



# Kony Management Quick Start Series Getting Started With EMM

## Release V8

### Document Relevance and Accuracy

This document is considered relevant to the Release stated on this title page and the document version stated on the Revision History page. Remember to always view and download the latest document version relevant to the software release you are using.

Copyright © 2017 Kony, Inc.

All rights reserved.

October, 2017

This document contains information proprietary to Kony, Inc., is bound by the Kony license agreements, and may not be used except in the context of understanding the use and methods of Kony, Inc., software without prior, express, written permission. Kony, Empowering Everywhere, Kony Fabric, Kony Nitro, and Kony Visualizer are trademarks of Kony, Inc. MobileFabric is a registered trademark of Kony, Inc. Microsoft, the Microsoft logo, Internet Explorer, Windows, and Windows Vista are registered trademarks of Microsoft Corporation. Apple, the Apple logo, iTunes, iPhone, iPad, OS X, Objective-C, Safari, Apple Pay, Apple Watch, and Xcode are trademarks or registered trademarks of Apple, Inc. Google, the Google logo, Android, and the Android logo are registered trademarks of Google, Inc. Chrome is a trademark of Google, Inc. BlackBerry, PlayBook, Research in Motion, and RIM are registered trademarks of BlackBerry. SAP® and SAP® Business Suite® are registered trademarks of SAP SE in Germany and in several other countries. All other terms, trademarks, or service marks mentioned in this document have been capitalized and are to be considered the property of their respective owners.

## Revision History

Date	Document Version	Description of Modifications/Release
10/09/2017	1.0	Document published for V8 GA

# Table of Contents

---

<b>1. Preface</b> .....	<b>7</b>
1.1 Purpose .....	7
1.2 Intended Audience .....	7
1.3 Formatting conventions used in this guide .....	7
1.4 Supported Platforms .....	9
1.5 Contact Us .....	9
<b>2. Overview</b> .....	<b>10</b>
<b>3. Introduction</b> .....	<b>11</b>
<b>4. Certificates</b> .....	<b>12</b>
4.1 Platform Specific Settings .....	13
<b>5. Optional Third Party Server Integration</b> .....	<b>15</b>
<b>6. Optional Settings</b> .....	<b>17</b>
6.1 Device Settings .....	17
6.2 Branding .....	19
6.3 Admin Email Settings .....	19
<b>7. Access Management</b> .....	<b>20</b>
7.1 Adding Users and Groups .....	20
7.2 Permission Sets .....	21
<b>8. Enroll Devices</b> .....	<b>23</b>
8.1 Enrollment Process .....	23

---

8.2 Platform Specifics .....	23
8.3 Monitoring the System .....	25
<b>9. Devices .....</b>	<b>27</b>
<b>10. Device Details .....</b>	<b>28</b>
10.1 View Device Policy .....	29
<b>11. Device Policy Creation .....</b>	<b>31</b>
<b>12. Device Set Creation .....</b>	<b>32</b>
12.1 Device Set Conditions .....	32
12.2 Device Set Rule Definition .....	33
12.3 Device Set Publication .....	33
12.4 Post Publication .....	33
<b>13. App Creation and Upload to Enterprise Store .....</b>	<b>34</b>
<b>14. App Policy Creation .....</b>	<b>36</b>
<b>15. Generating Certificates .....</b>	<b>37</b>
15.1 Implications of renewing iOS certificates on Launchpad and Child apps .....	37
15.2 Creating the Apple Enterprise Wild Card Distribution Certificate .....	38
15.3 Recreate Apple Wild Card Distribution Certificate .....	48
15.4 Creating the Apple Enterprise Wild Card Provisioning Profile .....	49
15.5 Recreate Apple Wild Card Provisioning Profile .....	57
15.6 Creating the Apple Application Manager (Launchpad app) Push Certificate .....	58
15.7 Recreate Apple Application Manager (Launchpad app) Push Certificate .....	69

---

15.8	Creating the Apple Application Manager (Launchpad app) Provisioning Profile .....	70
15.9	Recreate Apple Application Manager Provisioning Profile .....	76
15.10	Assigning App Resources in the Kony Management Cloud Administrator Console .....	76
15.11	Creating the Apple Push Notification Certificate (APNS) .....	77
15.12	Renew Apple Push Notifications Certificate .....	88
15.13	Generating Certificate Signing Request (CSR) in Windows .....	88
15.14	Generating Certificate Signing Request (CSR) in Linux .....	91
15.15	Creating Android Certificates and Keys .....	93
15.16	Re-creating Android Certificates and Keys .....	105
<b>16.</b>	<b>Generating Package Family Name .....</b>	<b>106</b>

# 1. Preface

Kony Enterprise Mobility Manager (EMM) is an all-encompassing approach to the secure use of company-owned and employee-owned mobile devices. EMM typically involves combination of Mobile Application Management (MAM), Mobile Device Management (MDM), and Mobile Access Management.

EMM solution: Scenarios

- For employees who need to install and use the enterprise apps on their own devices.
- For an enterprise that intends to manage its applications through a web console.
- For applications that can be managed with policies based on the latest IT guidelines within the organization.

## 1.1 Purpose

This document helps you familiarize with Kony Enterprise Mobile Management and provide procedural information to use Management console, Self-service console, and Launchpad.

## 1.2 Intended Audience

The information in this guide is intended primarily for:

- **System Administrators:** Employees who implement and enforce the security structure, responsible for maintaining multi-user computer system, including a local area network (LAN), setting up user accounts, installing system-wide software, adding and configuring new workstations and so on.
- **Users:** Employees who use the EMM where the application is running and can access some or all of its features.

## 1.3 Formatting conventions used in this guide

The following typographical conventions are used throughout the document:

Click here

Conventions	Explanation
Monospace	<ul style="list-style-type: none"> <li>■ User input text, system prompts and responses</li> <li>■ File Path</li> <li>■ Commands</li> <li>■ Program Code</li> <li>■ File Names</li> </ul>
<i>Italic</i>	<ul style="list-style-type: none"> <li>■ Emphasis</li> <li>■ Names of Books and Documents</li> <li>■ New Terminology</li> </ul>
<b>Bold</b>	<ul style="list-style-type: none"> <li>■ Windows</li> <li>■ Menus</li> <li>■ Buttons</li> <li>■ Icons</li> <li>■ Fields</li> <li>■ Tabs</li> <li>■ Folders</li> </ul>
<a href="#">URL</a>	Active link to a URL.
<i>Note</i>	Provides helpful hints or additional information.
<i>Important</i>	Highlights actions or information that might cause problems to systems or data.



## 1.4 Supported Platforms

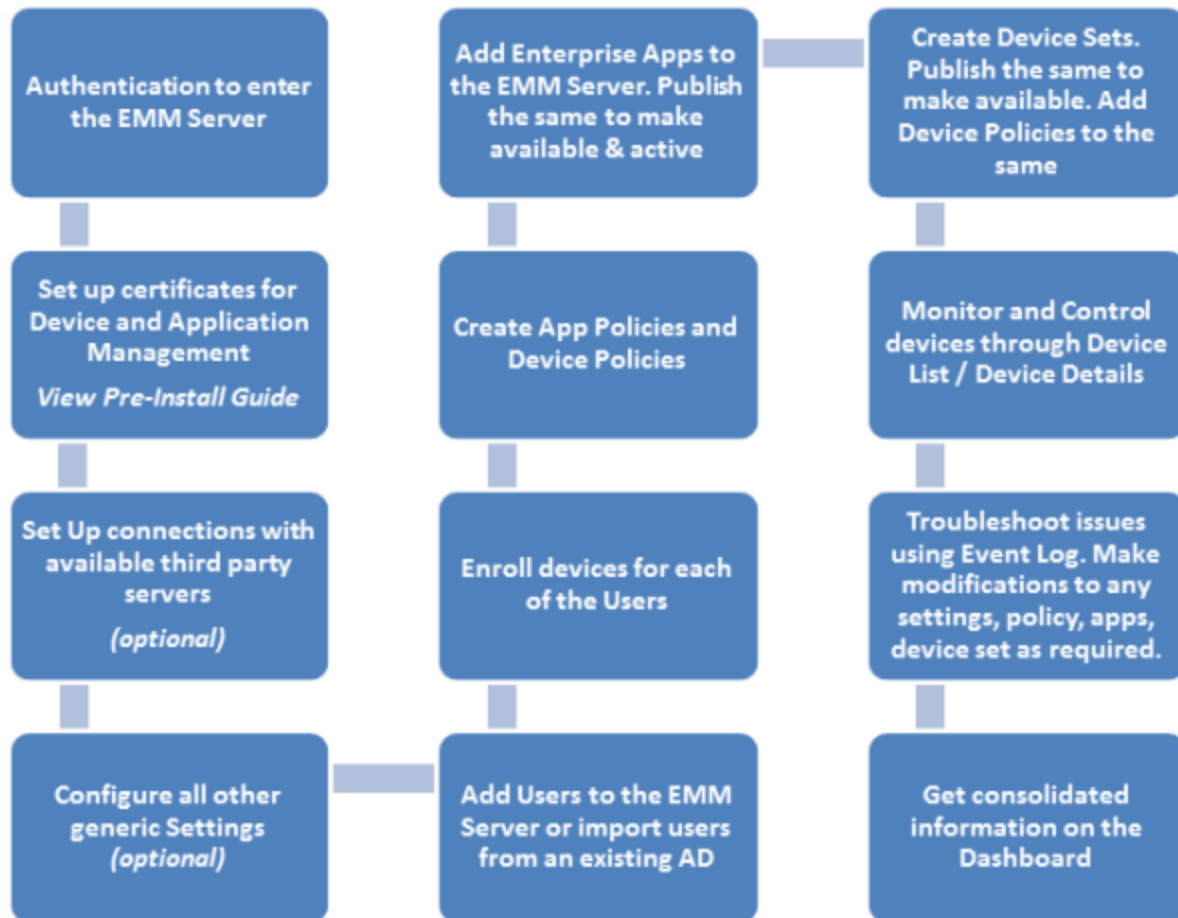
Supported Platforms are iOS, iPad, Android, Android Tablet, and Windows Phone 8.1. Other Device Operating Systems are not supported.

## 1.5 Contact Us

We welcome your feedback on our documentation. Write to us at [techpubs@kony.com](mailto:techpubs@kony.com). For technical questions, suggestions, comments or to report problems on Kony's product line, contact [support@kony.com](mailto:support@kony.com).

## 2. Overview

The steps described below are a quick guide on how to set up and get ready to use Kony EMM



## 3. Introduction

The Kony Management Cloud is an Enterprise Mobility Management (EMM) software suite that provides policy configuration and management tools for mobile handheld devices and corresponding applications on smartphones and tablets. EMM helps enterprises to manage complex communications among mobile devices by supporting security, network services, and software and hardware management across multiple OS platforms.

EMM also supports bring your own device (BYOD) initiatives that has become the focus of many enterprises. It can support corporate and personal devices, and helps to support a more complex and heterogeneous environment.

This guide will help you with first time configuration tasks as well as providing an overview to successfully using Kony EMM to enroll devices, create policies, and manage apps.

## 4. Certificates

Several Certificates are required to make the EMM system functional.

Platform	Certificates
iOS	<ul style="list-style-type: none"><li>• APNS</li><li>• Apple Enterprise Certificates<ul style="list-style-type: none"><li>◦ Wild Card Distribution Certificate</li><li>◦ Wild Card Provisioning Profile</li></ul></li><li>• Apple App Manager<ul style="list-style-type: none"><li>◦ Provisioning Profile</li><li>◦ Push Certificate</li></ul></li></ul>
Android	<ul style="list-style-type: none"><li>• Google ID</li><li>• Key Store</li><li>• Key Store Pass phrase</li><li>• Certificate Alias</li><li>• Certificate Pass Phrase</li><li>• GCM Key</li><li>• Sender ID</li></ul>

**Note:** The APNS certificate is required if you plan to enroll iOS devices.

## 4.1 Platform Specific Settings

The Apple Enterprise and App Manager certificates are required for the iOS EMM enrollment app (Launchpad), iOS enterprise app distribution, and iOS app wrapping. iOS devices can be enrolled without the Enterprise App Store using the browser, if these certificates are not available.

The Android Key Store file and details are required for the Android EMM enrollment app (Launchpad), Android enterprise app distribution, and Android app wrapping. Android cannot be enrolled using the browser like iOS devices; without these entries Android devices cannot be supported.

These certificates are entered in the Kony Management Cloud administrative console through **Settings > Applications Settings > Certificates** tab.

### Application Settings

Certificates Usage Settings Policy Error Messages Encryption Key

#### iOS

##### Enterprise Certificates

*Wrap is in progress. Cannot alter details till complete.*

Wild Card Distribution Certificate	KonyEnterpriseDistribution.p12	3.15 KB	
Certificate Pass Phrase	••••••••		
<input type="button" value="Certificate Details"/>			
Wild Card Provisioning Certificate	MAMEnterpriseDistribution.mobileprovision	7.37 KB	
<input type="button" value="Certificate Details and Errors"/>			

##### Launchpad

Note: The Bundle Identifier should end with .containerapp

Provisioning Certificate	containerapp_Push.mobileprovision	7.41 KB	
<input type="button" value="Certificate Details and Errors"/>			
Push Certificate	ContainerPushCertificate.p12	3.18 KB	
Push Certificate Pass Phrase	••••••••		
<input type="button" value="Certificate Details"/>			

#### Android

Key Store	debug.keystore	1.27 KB	
Key Store Pass Phrase	••••••••		
Certificate Alias	androiddebugkey		
Certificate Pass Phrase	••••••~		
<input type="button" value="Certificate Details"/>			
GCM key for Android	AlzaSyBdBp3Z2_8qza6c9eF		
Project number (Sender ID)	991045329872		
Google ID	konysolutions@gmail.com		

Step by step instructions for creating all resources are under [Generating Certificates](#) section of this document.

## 5. Optional Third Party Server Integration

There are several third party servers to which EMM can connect. The administrator is required to provide the necessary details to establish a connection between the EMM Server and these servers if the relevant feature is desired.

- **Active Directory:** This can be set up on the **Directory Settings** page under Settings. If no LDAP connector is set up, the Administrator must create all users and groups locally within the EMM Server.
- **Exchange ActiveSync:** Exchange ActiveSync can be set up on the **Exchange Settings** page in the Settings section. If no Exchange ActiveSync server connection is set up, the EMM server cannot automate access or denial of devices communicating with Exchange via ActiveSync. This can still be done manually by an Exchange administrator until connectivity is established between Exchange and the Kony EMM server.

**Note:** It is recommended not to establish this connection during a trial period, as once established only devices enrolled to Kony EMM and compliant with designated rules will be capable of receiving enterprise email.

- **BlackBerry Enterprise Server (BES):** To connect the EMM Server to the BES, the administrator must configure the connection settings. This can be done through **Device Settings > Communication Configuration**. If this is not configured, BlackBerry devices cannot be supported by EMM. If there are existing devices in the enterprise, they cannot be synced with the EMM Server and therefore cannot be monitored through EMM.
- **Windows MDM Server:** To provide support for Windows 6.x devices, the administrator must configure the connection settings to the Windows MDM server. This can be done through **Device Settings > Communication Configuration**. Details on installing a Windows MDM server are included in the Kony EMM Pre-Install Guide.

### Configure Windows 6.x

**Warning:** Without providing these fields Windows 6.x devices cannot be enrolled.

Windows 6.x Service URL*	<input type="text" value="https://winmdm.manage.lov"/>
Windows MDM Service Key*	<input type="text" value="mykey"/>
Windows MDM Service Secret*	<input type="password" value="••••••••"/>
My (Kony Console) Key*	<input type="text" value="key"/>
My(Kony Console) Secret*	<input type="password" value="••••••"/>

### Configure Windows Phone 8

**Warning:** Without providing these fields Windows Phone 8 devices cannot be enrolled.

Company Name	<input type="text" value="Ruther &amp; Ford"/>
Device Sync Interval (in minutes)	<input type="text" value="30"/>



## 6. Optional Settings

The following settings have defaults, and are not mandatory to change to make the EMM system function. It is recommended that they are customized before adding devices. Creating them once the system is functioning can put a significant load on the devices enrolled as new details must be synced from the EMM Server. This may lead to a poor experience for device users.

### 6.1 Device Settings

#### 6.1.1 Usage Settings

There are several parameters that define how EMM behaves.

##### 6.1.1.1 Time Zone

The default Time Zone set is EST. If your company follows a different time zone, set the same. Specifying this displays all time related attributes according to this time zone. Showing the right time zone avoids a lot of unwanted confusion.

##### 6.1.1.2 Device Agent Settings

**Timeout Period:** This specifies the maximum idle time of the device after which the user is logged out of the Device Agent.

##### 6.1.1.3 Enrollment Settings

#### Allowed Enrollment Methods

- Administrator can choose to allow **Admin Initiated**, **Device Initiated** or **Self Service Portal** Initiated enrollment processes. Only the selected mechanism is allowed to enroll. Enrollment through unauthorized mechanisms fails. By default all the three are selected. If none of them are selected, then no enrollment is possible.
- **Verify User Presence in AD Group (Optional):** This is marked **No** by default. If it is marked **Yes**, it means that to enroll their device, the User must be in an AD Group as chosen by the

Administrator.

- Only if the above choice is Yes, Enforce AD Group for Enrollment can be specified. The Administrator must choose only one group among all AD Groups.
- This group is used only at the time of enrollment. Should the members of the group change post-enrollment; it is not taken into consideration.

### Enrollment Denied List (Optional)

All the devices for which future enrollment is denied are part of this list. The Administrator can view and modify the list. By deleting devices from this list, they are allowed to enroll again. The Administrator cannot add any devices to this list directly.

## 6.1.2 Terms and Conditions

The Terms and Conditions of the Company can be provided in the Terms and Conditions tab of Device Settings from the Settings area of the Administrator console. An acceptance sign off is sought by every device user on these terms and conditions. If detailed Terms and Conditions are not yet written, a welcome message is typically placed here instead.

The screenshot displays the 'Device Settings' application. At the top, there are four tabs: 'Usage Configuration', 'Terms and Conditions', 'Message Templates', and 'Communication Configuration'. The 'Terms and Conditions' tab is selected. Below the tabs is a window titled 'Employee Terms'. This window contains a rich text editor with a toolbar at the top. The toolbar includes dropdown menus for 'Format', 'Font', and 'Size', followed by buttons for Bold (B), Italic (I), Underline (U), and Strikethrough (I<sub>x</sub>). There are also buttons for bulleted and numbered lists, indenting and outdenting, link, unlink, and image insertion. The main text area of the editor contains the text 'Terms & Conditions'. At the bottom of the 'Employee Terms' window, there are 'Save' and 'Cancel' buttons.

## 6.2 Branding

This page allows you to replace the existing Kony icons with your own. If this is not configured, it reflects Kony branding by default for all the placeholders provided.

## 6.3 Admin Email Settings

This allows you to set the default email id to, which all the support queries are sent. If this is not configured, the default support email id is not configured. It results in an error on the device every time a user initiates Contact Support.

## 7. Access Management

Access Management includes adding users and groups and applying permission sets to them.

The screenshot shows the 'Users' management interface. At the top, there are buttons for '+ New User' and 'Import From Active Directory'. Below this is a table of users with the following data:

Display Name	User ID	Source	Email	Status	Permission Set
anupam	anupam	Active Directory	anupam@mdmtest.local	Active	Admin Permissions
Akram Ali	mdmadmin@kony.com	Local <a href="#">Reset Password</a>	mdmadmin@kony.com	Active	Admin Permissions
sunil_123!@#	sunil_123!@#	Active Directory	sunil.meda@kony.com	Active	None
Aravind Kony	aravindakony	Active Directory	aravindakony@mdmtest.local	Active	None
HSIREESHA	HSIREESHA	Active Directory	sireesha.haripanthula@kony.com	Active	None

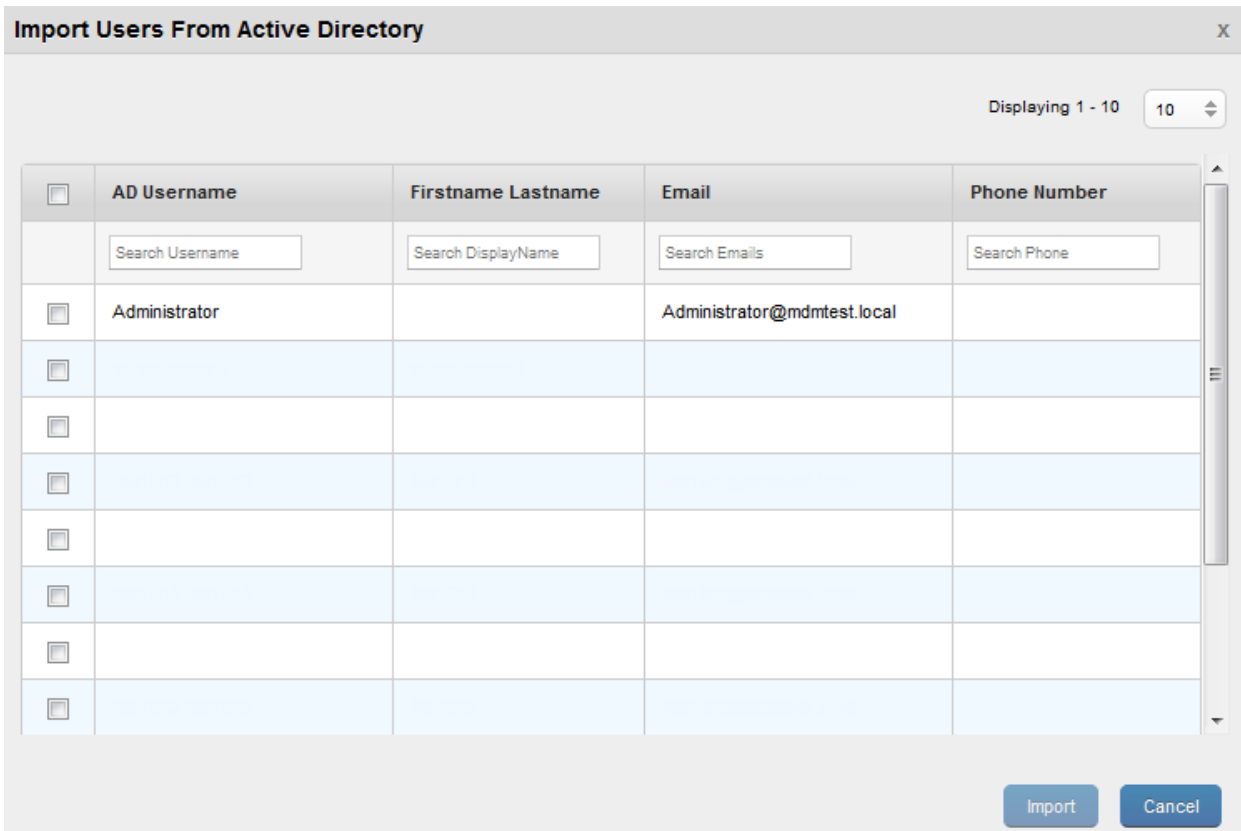
At the bottom of the table, there are 'Sync Selected Users', 'Previous', 'Page (1/3)', and 'Next' buttons.

### 7.1 Adding Users and Groups

There are two ways to add Users or Groups:

- Create Local Users or Groups (on the EMM Server)
- Import from Active Directory: This requires prior integration with the existing company AD. If it is not present, only Local Users and Groups can be created.

Users and Groups can have permission sets applied to them. With every User, the list of permissions is shown with their details.



Import Users From Active Directory

Displaying 1 - 10 10

<input type="checkbox"/>	AD Username	Firstname Lastname	Email	Phone Number
	<input type="text" value="Search Username"/>	<input type="text" value="Search DisplayName"/>	<input type="text" value="Search Emails"/>	<input type="text" value="Search Phone"/>
<input type="checkbox"/>	Administrator		Administrator@mdmtest.local	
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				

Import Cancel

## 7.2 Permission Sets

Typically, the first user of the system is a super administrator with rights to view all the pages and perform all the actions.

If other Administrators are added, super administrator can create permission sets that allow access to certain pages and privileges to perform certain actions. Every Administrator must have a Permission Set applied to them to access the Management Console. Creating different permission sets can help to define roles of different administrators more clearly.

Users by default have access to the Self Service Portal.

## Permission Details

[Permissions](#) > Sample Permission Set

### Device Management Page Permissions

Device List  Allow  Denied

Device Enrollment  Allow  Denied

Device Set  Allow  Denied

Device Policy  Allow  Denied

Device Settings  Allow  Denied

Event Log  Allow  Denied

### App Management Page Permissions

Apps  Allow  Denied

Policies  Allow  Denied

Categories  Allow  Denied

Application Settings  Allow  Denied

### Device Management Action Permissions

Lock Device  Allow  Denied

Reset Passcode  Allow  Denied

Wipe Device  Allow  Denied

Force Check-in  Allow  Denied

Approve Device Set  Allow  Denied

Publish/Unpublish Device Set  Allow  Denied

Approve Device Policy  Allow  Denied

Publish/Unpublish Policy  Allow  Denied

Change Priority  Allow  Denied

### App Management Action Permissions

Review/Approve App  Allow  Denied

Publish App  Allow  Denied

Review/Approve Policy  Allow  Denied

Publish Policy  Allow  Denied

### Common Permissions

Dashboards  Allow  Denied

Geo and Time Fences  Allow  Denied

Users & Groups  Allow  Denied

Permission Sets  Allow  Denied

AD Settings  Allow  Denied

Branding  Allow  Denied

Log Levels  Allow  Denied

Reset Password  Allow  Denied

Exchange Settings  Allow  Denied

Admin Email Settings  Allow  Denied

Enterprise Resources  Allow  Denied

System Status  Allow  Denied

© 2017 by Kony, Inc. All rights reserved

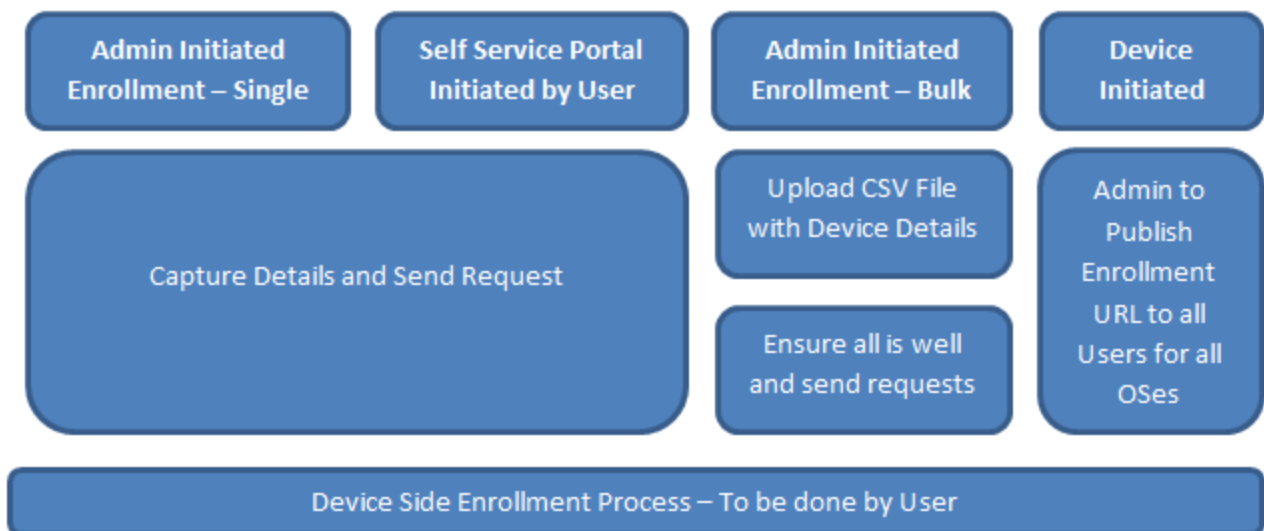
## 8. Enroll Devices

Based on the choices made in Enrollment Settings, devices can be enrolled in multiple ways.

Devices can only be enrolled with Users that are already part of the system - either local Users or Users imported from AD. Devices cannot be enrolled against Users who are not part of the system.

### 8.1 Enrollment Process

The process can be Admin initiated - from the **Device Enrollment** page or Self Service portal initiated from the **Devices** page. While enrolling devices the User ID must be selected, which auto-populates from respective field. Administrator must provide all other fields required.



Enrollment requests are sent to the User's email account. A typical enrollment request contains a set of instructions and a URL to be accessed from the device. The User must access the URL from the device to complete the enrollment process.

### 8.2 Platform Specifics

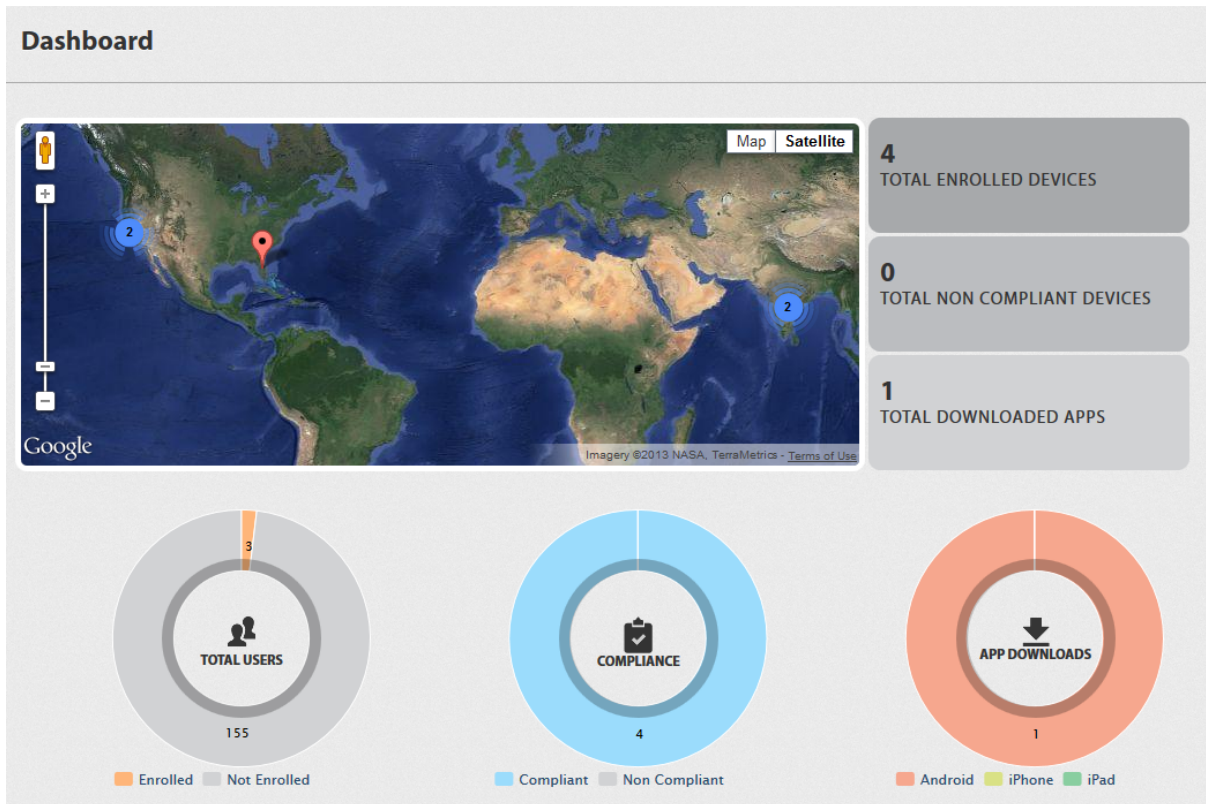
- For iOS and Android, all the three mechanisms are possible.
- For Windows 6.x devices, Device initiated is not possible.

- For Windows Phone 8, only Device Initiated Enrollment is possible. The URL to which the device must connect for the enrollment process must be provided to all the Users through email or any other means.
- For BlackBerry devices, enrollment only happens on the BES. The EMM Server will automatically sync these devices from BES, provided BES integration has been defined.



## 8.3 Monitoring the System

A dashboard is provided to for Administrators and others to view the salient activities of device users.



### 8.3.1 Location Distribution

The dashboard provides a quick snapshot of the devices and their distribution across the world. The Admin can quickly get to a pinpointed device, if its location seems off. The Admin can get to the device details to learn more about it and take several actions on the same.

#### Salient Metrics

- **Total Enrolled Devices:** This shows the total enrolled devices. This has a drilldown to all the devices enrolled.

- **Total Non-Compliant Devices:** This shows the number of devices that are currently out of compliance. This has a drilldown to the Device list with non-compliance devices.
- **Total Downloaded Apps:** This shows the total number of app downloads from the system. This includes all enterprise apps and any others pushed through the system.
- **Snapshots:** Charts display the company's current usage of EMM. Clicking on these charts brings up graph reports that can be exported to various formats.
  - **User Device Enrollment Summary:** A graph is shown with the Total Users on the system and a split of how many have enrolled their devices versus not.
  - **Compliance Summary:** A graph shows the total enrolled devices and a split of how many devices are within compliance versus not
  - **App Downloads:** A graph shows the split of app downloads based on the different platforms.

## 9. Devices

The list of all the devices is shown here. The list can be filtered to view only the devices that are of interest. The policies applied to the device can be viewed.

Devices								
Device Name ▼	Status	Device Owner	Ownership	Compliance	OS	Last Check-in	Date Enrolled	Policy Applied
<input type="text" value="Search Device Name"/>	All	<input type="text" value="Search Device Ow"/>	All	All	<input type="text" value="Search OS"/>	All	All	
<a href="#">test 4G iPhone</a>	Enrolled	test1	Employee	Non Compliant	🍏 iOS 6.1.3	31 Dec, 2013 20:30:08 EST	30 Dec, 2013 07:01:28 EST	<a href="#">View Policy</a>
<a href="#">emmqa21 9810</a>	Enrolled	emmqa21	Corporate	NA	📱 BB 7.0.0.261	31 Dec, 2013 08:36:10 EST	31 Dec, 2013 08:36:06 EST	
Sunil GT-P7510	Control Removed	Sunil Meda	Employee	Non Compliant	🤖 Android 4.0.4	31 Dec, 2013 08:11:16 EST	30 Dec, 2013 08:16:48 EST	

Displaying 1 - 10 of 25 - Display 10

Previous Page {1/3} Next

The details about the device can also be viewed, and several actions can be taken.

## 10. Device Details

From the **Device Management** section, click the **Device List** from the left panel. The Device List page is displayed. The page displays a list of devices. Click on any of the device name's to view its details.

**Device Details** Force Check-in

[Device List](#) > Device Details

**automationFn 5G iPhone**  
5G iPhone | iOS 6.1.4  
Device Status : Enrolled  
Serial Number: C37JNFZXTWD  
Last sync: 31 Oct, 2013 20:44:46 IST

Lock Device Clear Passcode  
Wipe Actions Block Email  
Remove App Data

Overview Messages Locate App Monitor Asset Properties

**Ownership** Corporate

**Manufacturer** Apple

**Home Carrier** Data Unavailable

**Current Carrier** Data Unavailable

**UDID** c6e224a5d25c103744182d5b95e74155b223a856

**Device model** 5G iPhone

**IMEI Number** 013407008318521

**SIM ID** Data Unavailable

**Storage Used** 3.30GB / 13.46GB

**Storage Available** 10.15GB / 13.46GB

**Phone Number** Data Unavailable

**Hardware Encryption** Unencrypted

**MDM Policy** View Policy

**Compliance State** Compliant

**Device Roaming Status** Non Roaming

Save & Exit Save & Continue Cancel

There are several details about the device that administrators have access to:

- The device internal statistics
- The Kony EMM system messages received by it

- The apps installed on the device
  - The admin can choose to remove apps if necessary
- The current and the last 5 locations of the device.

The Admin can also take required actions such as:

- Lock Device
- Reset Passcode
- Wipe Device (both corporate data only or factory reset)
- Block Email

## 10.1 View Device Policy

The Admin has the ability to view the policies applied on any device. In this view, the admin can see the policies that are resolved on the server side versus those acknowledged by the device.

The Administrator can also choose to look at more details. This shows the inheritance of the policies from their respective Device Sets. All the Device Sets, the device is part of are shown along with all the policies applied to the same with their priorities. Based on the priorities, the server resolves certain policies.

In some occasions, there can be a difference between the server resolved policies and device acknowledged policies. It is recommended to wait till the next heartbeat to see, if this synchronization occurs.

If absolutely essential, doing a Force Check-in enables the device to interact with the server and all the details are passed along then. It updates the device status to the Admin. Any discrepancies should be resolved.

**View Device Policy** x

Policy Type	Server Resolved Policy	Device Acknowledged Policy
Passcode Policy	-	-
Device Restrictions	-	-
Email and Calendar	Anupam Exchsng	Anupam Exchsng
Network	-	-
Certificate Distribution	-	-
Web Clips	-	-
Compliance Actions	-	-
App Policy	-	-
Policy Version	15	15

More DetailsOK

## 11. Device Policy Creation

In the **Device Policy** page of the **Device Management** section, the Admin can add a New Policy.



The Administrator must choose the policy type and provide a description for the same. Once it is done, the Admin is required to provide all the details with respect to that policy. There are eight types of device policies that can be created. All the policy types are not available for all device platforms (OSes). Within each policy, every supported platform is displayed in a tab.

For **Network** and **Certificate Distribution** policies, enterprise resources (for example, Wi-Fi networks, VPN networks) must be created before they can be added to these policies. These are created in the Enterprise Resources area under Settings.

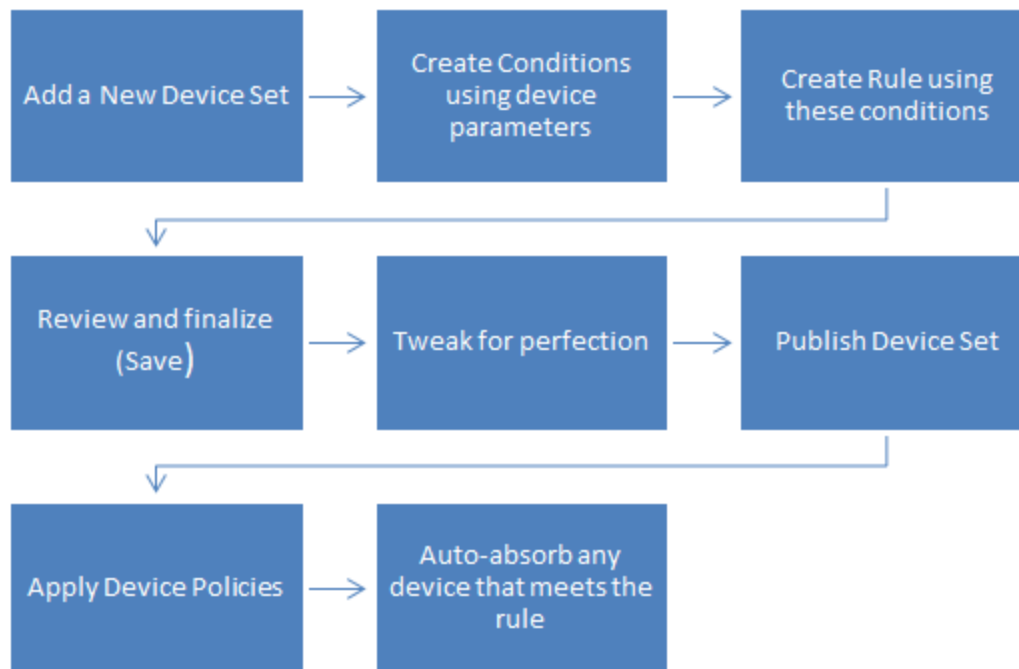
Once a policy is created, it must be activated and published. By default, the newly created policies are in an **Unpublished** state. To publish the policy, the state must be Active.

After publication, the policy is given the last priority in its type. Once published, policies are made available to be applied to the Device Sets.

## 12. Device Set Creation

Handling devices one at a time is a very tedious activity. To make the process easier, devices are organized into Device Sets using certain rules. These Device Sets are self-organizing and dynamic as all the devices as part of a Device Set adhere to the same set of rules.

Device Sets can be created from the Device Set page in Device Management.



### 12.1 Device Set Conditions

The most common sets are created by default. For custom sets that are needed, several parameters are available to customize conditions. The admin must choose these parameters and their values to build conditions. To create Device Sets, there could be one or more conditions.

Some sample conditions can include:

- Device OS = Android
- Device ownership = Corporate



## 12.2 Device Set Rule Definition

These conditions are then used to create rules. Operators such as AND, OR, NOT and Parentheses are available to build expressions (rules). The expressions are interpreted from right to left. In the order of precedence, it is parentheses “()”, NOT, AND, OR. The laws of Boolean algebra for basic operations apply.

The final expression created is the one rule that all devices must adhere to be part of the Device Set.

The Admin can search for all devices that satisfy the rule and modify both the conditions and rules to arrive at the final devices required to be part of the Device Set.

## 12.3 Device Set Publication

After creation, the default state of the Device Set is **Unpublished**. If the Admin is satisfied with the conditions and rules used to define the Device Set, they may choose to publish the same. Just like Device Policies, the state must be Active before the status can be changed to Published.

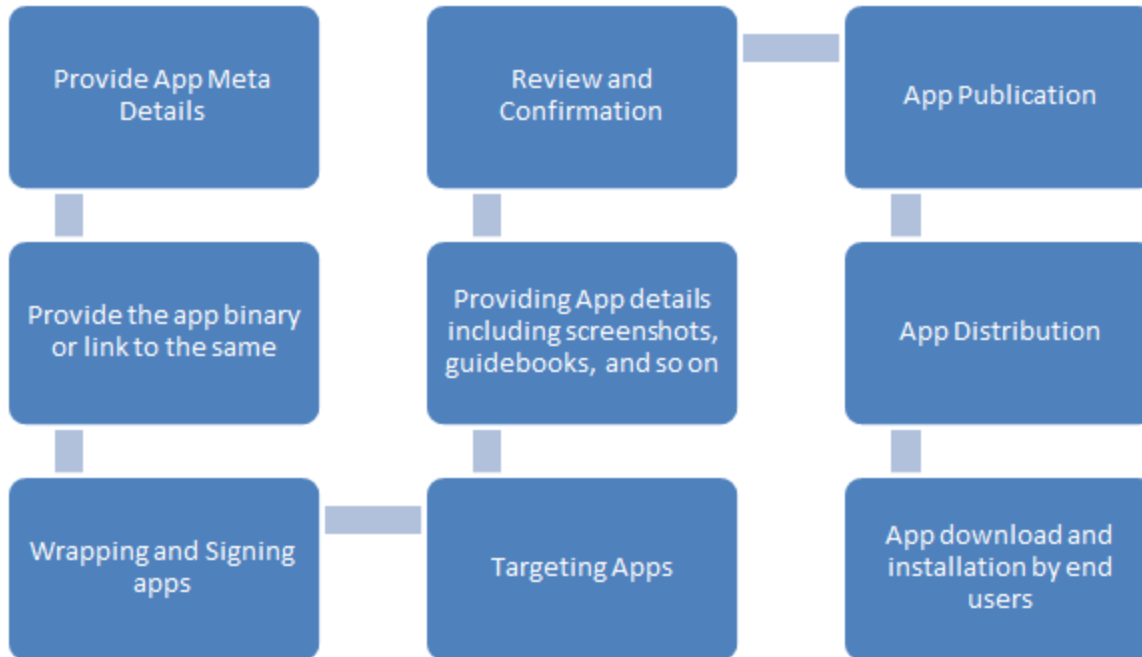
## 12.4 Post Publication

After it is published, devices that satisfy the rules of the Device Set automatically become a part of it. Therefore, devices can be part of multiple Device Sets.

Only after Device Set publication, policies can be assigned to the same. All devices that are part of the Device Set have the same set of policies applied to them.

If a device is part of multiple Device Sets, multiple policies may be applied to the device. To resolve this issue there are priorities for each policy. The policy with the highest priority wins and applied to the device.

## 13. App Creation and Upload to Enterprise Store



App Creation involves the following steps:

- Providing App meta details (such as name, version and description)
- Providing the app binary or link to the same
- Wrapping and or Signing Apps
- Targeting Apps
- Providing App details including screenshots, guidebooks, description and so on
- Review and Confirmation

Once the admin performs all these steps for an app, it is ready to undergo the publication process. Every app has a workflow state and publication status. By default, the workflow state is **Draft** and the publication status is **Unpublished**. The workflow state is shifted until it reaches **Approved**. At this point the publication status can be made **Published**.

Upon successfully publishing an app, it automatically appears in the Enterprise Store. It also appears in the store portion of the device agent for Users to whom it is targeted. An email notification is automatically sent to the user alerting them that the new app is available.

Users may choose to download and install any apps visible to them in their Store. If an app is flagged as mandatory for a user, it is automatically installed on the device the next time the user opens Enterprise App Store.

If you unpublish an app, it is removed from the store and from all the devices it is present on.

## 14. App Policy Creation

App Policies govern the behavior of an enterprise app. From the **Policies** page in the App Management section of the administrator console, the admin can add a new policy. The admin must provide the name and description of the app along with the policy configuration.



Once a policy is created, it must be published before it can be assigned to any app. All Policies have both State and Status. By default, when a policy is created, it is in **Draft** state and **Unpublished** status. The state must be approved before the policy can be published. With each change in the state and the status, a comment should be provided indicating the progress in the workflow.

After successful publishing, the policy is made available to apply on wrapped enterprise apps.

## 15. Generating Certificates

Before using the Kony Management Cloud EMM solution it is required to create and apply various Apple and Google resources. These resources allow the server access to the Apple mobile communication network, the Google mobile communication network, as well as Google Maps.

The resources you will create are:

- [Apple Enterprise Wild Card Distribution Certificate](#)
- [Apple Enterprise Wild Card Provisioning Profile](#)
- [Apple Application Manager \(Launchpad app\) Push Certificate](#)
- [Apple Application Manager \(Launchpad app\) Provisioning Profile](#)
- [Assigning App Resources in the Kony Management Cloud Administrator Console](#)
- [Apple Push Notification Certificate \(APNS\) for MDM](#)
- [Creating CSRs](#)
- [Android Certificates and Keys](#)
  - Android GCM Key with Sender ID
  - Google MAPSv2 Key
  - Android Key Store

### 15.1 Implications of renewing iOS certificates on Launchpad and Child apps

When you renew/re-create iOS certificates for any reasons, the following are the implications for the Launchpad and any Child apps in the Launchpad.

- **Wild Card Distribution Certificate:** When a certificate is generated again, wrapping will be initiated on the Launchpad and on Child apps.

If the wrapping is successful, a push message is sent to the user device to install the latest launchpad and its corresponding child apps on the device. The Launchpad and Child apps will be wrapped and signed with the latest Wild card distribution certificate.

If a child app is active, the user must save the data and logout from child app before the Launchpad upgrade.

- **Wild Card Provisioning Profile:** When a profile is generated again, wrapping will be initiated on the child app and no action is taken on Launchpad.

If the wrapping is successful, a push message is sent to the user device to install the corresponding latest app. The child app will be wrapped and signed with the new wild card provisioning profile certificate.

If a child app is active, the user must save the data and logout from child app before the Child app upgrade.

- **Push Certificate:** When a new push notification certificate is uploaded, wrapping is not initiated on either the Launchpad or the Child app.
- **Launchpad Provisioning Profile:** When a profile is generated again, wrapping (sign only) is initiated on the Launchpad. No action will be taken on the child app.

If the wrapping is successful, a push message is sent to the user device to install the Launchpad on the device. The Launchpad will be signed with the latest Launchpad Provisioning profile.

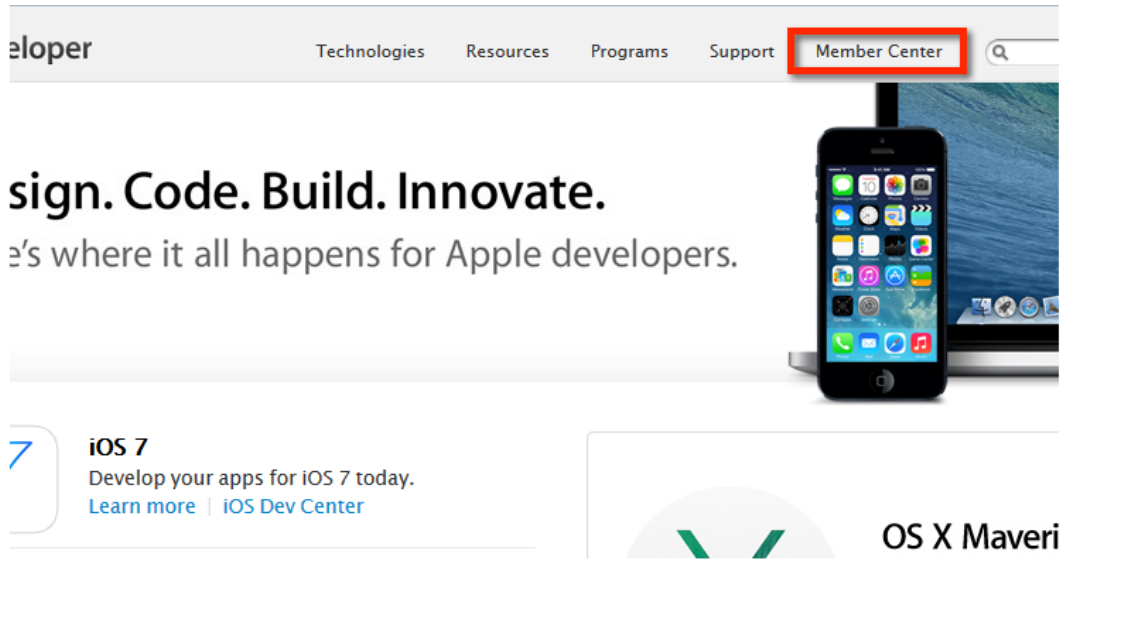
While upgrading launchpad, active child app will be moved to the background and can be launched after installation.

- **MDM APNS Certificate:** When a MDM APNS certificate is uploaded to the EMM console through Device Settings > Communication Settings page, no action is taken on the Launchpad the its child apps.

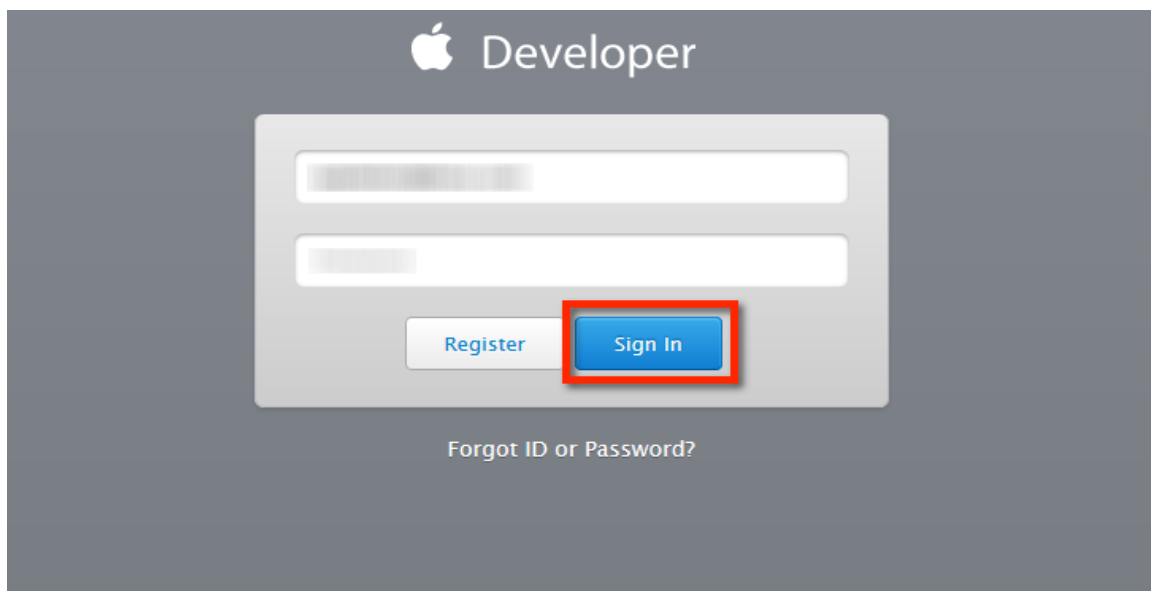
## 15.2 Creating the Apple Enterprise Wild Card Distribution Certificate

To create Apple Enterprise Wild Card Distribution Certificate, follow these steps:

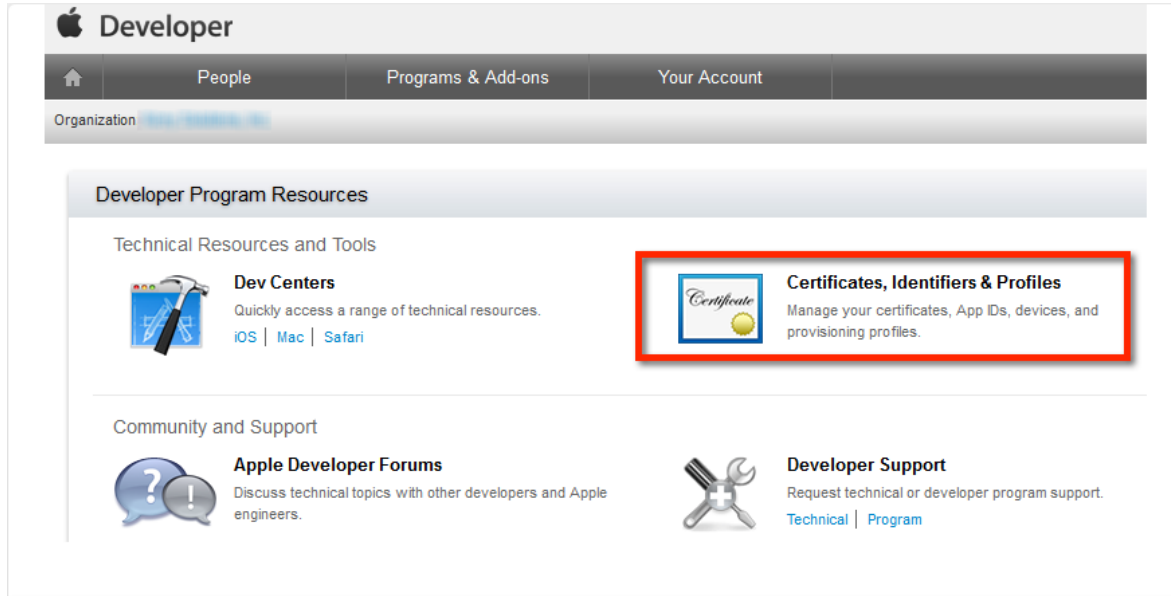
1. In a browser, go to <https://developer.apple.com>, and click **Member Center**.



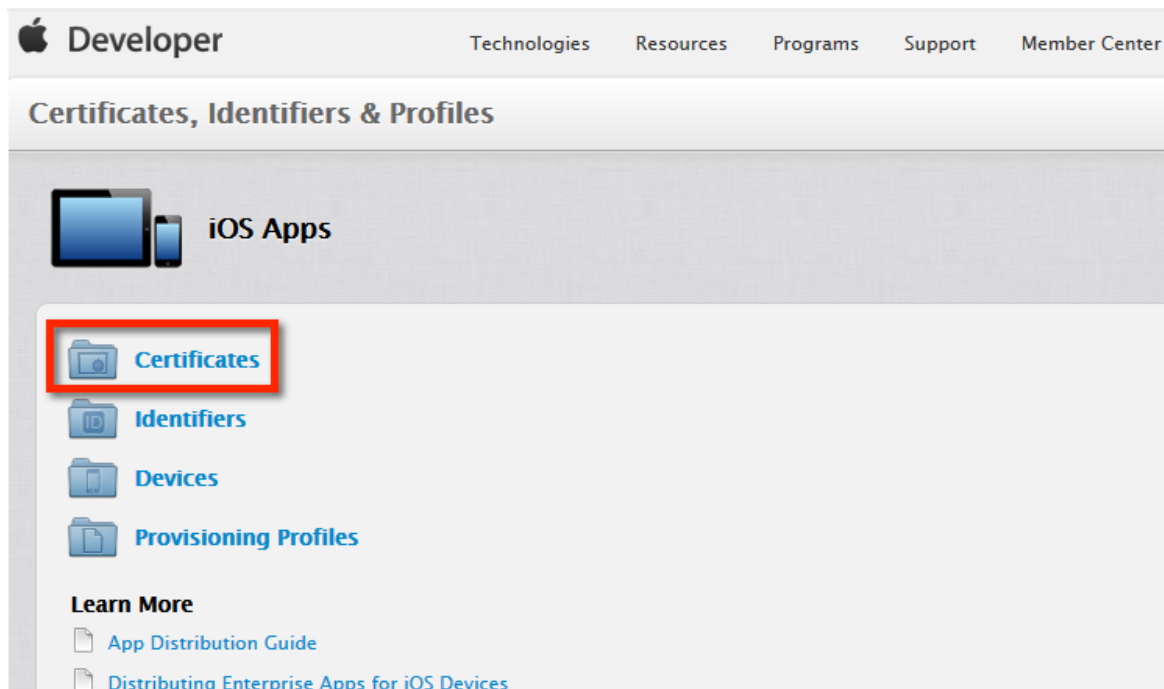
2. Enter your [Apple Developer Enterprise Program](#) credentials and click **Sign in**. If you do not have an account already, you need to create one. Developer Program Resources page appears.



3. Click **Certificates, Identifiers & Profiles** icon. The Certificates, Identifiers & Profiles page appears.

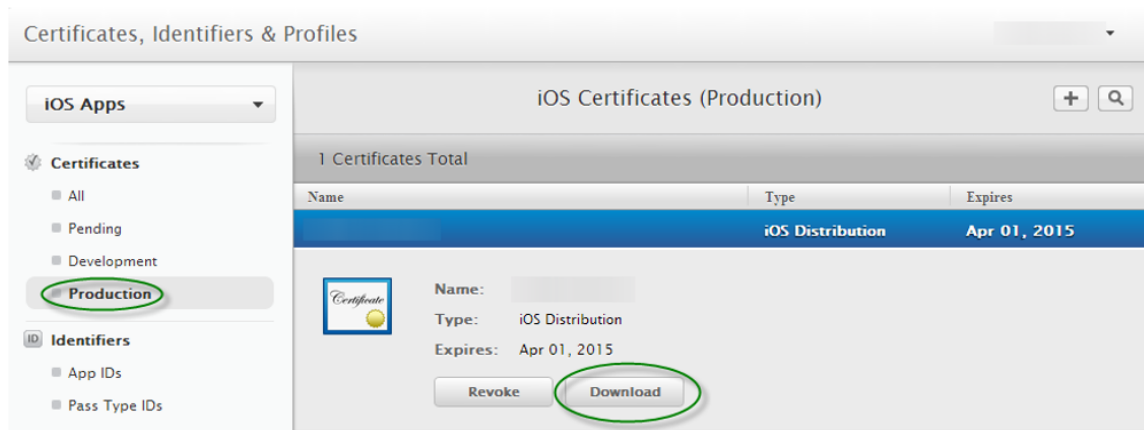


4. Click **Certificates**.

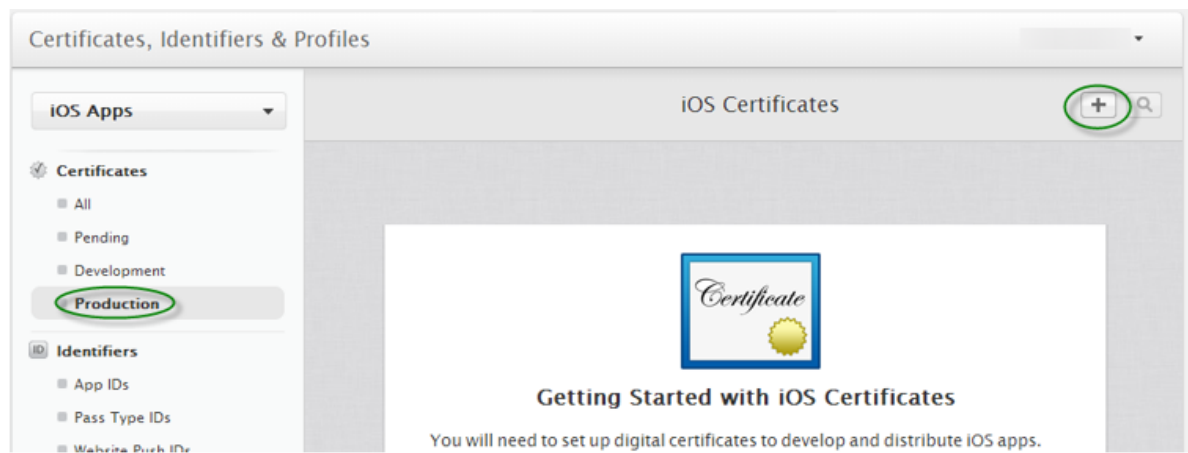




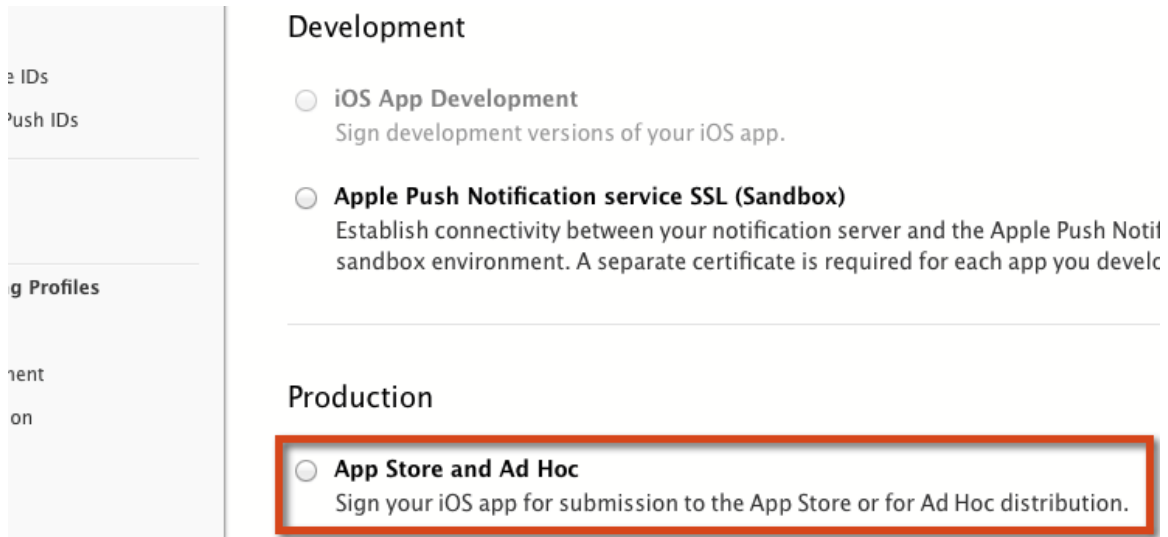
5. Verify whether there is an existing iOS Distribution Type certificate under the **Production** in the left column. If there is, highlight it choose **Download** to save the `ios_distribution.cer` into your local system.



6. If you just downloaded an existing iOS Distribution certificate, skip to [Step 16](#) below. If no iOS Distribution certificate exists, you will need to create one and then download it. To create a new certificate, continue to Step 7 below.
7. Click **Production** in the left column, and then the **+** icon next to iOS Certificates label to add a new certificate. Development window appears in the right column.



8. Choose the **App Store and Ad Hoc** option under Production. Instructions to generate a certificate appear.



9. Follow the displayed instructions to generate a **CSR** file and click the **Continue** button.

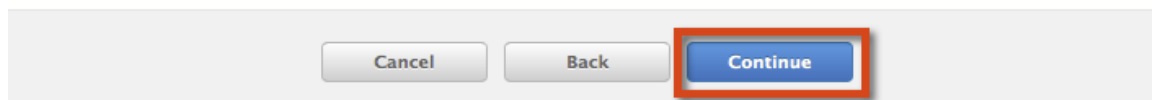
To manually generate a Certificate, you need a Certificate Signing Request (CSR) file from your Mac. To create a CSR file, follow the instructions below to create one using Keychain Access.

#### Create a CSR file.

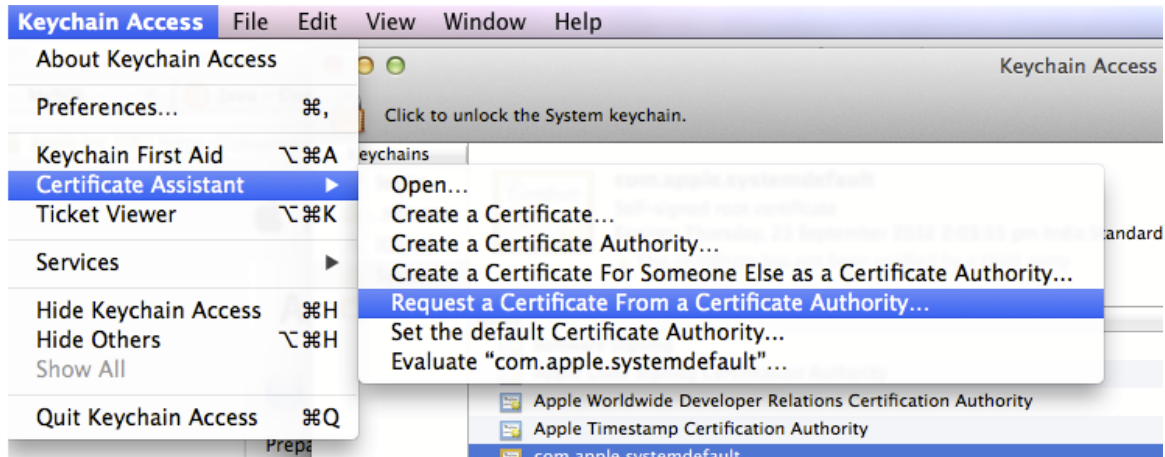
In the Applications folder on your Mac, open the Utilities folder and launch Keychain Access.

Within the Keychain Access drop down menu, select Keychain Access > Certificate Assistant > Request a Certificate from a Certificate Authority.

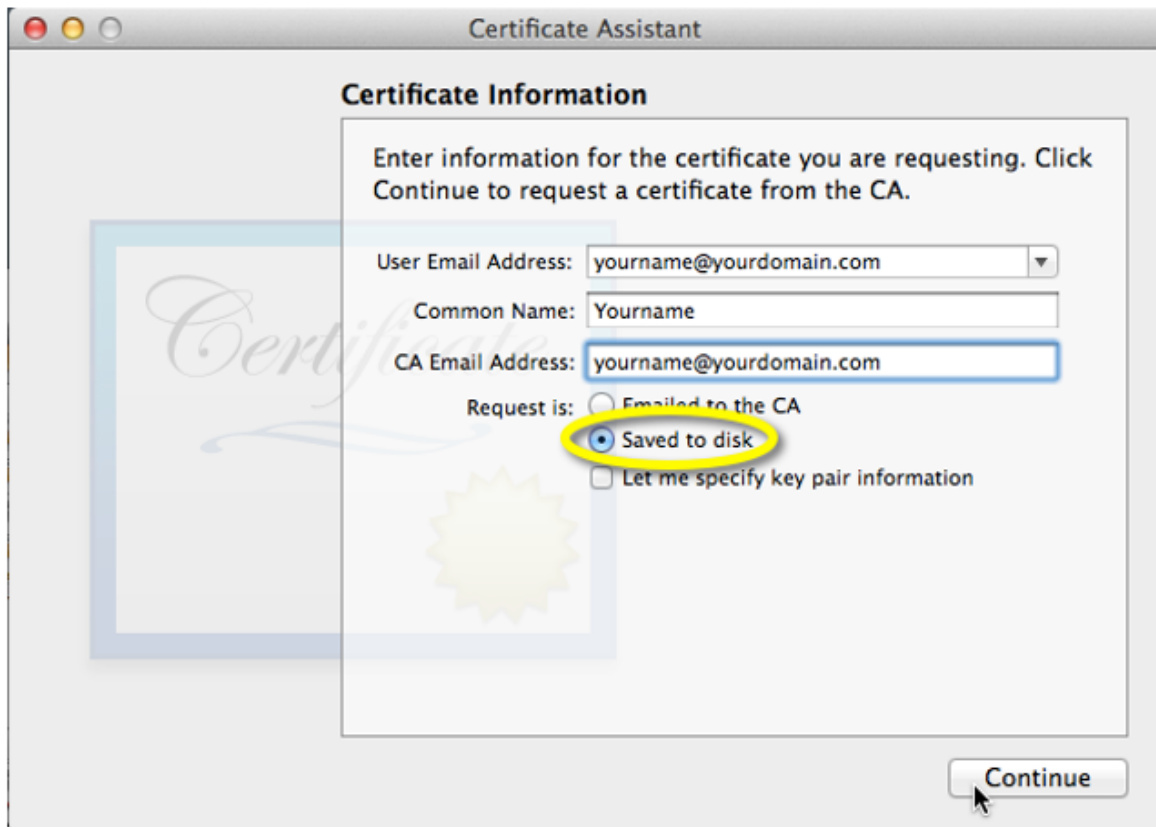
- In the Certificate Information window, enter the following information:
  - In the User Email Address field, enter your email address.
  - In the Common Name field, create a name for your private key (e.g., John Doe Dev Key).
  - The CA Email Address field should be left empty.
  - In the "Request is" group, select the "Saved to disk" option.
- Click Continue within Keychain Access to complete the CSR generating process.



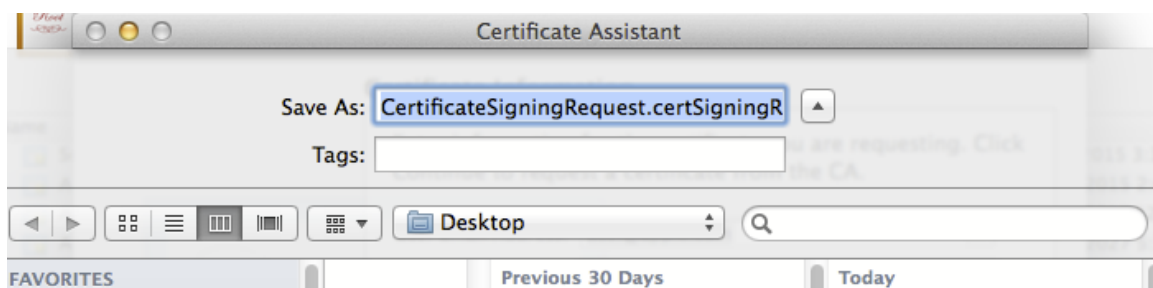
10. In the **Keychain Access** application on your Mac, under the **Certificate Assistant** menu, choose the **Request a Certificate From a Certificate Authority** option. Certificate Assistant page appears.



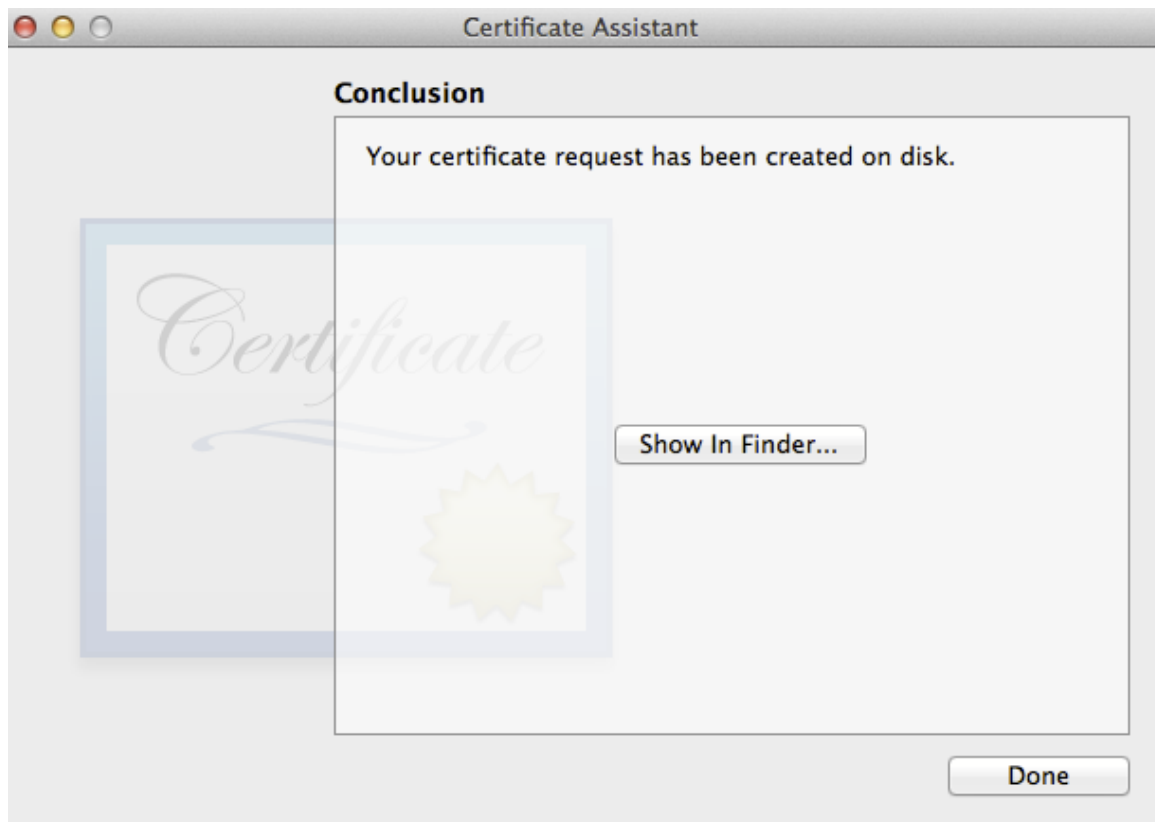
11. Fill out the displayed fields, choose **Save to disk** option, and then click **Continue**.



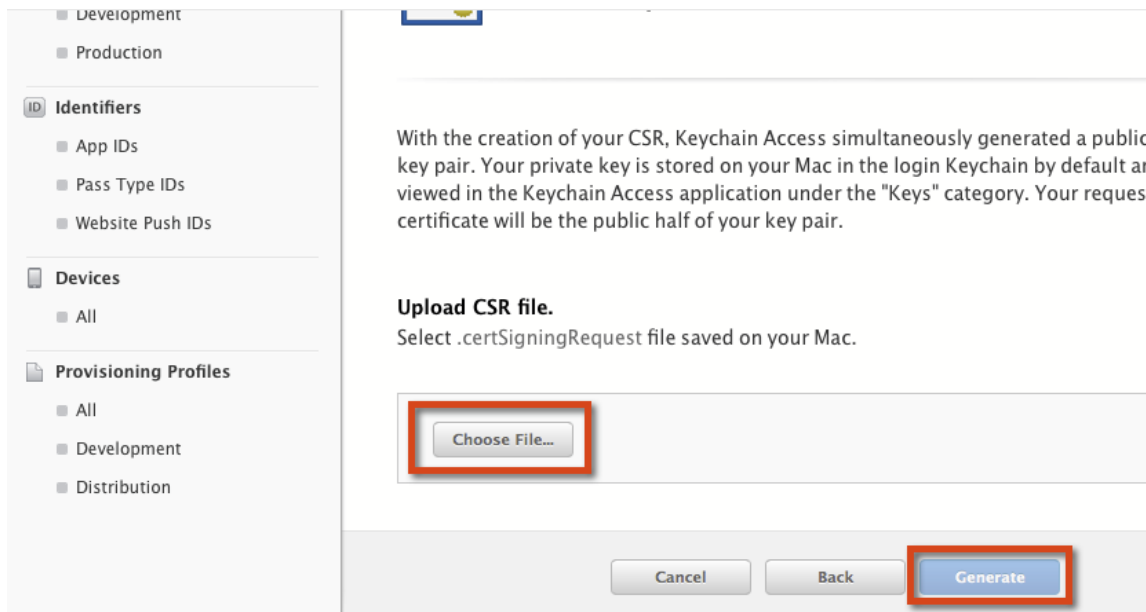
12. Save the **Certificate Signing Request (CSR)** to your local machine.



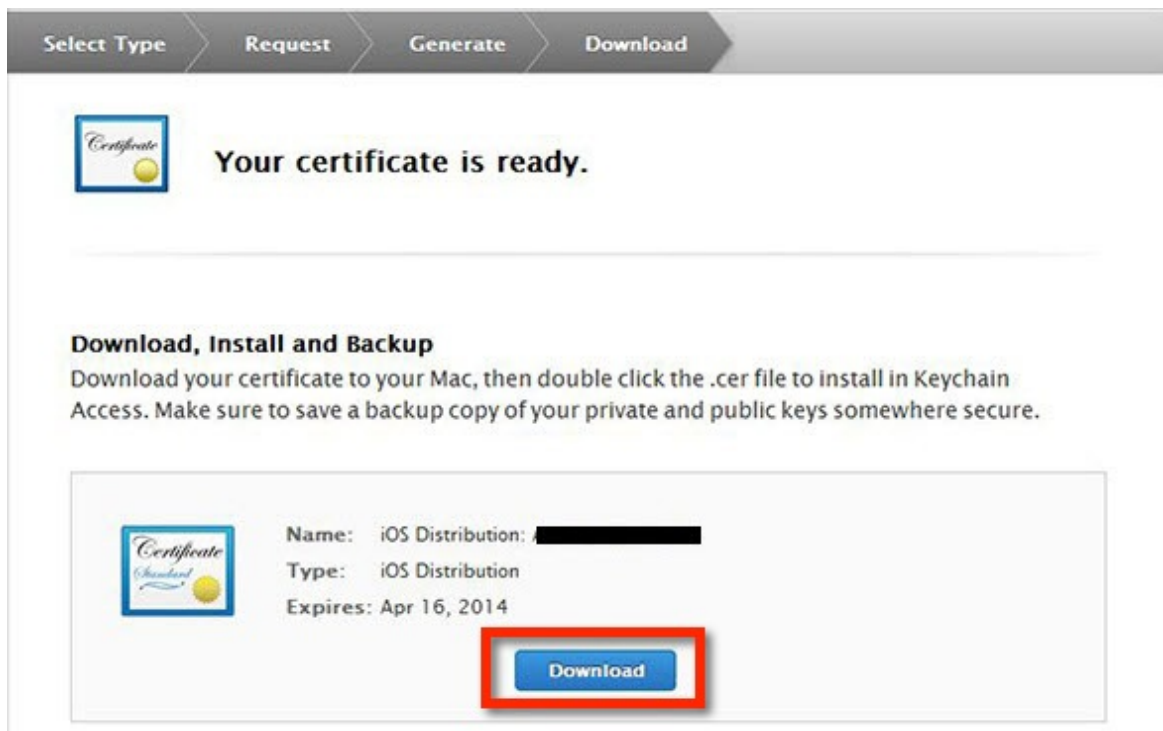
13. Click **Done**. You have now generated your CSR.



14. Back in the [developer.apple.com](https://developer.apple.com) site, click **Choose File** and upload your CSR file, and then click **Generate** to continue.

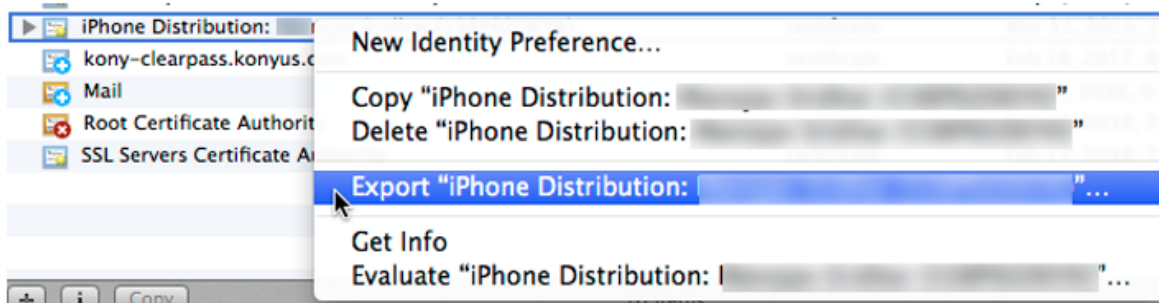


15. Click the **Download** button to download the **Distribution Certificate** you have just generated.

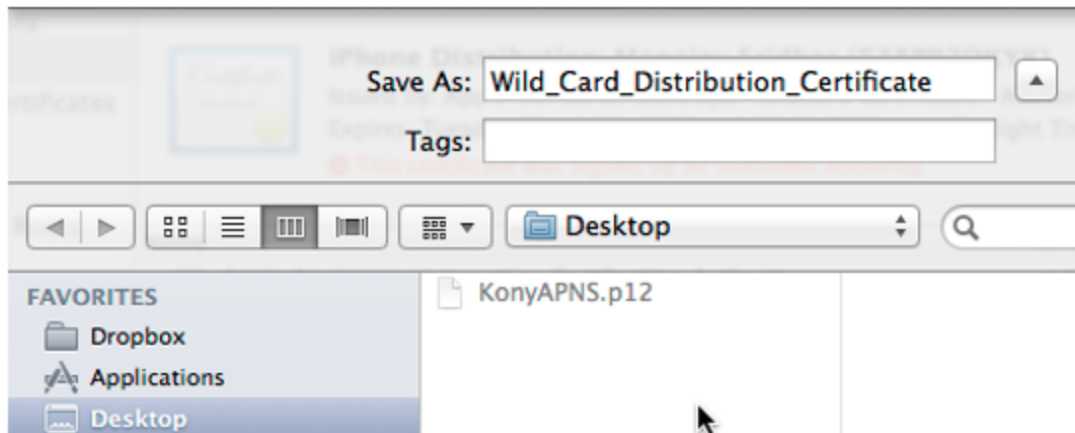


16. Double-click the downloaded **.cer** file to import it into the local Keychain.

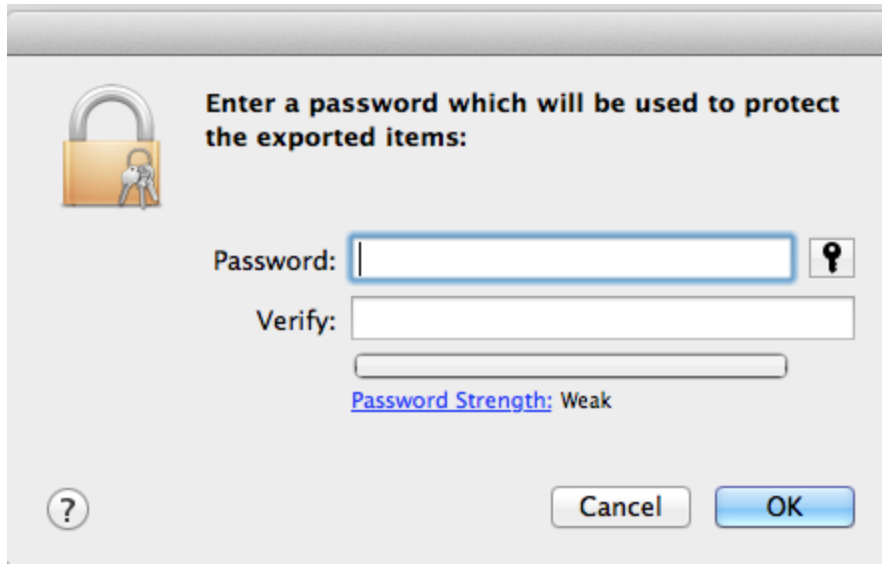
17. In the **Keychain Access** application on your Mac, select the iPhone Distribution certificate you just imported (a tip to identify it is to look at the expiration date; it will be one year exactly from today), right-click, and select **Export**.



18. Save the exported certificate in a secure location. It is recommended to name the saved certificate **Wild\_Card\_Distribution\_Certificate**.



19. You will be required to provide a certificate password. Make a note of this password for future use with this certificate.



20. Your Apple Wild Card Distribution Certificate is now complete. Store this file in a safe place to be used during your Management Cloud initial configuration. You can now delete the CSR saved locally, the `ios_distribution.cer` saved locally, and the iPhone Distribution entry imported to your Keychain.

### 15.3 Recreate Apple Wild Card Distribution Certificate

You cannot renew a certificate. You can only create a new certificate with the old certificate details. To recreate a certificate,

1. Go to your Apple developer [member center](#) in an internet browser.
2. In the Certificates, Identifiers & Profiles section, select **Certificates**.
3. Under the certificates section, select **Production**. All existing certificates appear.
4. Select the Certificate you want to recreate. If the certificate has expired, the **Revoke** button will be active.
5. Click **Revoke**. The certificate will be revoked.

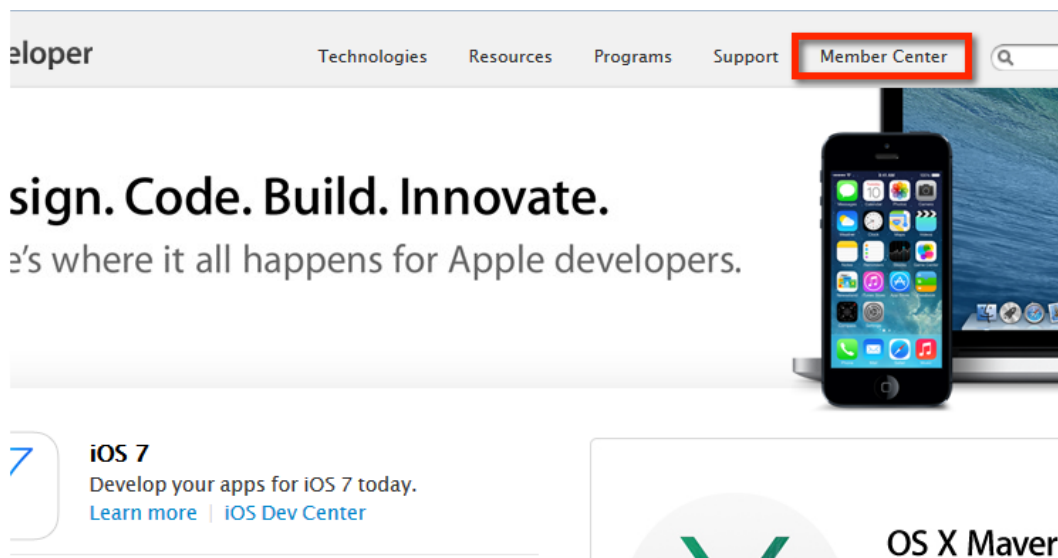


6. Once the certificate is revoked, create a new certificate with the details of the old expired certificate as explained in the previous section.

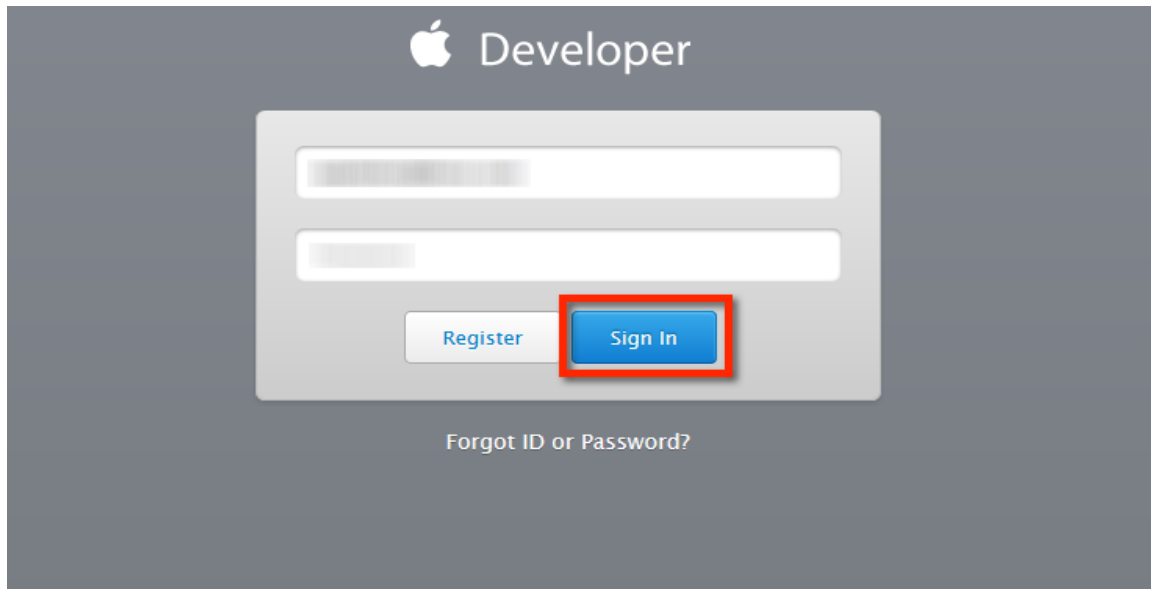
## 15.4 Creating the Apple Enterprise Wild Card Provisioning Profile

To create Apple Enterprise Wild Card Provisioning Profiles, follow these steps:

1. In a browser, go to <https://developer.apple.com>, and click **Member Center**.

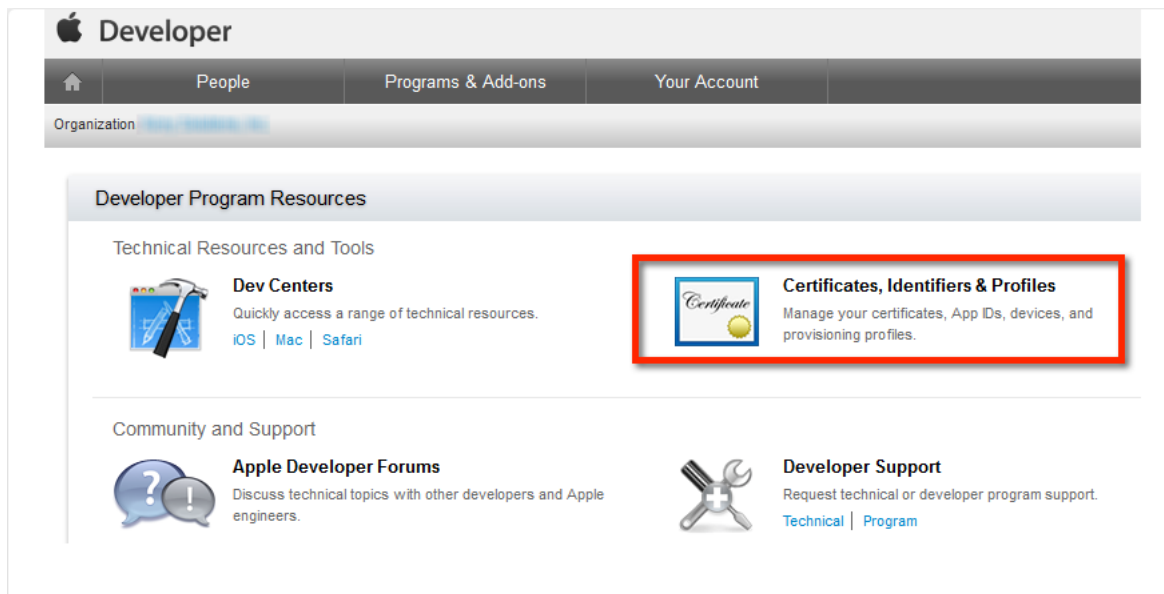


2. Enter your [Apple Developer Enterprise Program](#) credentials and click **Sign in**.

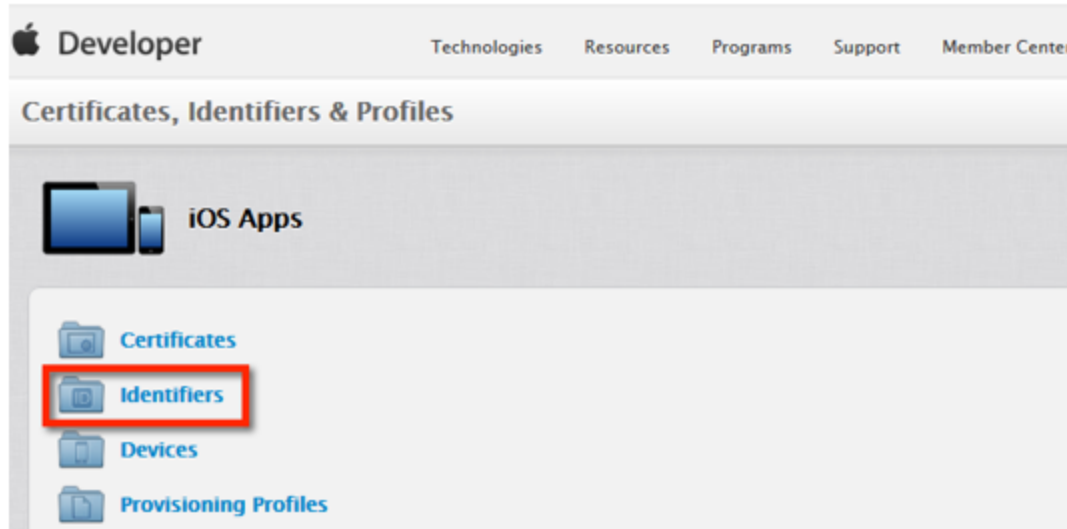


Apple Developer page appears.

3. Click the **Certificates, Identifiers & Profiles** icon.



4. Click **Identifiers**.



5. Click **App IDs** in the left column and then the + icon next to iOS App IDs label to create a new App ID.



6. Provide a description (**Wild Card Provisioning Profile** is the recommended description).

## App ID Description

Name:

You cannot use special characters such as @, &, \*, ', \*

7. Choose the option **Wildcard App ID** and provide this **Bundle ID** (substitute your company name in the middle segment instead of 'companyname'): **com.companyname.\***

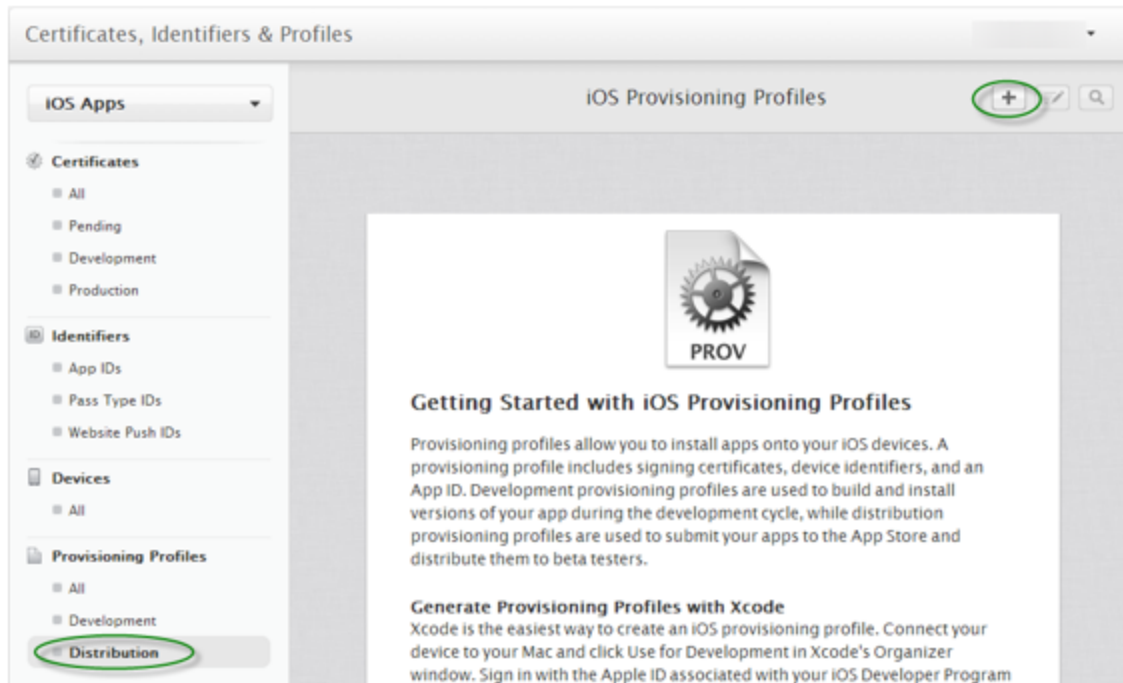
**Wildcard App ID**

This allows you to use a single App ID to match multiple apps. To create a wildcard App ID, enter an asterisk (\*) as the last digit in the Bundle ID field.

Bundle ID:

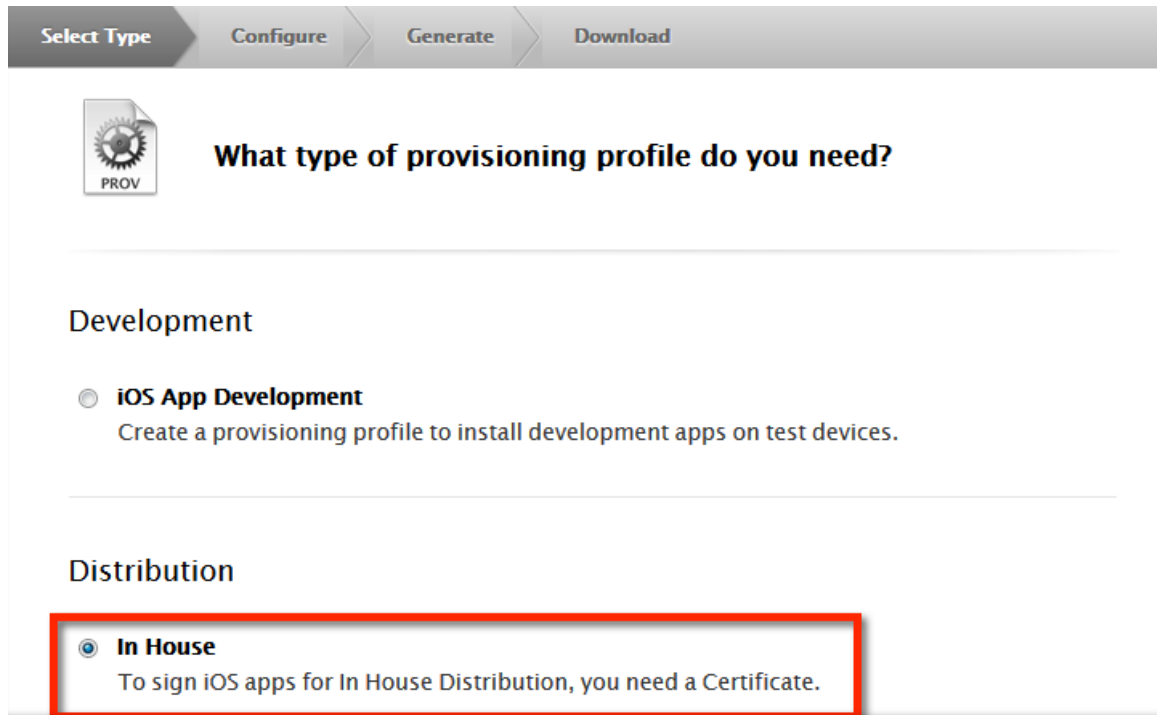
Example: com.domainname.\*

8. Review details and click the **Submit** button, and then click **Done**.
9. Click **Distribution** under the **Provisioning Profiles** menu, and then click the + icon next to iOS Provisioning Profiles label to create a Distribution Provisioning profile.



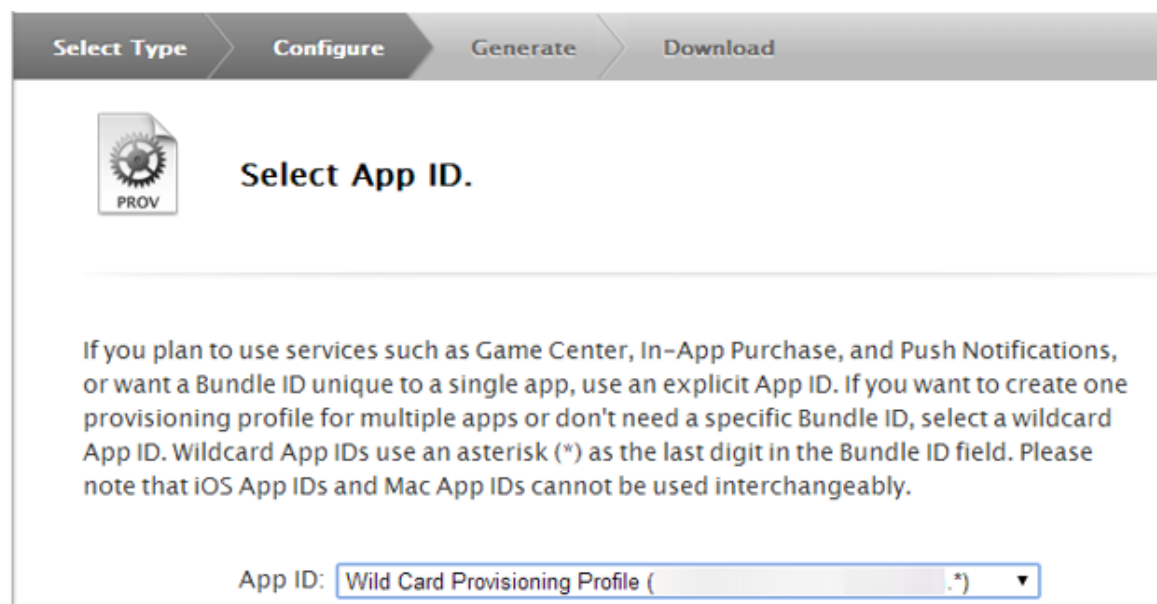
What type of provisioning profile do you need? Page appears.

10. Select **In House** under the **Distribution** heading. Select App ID page appears.



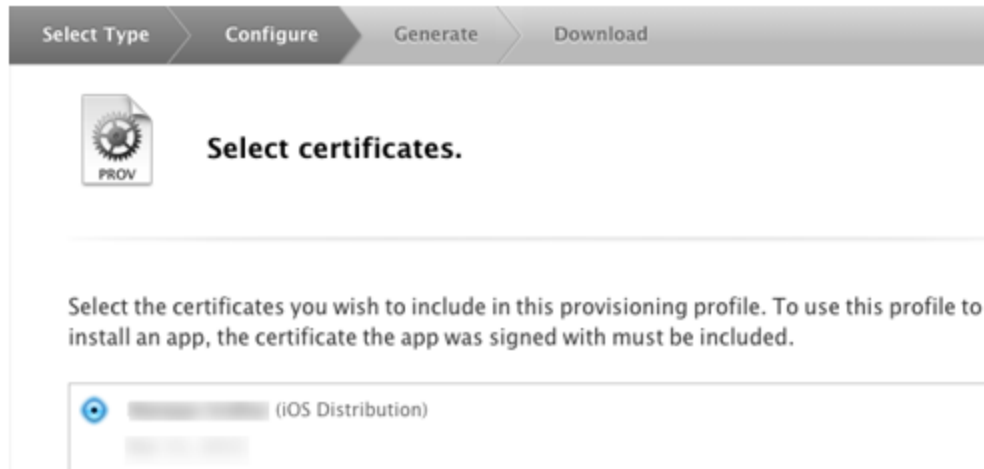
The screenshot shows a progress bar at the top with four steps: 'Select Type', 'Configure', 'Generate', and 'Download'. Below the progress bar is a document icon with a gear and the word 'PROV'. The main heading is 'What type of provisioning profile do you need?'. There are two sections: 'Development' and 'Distribution'. Under 'Development', there is a radio button for 'iOS App Development' with the description 'Create a provisioning profile to install development apps on test devices.' Under 'Distribution', there is a radio button for 'In House' which is selected and highlighted with a red box. The description for 'In House' is 'To sign iOS apps for In House Distribution, you need a Certificate.'

11. Select the **App ID** created above from the drop-down menu and click **Continue**.



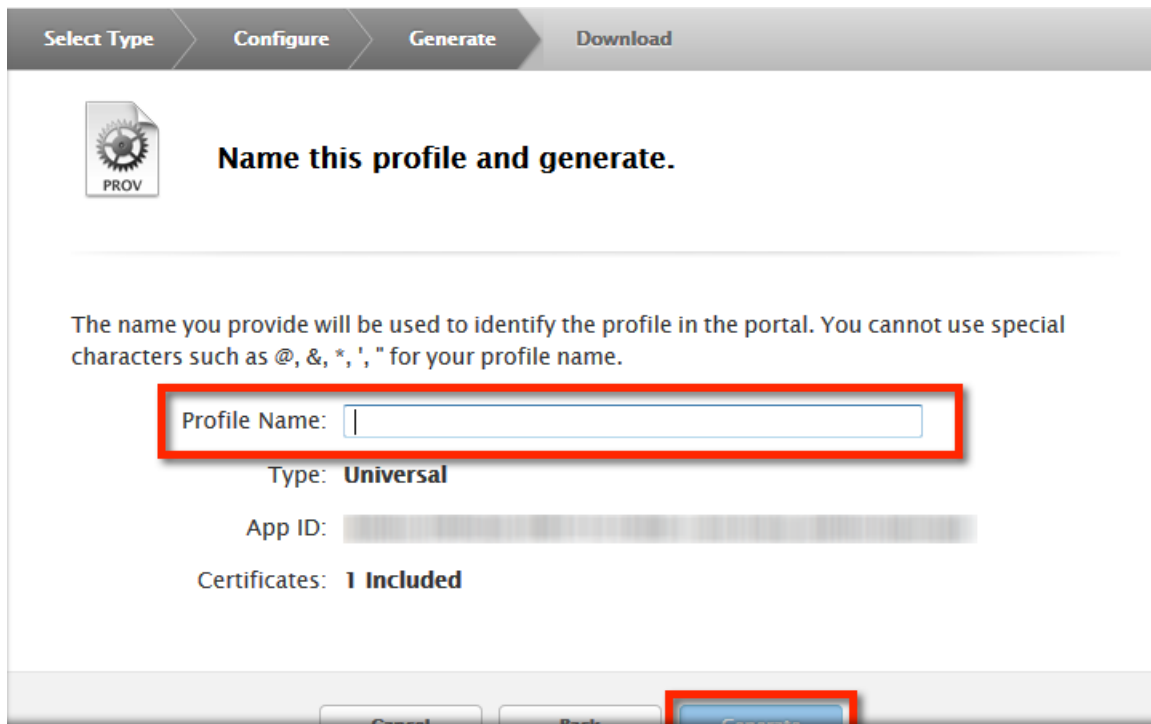
The screenshot shows the same progress bar as the previous screen. Below it is the 'PROV' icon and the heading 'Select App ID.'. A paragraph of text explains the use of explicit vs. wildcard App IDs. At the bottom, there is a label 'App ID:' followed by a drop-down menu. The selected option in the menu is 'Wild Card Provisioning Profile (.\*)'.

12. Choose the iOS Distribution certificate from the list to include in this provisioning profile and then choose **Continue**. If there are two certificates in this list, make note of the expiration date to ensure you choose the same certificate when you create another Provisioning Profile in a later step.




Name this profile and generate page appears.

13. Enter a Name for this **Profile** (**Wild Card Provisioning Profile** is the recommended Profile name), and then click **Generate** to continue.



Select Type > Configure > **Generate** > Download

 **Name this profile and generate.**

The name you provide will be used to identify the profile in the portal. You cannot use special characters such as @, &, \*, ', " for your profile name.

Profile Name:

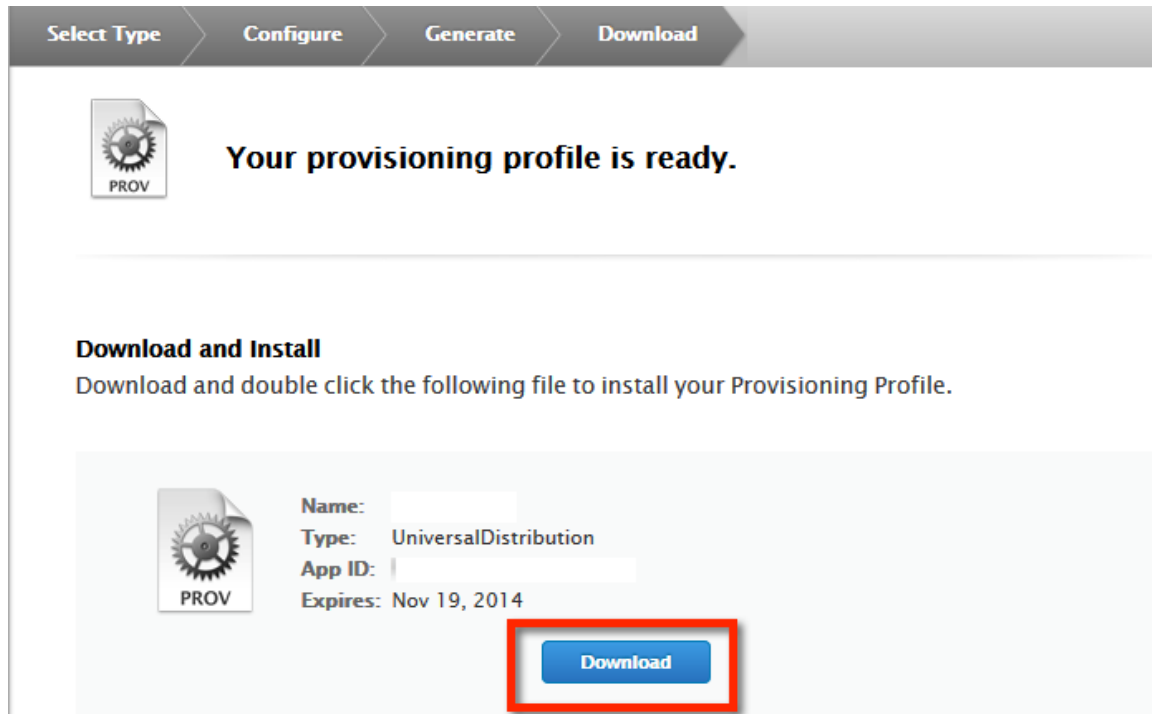
Type: **Universal**

App ID: [blurred]

Certificates: **1 Included**

14. Click the **Download** button to download the Wild Card Provisioning Profile.





15. Your Apple Wild Card Provisioning Profile is now complete. Store this file in a safe place to be used during your Management Cloud initial configuration.

## 15.5 Recreate Apple Wild Card Provisioning Profile

You cannot renew a provisioning profile. You can renew an expired provisioning profile by editing and re-generating it. To recreate a provisioning profile,

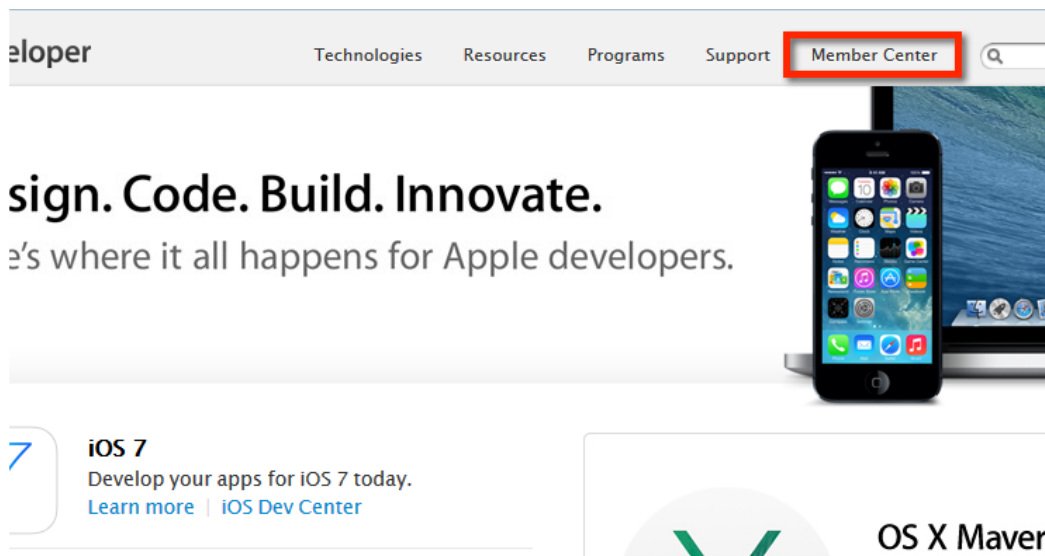
1. Go to your Apple developer [member center](#) in an internet browser.
2. In the Certificates, Identifiers & Profiles section, select **Provisioning Profiles**.
3. Under the Provisioning Profiles, select **All**, **Development** or **Distribution**. All existing provisioning profiles appear.
4. Select the provisioning profile you want to modify and click **Edit**. The Edit Provisioning Profile page appears.

5. Select the app id (com.CompanyName.\*) and the certificate the provisioning profile corresponds to, and click **Generate**. The Add iOS Provisioning Profile page appears.
6. Click **Download**. The provisioning profile will be downloaded.

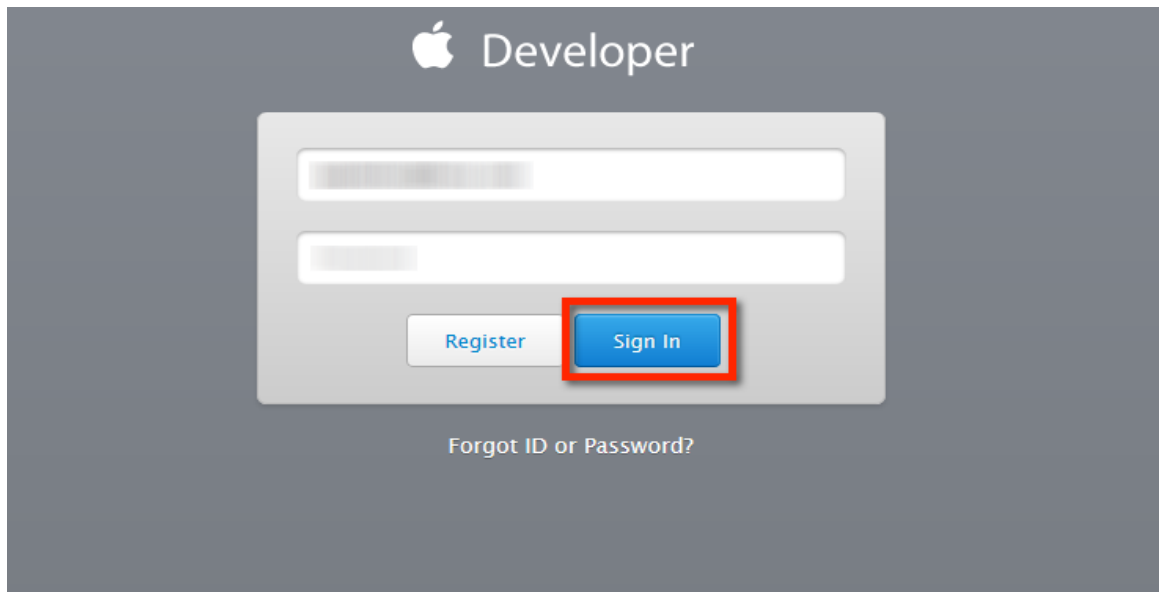
## 15.6 Creating the Apple Application Manager (Launchpad app) Push Certificate

To create Apple Application Manager (Launchpad app) Push Certificate, follow these steps:

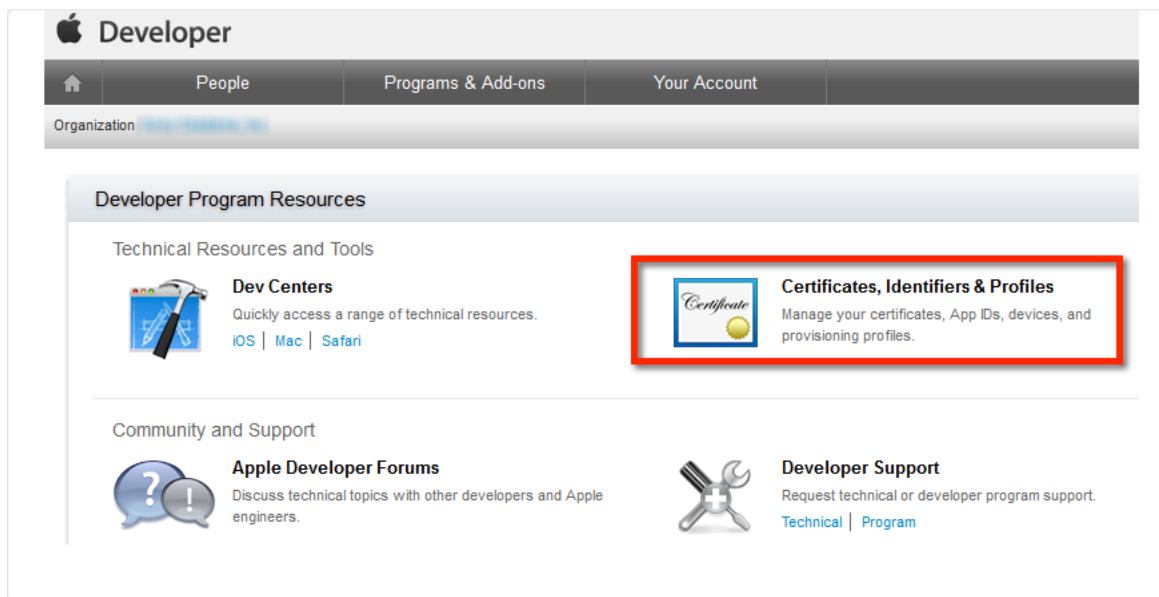
1. In a browser, go to <https://developer.apple.com> and click **Member Center**. Apple Developer Home page appears.

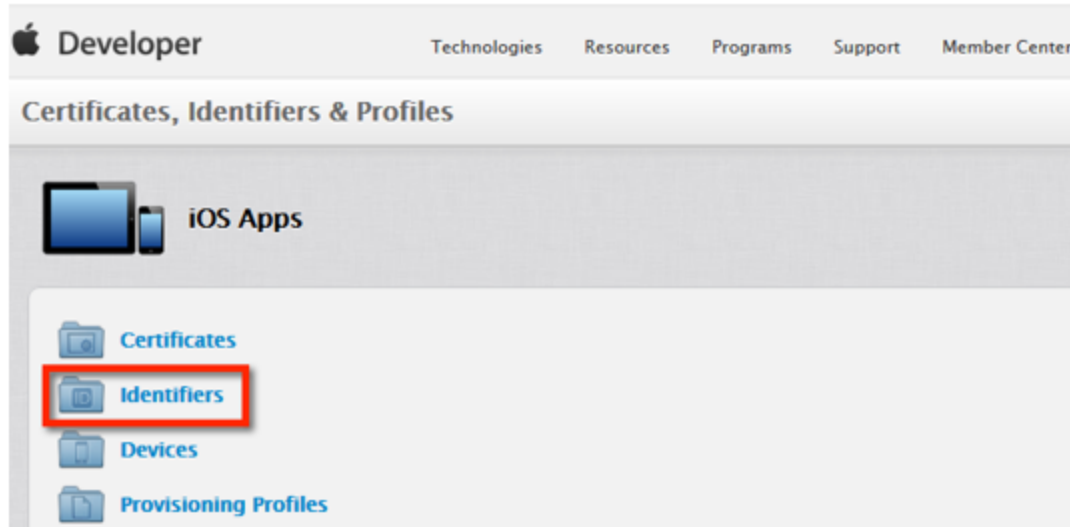


2. Enter your [Apple Developer Enterprise Program](#) credentials and click **Sign in**. Developer Program Resources page appears.



3. Click the **Certificates, Identifiers & Profiles** icon.



4. Click **Identifiers**5. Click **App IDs** in the left column, then click the + symbol to create a new App ID.6. Choose the option **Explicit App ID** and provide an **App ID description** (Launchpad is recommended).

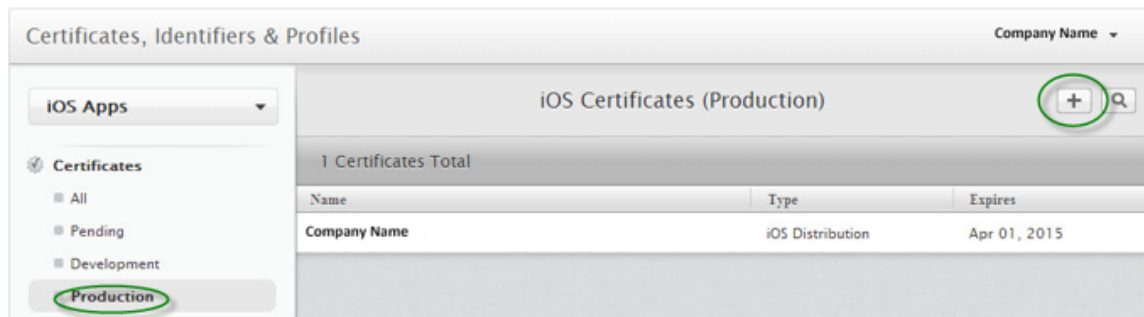
Name:

7. Provide the **Bundle ID**: `com.companyname.containerapp` (Note that the **com.** as the first segment and the **.containerapp** as the third segment are required. Companyname in the middle segment must match the middle segment choice of the Wild Card App ID chosen in a previous step. A different companyname entry between the IDs here will cause a Bundle ID Mismatch error later)

Bundle ID:

We recommend using a reverse-domain name style string (i.e., `com.domainname.appname`). It cannot contain an asterisk (\*).

8. Review details and choose **Submit**.
9. In the Certificates section, click on **Production**, then click the **+** symbol to add a new certificate. The Production page appears.




10. Choose the **Apple Push Notification service SSL (Production)** option.

## Production

- In-House and Ad Hoc**  
Sign your iOS app for In-House or for Ad Hoc distribution.
- MDM CSR**  
For signing certificate signing requests from MDM solution customers for MDM certificate issuance at [identity.apple.com](https://identity.apple.com). For more information, read the [Mobile Device Management Protocol Reference](#).
- Apple Push Notification service SSL (Production)**  
Establish connectivity between your notification server and the Apple Push Notification service production environment. A separate certificate is required for each app you distribute.
- Pass Type ID Certificate**  
Sign and send updates to passes in Passbook

11. Choose the **App ID** created previously (the ID ends with `.containerapp`) from the drop-down list and click **Continue**.

Select Type Request Generate Download



### Which App ID would you like to use?

All App IDs you enable to receive push notifications require its own individual Push SSL Certificate. The App ID-specific Push SSL certificate allows your notification server to connect to the Apple Push Notification Service. Note that only explicit App IDs with a specific Bundle Identifier can be used to create an Push SSL Certificate.

**Select an App ID for your Push SSL Certificate (Production)**

App ID: .com. .containerapp ▼

12. Follow the displayed instructions to generate a CSR file and click **Continue**.

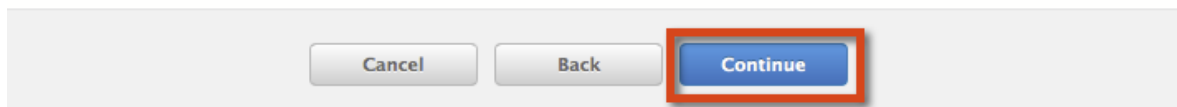
To manually generate a Certificate, you need a Certificate Signing Request (CSR) file from your Mac. To create a CSR file, follow the instructions below to create one using Keychain Access.

### Create a CSR file.

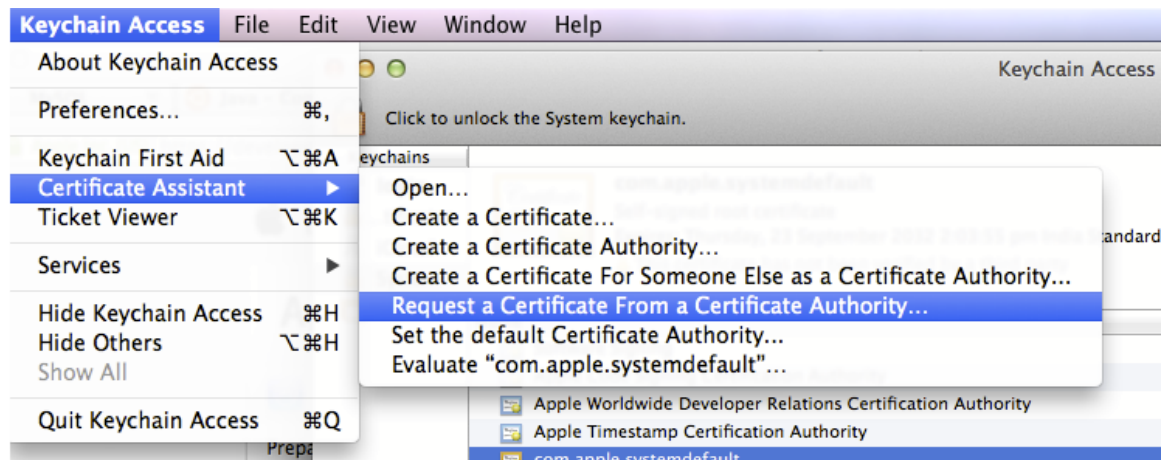
In the Applications folder on your Mac, open the Utilities folder and launch Keychain Access.

Within the Keychain Access drop down menu, select Keychain Access > Certificate Assistant > Request a Certificate from a Certificate Authority.

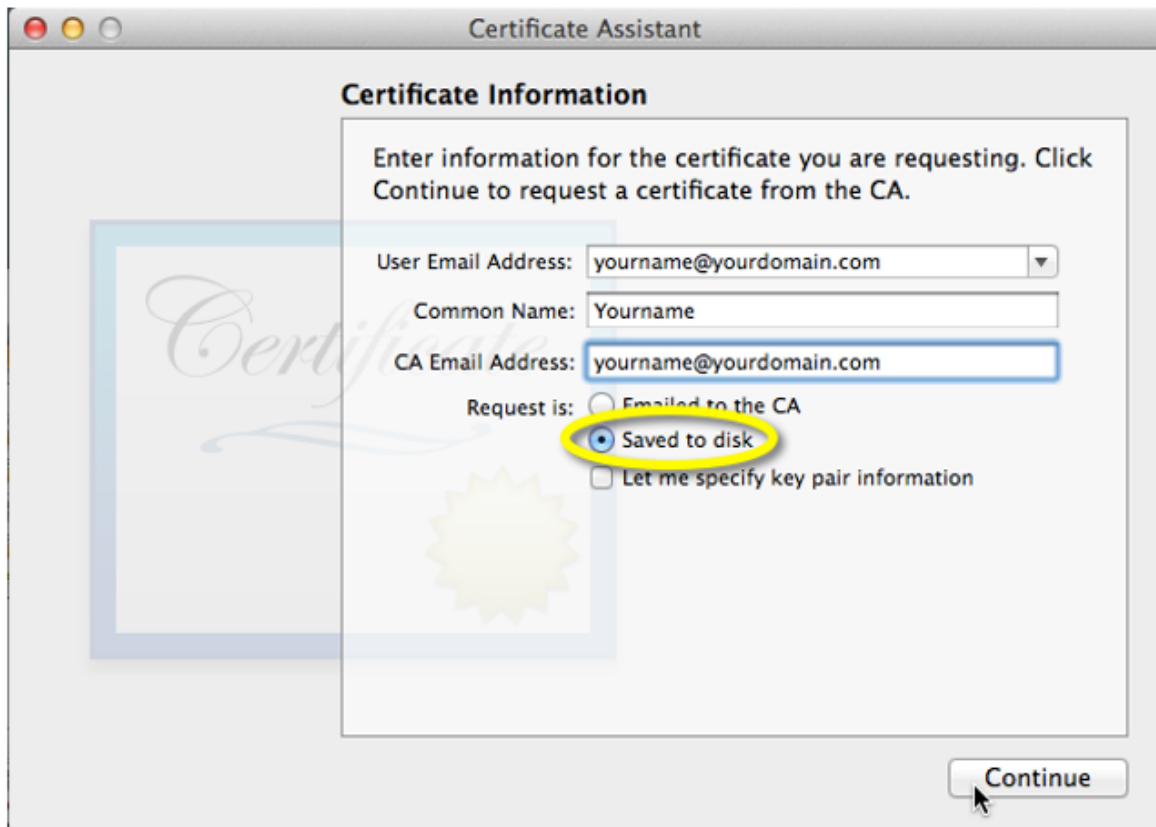
- In the Certificate Information window, enter the following information:
  - In the User Email Address field, enter your email address.
  - In the Common Name field, create a name for your private key (e.g., John Doe Dev Key).
  - The CA Email Address field should be left empty.
  - In the "Request is" group, select the "Saved to disk" option.
- Click Continue within Keychain Access to complete the CSR generating process.



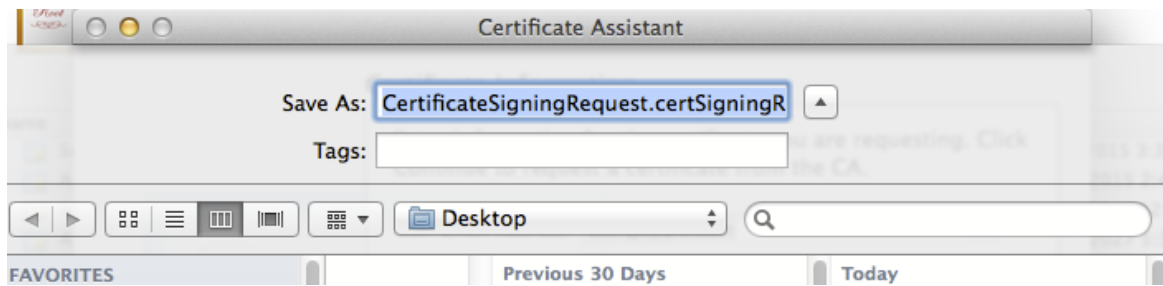
13. In the **Keychain Access** application on your Mac, under the **Certificate Assistant** menu, choose the **Request a Certificate From a Certificate Authority** option.



14. Enter details for the following fields, choose **Saved to disk**, and click **Continue**:

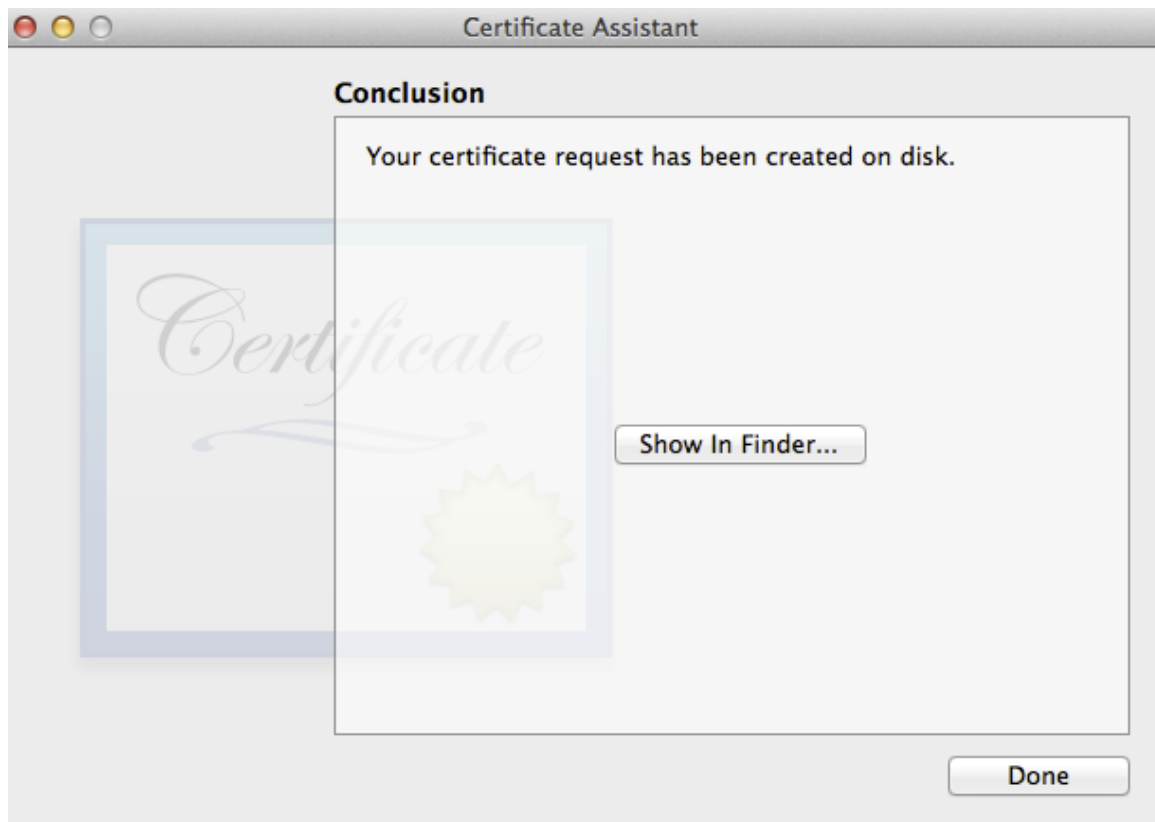


15. Save the **Certificate Signing Request (CSR)** to your local machine.

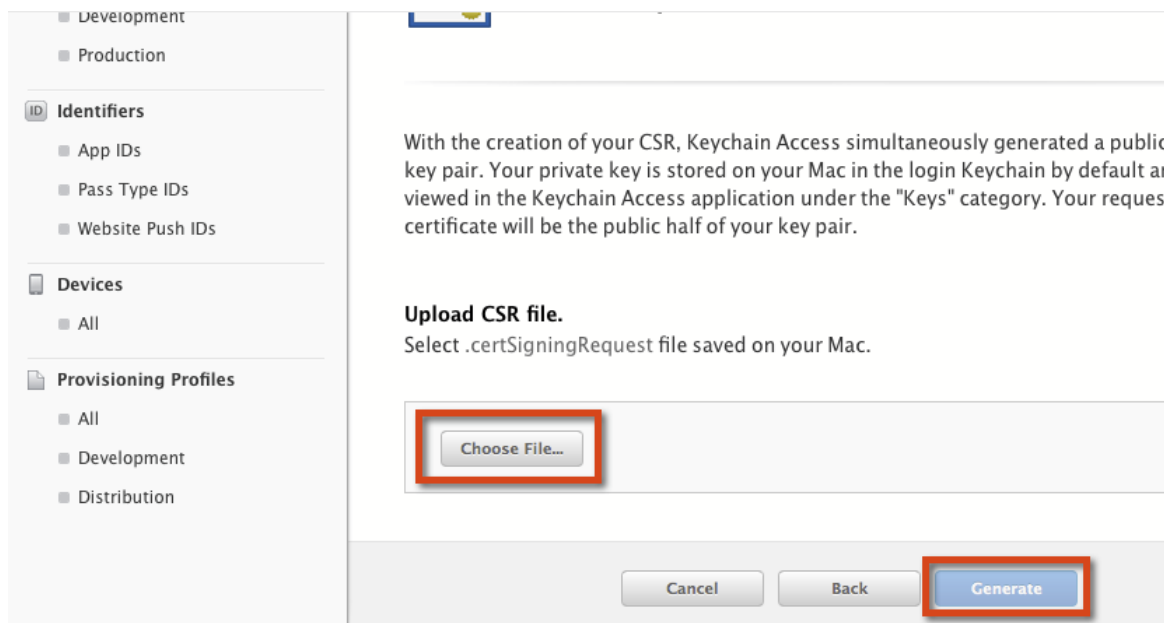




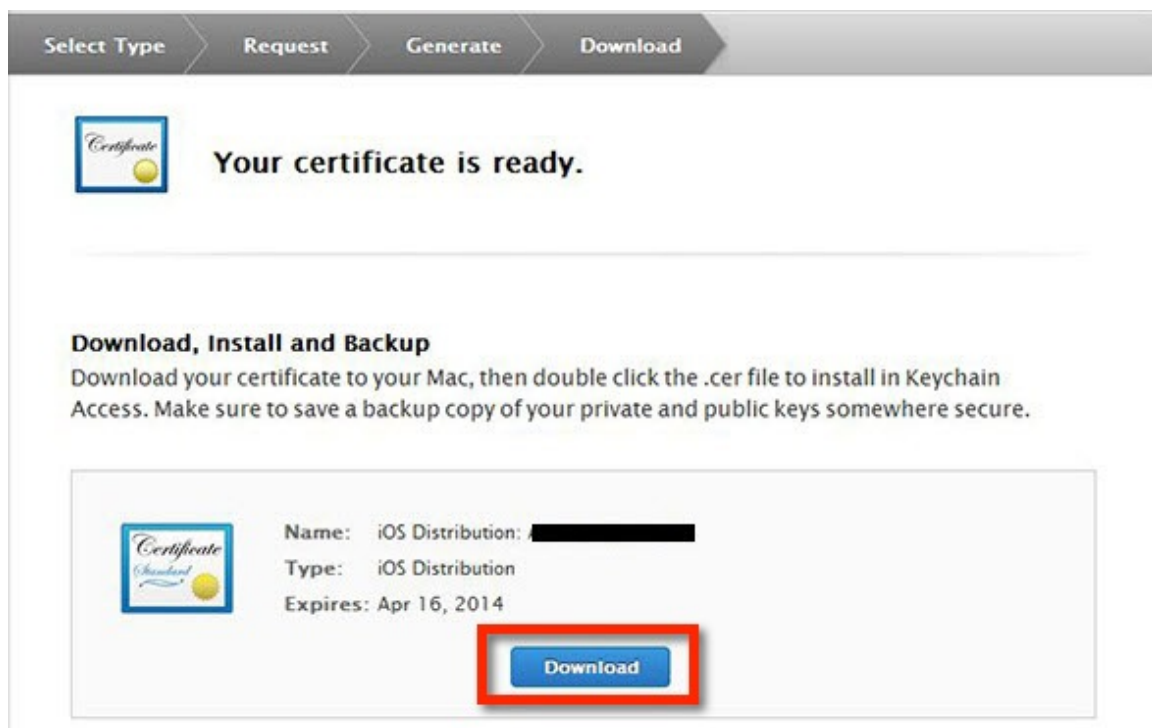
16. Click **Done**. You have now generated your CSR.



17. Back in your [developer.apple.com](https://developer.apple.com) site, click the **Choose File** button and upload your **CSR** file, and then click the **Generate** button.

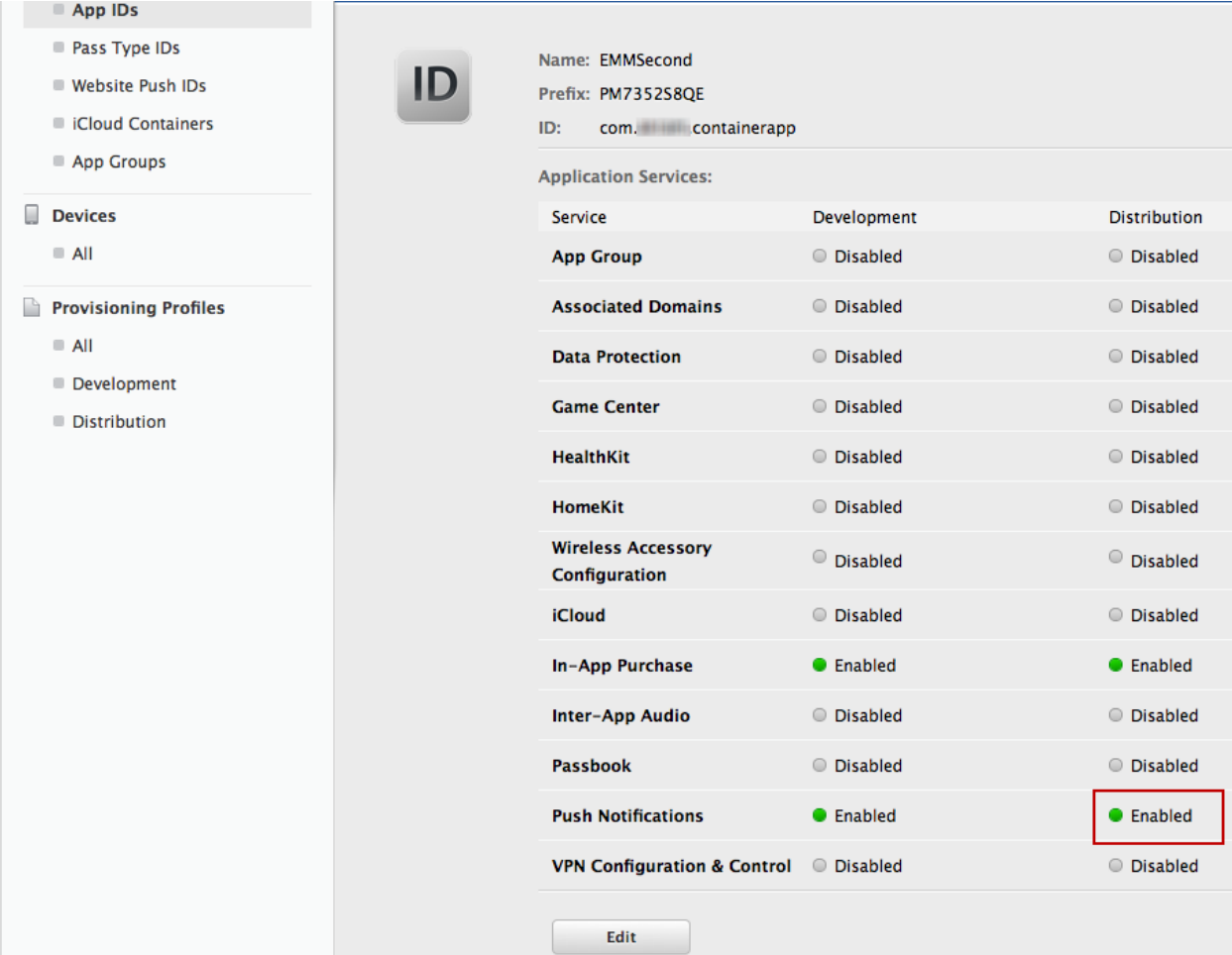


18. Click the **Download** button to download the Certificate you have just generated.

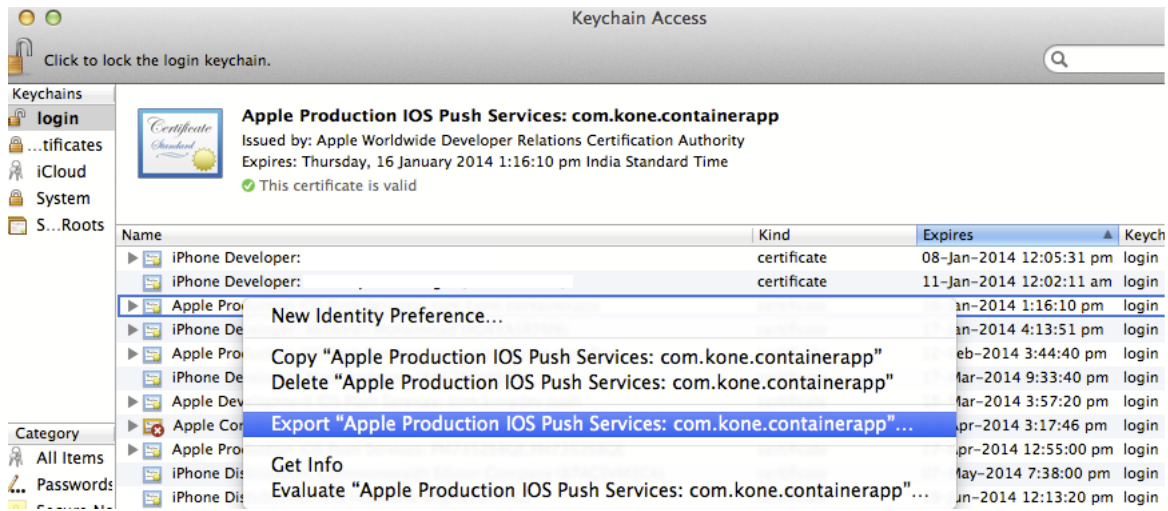


19. To verify that push is enabled, navigate to **Identifiers > App IDs**.

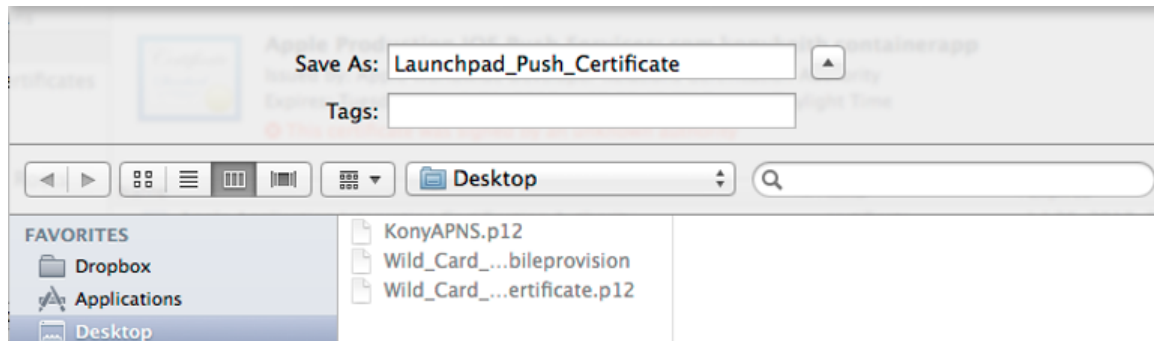
- 20. Select the certificate you created. Push Notifications will be enabled. You should see details as below.



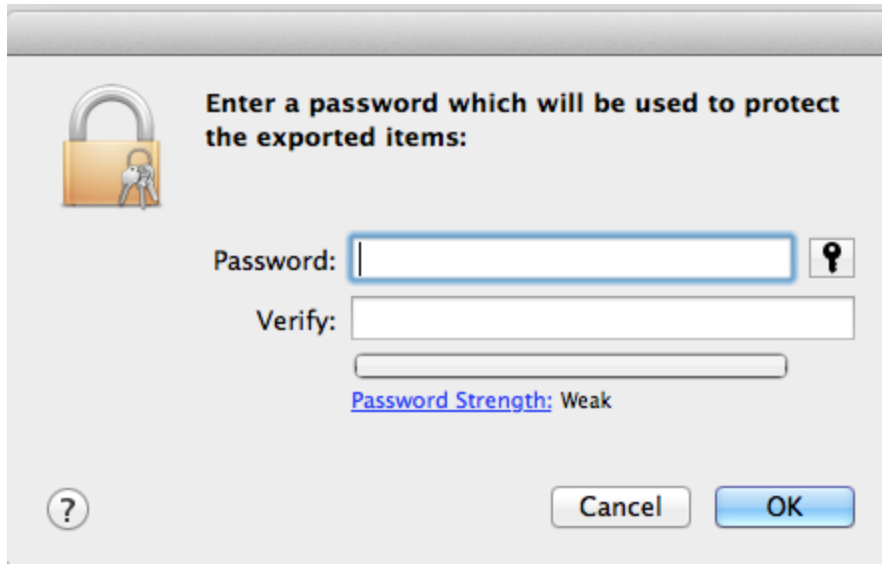
- 21. Import the downloaded .cer file to the local Keychain by double-clicking it.
- 22. In the **Keychain Access** application on your Mac, select the **Push certificate** downloaded (it will be listed as Apple Production iOS Push Services: com.companyname.containerapp), right-click it, and then select **Export**.



23. Save the exported certificate in a secure location. It is recommended to name this file **Launchpad\_Push\_Certificate**



24. You will be required to provide a certificate password. Make a note of this password for future use with this certificate.



25. Your Apple Application Manager (Launchpad app) Push Certificate is now complete. Store this file in a safe place to be used during your Management Cloud initial configuration.

You can now delete the CSR file saved locally, the `aps_production.cer` file saved locally, and the imported entry from your keychain.

## 15.7 Recreate Apple Application Manager (Launchpad app) Push Certificate

You cannot renew a certificate. You can only create a new certificate with the old certificate details. To recreate an Apple Application Manager Push Certificate,

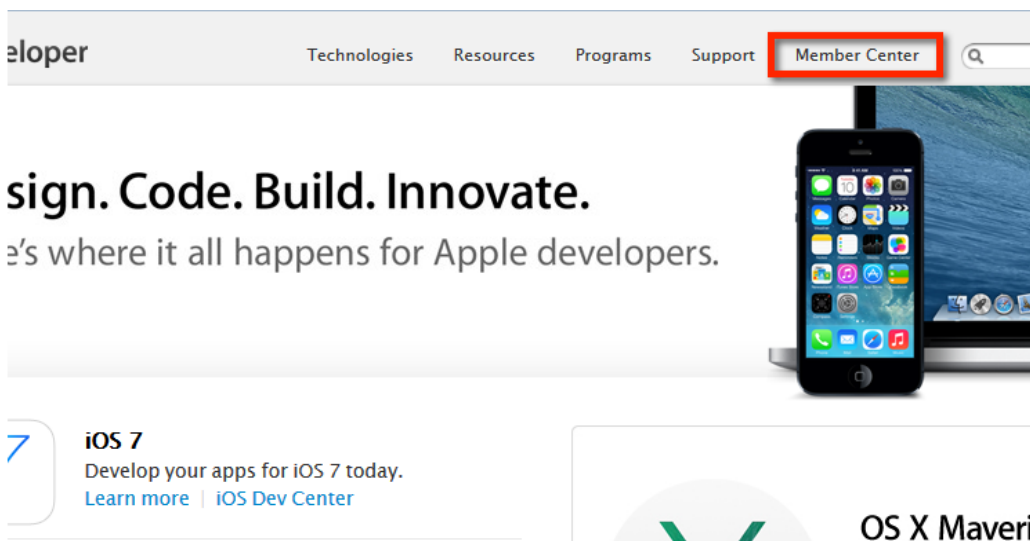
1. Go to your Apple developer [member center](#) in an internet browser.
2. In the Certificates, Identifiers & Profiles section, select **Certificates**.
3. Under the certificates section, select **Production**. All existing certificates appear.
4. Select the Certificate you want to recreate. If the certificate has expired, the **Revoke** button will be active.

5. Click **Revoke**. The certificate will be revoked.
6. Once the certificate is revoked, create a new certificate with the details of the old expired certificate as explained in the previous section.

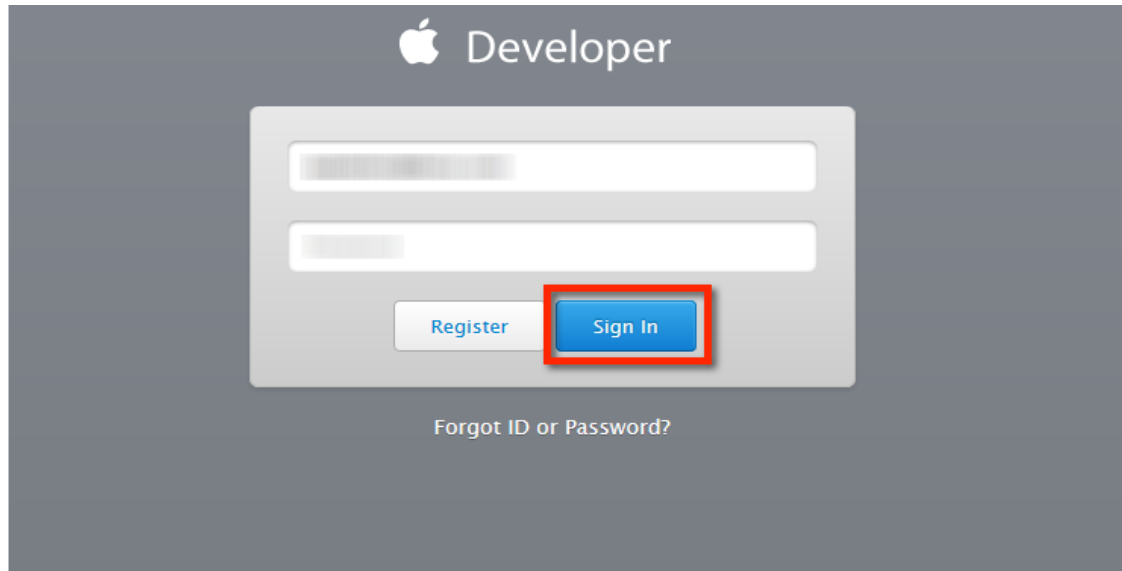
## 15.8 Creating the Apple Application Manager (Launchpad app) Provisioning Profile

To create Apple Application Manager (Launchpad app) Provisioning Profile, follow these steps:

1. In a browser, go to <https://developer.apple.com> in a web browser and click **Member Center**.

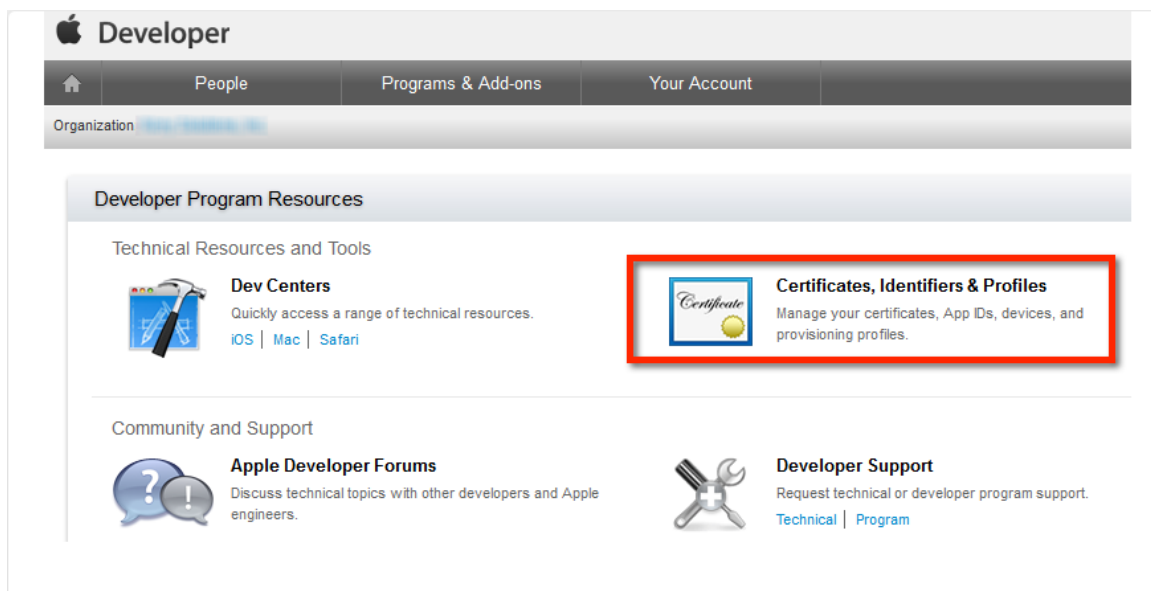


2. Enter your [Apple Developer Enterprise Program](#) credentials and click **Sign in**.

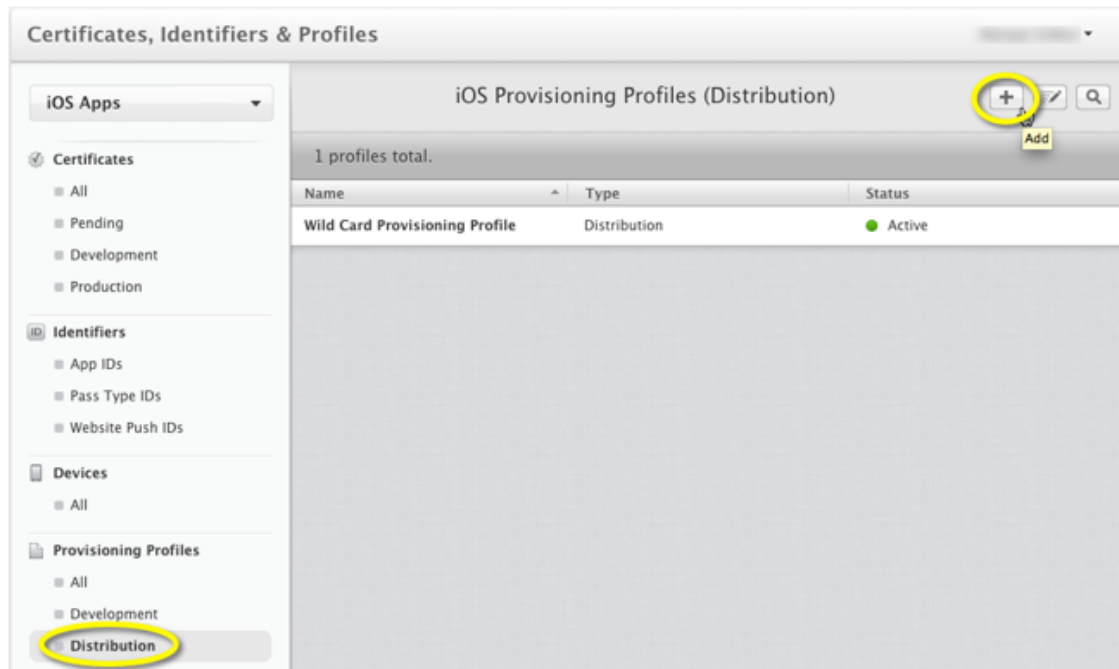


Apple Developer Home page appears.

3. Click the **Certificates, Identifiers & Profiles** icon.

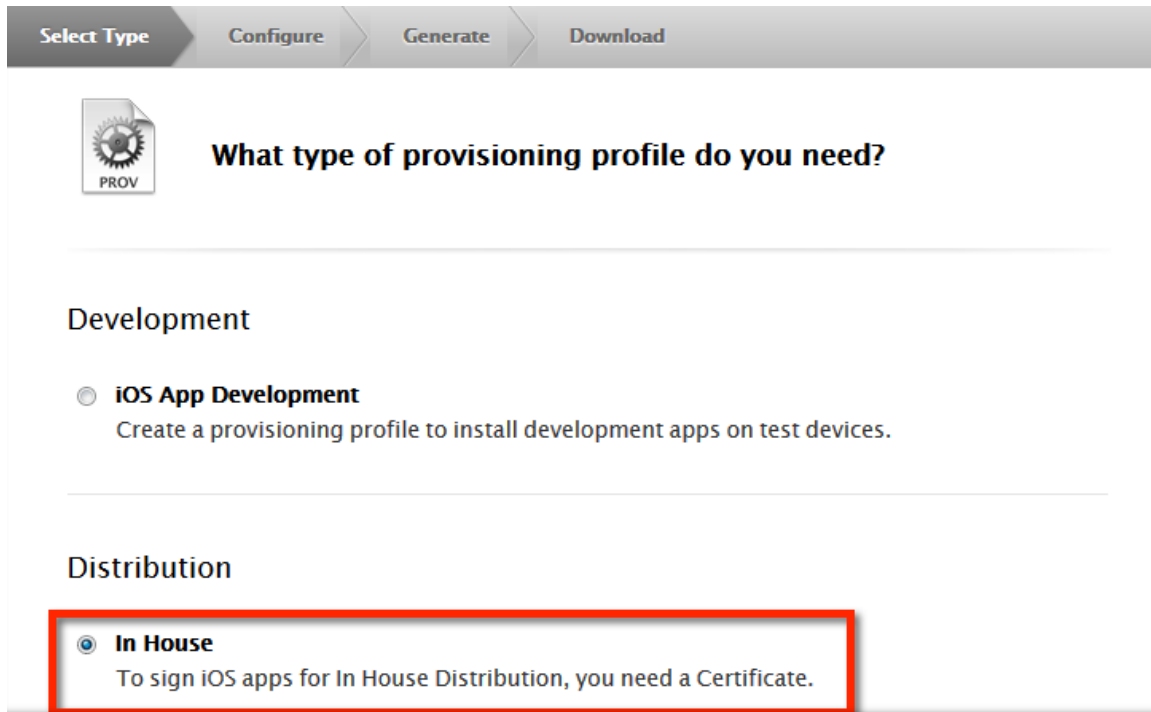


4. Click on **Distribution** under the **Provisioning Profiles** menu, then click the **+** symbol to create a Distribution Provisioning Profile.






5. Choose the **In House** option under the **Distribution** heading.



Select Type   Configure   Generate   Download

 **What type of provisioning profile do you need?**

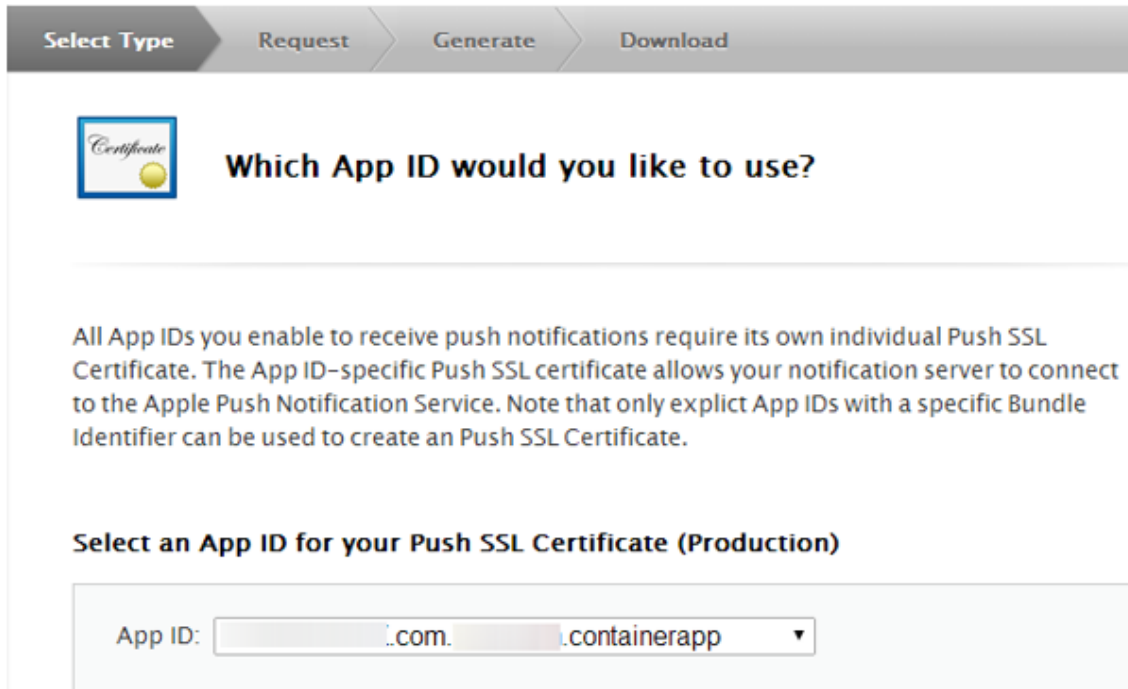
Development

**iOS App Development**  
Create a provisioning profile to install development apps on test devices.

Distribution

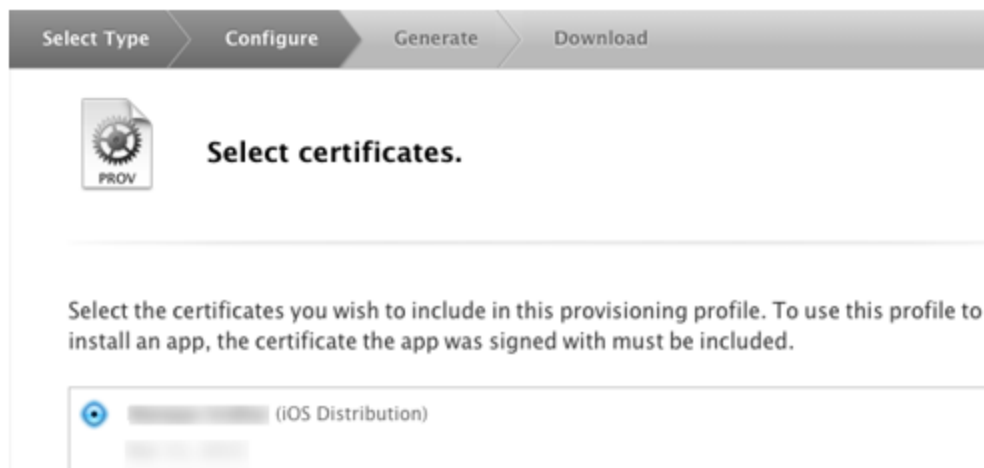
**In House**  
To sign iOS apps for In House Distribution, you need a Certificate.

6. Select the **App ID** ending with `.containerapp` from the drop-down list and click **Continue**.



The screenshot shows a progress bar with four steps: 'Select Type', 'Request', 'Generate', and 'Download'. The 'Request' step is active. Below the progress bar is a 'Certificate' icon and the heading 'Which App ID would you like to use?'. A paragraph explains that all App IDs require their own Push SSL Certificate. Below this is a section titled 'Select an App ID for your Push SSL Certificate (Production)' with a dropdown menu showing 'App ID: [redacted].com.[redacted].containerapp'.

7. Choose the iOS Distribution certificate from the list to include in this provisioning profile and then choose **Continue**. If there are two in this list, make note of the expiration date to ensure you choose the same certificate when you create another Provisioning Profile in a later step.



The screenshot shows a progress bar with four steps: 'Select Type', 'Configure', 'Generate', and 'Download'. The 'Configure' step is active. Below the progress bar is a 'PROV' icon and the heading 'Select certificates.'. A paragraph instructs the user to select certificates to include in the provisioning profile. Below this is a list of certificates, with one selected (indicated by a blue circle) and labeled '(iOS Distribution)'.

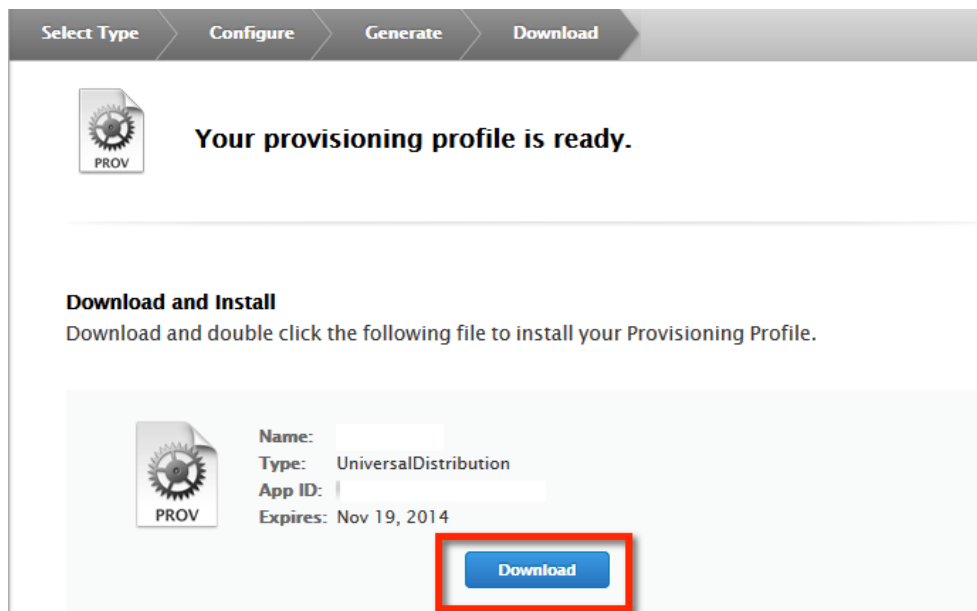
8. Enter the Profile Name (**Launchpad Provisioning Profile** is the recommended Profile Name),

then click the **Generate** button.



The screenshot shows a progress bar at the top with four steps: 'Select Type', 'Configure', 'Generate' (which is highlighted), and 'Download'. Below the progress bar is a document icon with a gear and the word 'PROV'. The main heading reads 'Name this profile and generate.' Below this is a text input field with the text 'Launchpad Provisioning Profile' entered. A note below the input field states: 'The name you provide will be used to identify the profile in the portal. You cannot use special characters such as @, &, \*, ', " for your profile name.'

9. Click the **Download** button to download the **Application Manager Provisioning Profile**.



The screenshot shows the progress bar with 'Download' highlighted. Below it is a document icon with a gear and 'PROV'. The heading reads 'Your provisioning profile is ready.' Underneath is the section 'Download and Install' with the instruction: 'Download and double click the following file to install your Provisioning Profile.' Below this is a card containing a document icon with a gear and 'PROV', and the following details: 'Name: [redacted]', 'Type: UniversalDistribution', 'App ID: [redacted]', and 'Expires: Nov 19, 2014'. A blue 'Download' button is highlighted with a red border.

10. Your Apple Wild Card Provisioning Profile now complete. Store this file in a safe place to be used during your Management Cloud initial configuration.

## 15.9 Recreate Apple Application Manager Provisioning Profile

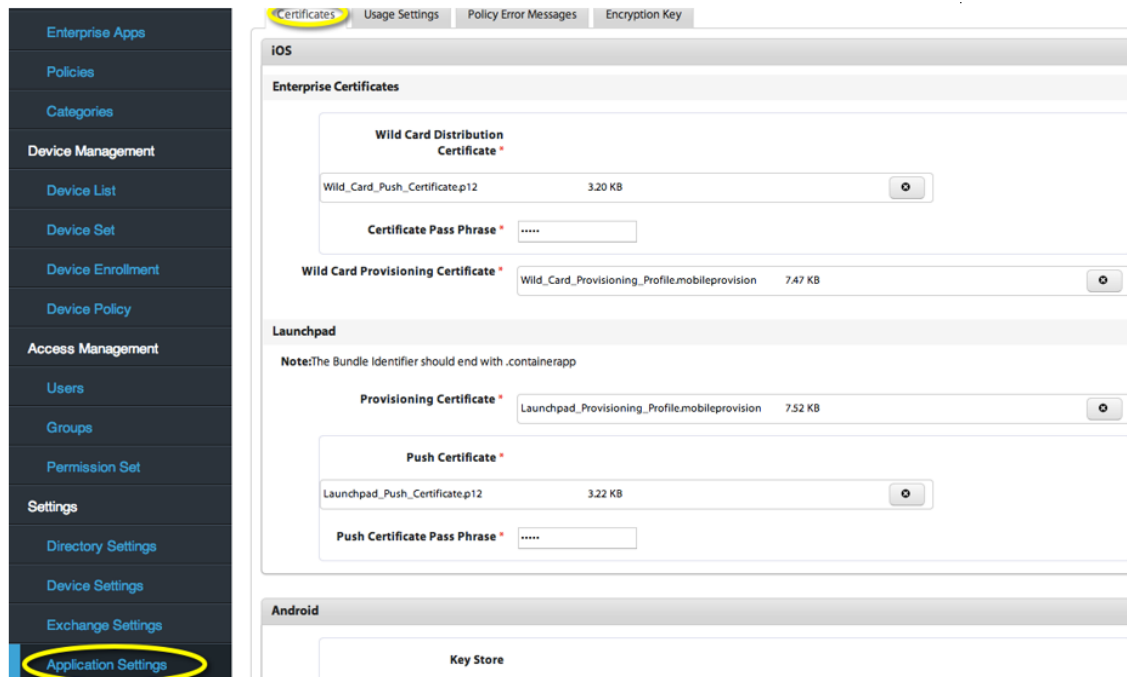
You cannot renew a provisioning profile. You can renew an expired provisioning profile by editing and re-generating it. To recreate a provisioning profile,

1. Go to your Apple developer [member center](#) in an internet browser.
2. In the Certificates, Identifiers & Profiles section, select **Provisioning Profiles**.
3. Under the Provisioning Profiles, select **All**, **Development** or **Distribution**. All existing provisioning profiles appear.
4. Select the provisioning profile you want to modify and click **Edit**. The Edit Provisioning Profile page appears.
5. Select the app id (com.CompanyName.\*) and the certificate the provisioning profile corresponds to, and click **Generate**. The Add iOS Provisioning Profile page appears.
6. Click **Download**. The provisioning profile will be downloaded.

## 15.10 Assigning App Resources in the Kony Management Cloud Administrator Console

To assign app resources in the Kony Management Cloud admin console, follow these steps:

1. You are now finished with creating resources on <https://developer.apple.com>. When you have your Kony Management Cloud administrator console available, launch it and log in to apply these resources.
2. From the Kony Management Cloud administrator console, choose **Application Settings** from the left navigation panel, then browse to the equally named four resources you have created. You will need to supply the certificate passwords chosen during the exports in this step.



- Once both certificates and pass phrases as well as both Provisioning Profiles are assigned, choose **Save**. After about 15 seconds a message will pop-up saying the Launchpad app is being wrapped. You will need to wait about 60 seconds for the process to finish. You can see the progress by choosing **Enterprise Apps** from the left navigation panel. When the wrap status changes to **Success** and the Published status changes to **Published**, you are finished assigning resources and updating the iOS versions of the Launchpad app.

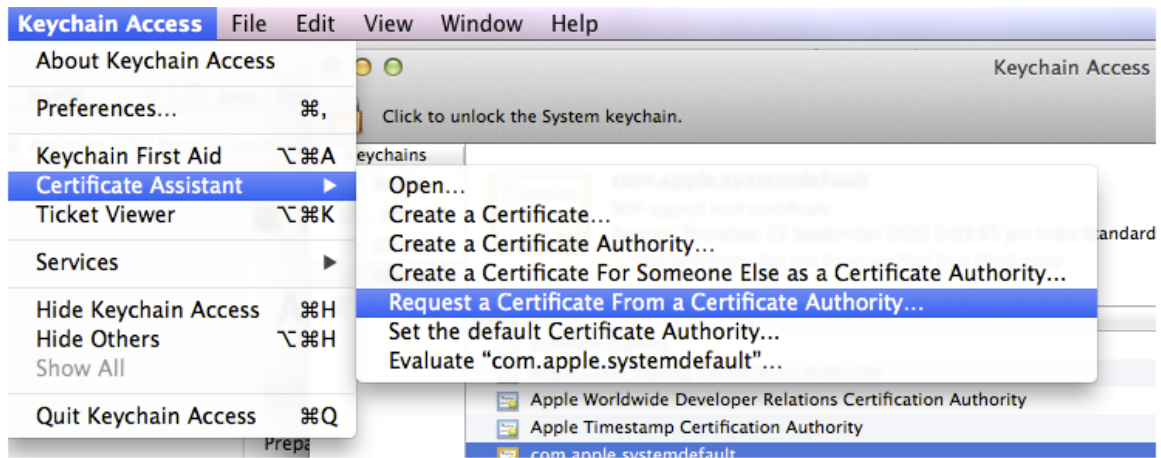
## 15.11 Creating the Apple Push Notification Certificate (APNS)

The APNS certificate is required before an iOS device can be enrolled. Completing this process requires access to the Kony Management Cloud administrator console. If you are preparing resources as a pre-install checklist, complete **Steps 1 through 4** only; **Steps 4 through 16** can be completed once you are able to log into the administrator console.

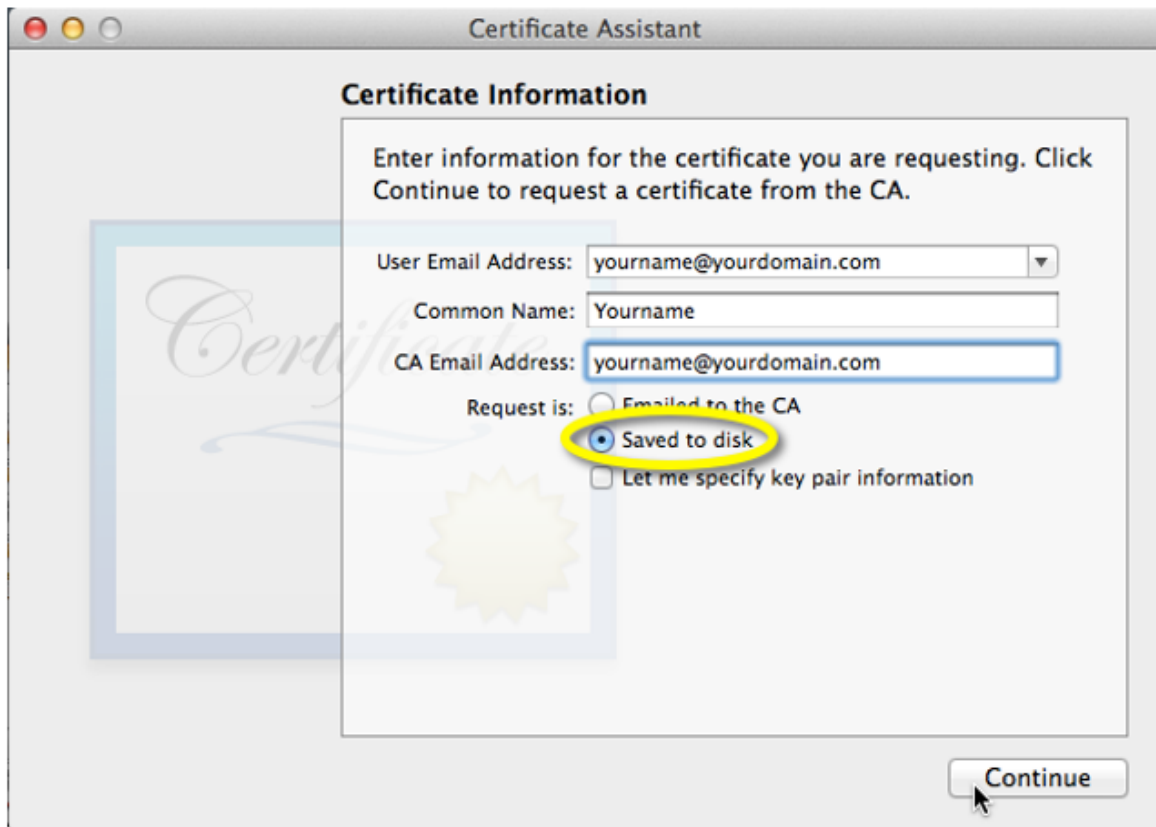
An Apple device APNs certificate can be created using the same Apple credentials as the previously used Developer Enterprise account, or any free iTunes account. You do not need a paid account for this step, but it should not be a personal account. Either the Enterprise account or a free account created as a service account should be used.

To create Apple Push Notification Certificate, follow these steps:

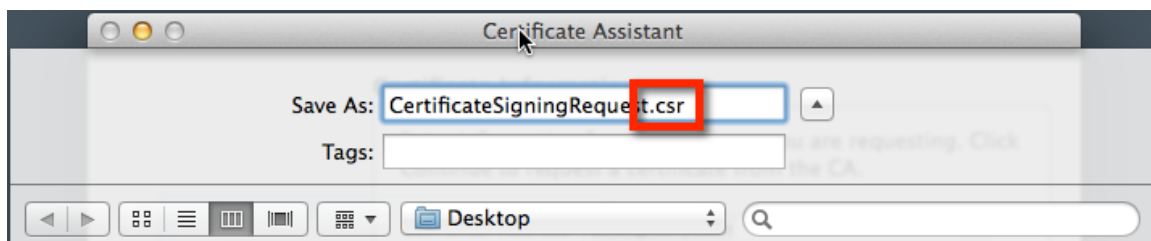
1. In the **Keychain Access** application on your Mac, under the **Certificate Assistant** menu, choose the **Request a Certificate From a Certificate Authority** option.



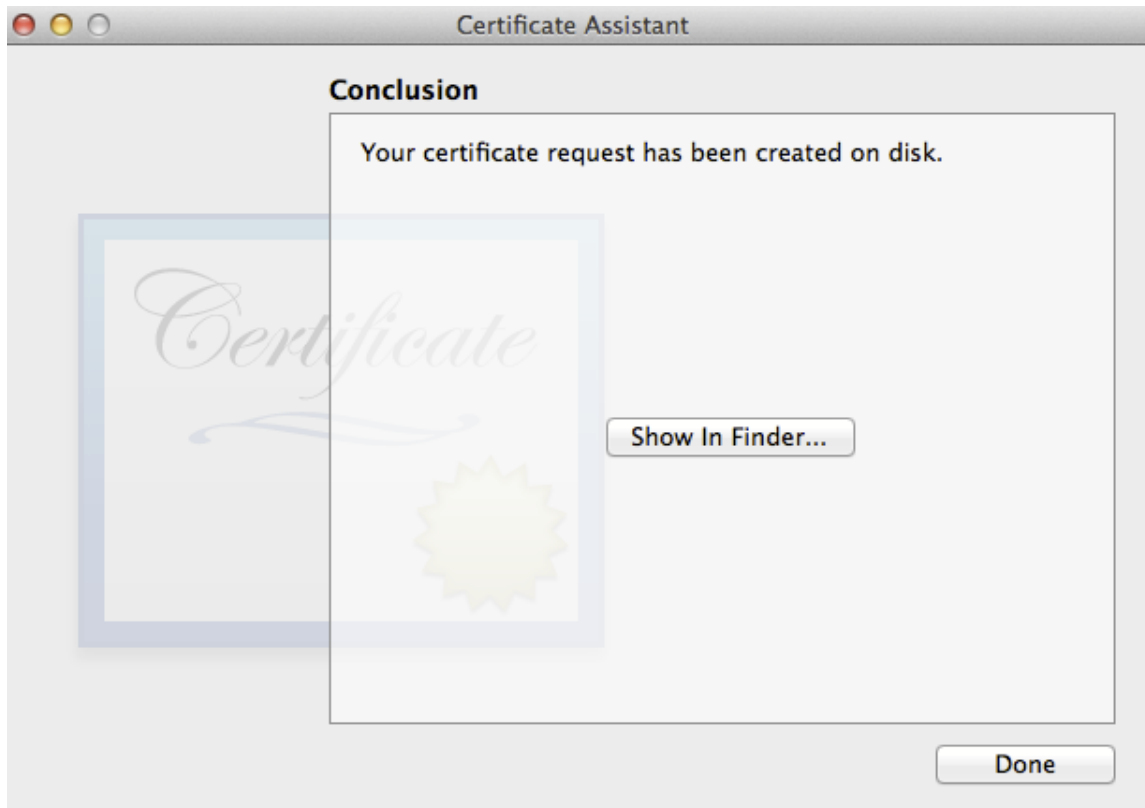
2. Enter details for the following fields, choose **Saved to disk**, and click **Continue**.



3. Save the Certificate Signing Request (CSR) to your local machine. You must change the extension from .certSigningRequest to `.csr`

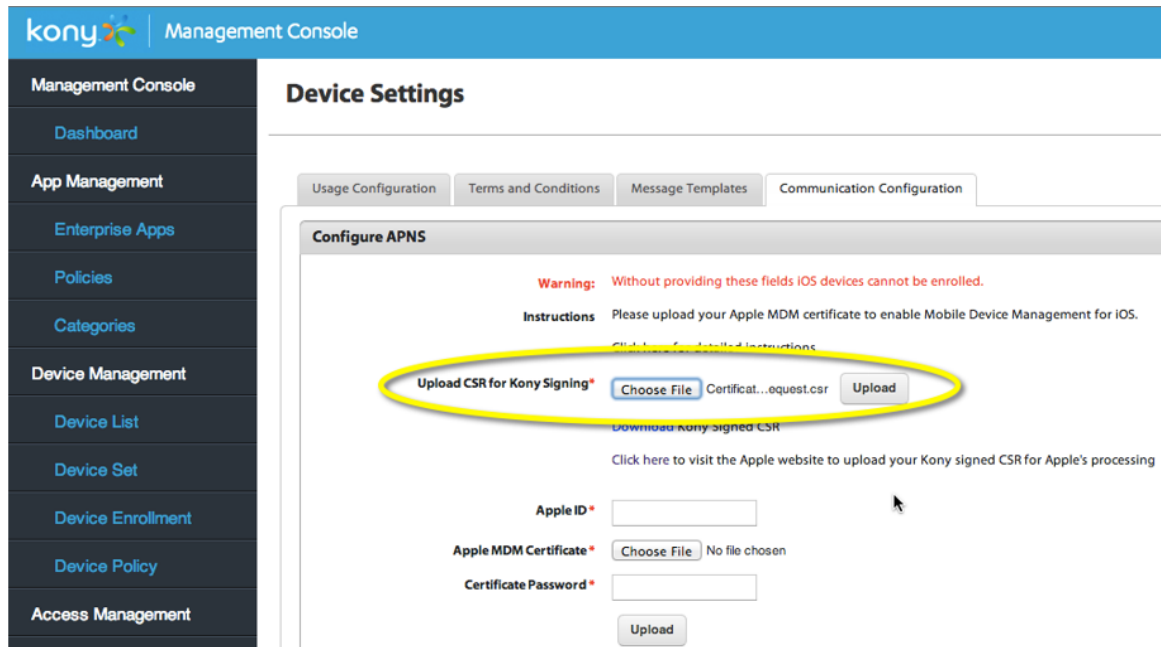


4. Click **Done**. You have now generated your CSR.



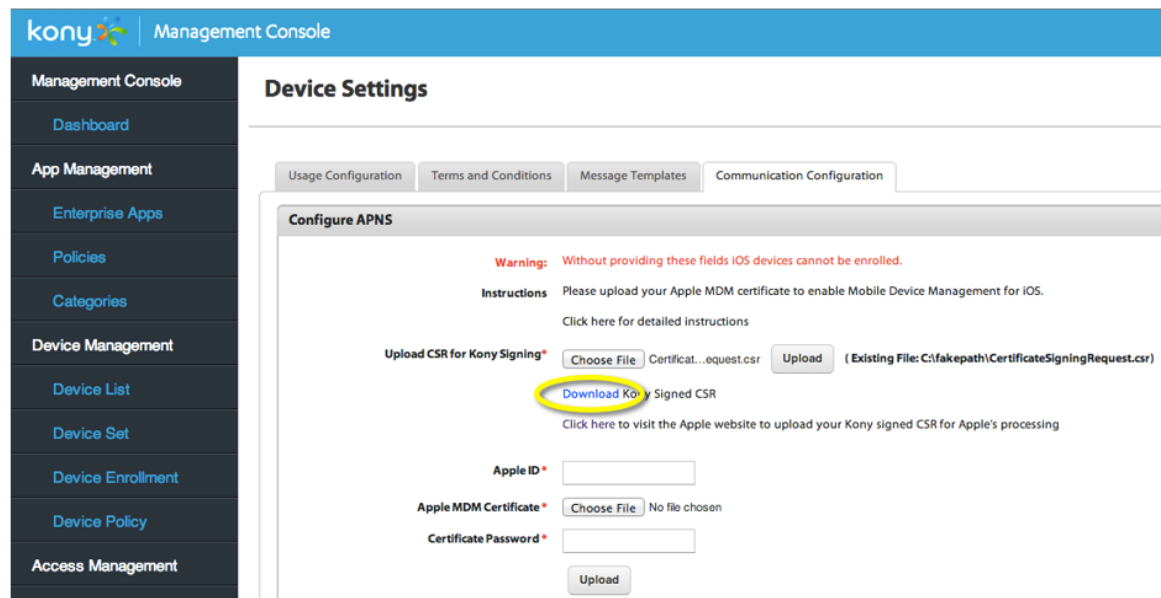
5. In the Kony Management Cloud administration console, select **Device Settings** from the **Settings** menu, then choose the **Communication Configuration** tab and browse to your saved CSR file, then choose the **Upload** option.





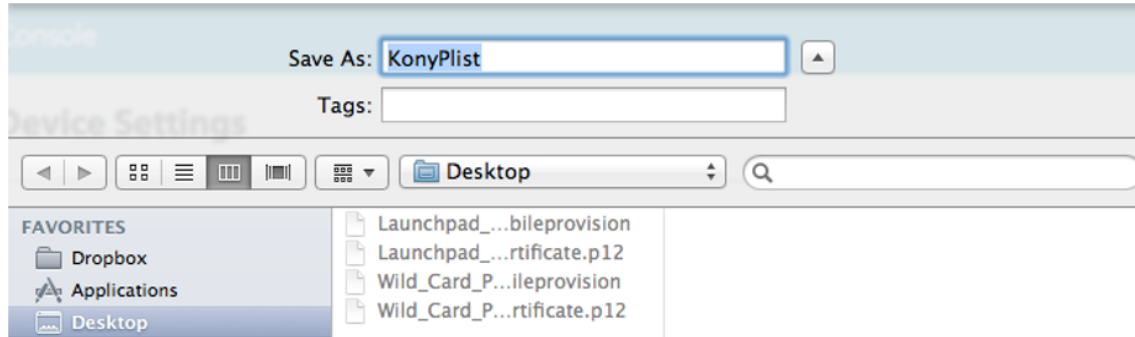
The screenshot shows the 'Device Settings' page in the Kony Management Console. The left sidebar contains navigation options: Management Console, Dashboard, App Management, Enterprise Apps, Policies, Categories, Device Management, Device List, Device Set, Device Enrollment, Device Policy, and Access Management. The main content area is titled 'Device Settings' and has tabs for Usage Configuration, Terms and Conditions, Message Templates, and Communication Configuration. The 'Configure APNS' section contains a warning: 'Warning: Without providing these fields iOS devices cannot be enrolled.' Below this are instructions: 'Please upload your Apple MDM certificate to enable Mobile Device Management for iOS. Click here for detailed instructions.' The 'Upload CSR for Kony Signing' field is highlighted with a yellow oval, showing a 'Choose File' button, the text 'Certificat...equest.csr', and an 'Upload' button. Below this is a 'Download Kony signed CSR' link. Further down are fields for 'Apple ID', 'Apple MDM Certificate' (with a 'Choose File' button and 'No file chosen' text), and 'Certificate Password', each with an 'Upload' button.

6. Once the CSR is uploaded, choose the **Download Kony Signed CSR** option to save the signed CSR to your local machine.

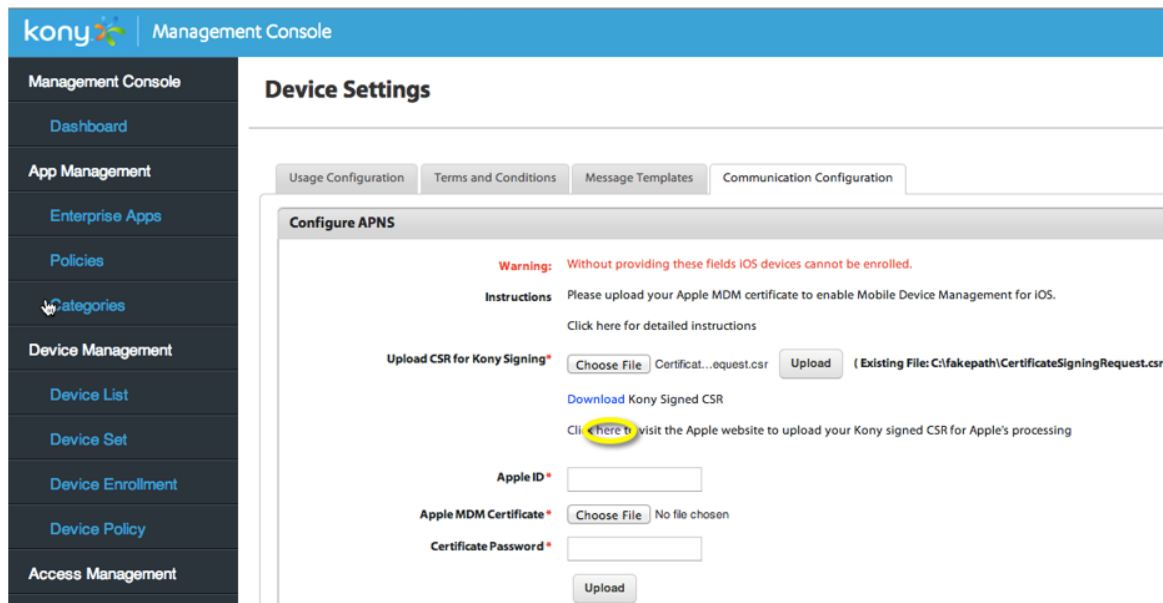


The screenshot shows the 'Device Settings' page in the Kony Management Console, similar to the previous one. The 'Upload CSR for Kony Signing' field now shows '(Existing File: C:\fakepath\CertificateSigningRequest.csr)' next to the 'Upload' button. The 'Download Kony Signed CSR' link is highlighted with a yellow oval. The rest of the page content remains the same.

7. Save the .plist to your local machine.

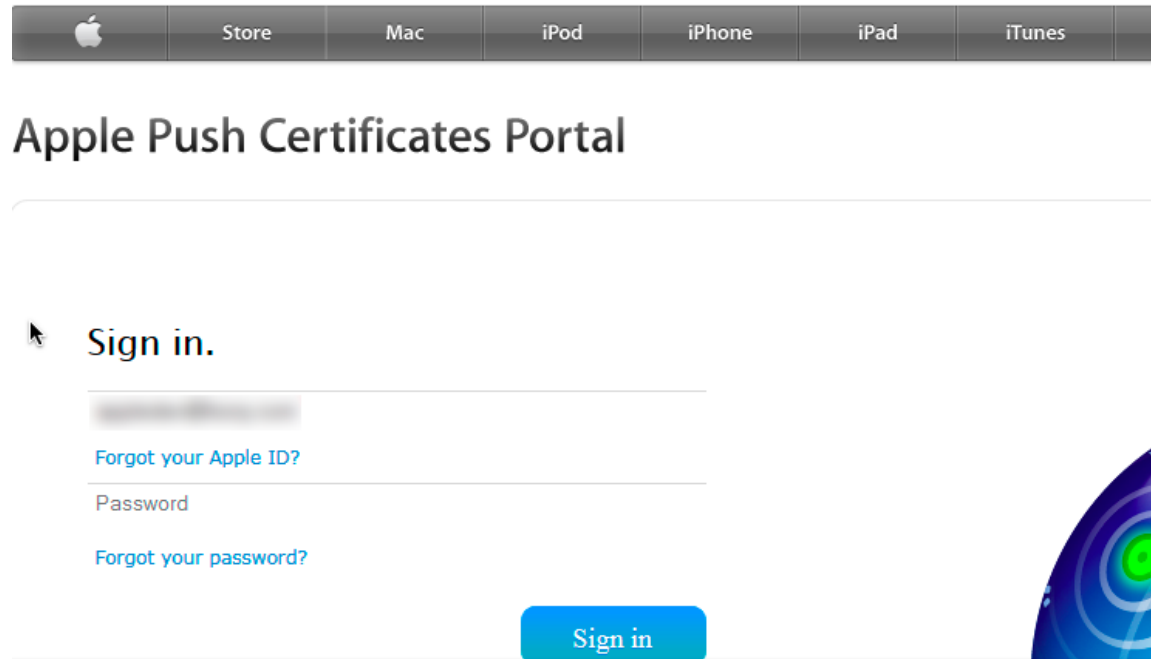


8. With the .plist (signed CSR) downloaded, choose the **Click here** option in the Kony Management Cloud administrator console to visit the Apple Certificate site, or manually go to the [identity.apple.com/pushcert](https://identity.apple.com/pushcert) site in a browser and sign in. These credentials can be a free Apple account, but should not be a personal account. It is recommended to create an account to be your Apple service account, or use your Apple Developer account from above.

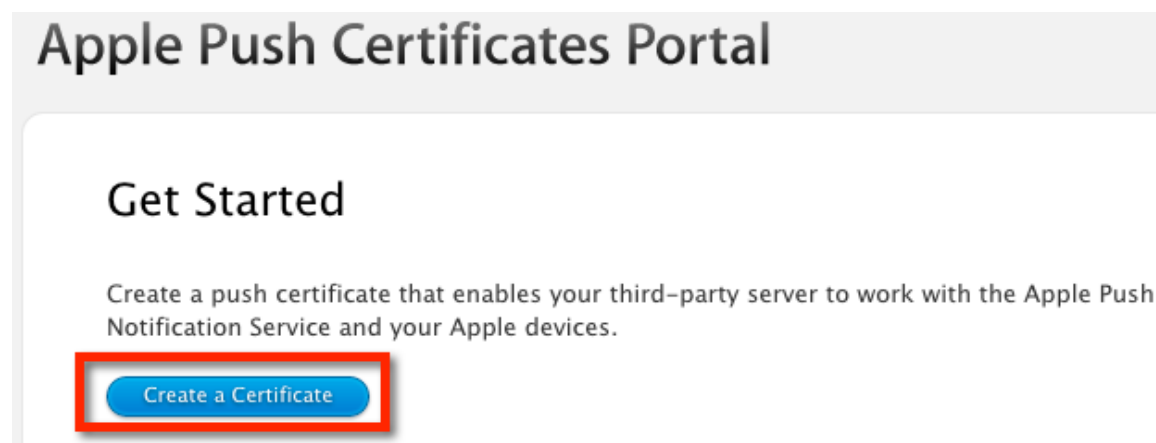


9. Provide your Apple credentials. Again, these credentials can be a free Apple account, but should not be a personal account. It is recommended to use your Apple Developer account, or

create an account to be your Apple service account.



10. Click **Create a Certificate** button.



11. Select the check box to agree to the terms and conditions, and then click **Accept**.

You accept and agree to the terms of this License Agreement on Your company's, organization's, educational

I have read and agree to these terms and conditions.

[Printable Version >](#)

Decline

Accept

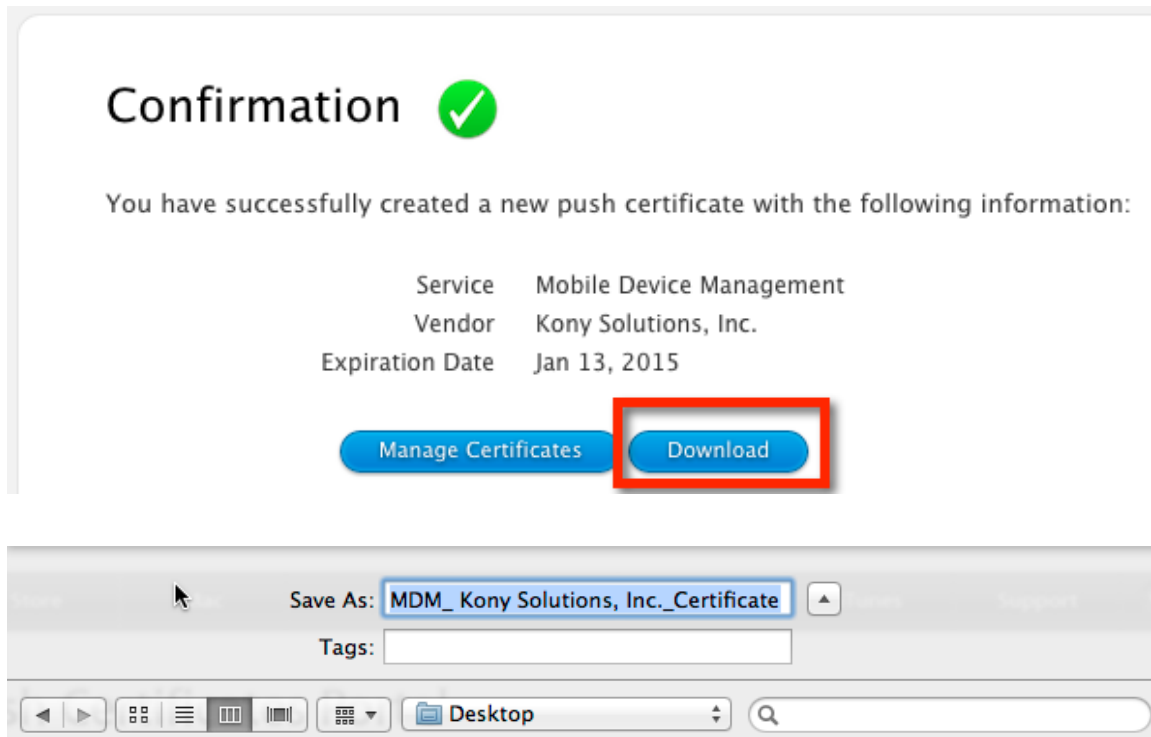
12. Browse to your downloaded .plist file (signed CSR) saved in [Step 7](#) above and choose the **Upload** option.

## Create a New Push Certificate

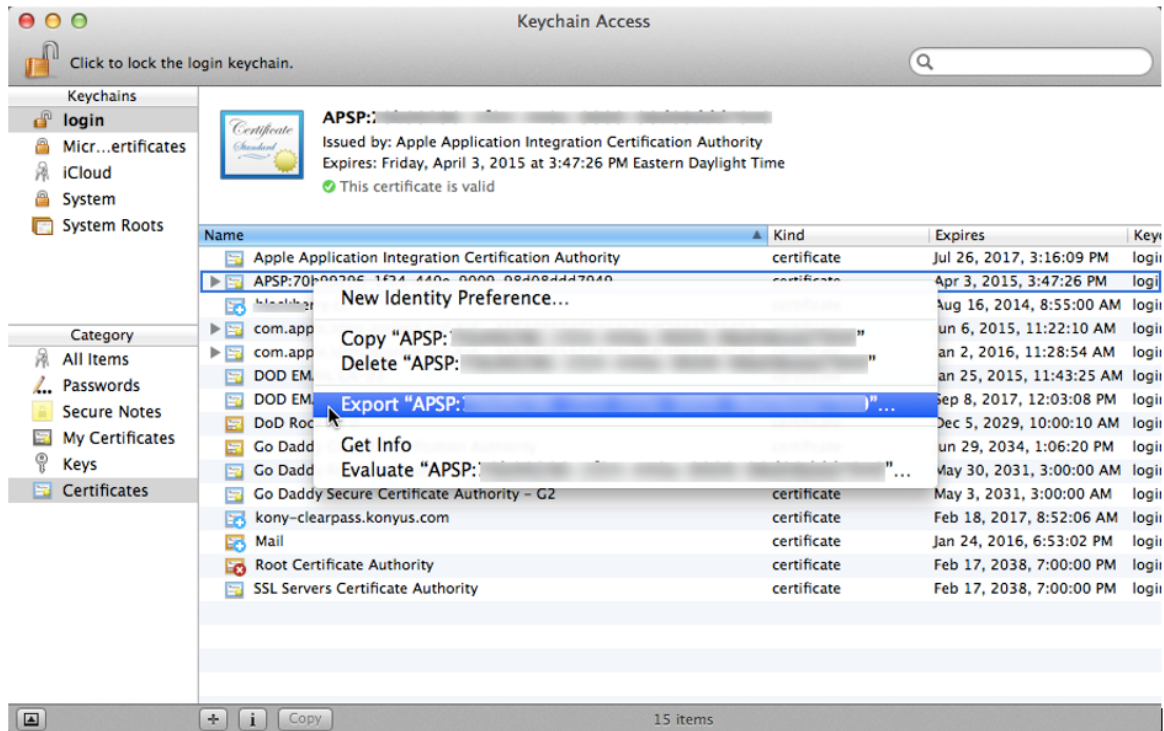
Upload your Certificate Signing Request signed by your third-party server vendor to create a new push certificate.

KonyPlist

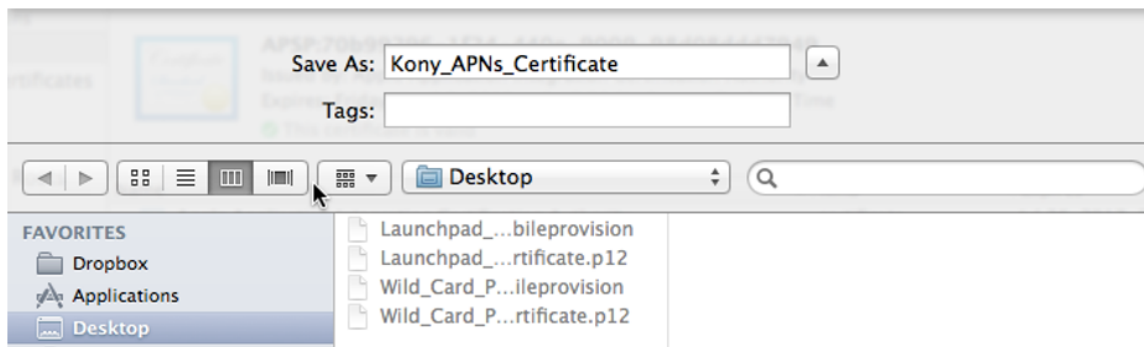
13. Next, choose the **Download** option and save the certificate to your local machine.



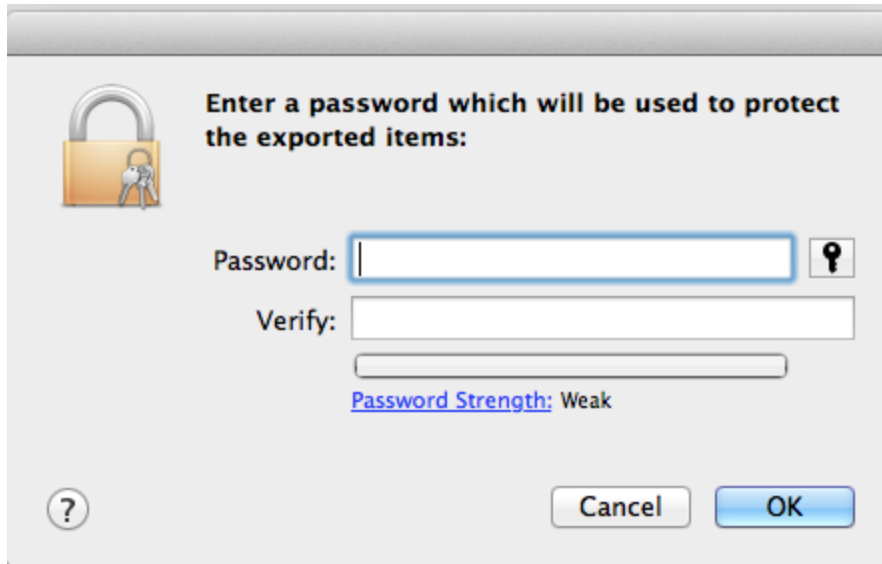
14. Double-click the downloaded certificate to add it to the Mac Keychain.
15. In the **Keychain Access** application on your Mac, select the APNS certificate downloaded (it will be listed as APSP:xxxxxxx), right click, and then select **Export**.



16. Save the exported .p12 certificate in a secure location.



17. You will be required to provide a certificate Password. Make a note of this password for future use with this certificate.



18. Your Apple Push Notification Certificate is now created. You can now delete the CSR file saved locally, the KonyPlist file saved locally, the MDM\_Kony\_Solutions,\_Inc.\_Certificate.cer file saved locally, and the imported entry from your keychain.
19. Upload Kony\_APNs\_Certificate.p12 now to the Kony Management Cloud console by selecting **Device Settings** from the **Settings** menu, then choose the **Communication Configuration** tab. Designate the Apple ID used to create the certificate in the designated field, browse to the .p12 certificate saved locally, and provide the password chosen in above, and then click **Upload**.

The screenshot shows the Kony Management Console interface. On the left is a navigation menu with options like Management Console, Dashboard, App Management, Enterprise Apps, Policies, Categories, Device Management, Device List, Device Set, Device Enrollment, Device Policy, and Access Management. The main content area is titled 'Device Settings' and has tabs for Usage Configuration, Terms and Conditions, Message Templates, and Communication Configuration. The 'Communication Configuration' tab is active, showing the 'Configure APNS' section. A warning message states: 'Warning: Without providing these fields iOS devices cannot be enrolled.' Below this are instructions: 'Please upload your Apple MDM certificate to enable Mobile Device Management for iOS. Click here for detailed instructions'. There are two file upload fields: 'Upload CSR for Kony Signing\*' with a 'Choose File' button and 'No file chosen' text, and 'Apple MDM Certificate\*' with a 'Choose File' button and a file name 'Kony\_APNS\_...icate.p12'. There is also a 'Certificate Password\*' field with masked characters. An 'Upload' button is highlighted with a yellow circle.

## 15.12 Renew Apple Push Notifications Certificate

To renew a certificate which is yet to expire, do the following:

1. In the EMM Management console, under **Settings**, click **Device Settings**.
2. Click **Communication Configuration** tab. The communication configuration tab appears.
3. In the **Configure APNS** section, click the help icon under the **APNS Certificate Expiry**. The APNS Certificate Renewal Steps page appears.
4. Follow the steps on the screen to renew the certificate.

## 15.13 Generating Certificate Signing Request (CSR) in Windows

To generate a CSR in Windows, follow these steps:

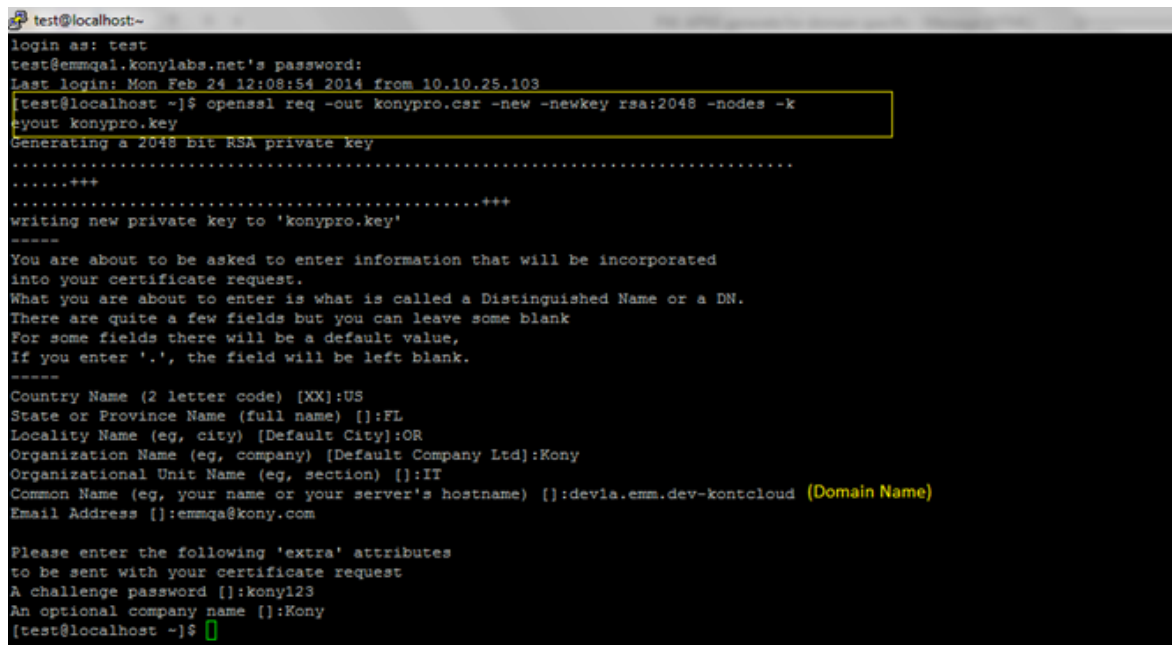
Make sure that you have OpenSSL enabled in your system.



1. Open command prompt and enter the command in the below format.

```
openssl req -out <csrname.csr> -new -newkey rsa:2048 -nodes -
keyout <keyname.key> For example, openssl req -out konypro.csr -new -
newkey rsa:2048 -nodes -keyout konypro.key.
```

2. If required, enter details for extra attributes. For example, a challenge password, or Optional company name and others. A CSR is generated. Save it to you desktop.



```
test@localhost~
login as: test
test@emmqa1.konylabs.net's password:
Last login: Mon Feb 24 12:08:54 2014 from 10.10.25.103
[test@localhost ~]$ openssl req -out konypro.csr -new -newkey rsa:2048 -nodes -k
eyout konypro.key
Generating a 2048 Bit RSA private key
.....+++
.....+++
writing new private key to 'konypro.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:FL
Locality Name (eg, city) [Default City]:OR
Organization Name (eg, company) [Default Company Ltd]:Kony
Organizational Unit Name (eg, section) []:IT
Common Name (eg, your name or your server's hostname) []:devia.emm.dev-kontcloud (Domain Name)
Email Address []:emmqa@kony.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:kony123
An optional company name []:Kony
[test@localhost ~]$
```

3. Upload the CSR for Kony signing in the EMM Management console Device Settings > Communication Configuration tab.

4. After uploading the CSR, download the Kony Signed CSR to your desktop.

**Configure APNS**

**Warning:** Without providing these fields iOS devices cannot be enrolled.

**Instructions:** Please upload your Apple MDM certificate to enable Mobile Device Management for iOS.  
Click here for detailed instructions

Upload CSR for Kony Signing\*  No file selected.  ( Existing File: konypro.csr)

Click here to visit the Apple website to upload your Kony signed CSR for Apple's processing

Apple ID\*

Apple MDM Certificate\*  No file selected.

Certificate Password\*

5. Upload the signed CSR to your Apple Push Certificates portal using your Apple developer credentials.
6. After the upload, download the pem file from Apple portal.
7. Rename the file to your company identifiable name. For example, yourcompany.pem.
8. Convert the downloaded .pem file to a.p12 file using the following command format.

```
openssl pkcs12 -export -inkey <yourcompany.key> -in  
<yourcompany.pem> -out <yourcompany.p12> For example, openssl pkcs12 -  
export -inkey konypro.key -in konypro.pem -out konypro.p12.
```

9. If prompted, provide a password while creating the .p12 file.
10. Upload the .p12 file to the Kony Management Suite portal under Device Settings > Communication Configuration.

11. Enter the password you created during .p12 file creation process.

**Configure APNS**

**Warning:** Without providing these fields iOS devices cannot be enrolled.

**Instructions:** Please upload your Apple MDM certificate to enable Mobile Device Management for iOS.  
Click here for detailed instructions

Upload CSR for Kony Signing\*  No file selected.  (Existing File: konypro.csr)

[Download Kony Signed CSR](#)

[Click here to visit the Apple website to upload your Kony signed CSR for Apple's processing](#)

Apple ID\*

Apple MDM Certificate\*  No file selected. (.p12 file)

Certificate Password\*  (.p12 Password)

## 15.14 Generating Certificate Signing Request (CSR) in Linux

To generate a CSR in Linux, follow these steps:

Make sure that you have OpenSSL enabled in your system.

1. Open command prompt and enter the command in the below format.

```
openssl req -out <csrname.csr> -new -newkey rsa:2048 -nodes -  
keyout <keyname.key> For example, openssl req -out konypro.csr -new -  
newkey rsa:2048 -nodes -keyout konypro.key.
```

2. If required, enter details for extra attributes. For example, a challenge password, or Optional company name and others. A CSR is generated. Save it to you desktop.

```
test@localhost:~$ ssh test@emmqa1.konylabs.net
login as: test
test@emmqa1.konylabs.net's password:
Last login: Mon Feb 24 12:08:54 2014 from 10.10.25.103
test@localhost ~]$ openssl req -out konypro.csr -new -newkey rsa:2048 -nodes -k
onyout konypro.key
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'konypro.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:FL
Locality Name (eg, city) [Default City]:OR
Organization Name (eg, company) [Default Company Ltd]:Kony
Organizational Unit Name (eg, section) []:IT
Common Name (eg, your name or your server's hostname) []:dev1a.emm.dev-kontcloud (Domain Name)
Email Address []:emmqa@kony.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:kony123
An optional company name []:Kony
test@localhost ~]$
```

3. Upload the CSR for Kony signing in the EMM Management console Device Settings > Communication Configuration tab.
4. After uploading the CSR, download the Kony Signed CSR to your desktop.

**Configure APNS**

**Warning:** Without providing these fields iOS devices cannot be enrolled.

**Instructions:** Please upload your Apple MDM certificate to enable Mobile Device Management for iOS.  
[Click here for detailed instructions](#)

Upload CSR for Kony Signing\*  No file selected.  ( Existing File: konypro.csr)

**Download Kony Signed CSR**

[Click here to visit the Apple website to upload your Kony signed CSR for Apple's processing](#)

Apple ID\*

Apple MDM Certificate\*  No file selected.

Certificate Password\*

5. Upload the signed CSR to your Apple Push Certificates portal using your Apple developer credentials.
6. After the upload, download the pem file from Apple portal.

7. Rename the file to your company identifiable name. For example, yourcompany.pem.
8. Convert the downloaded .pem file to a.p12 file using the following command format.

```
openssl pkcs12 -export -inkey <yourcompany.key> -in
<yourcompany.pem> -out <yourcompany.p12> For example, openssl pkcs12 -
export -inkey konypro.key -in konypro.pem -out konypro.p12.
```

9. If prompted, provide a password while creating the .p12 file.
10. Upload the .p12 file to the Kony Management Suite portal under Device Settings > Communication Configuration.
11. Enter the password you created during .p12 file creation process.

**Configure APNS**

**Warning:** Without providing these fields iOS devices cannot be enrolled.

**Instructions:** Please upload your Apple MDM certificate to enable Mobile Device Management for iOS.  
Click here for detailed instructions

**Upload CSR for Kony Signing\***  No file selected.  (Existing File: konypro.csr)

[Download Kony Signed CSR](#)  
[Click here to visit the Apple website to upload your Kony signed CSR for Apple's processing](#)

**Apple ID \***

**Apple MDM Certificate \***  No file selected. (.p12 file)

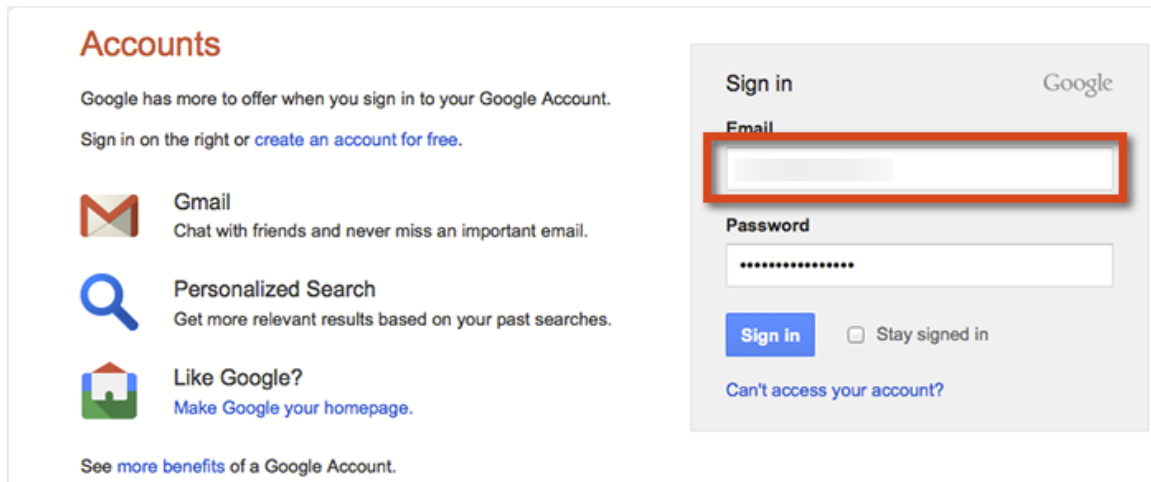
**Certificate Password \***  (.p12 Password)

## 15.15 Creating Android Certificates and Keys

To create Android Certificates and Keys follow these steps:


### 15.15.1 Create an Android GCM Key


1. In a browser, go to <https://code.google.com/apis/console>, and log in. If you do not have an account already, you need to create one. Note that this Google ID will be input into the Management Cloud configuration setup. You should not use a personal account.




**Accounts**

Google has more to offer when you sign in to your Google Account.  
Sign in on the right or [create an account for free](#).

 **Gmail**  
Chat with friends and never miss an important email.

 **Personalized Search**  
Get more relevant results based on your past searches.

 **Like Google?**  
Make Google your homepage.

See [more benefits](#) of a Google Account.

**Sign in** Google

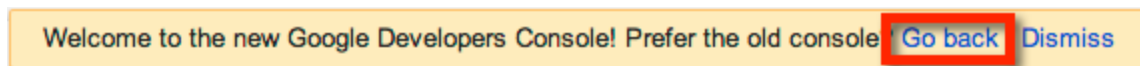
**Email**

**Password**

[Sign in](#)  Stay signed in

[Can't access your account?](#)

2. the **Go Back** option.



Welcome to the new Google Developers Console! Prefer the old console [Go back](#) [Dismiss](#)

3. Click the **Create Project** button.



**Google apis**

**Start using the Google APIs console**  
to manage your API usage

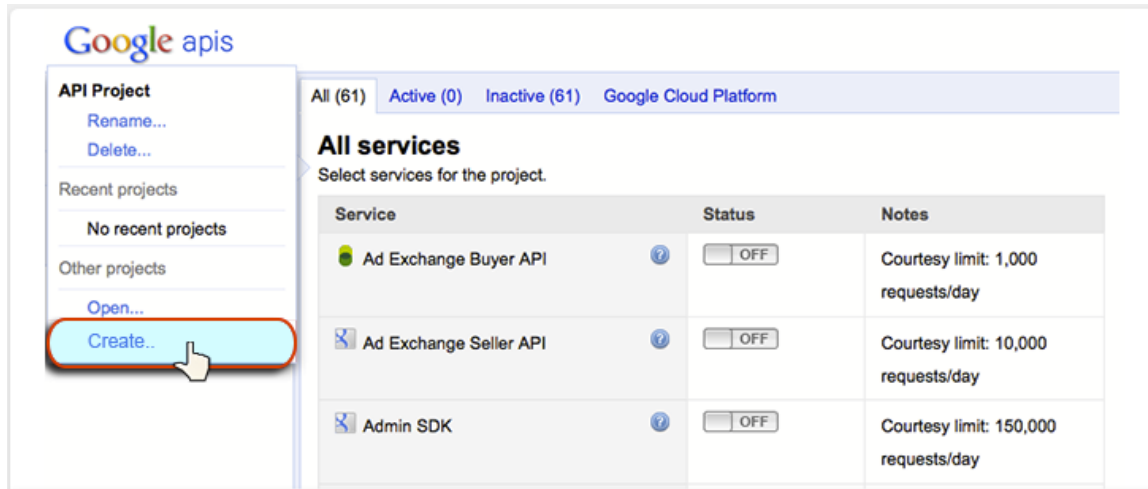


Creating an **APIs project** will let you:

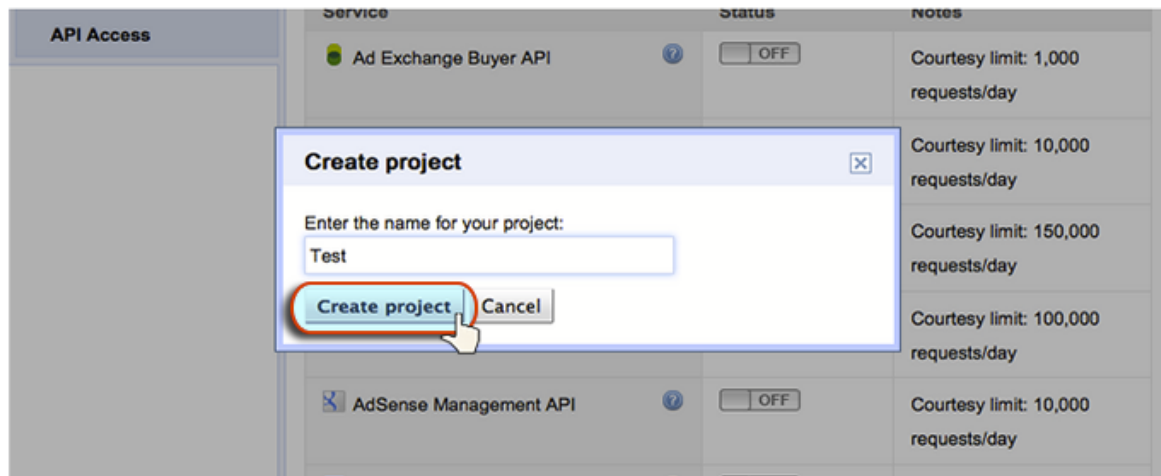
- Use Google APIs beyond anonymous limits.
- Monitor API usage and control API access.
- Share API management with a team.

[Create project...](#)

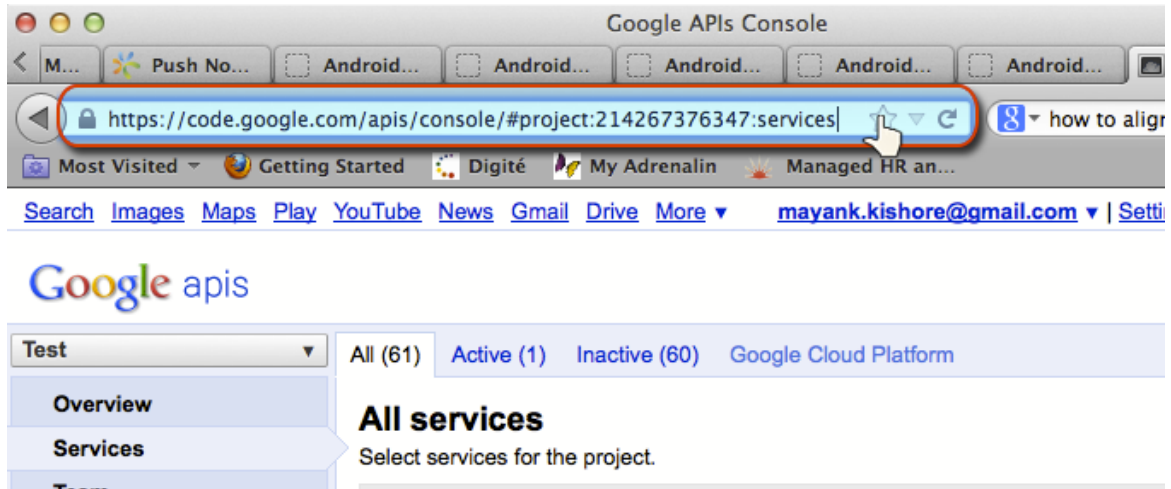
4. On the left side of the API's Dashboard page, click the drop-down menu and select **Create**.













5. Enter a name and choose **Create project**. Your browser will refresh and display a new URL.



6. The number after #project (Ex: 214267376347) should be noted. This is your Sender ID, and will be needed in the Management Cloud console during initial configuration.



7. Scroll down the page to **Google Cloud Messaging for Android** and set the ON/OFF widget to ON.

			requests/day
	Google Cloud Datastore API 	<input type="checkbox"/> OFF	Courtesy limit: 10,000,000 requests/day
	Google Cloud Messaging for Android 	<input checked="" type="checkbox"/> ON	
	Google Cloud Messaging for Chrome 	<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day
	Google Cloud SQL 	<input type="checkbox"/> OFF	<a href="#">Pricing</a>
	Google Cloud Storage 	<input type="checkbox"/> OFF	<a href="#">Pricing</a>

8. Agree to the terms and conditions and click **Accept**.



These terms outline your rights and responsibilities when using our APIs, so read them carefully. Additional terms may apply to the use of an API, including additional terms of service, terms within the accompanying API documentation, and any applicable policies or guidelines. If there is a conflict between these terms and the additional terms, the additional terms apply for that conflict. If you use the APIs as an interface to, or in conjunction with other Google products and services, then the terms for such products and services also apply.

**Section 1: Account and Registration**

**Accepting the Terms.** You may not use the APIs and may not accept the Terms if (a) you are not of legal age to form a binding contract with

I agree to these terms.

Accept Decline

**0 of 2 terms of service accepted.**  
Google APIs  
Google Cloud Messaging for Android

↓

**1 of 2 terms of service accepted.**  
**Accepted** Google APIs  
Google Cloud Messaging for Android

[Code Home - Privacy Policy](#)

- At the bottom of the API's home page, click the **Create new Server key** button.

Test

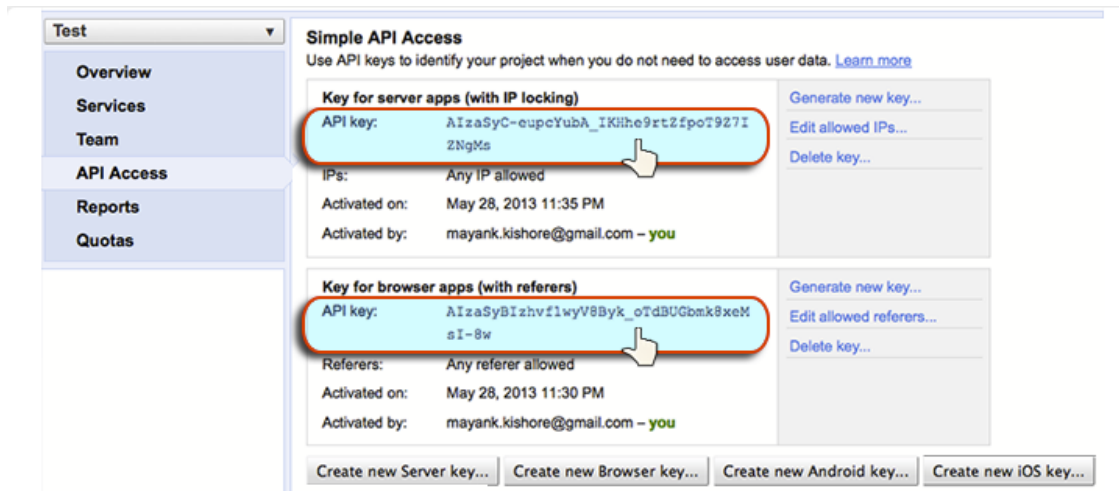
**Simple API Access**  
Use API keys to identify your project when you do not need to access user data. [Learn more](#)

**Key for browser apps (with referers)**

API key:	AIzaSyBIzhvf1wyV8Byk_oTdBUGbmK8xeMsI-8w	<a href="#">Generate new key...</a>
Referers:	Any referer allowed	<a href="#">Edit allowed referers...</a>
Activated on:	May 28, 2013 11:30 PM	<a href="#">Delete key...</a>
Activated by:	mayank.kishore@gmail.com - you	

[Create new Server key...](#) [Create new Browser key...](#) [Create new Android key...](#)  
[Create new iOS key...](#)

- Choose the **Server Key** option and click **Create**.
- Two GCM Keys are now displayed. Record the top one labeled "*for server apps*".



12. Store this key in a safe place to be used during your Management Cloud initial configuration. You will apply it with the Keystore created in the next steps below.

### 15.15.2 Google Maps API Key

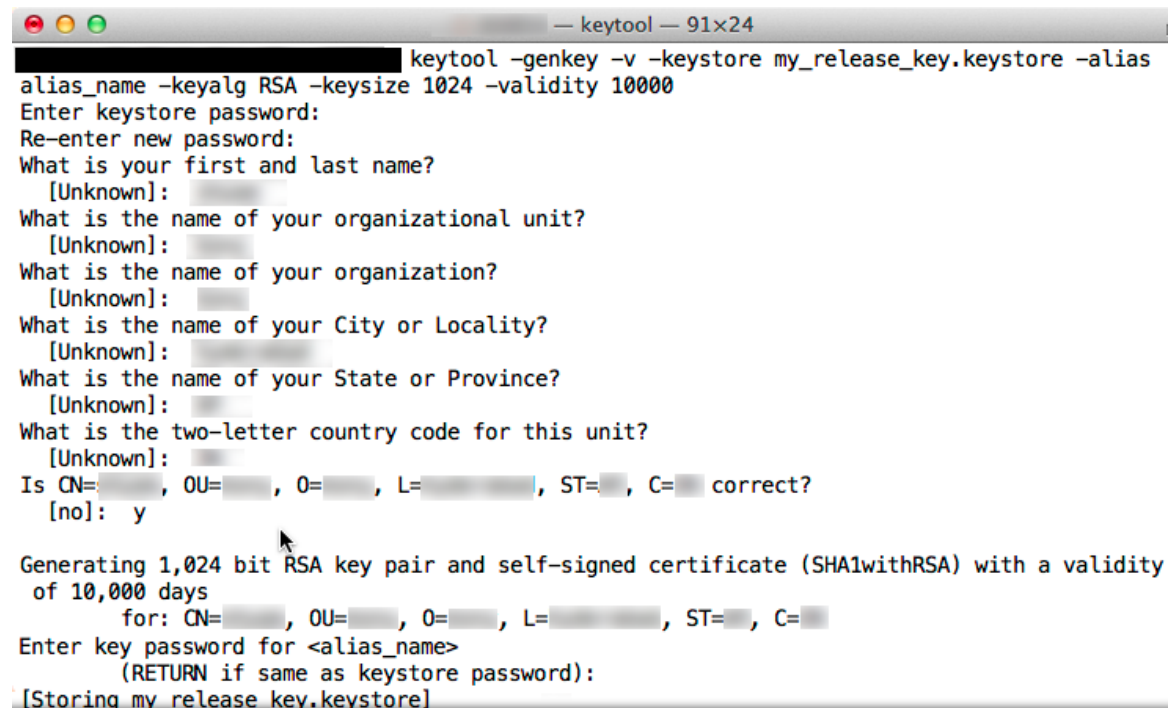
To create your API key, follow these steps:

1. Visit the APIs Console at <https://code.google.com/apis/console> and log in with your Google Account.
2. Click the **Services** link from the left-hand menu.
3. Activate the **Google Maps API v3** service.
4. Click the **API Access** link from the left-hand menu. Your API key is available from the **API Access** page, in the Simple API Access section. Maps API applications use the **Key**.

### 15.15.3 Creating an Android Key Store

1. Verify that your local computer you will use to generate the Key Store has Keytool installed and available. Keytool is a component of the [Java Development Kit \(JDK\)](#).
2. You will need to launch Keytool from the Java install bin directory (if Windows machine), or the root path (if Mac machine), as Keytool is included by default in Xcode on a Mac. You can also use Keytool from Java on a Mac if desired by using the `./keytool` command from the Java bin directory.
3. Enter this command: `keytool -genkey -v -keystore my_release_key.keystore -alias alias_name -keyalg RSA -keysize 1024 -validity 10000`

Note that 'my\_release\_key' and 'alias\_name' can be customized to preference.



```
keytool -genkey -v -keystore my_release_key.keystore -alias
alias_name -keyalg RSA -keysize 1024 -validity 10000
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]:
What is the name of your organizational unit?
[Unknown]:
What is the name of your organization?
[Unknown]:
What is the name of your City or Locality?
[Unknown]:
What is the name of your State or Province?
[Unknown]:
What is the two-letter country code for this unit?
[Unknown]:
Is CN=, OU=, O=, L=, ST=, C= correct?
[no]: y

Generating 1,024 bit RSA key pair and self-signed certificate (SHA1withRSA) with a validity
of 10,000 days
for: CN=, OU=, O=, L=, ST=, C=
Enter key password for <alias_name>
(RETURN if same as keystore password):
[Storing my_release_key.keystore]
```

4. Provide a password for the keystore and make note of this password.

```

keytool -genkey -v -keystore my_release_key.keystore -alias
alias name -keyalg RSA -keysize 1024 -validity 10000
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:
What is the name of your organizational unit?
  [Unknown]:
What is the name of your organization?
  [Unknown]:
What is the name of your City or Locality?
  [Unknown]:

```

5. Provide desired details for each question.

```

Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:
What is the name of your organizational unit?
  [Unknown]:
What is the name of your organization?
  [Unknown]:
What is the name of your City or Locality?
  [Unknown]:
What is the name of your State or Province?
  [Unknown]:
What is the two-letter country code for this unit?
  [Unknown]:
Is CN=, OU=, O=, L=, ST=, C= correct?
[no]: y

```

Generating 1,024 bit RSA key pair and self-signed certificate (SHA1withRSA) with a

6. A certificate will be generated with the Alias chosen in the Keytool command. Provide a password for this certificate, or hit enter to reuse the password chosen for the keystore in Step 4 above. The Keystore is now stored with the name chosen in the Keytool command.

```

Generating 1,024 bit RSA key pair and self-signed certificate (SHA1withRSA) with a validity
of 10,000 days
  for: CN=, OU=, O=, L=, ST=, C=
Enter key password for <alias name>
  (RETURN if same as keystore password):
[Storing my_release_key.keystore]

```

7. Store this keystore and info in a safe place to be used during your Management Cloud initial configuration.
8. You can display a certificate's SHA-1 fingerprint using the keytool program with the `-v` parameter.

### Command

```
keytool -list -v -keystore my_release_key.keystore -alias <alias name> -storepass <storepass name> -keypass <keypass name>
```

### Response

```
Owner: CN=ll, OU=ll, O=ll, L=ll, S=ll, C=ll
  Issuer: CN=ll, OU=ll, O=ll, L=ll, S=ll, C=ll
  Serial Number: 59092b34
  Valid from: Thu Sep 25 18:01:13 PDT 1997 until: Wed Dec 24
17:01:13 PST 1997
  Certificate Fingerprints:
    MD5: 11:81:AD:92:C8:E5:0E:A2:01:2E:D4:7A:D7:5F:07:6F
    SHA1:
45:B5:E4:6F:36:AD:0A:98:94:B4:02:66:2B:12:17:F2:56:26:A0:E0
```

Use the command above and copy the SHA1 from the response to your clipboard. The SHA1 is used in the steps below.

9. In your **Google account > API Access** page, you need to tag every enterprise app with a unique bundle identifier with the fingerprint generated. This needs to be done for both Android and iOS. This ensures that apps can show Google Maps and use other Google resources. To configure fingerprint, do the following:

**API Access**

To prevent abuse, Google places limits on API requests. Using a valid OAuth token or API key allows you to exceed

**Authorized API Access**

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private. A single project may contain up to 20 client IDs. [Learn more](#)

[Create an OAuth 2.0 client ID...](#)

**Simple API Access**

Use API keys to identify your project when you do not need to access user data. [Learn more](#)

[Create new Server key...](#) [Create new Browser key...](#) [Create new Android key...](#) [Create new iOS key...](#)

- Click the **Create new Android key** button. The Configure Android Key for API Project dialog appears.
- Enter the fingerprint details you copied from the previous [Step 8](#) in the above text box and click **Create**.

One SHA1 certificate fingerprint and package name (separated by a semicolon) per line.

Example:

```
45:B5:E4:6F:36:AD:0A:98:94:B4:02:66:2B:12:17:F2:56:26:A
0:E0;com.companyname.containerapp
```

The API Key is generated as shown below:

Key for Android applications	
API key	<code>AIzaSyA7VnL2FV80ggj0R40r_v020uk3ea007v</code>
Android applications	<code>71:9C:49:55:96:CC:A3:0B:97:11:62:9C:FD:77:BF:15:00:5C:66:8E</code> ; com.companyname.containerapp
Activation date	Mar 16, 2014 11:06 PM
Activated by	<a href="#">[User Name]</a> (you)
<a href="#">Edit allowed Android applications</a> <a href="#">Regenerate key</a> <a href="#">Delete</a>	

The API Key must be copied and provided in the EMM Console.

10. Provide the Google Maps Android API Key generated from [Step 9](#) along with other details in the **EMM Console > Application Settings > Certificates > Android** section.

**Android**

**GCM Key**

Google ID

GCM key for Android

Project number (Sender ID)

**Key Store Credentials**

Key Store

Key Store Pass Phrase

Certificate Alias

Certificate Pass Phrase

**Google Maps API**

Google Maps Android API V2 Key

**Note** Kony will re-sign the Android Launchpad app based on the details provided here. Please click 'Certificate Details' and make sure that the SHA1 fingerprint and the launchpad package name (com.kony.mdclient) are appropriately associated with your Google account. In case any Android apps are submitted to EMM and they use Maps, then please make sure the SHA1 fingerprint and application package name are associated to that app's corresponding Google account. The SHA1 fingerprint is replaced as part of the app signing process, hence this task is necessary.

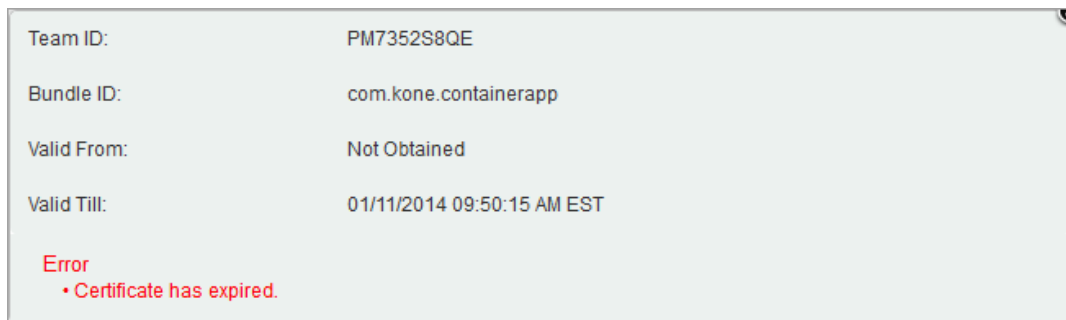
- a. **Google ID**: Enter your email account ID.
- b. **GCM Key for Android**: Enter the Google Cloud Messaging (GCM) Key.
- c. **Project Number (Sender ID)**: Enter the Sender ID.
- d. **Key Store**: Click the **+Add** button to select the certificate from its location and click the **Open** button. The selected certificate with size in KB appears next to Key Store label.

Click the **Close**



icon if you wish to close the selected certificate details.

- e. **Key Store Pass Phrase:** Enter the password that you need to enter while accessing the Certificate.
- f. **Certificate Alias:** Enter a suitable called name for the Certificate.  
The keystore protects each certificate with its individual password. For example, when you sign an Android application using the Key Store Pass Phrase, you are asked to select a keystore first, and then asked to select a single alias from that keystore. After providing the passwords for both the keystore and the chosen alias, the app is signed and the public key (the certificate) for that alias is embedded into the APK.
- g. **Certificate Pass Phrase:** Enter the password that you need to enter while accessing the Certificate.



- h. Click the **Certificate Details** button to view the respective certificate details and associated error, if any.
- i. Click the **Save** button to save the entered details. In the confirmation message that appears, click OK to return to the main page.

For more information about How to create Google Maps key, see <https://developers.google.com/maps/documentation/android/start>



## 15.16 Re-creating Android Certificates and Keys

You can not renew Android certificates and keys. You can renew an expired certificate or key by re-create them with previous certificate/key details. To recreate Android Certificates and Keys, you must navigate to your Android developer member center in an internet browser. and generate the key or certificate all over again.

If you renew any Android certificates or keys, the Launchpad and Child apps (if any) will be wrapped and signed with the new certificates and keys.

**Important:** Make sure that you are using the same google account you previously used to generate the certificates and keys.

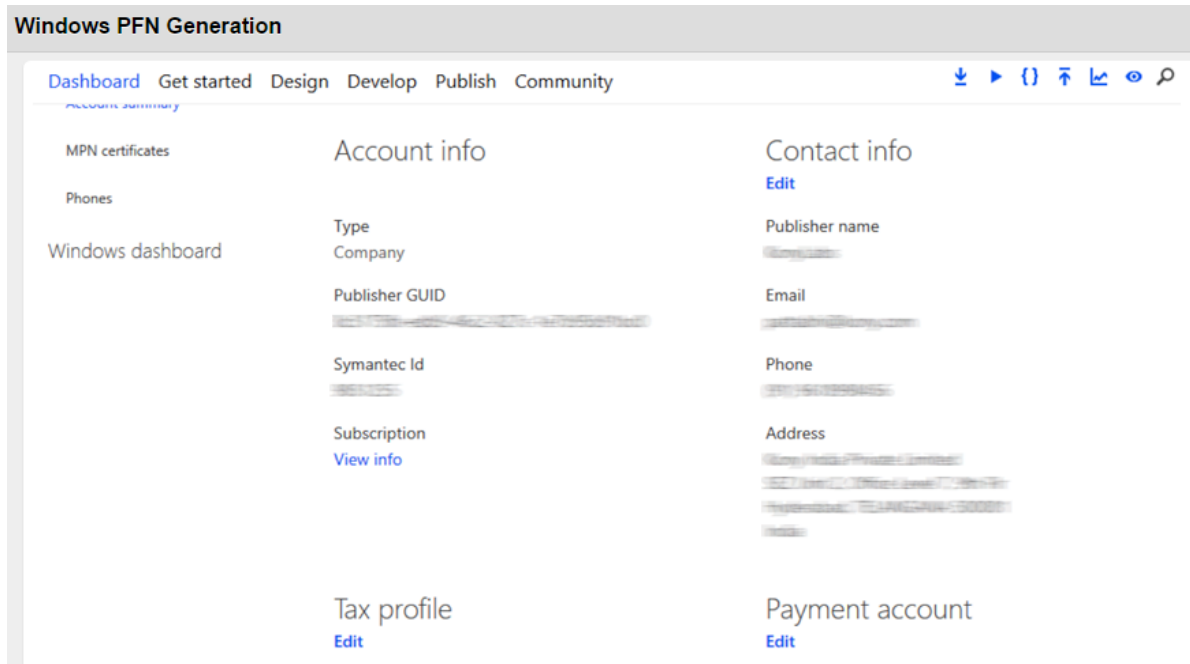
**Important:** If you create any of the certificate or key with a new google account, you must re-enroll your Android devices into Kony Management Suite EMM console.

## 16. Generating Package Family Name

To create Package Family name for Windows Phone 8.x devices and Windows 8.1. follow these steps:

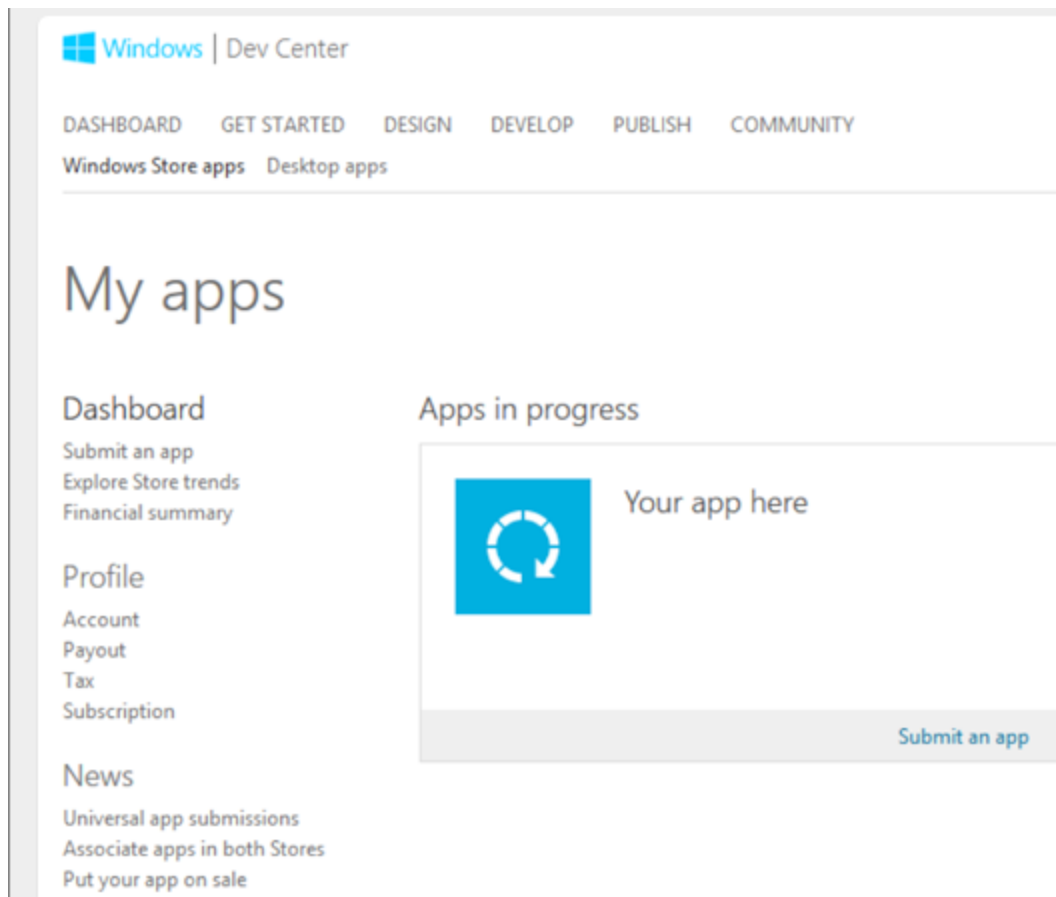
1. In a browser, navigate to <http://dev.windows.com/en-us/> and click on **Sign in**. The Sign in page appears.
2. Enter your windows credentials and click **Sign in**.  
The Windows Developer page appears.
3. Click **Dashboard**. Choose your dashboard page appears.
4. Click **Windows Phone Store**. If you are not registered as an app developer already, you will be prompted to register as an app developer. F
5. Click **Accept and continue**. The Getting Started page appears.
6. Click **Join Now**. The Account Type page appears.
7. From the **Country/region** drop-down list, select your country. Pick account type option appears. Two types of accounts are available. Individual and Corporate.
8. Select your account type and click **Enroll now**. The **Account info** page appears.
9. Provide all required information in the Account info form.
10. Click **Next**. The Approver info page appears.
11. Provide all required information in the Approver info form.
12. Click **Next**. The Agreement page appears.
13. Accept Terms and Conditions and then click **Next**. The Payment options page appears.
14. Select **Pay for the account** and then click **Next**. The Purchase page appears.
15. Review your purchase details and click **Purchase**. The You're done! page appears.

16. Click **Done**. Dashboard details appear.

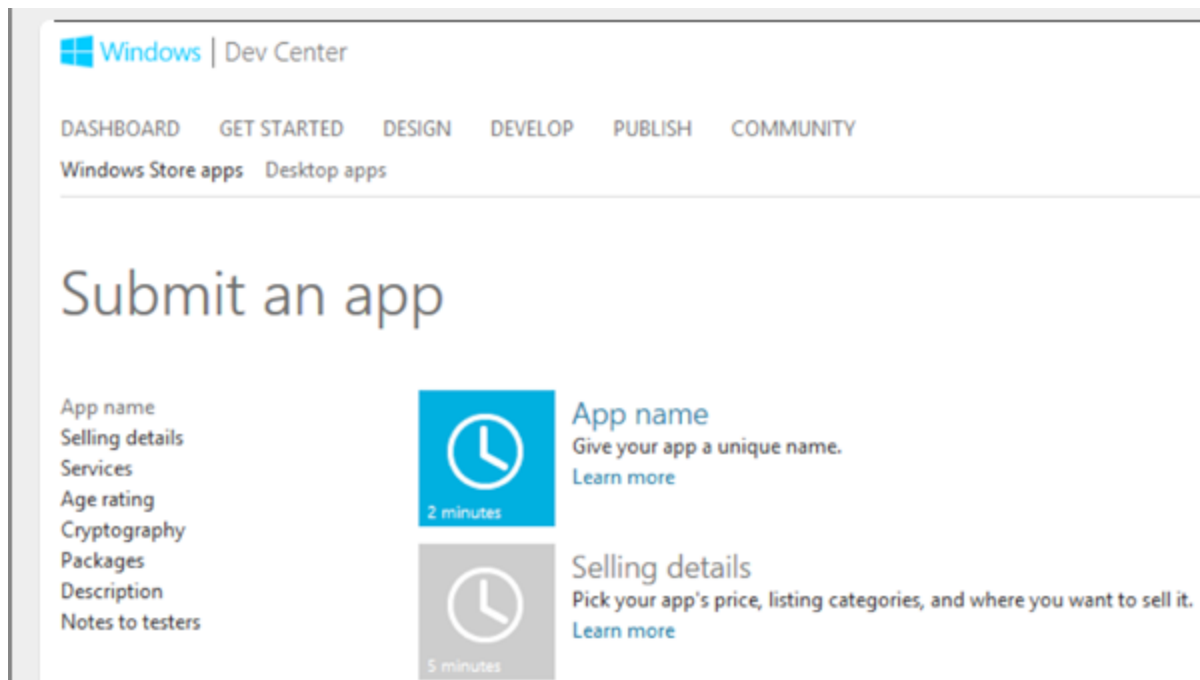


17. Make a note of **Publisher GUID** details.

18. In the Dashboard, click **Windows Store apps**. Windows store apps details appear.



19. Click **Submit an app**. The Submit an app page appears.



20. Click **App Name**. App name details appear.
21. In the App name text box, enter a unique app name and click **Reserve app name**. App name details appear.
22. Click **Save**. Confirmation message displays and App name details appear in the dashboard.
23. Click **Services**. Services details appear.
24. Click **Live Services site**. Configure WNS details in this page.

25. Navigate to **Settings > App Settings**.

Home My apps Docs Downloads Support

My applications > KonyDevLaunchpad > App Settings

## KonyDevLaunchpad

Settings

- Basic Information
- API Settings
- App Settings**
- Localization

To protect your app's security, Windows Push Notification Services (WNS) and services using Microsoft account use client secrets to authenticate the communications from your server.

**Package SID:**  
ms-app://s-1-15-2-2010508214-3624698978-1756763660-270412976-2102698090-2867670812-3063674740  
This is the unique identifier for your Windows Store app.  
[Link to different app](#)

**Application identity:**  
<Identity Name="A2DB1A53.KonyDevLaunchpad"  
Publisher="CN=4D268CEA-C9A4-43E7-B37F-BDB12E1C8442" />

To set your application's identity values manually, open the AppManifest.xml file in a text editor and set these attributes of the <identity> element using the values shown here.

**Client ID:**  
000000048127023  
This is a unique identifier for your application.

**Client secret:**  
TyC0AWtNokVkuKbXuMCzv/kJSK4rBRcA  
For security purposes, don't share your client secret with anyone.

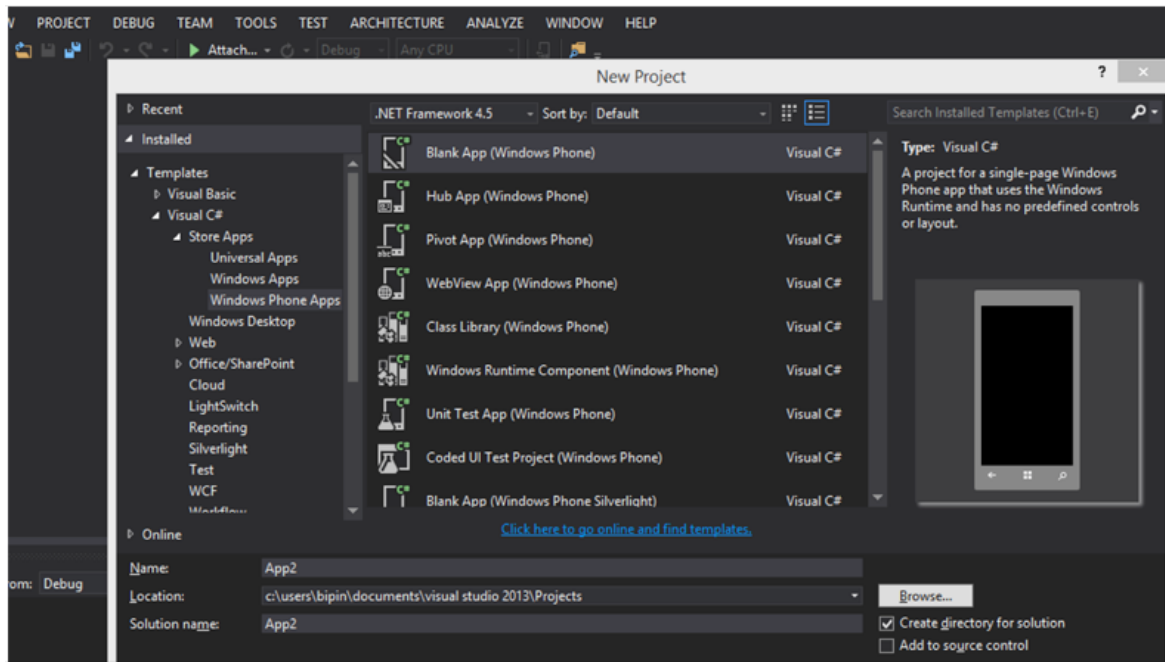
If your client secret has been compromised or your organization requires that you periodically change client secrets, create a new client secret

26. Make a note of **Package SID** and **Client Secret** details. You need these details to enter in the EMM portal.

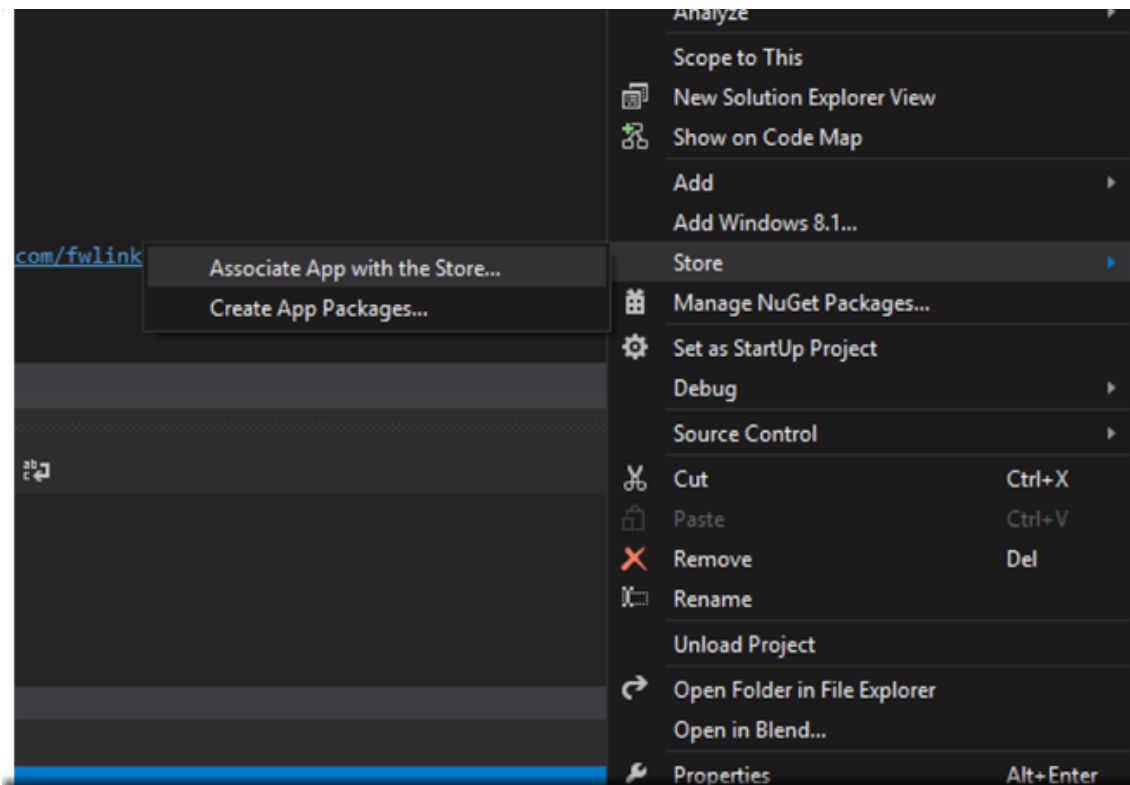
27. Open Microsoft Visual Studio.

28. Open a new project. The New project page appears.

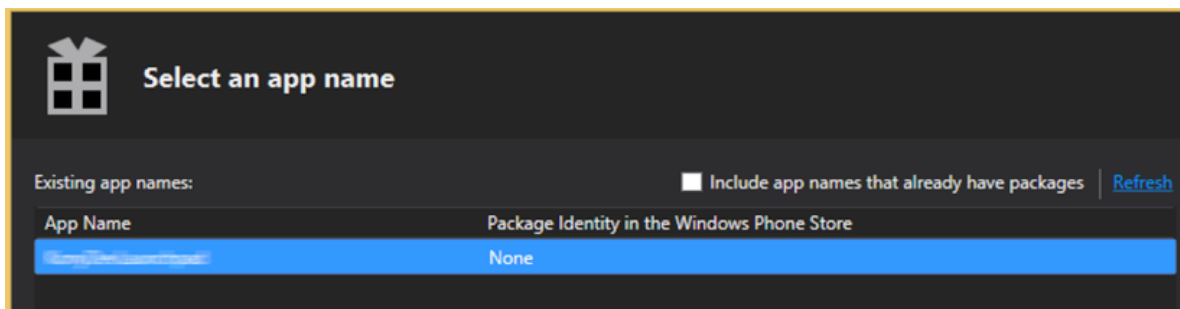
29. In the left pane, choose Visual C#, then Store Apps, and then Windows Phone Apps.



30. Choose Blank App (Windows Phone).
31. In the Name text box, provide the app name and then click OK.
32. In the right pane, right click on the app and select **Associate App with the Store**. The Associate your app page appears.



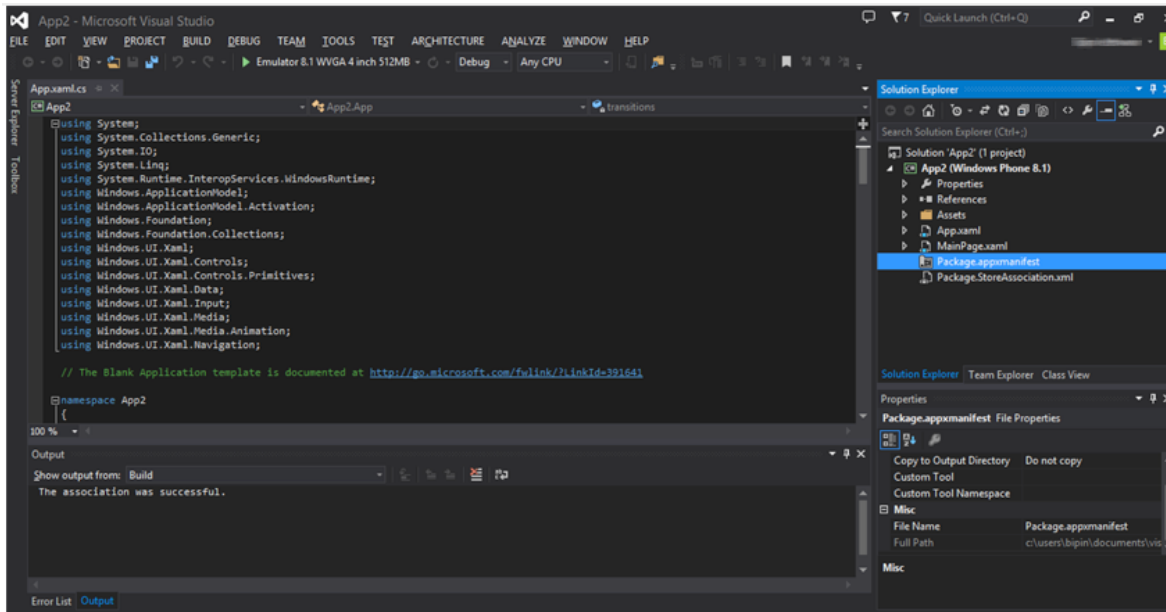
33. Click **Next**. The Sign in to the Windows Phone Store page appears.
34. Enter your Microsoft credentials and click **Sign in**. The Select an app name page appears.



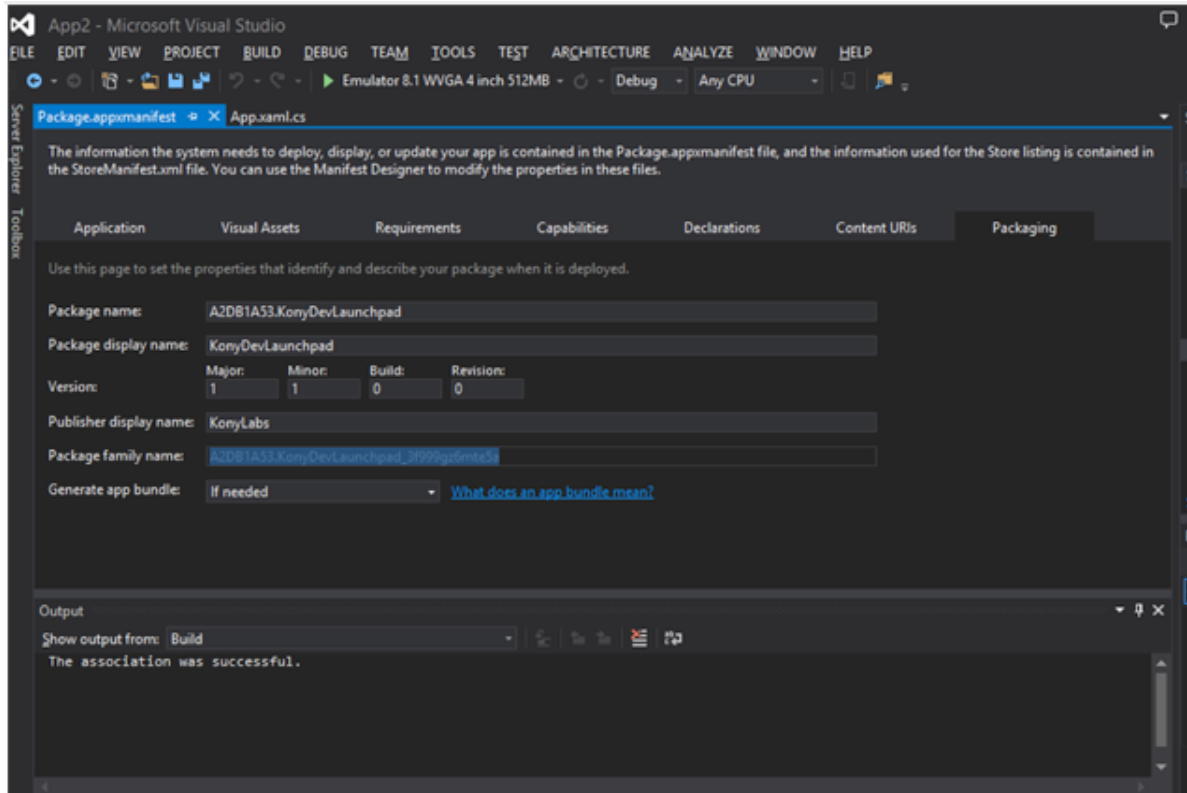
35. Select the app name which you reserved and click **Next**. The Associate you app page appears.



36. Review the details and click **Associate**. A success message appears.



37. In the app, open **Package.appxmanifest** file. Package Family Name is highlighted.



38. Copy Package Family Name details. You need them later to enter into the EMM console.