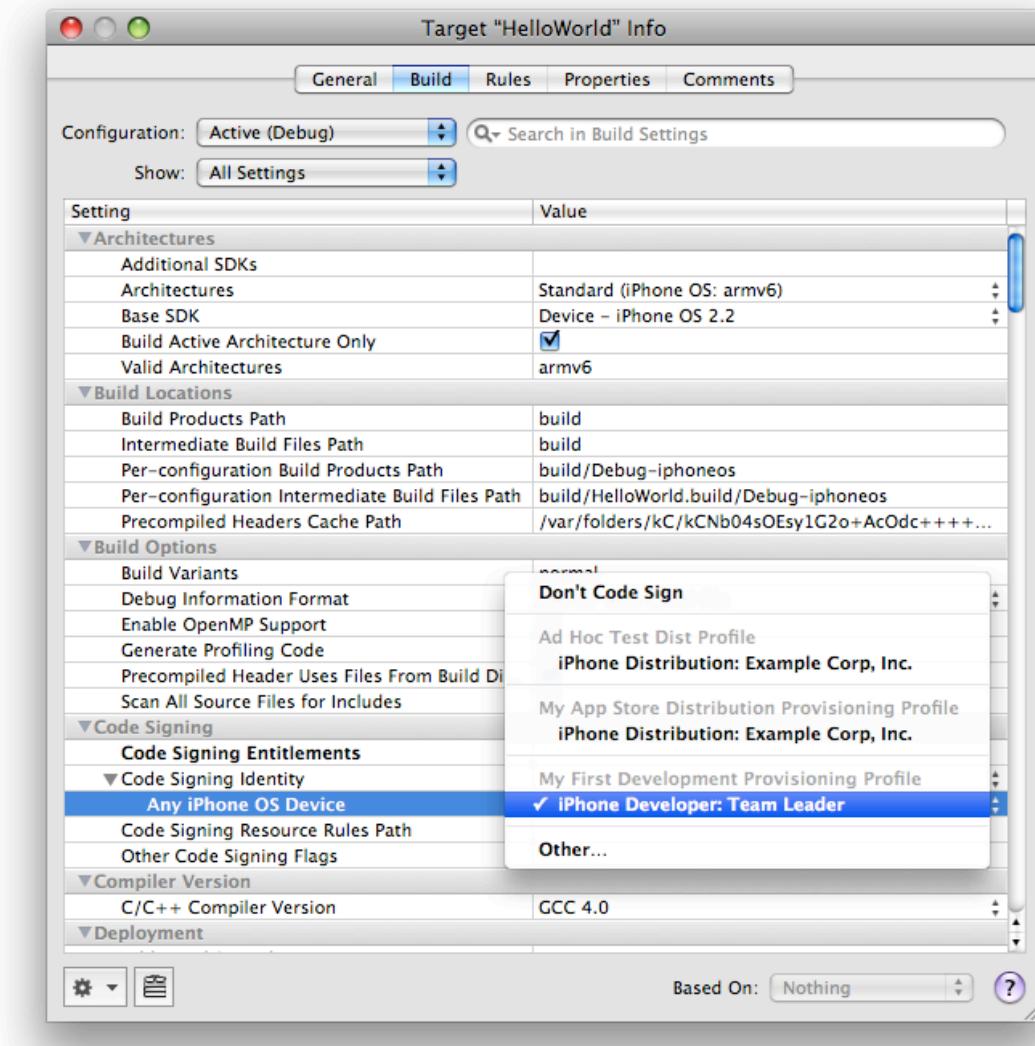
1. Launch Xcode and open your project.
2. In the Project Window, select 'Device - iOS' from the 'Device | Debug' drop down menu in the upper-left hand corner.



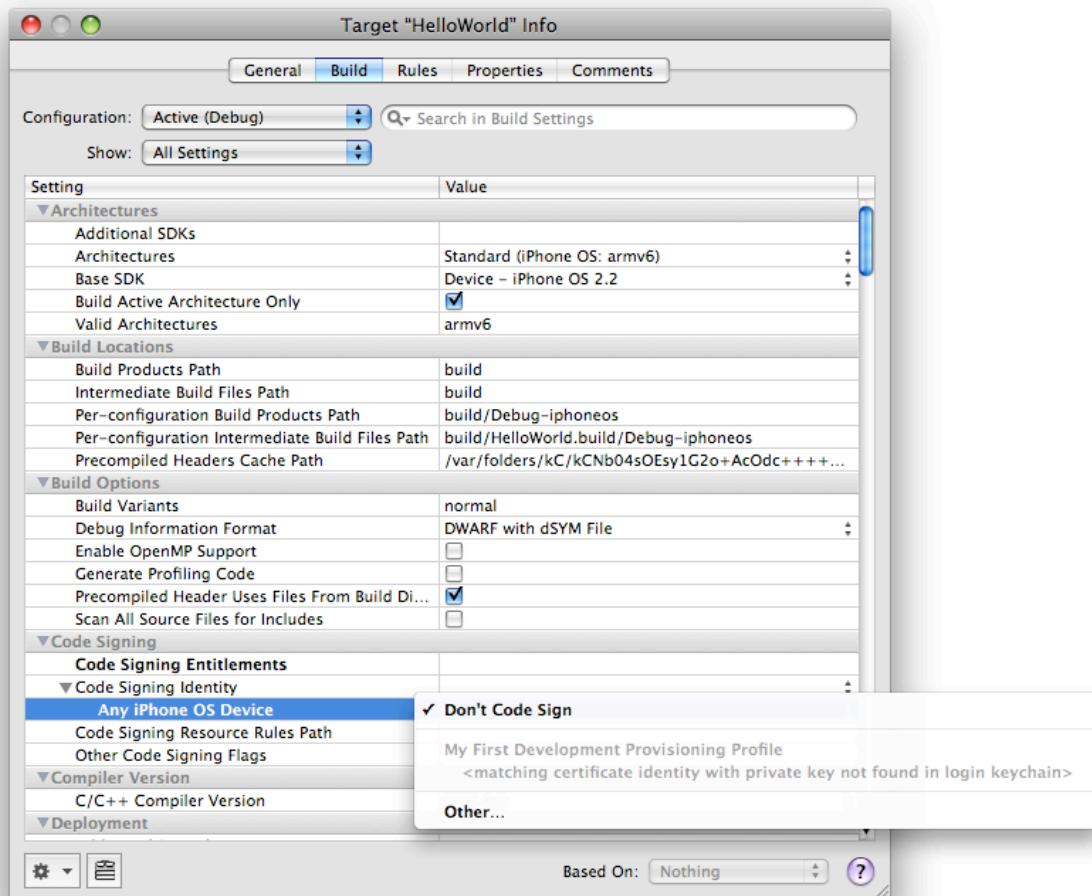
3. Highlight the project Target and select the 'Info' icon from the top menu bar.



4. 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 Development Certificate/Provisioning Profile pair you wish to sign and install your code with. Your iOS Development Certificate will be in bold with the Provisioning Profile associated with it in grey above. In the example below, 'iPhone Developer: Team Leader' is the Development Certificate and 'My First Development Provisioning Profile' is the .mobileprovision file paired with it.



Note: If the private key for your iOS Development Certificate is missing, or if your iOS Development Certificate is not included in a provisioning profile, you will be unable to select the iOS Development Certificate/Provisioning Profile pair and you will see the following. Re-installing the private key or downloading a provisioning profile with your iOS Development Certificate included in it will correct this.



5. Click 'Build and Go' to install the application on your Apple device.



Distribution

The distribution area of the iOS Developer Program Portal is where you will prepare and learn how to submit your iPhone and/or iPod touch application for delivery via the App Store or for Ad Hoc distribution. **Only Team Agents are authorized to prepare and submit applications for distribution.** The iOS Developer Program offers two ways for you to Distribute your application; either through the App Store, utilizing an iTunes Connect account created just for you, or through the Ad Hoc Distribution method.

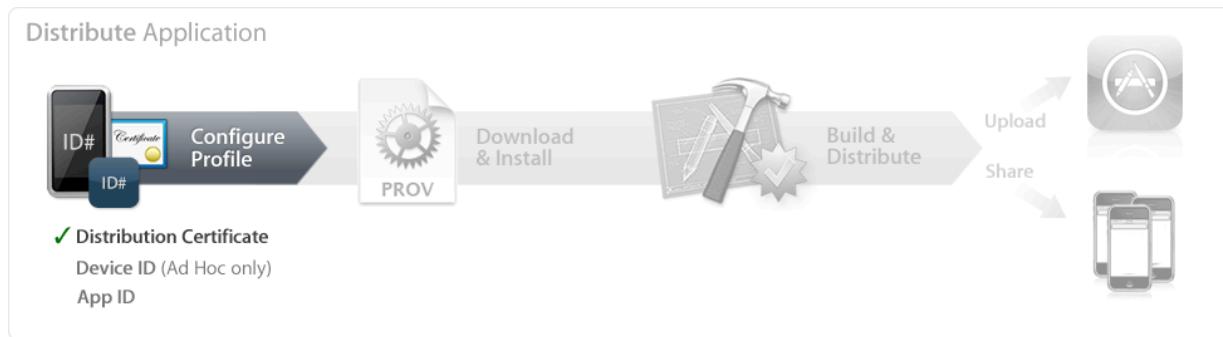
Whichever method you choose to distribute an application, Team Agents will need to complete the following:

- Create and Download an iOS Distribution Certificate
- Create and Download a Distribution Provisioning Profile
- Build your final application in Xcode

Preparing your Application for Distribution

Prior to distributing your application, there are several steps you will need to complete.

Obtaining your iOS Distribution Certificate

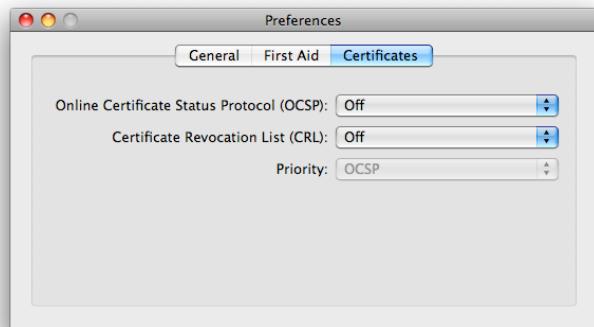


In order to distribute your iOS application, the Team Agent is required by Apple to create an iOS Distribution Certificate. Only the Team Agent for your team will be able to create this certificate and only this certificate will enable application submission.

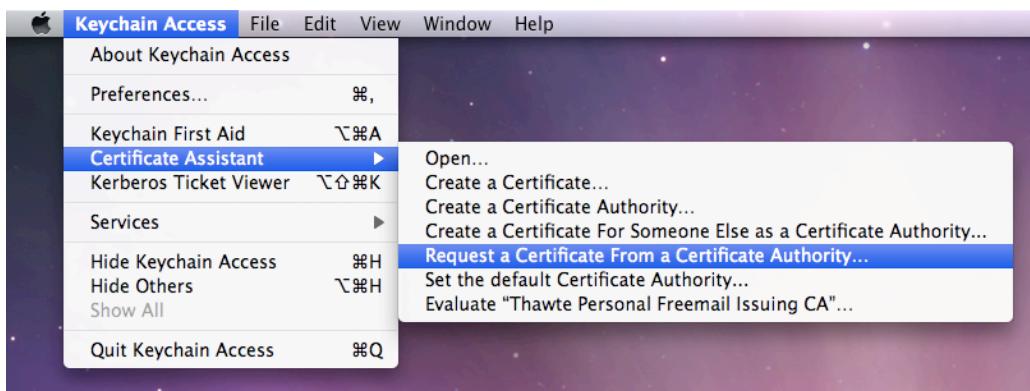
Generating a Certificate Signing Request

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 iPhone 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:

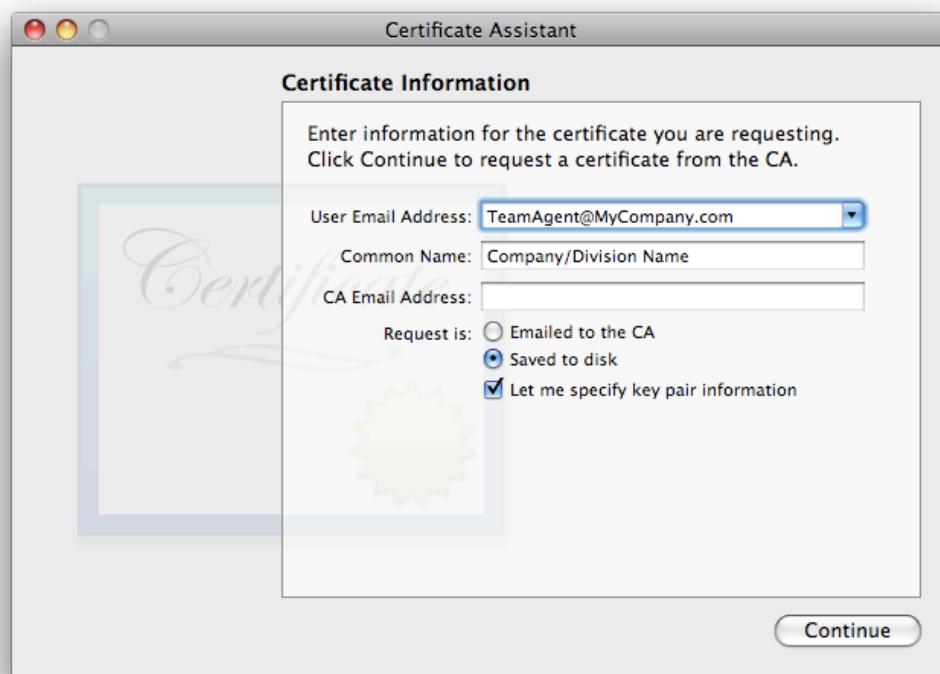
1. In your Applications folder, open the Utilities folder and launch Keychain Access.
2. In the Preferences menu, set Online Certificate Status Protocol (OSCP) and Certificate Revocation List (CRL) to "Off".



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 Program 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>..."



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 iPhone 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 iPhone 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'.



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'.



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 Developer Program Portal and navigate to 'Certificates' -> 'Distribution' and click the 'Request Certificate' button.

The screenshot shows the 'Distribution' tab selected in the top navigation bar. A message states: 'You currently do not have a valid distribution certificate' with a 'Request Certificate' button. The sidebar on the left lists 'Certificates' as the active category, along with other options like 'Devices', 'App IDs', 'Provisioning', and 'Distribution'.

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.

The screenshot shows the 'Create iPhone Distribution Certificate' page in the iPhone Developer Program. The left sidebar is titled 'Program Portal: Example Corp, Inc.' and includes links for Home, Team, Certificates (which is selected), Devices, App IDs, Provisioning, and Distribution. The main content area has tabs for Development, Distribution (which is selected), History, and How To. The 'How to create a Distribution certificate:' section contains three numbered steps:

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.

Below the steps is a file input field labeled '(Choose File)' with a placeholder 'CertificateSigningRequest'. At the bottom right is a blue 'Submit' button. The footer of the page includes links for the Apple Online Store, Apple Retail Store, Mailing Lists, RSS Feeds, Terms of Use, and Privacy Policy.

3. Approve your iOS Distribution Certificate.

The screenshot shows the "Program Portal: Example Corp, Inc." section of the iPhone Dev Center. The left sidebar has "Certificates" selected. The main content area is titled "Current Distribution Certificate". A message box says "The Certificate Request has been submitted for approval." Below it is a table with one row:

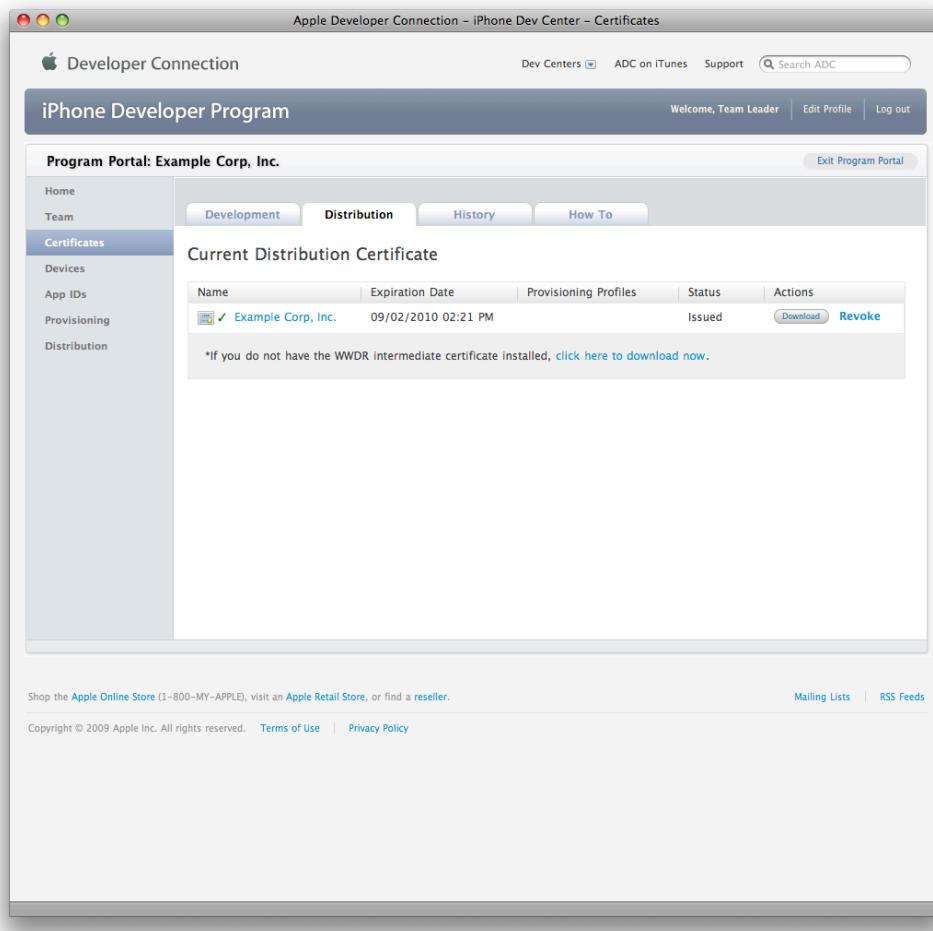
Name	Expiration Date	Provisioning Profiles	Status	Actions
Example Corp, Inc.			Pending Approval	Approve Reject

*If you do not have the WWDR intermediate certificate installed, [click here to download now](#).

At the bottom, there are links to the Apple Online Store, Apple Retail Store, reseller, Mailing Lists, and RSS Feeds.

Downloading and Installing iOS Distribution Certificates

1. In the Certificates -> Distribution tab of the Program Portal, click on the name of the iOS Distribution Certificate to download.
2. On your local machine, double-click the downloaded .cer file to launch Keychain Access and install your certificate.



Saving your Private Key and Transferring to Other Systems

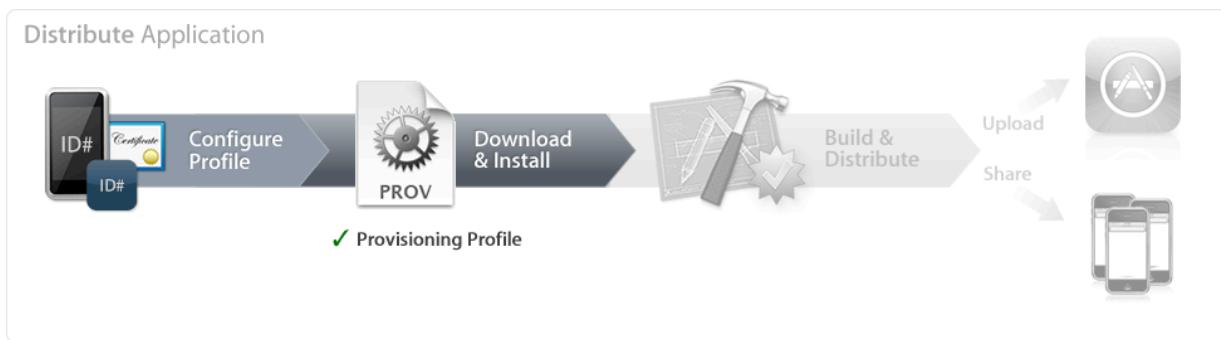
It is critical that you save your private key somewhere safe in the event that you need to build your application on multiple Macs or decide to reinstall your system OS. **Without your private key, you cannot sign binaries in Xcode and there you will be unable to upload your application to the App Store 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.

Managing Certificate Expiration

iOS Distribution Certificates are valid for one year from date of issue. After your iOS Distribution Certificate expires, you will no longer be able sign and install iOS applications on your devices. To continue distribution, you can revoke your iOS Distribution Certificate in the Apple Developer Member Center or iPhone Provisioning Portal. After your certificate has been revoked, you can create a new Certificate Signing Request and obtain a new iOS Distribution Certificate. **In the iPhone Developer Standard Program, revoking iOS Distribution Certificates will NOT affect applications already running on your device or applications that are already available on the App Store.** You can continue to submit application updates to the App Store with a new iOS Distribution Certificate. However, any applications you have distributed via Ad Hoc Distribution will no longer run after your iOS Distribution Certificate expires.

Creating and Downloading a 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. Therefore, you will not be able to install an App Store Provisioning Profile or Distribution build of your application on your 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 Program Portal and select the 'Distribution' tab and click 'Add Profile'.
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 the App ID for the application (or group of applications) you wish to distribute.
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 it on your Mac.

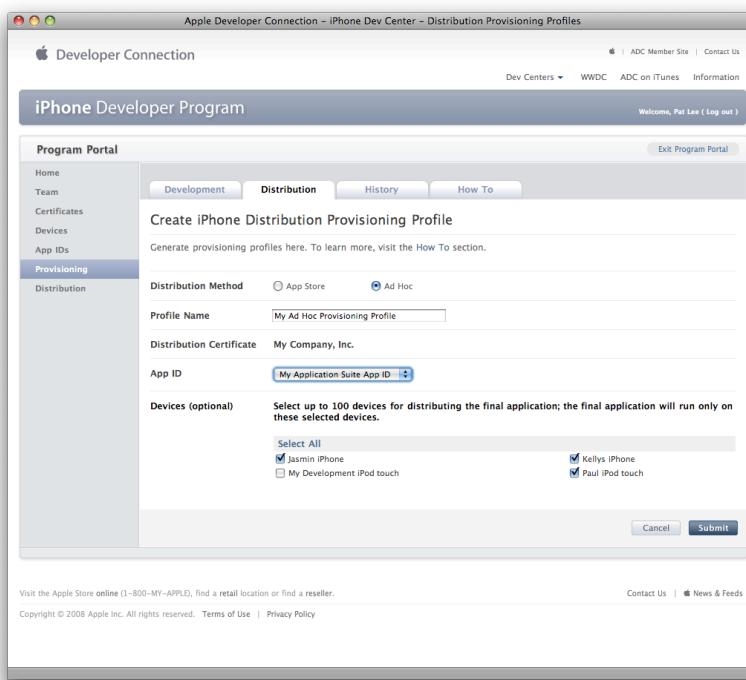
The screenshot shows the 'Create iPhone Distribution Provisioning Profile' interface in the iPhone Dev Center. The form fields are as follows:

- Distribution Method:** App Store (radio button selected)
- 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. (checkbox for My Development iPod touch)

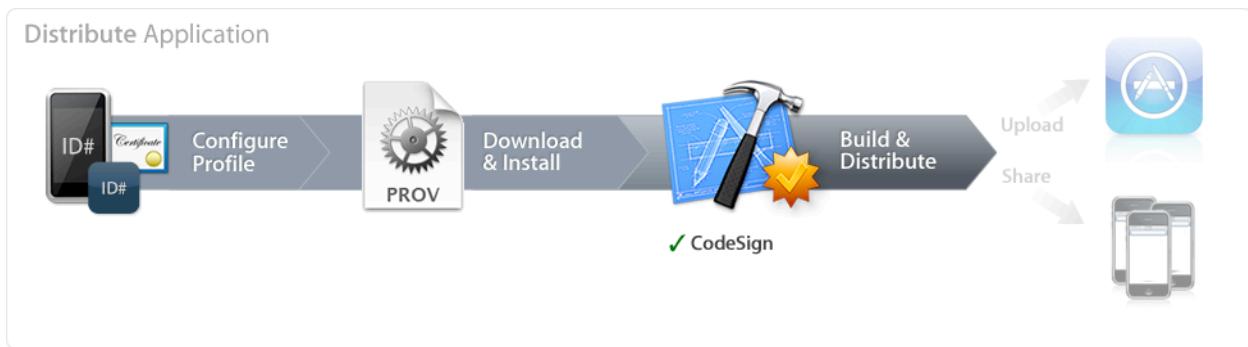
Creating and Downloading a Distribution Provisioning Profile for Ad Hoc Distribution

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

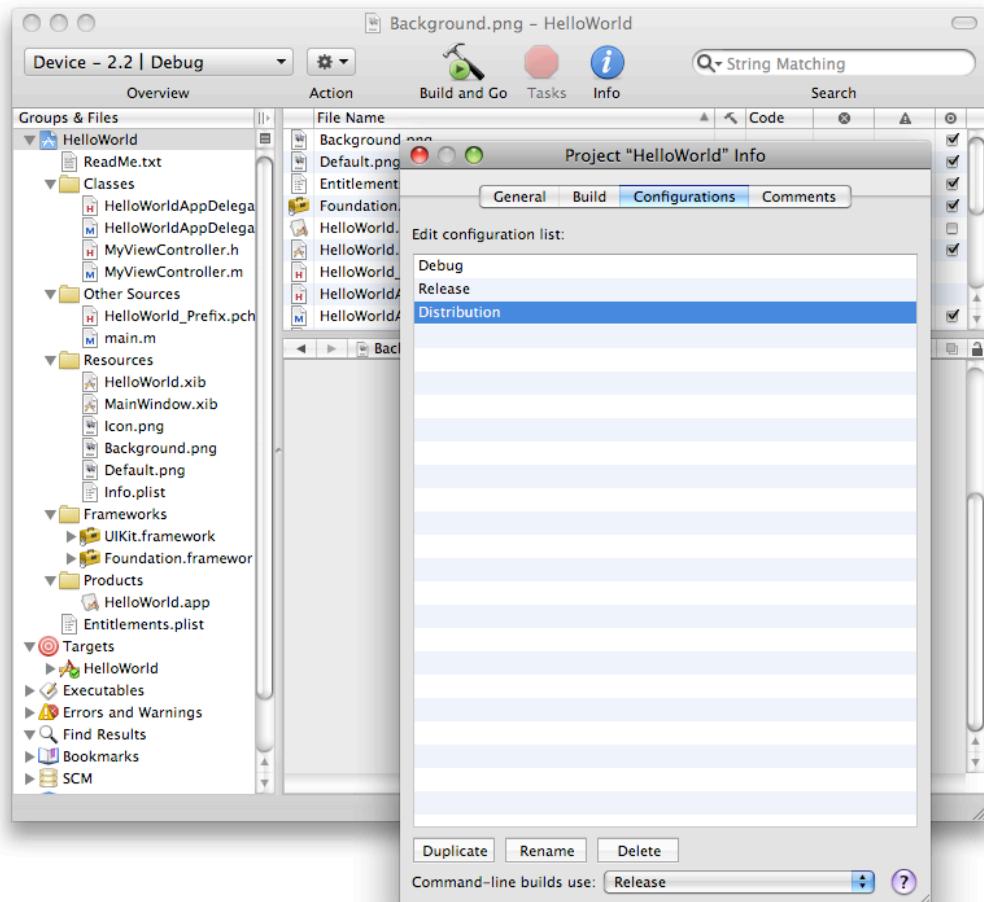
1. Team Agents should navigate to the 'Provisioning' section of the Program Portal.
2. Select the 'Ad Hoc' radio button.
3. Enter the name for your Ad Hoc Distribution Provisioning Profile.
4. Confirm your iOS Distribution Certificate has been created and is displayed.
5. Select the App ID for the application (or group 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.



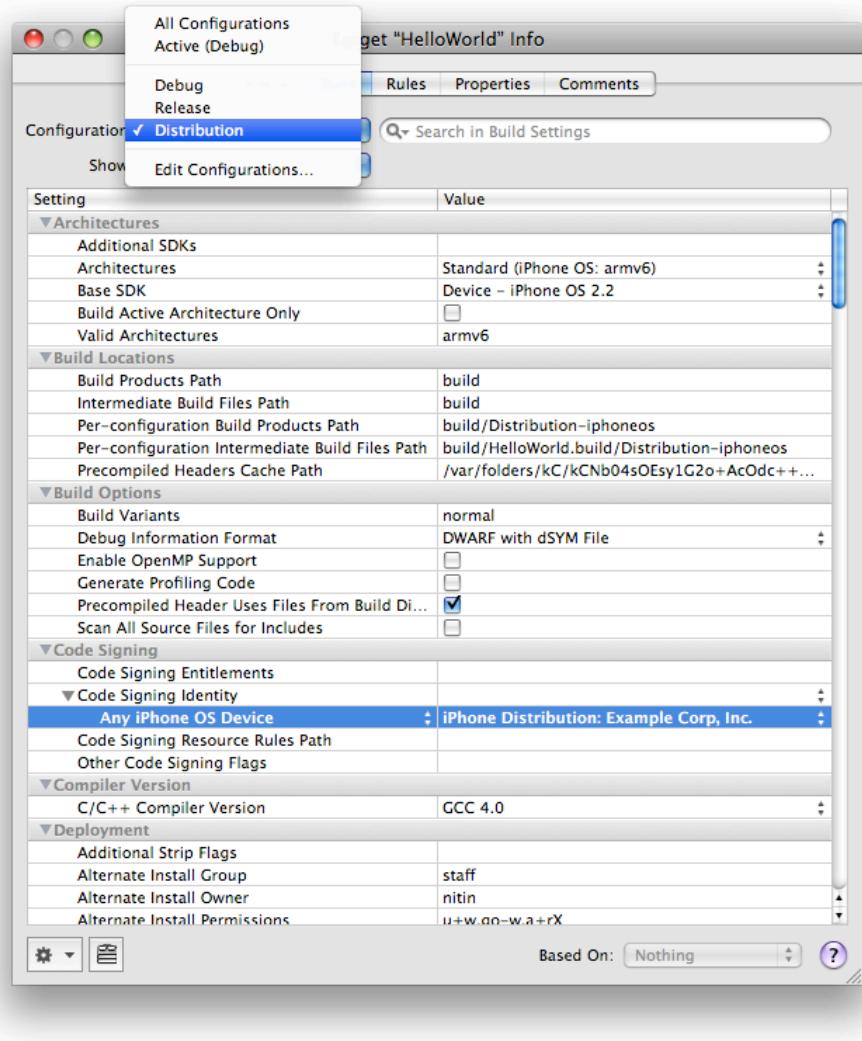
Building your Application with Xcode for Distribution



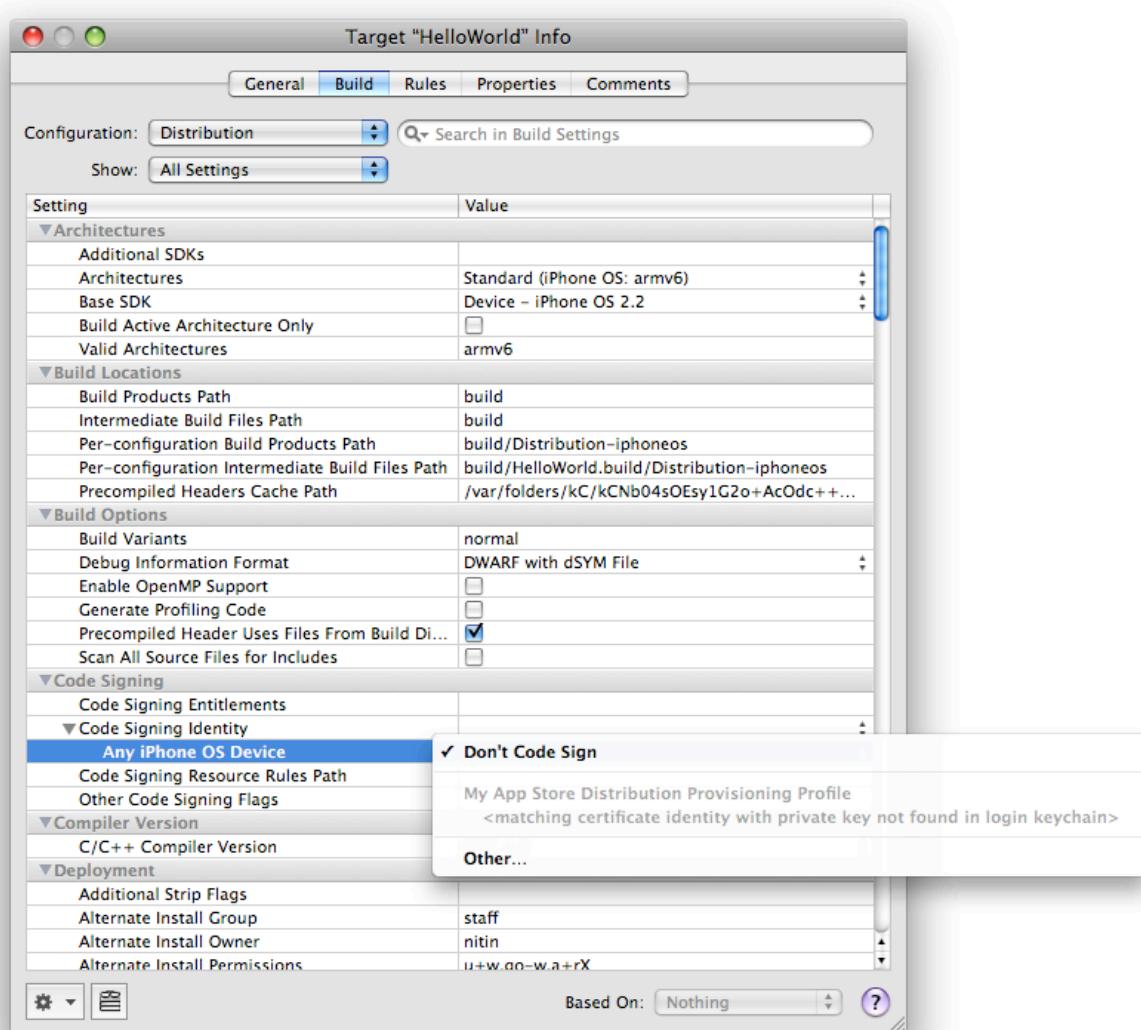
1. Launch Xcode and open your project.
2. If you have not already done so, drag the Distribution Provisioning Profile downloaded from the Program Portal onto the Xcode or iTunes icon in the dock (or, drag into '~/Library/MobileDevice/Provisioning Profiles' directory.)
3. Open the Xcode project and Duplicate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



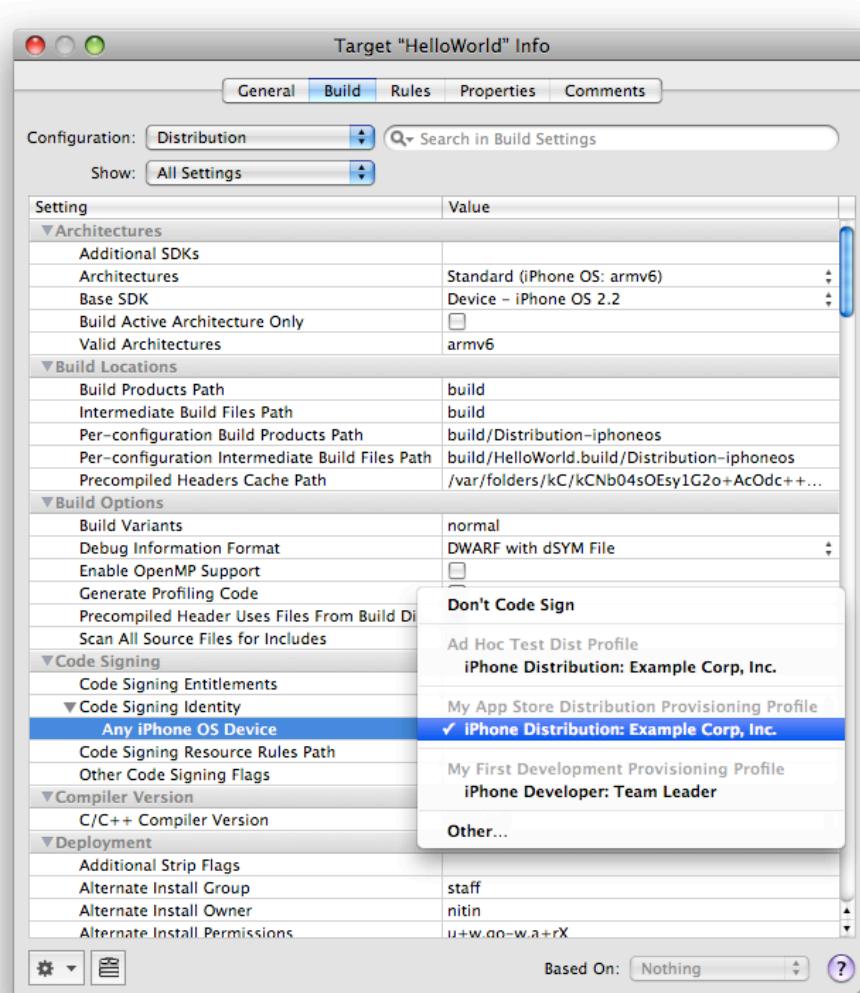
4. In the Target Info window, select the 'Build' tab and set the 'Configuration' to 'Distribution'



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, 'iPhone Distribution: Example Corp, Inc.' is the Distribution Certificate and 'My App Store Distribution Provisioning

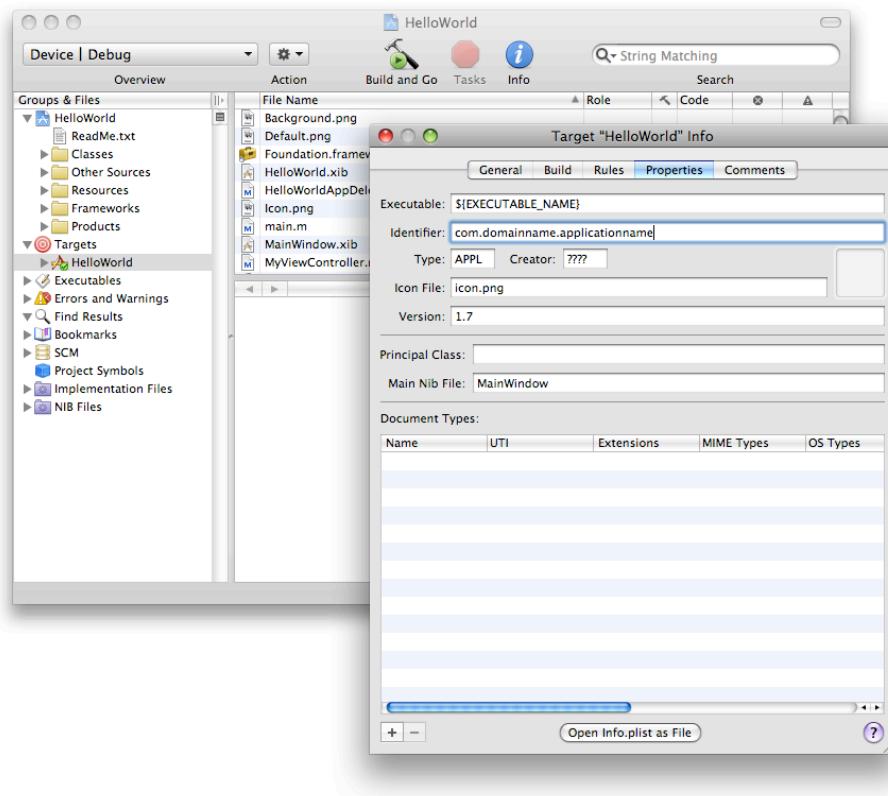


Profile' is the .mobileprovision file paired with it.



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.

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, you can replace the asterisk with whatever string you choose.



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.*

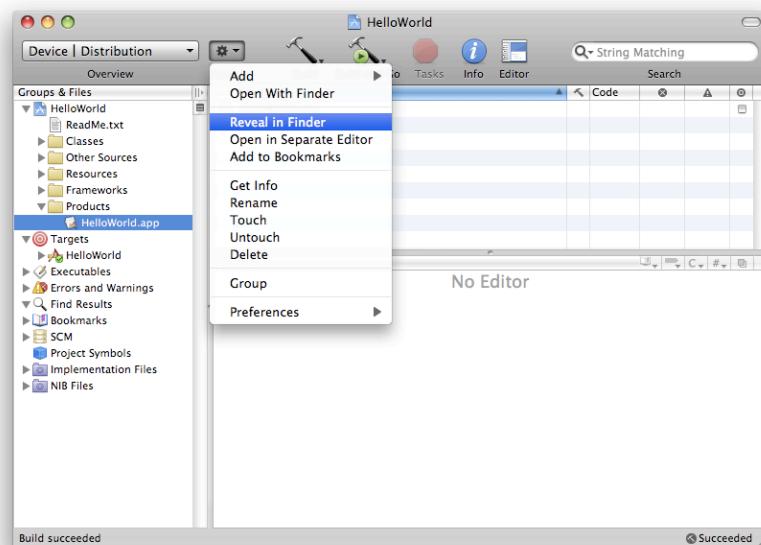
Identifier to enter in Xcode: com.domainname.<name_of_application>

Example App ID: A1B2C3D4E5.*

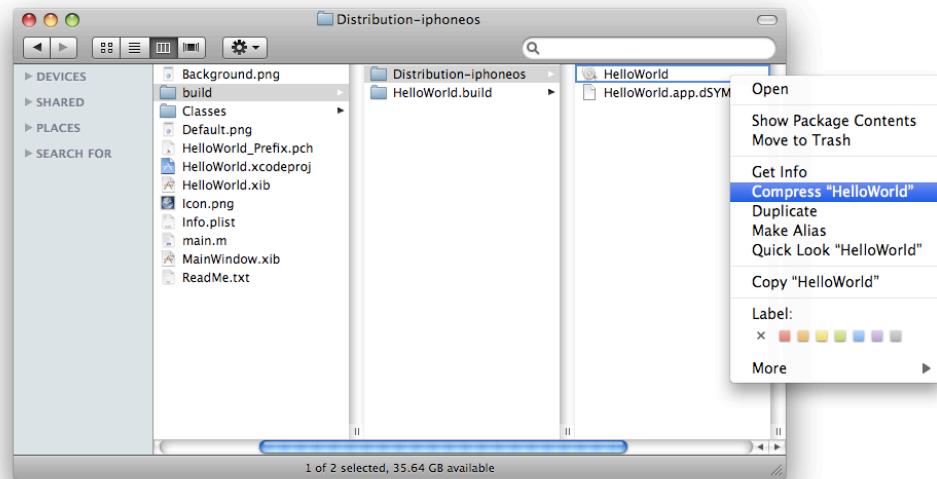
Identifier to enter in Xcode: <full_reverse_dns_company_and_application>

7. In the project window, select the Distribution Active Configuration from the overview popup and set the Active SDK to the desired Device.
8. 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.)

9. Highlight the app located within the "Products" sub-folder and select 'Reveal in Finder' from the Action popup.



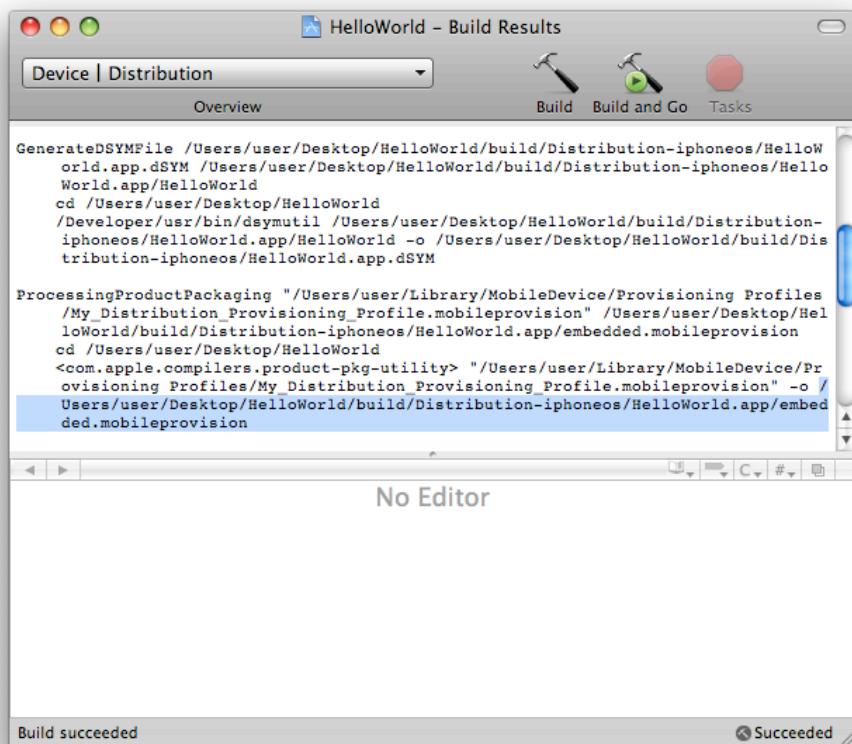
10. Use the compress option in Finder to create a .zip file containing your application. Be sure to compress only the .app file only and not the entire build folder. Using the command line to compress your file may result in problems when uploading to the App Store so best practice is to use the Finder to compress your application.



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.mobileprovision 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.



The screenshot shows the Xcode Build Results window titled "HelloWorld - Build Results". The "Device | Distribution" tab is selected. The main pane displays the build log output:

```

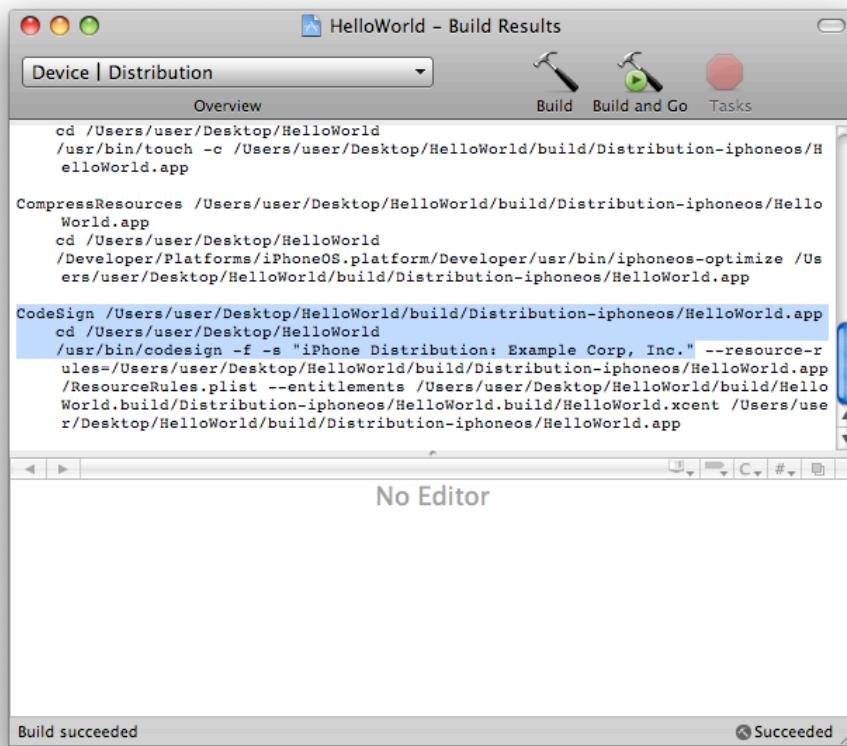
GenerateDSYMFile /Users/user/Desktop/HelloWorld/build/Distribution-iphoneos>HelloWorld.app.dSYM /Users/user/Desktop/HelloWorld/build/Distribution-iphoneos>HelloWorld.app/HelloWorld
cd /Users/user/Desktop/HelloWorld
/Developer/usr/bin/dsymutil /Users/user/Desktop/HelloWorld/build/Distribution-iphoneos/HelloWorld.app/HelloWorld -o /Users/user/Desktop/HelloWorld/build/Distribution-iphoneos/HelloWorld.app.dSYM

ProcessingProductPackaging "/Users/user/Library/MobileDevice/Provisioning Profiles/My_Distribution_Provisioning_Profile.mobileprovision" /Users/user/Desktop/HelloWorld/build/Distribution-iphoneos/HelloWorld.app/embedded.mobileprovision
cd /Users/user/Desktop/HelloWorld
<com.apple.compilers.product-pkg-utility> "/Users/user/Library/MobileDevice/Provisioning Profiles/My_Distribution_Provisioning_Profile.mobileprovision" -o /Users/user/Desktop/HelloWorld/build/distribution-iphoneos/HelloWorld.app/embedded.mobileprovision

```

The status bar at the bottom indicates "Build succeeded" and "Succeeded".

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 iPhone Certificate.



The screenshot shows the Xcode 'Build Results' window for a project named 'HelloWorld'. The window title is 'HelloWorld - Build Results'. The top menu bar includes 'Device | Distribution', 'Build', 'Build and Go', and 'Tasks'. The main pane displays the terminal command history for the build process:

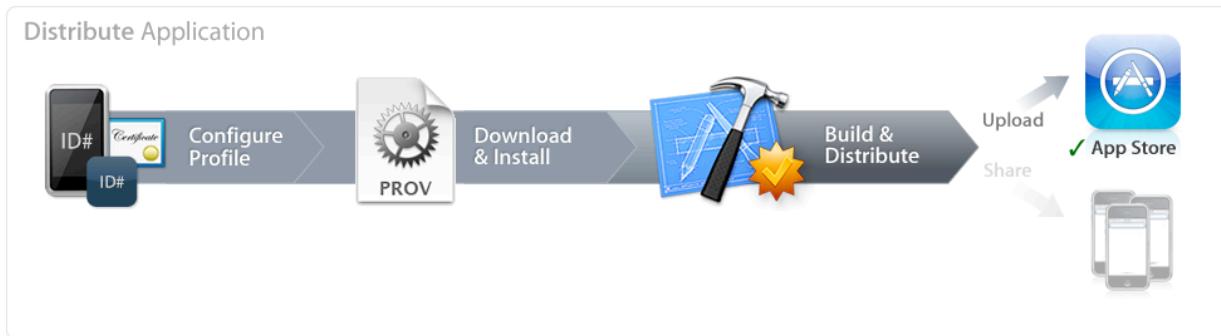
```
cd /Users/user/Desktop/HelloWorld  
/usr/bin/touch -c /Users/user/Desktop/HelloWorld/build/Distribution-iphoneos/H  
elloWorld.app  
  
CompressResources /Users/user/Desktop/HelloWorld/build/Distribution-iphoneos/Hello  
World.app  
cd /Users/user/Desktop/HelloWorld  
/Developer/Platforms/iPhoneOS.platform/Developer/usr/bin/iphoneos-optimize /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/us  
er/Desktop/HelloWorld/build/Distribution-iphoneos/HelloWorld.app
```

The status bar at the bottom indicates 'Build succeeded' and 'Succeeded'.

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. In the Xcode Build Menu, select Clean all Targets.
3. Delete any existing build directories in your Xcode project using Finder.
4. Re-launch Xcode and open your Project.
5. Select the iOS Distribution Certificate/provisioning profile pair in the Target Build Settings Inspector.
6. Rebuild your Project.

Submitting your application via iTunes Connect



iTunes Connect is a suite of web-based tools created for developers to submit and manage their applications for sale via the App Store. In iTunes Connect you will be able to check the status of your contract, manage users, submit your binary, metadata, and assets.

Team Agents will find a link to iTunes Connect in the ‘Distribution’ section of the Program Portal. Clicking this link will take you directly to the login page of iTunes Connect. Enter the same AppleID and password you use to log in to the iOS Developer Program Portal to access iTunes Connect.

Prior to uploading your application to iTunes Connect, you will be asked to review iTunes Distribution Terms & Conditions as well as gather metadata pertinent to your application. View the [iTunes Connect User Guide](#) for more information.

Distribution Terms & Conditions

Before beginning the application submission process, it is critical that you read and review the distribution Terms and Conditions. If your company has a legal department, it is recommended that you provide this document to them for review. If you plan to sell your application, you need to have your paid commercial agreement in place, so that it can be posted to the App Store. If not, your application will not be posted to the App Store until your commercial agreement is signed. If your application is free, you've already entered into the freeware distribution agreement by being accepted into the iOS Developer Program.

Gathering your Application Metadata

Prior to submitting your application, it will be helpful for you to collect all the information about your application that will be displayed on the App Store. Some data that you will want to gather are:

- Application Name (Must conform to [guidelines for using Apple trademarks and copyrights](#))
- Application Description (up to 4000 characters)
- Device type(s) the Application can be run on (iPhone and/or iPod touch)
- Unique Stock Keeping Unit (SKU) for each major version of your application
- Primary Category
- Secondary Category (Optional)
- Game Rating (Optional)
- Unique Version #
- Application copyright holder
- Support URL for the Company
- Support Email Address (for use by Apple only)
- End User License Agreement (Optional) If a EULA is not provided, standard iTunes App Store EULA will be applied.
- Territories application to be distributed in
- Application Availability Date
- Application Price
- Localization desired (English, Dutch, French, German, Italian, Spanish, Japanese)

Preparing your Application Artwork

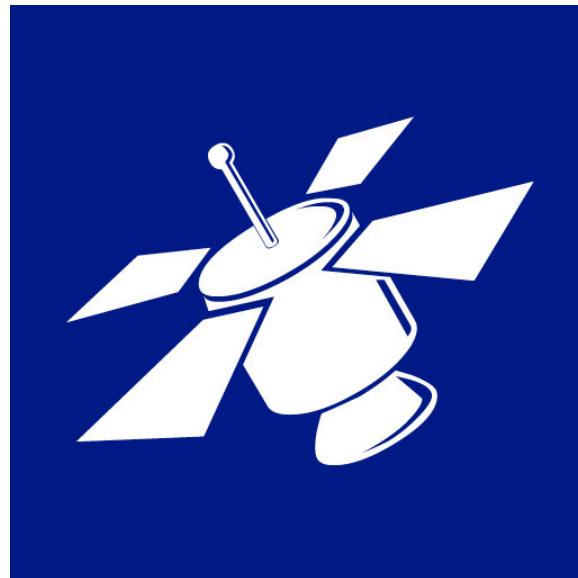
You have the ability to provide several pieces of artwork that will be visible to customers visiting the App Store. Some of the assets you may wish to submit with your application are:

- **iPhone/iPod touch Home Screen Icon** - Your application binary must contain a flattened, square-image icon that is 57x57 pixels. This icon is what will be displayed on the iPod touch or iPhone home screen.



- **Large Application Icon** - This will be used to feature your application on the App Storefront. To be featured prominently on the App Store we recommend you provide an attractive, original icon that meets the following requirements:

- 512 x 512 pixels (flattened, square image)
- 72 DPI
- .jpg, .jpeg, or .tiff format



- **Primary Screenshot** - The primary screenshot will be used as the main screenshot visible on your application product page on the App Store, and the one that will also be displayed on the device. For best results, don't include the iPhone status bar in your screenshots and follow these requirements:

- 320x460 portrait (without status bar) minimum
- 480x300 landscape (without status bar) minimum
- 320x480 portrait (full screen)

Uploading your Application

After completing the above steps, iTunes will present an upload mechanism for you to submit your .zip file to Apple.

Guidelines for Submitting Updates to Your iPhone Application

Keeping your iPhone application on the App Store up-to-date is important, however submitting these updates at a high frequency may cause an unnecessary delay in your update posting to the App Store. By planning and scheduling your updates, you can increase the impact of each update and improve the experience for your customers.

Use these guidelines when planning updates to your iPhone application:

High Frequency: Crashes or data loss

Updates to your application that address crashing and data loss should be submitted as frequently as necessary. Fixing as many related bugs as possible in each update is highly recommended.

Medium Frequency: Minor enhancements and usability improvements

Consider a release schedule between two to four weeks that groups together updates which do not affect the core functionality of your application, such as user interface improvements, spelling corrections, and minor functionality enhancements.

Low Frequency: New features

Applications with new features should be submitted on a periodic, monthly basis. A high frequency of new feature updates suggests poor development planning and can be confusing to your customers.

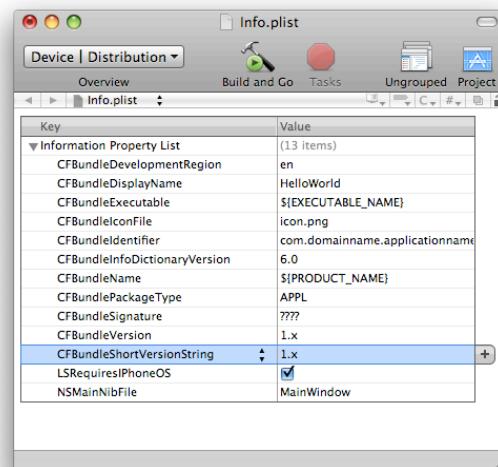
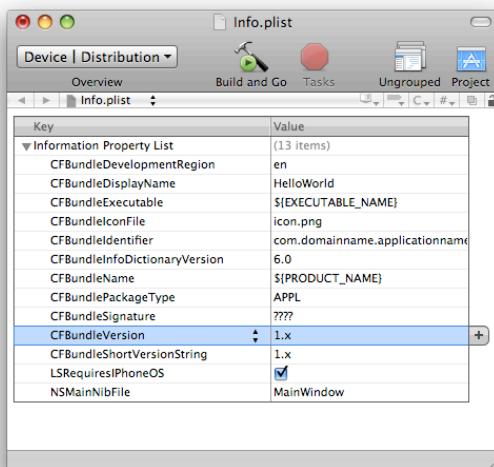
Using Ad Hoc Distribution to test your updates

As an enrolled iPhone Developer, you can use Ad Hoc Distribution to share your application with up to 100 iPhone or iPod touch users as a method to obtain feedback on the changes you've made.

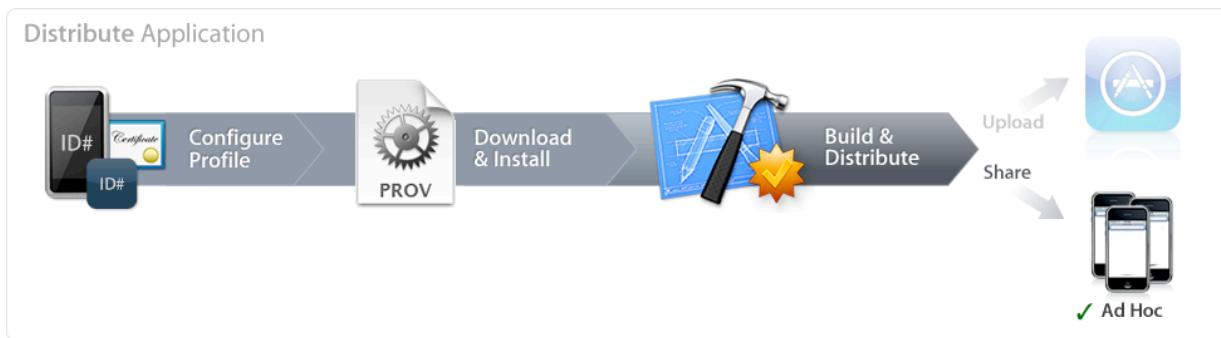
Updating your Application

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).



Distributing your application with Ad Hoc Distribution



In addition to distributing applications through the App Store, iPhone Developers enrolled in the Standard program have the opportunity to distribute their application outside of the App Store on up to 100 different devices.

To distribute your application for Ad Hoc Distribution:

1. Create and Download an Ad Hoc Distribution Provisioning Profile in the 'Provisioning'->'Distribution' tab as described in the 'Creating and Downloading a Distribution Provisioning Profile for Ad-hoc Distribution' section above.
2. Build your application in Xcode using the build process described in the 'Building your Application with Xcode for Distribution' section.
3. To share your application, provide both the compiled application binary AND the Ad Hoc Distribution Provisioning profile to the owners of the devices specified in the Ad Hoc Distribution Provisioning Profile.
4. Recipients of the .app bundle and .mobileprovision file can drag both files onto the iTunes icon in the dock and sync their device to install. Both the application and the Ad Hoc Distribution Provisioning Profile will be installed on the device. Windows users should drag both the application and the Ad Hoc Distribution Provisioning Profile into the Library section of iTunes and then sync their device to install. The [iPhone Configuration Utility](#) can also be used for Ad Hoc application installation.

Glossary

Below are a list of commonly used terms in the iOS Developer Program:

App ID - A unique digital fingerprint that iOS uses to grant your application access to a portion of the Keychain and one part of a Development and Distribution Provisioning Profile.

Apple device - An iPhone or iPod touch

Development Team - A Development team is either a single person enrolled in the Individual Program or a Team Agent and group of Team Admins and Team Members in the Standard Program.

iPhone - A combination mobile phone, multimedia player, and wireless Internet device from Apple Inc.

iOS Development Certificate - An electronic document that associates a digital identity with other information, including a name, email address, or business. An iOS Development Certificate is restricted to application development only and is valid for a limited amount of time.

iOS Distribution Certificate - An electronic document that associates a digital identity with other information, including a name, email address, or business. An iOS Distribution Certificate is restricted to application distribution only and is valid for a limited amount of time.

iPod touch - A portable multimedia player and wireless Internet device from Apple Inc.

iTunes Connect - A suite of web-based tools created for developers to submit and manage their applications for sale via the App Store.

iOS - iOS comprises the operating system and technologies that you use to run applications natively on iPhone and iPod touch devices. Although it shares a common heritage and many underlying technologies with Mac OS X, iOS was designed to meet the needs of a mobile environment, where user's needs are slightly different.

Provisioning Profile - A set of entities and entitlements allowing applications to be installed and run on an Apple device.

Team Agent - The original person accepted into the iOS Developer Program for all membership types. The Team Agent has access to all functionality in the iOS Developer Program Portal and is the sole person capable of Distributing an application.

Team Admin - A person on an iPhone Development Team designated by the Team Agent or another Team Admin who has the capability to approve iOS Development Certificate requests, invite Team Members, add Apple devices and create App IDs and Development Provisioning Profiles.

Team Member - A person on an iPhone Development Team who can request iOS Development Certificates and install iOS applications on an Apple device.

Unique Device Identifier (UDID) - A 40 character string used to identify a specific Apple device.

Xcode - Apple's premiere development environment for Mac OS X and iOS and includes all the tools needed to create, debug, and optimize Mac OS X and iOS applications.

Revision History

Version 2.7 - September 1, 2010

Minor Document Update

Version 2.6 - March 4, 2010

Minor Document Update

- Updated instructions for Team Invitations
- New instructions for managing certificate expiration

Version 2.5 - September 9, 2009

Major Document Update

- Updated Screenshots
- App ID Instructions Updated
- Updated description of Provisioning Profile installation instructions
- Addition of Apple Push Notification, In App Purchase, Keychain Data Sharing instructions.