Building, Archiving, and Submitting Your App

iOS and OS X tips, tricks, and best practices

Session 414

Rob Marini
Xcode

These are confidential sessions—please refrain from streaming, blogging, or taking pictures

Distribution

Roadmap

Distributing Outside the App Store

Distributing in the App Store

iOS Application Best Practices

Roadmap

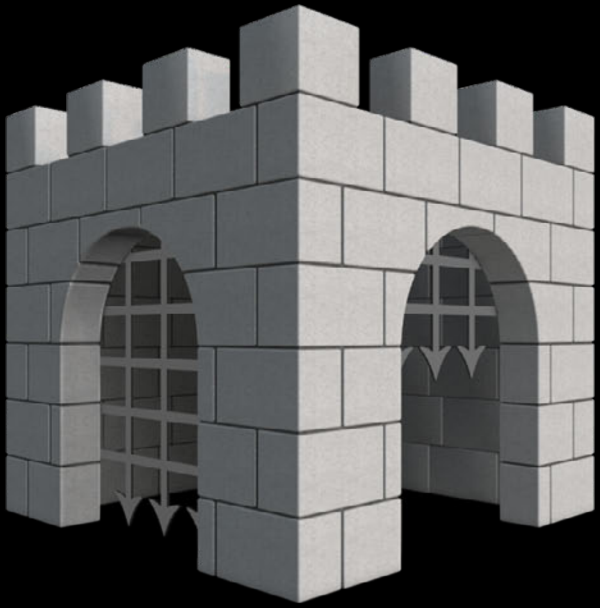
Roadmap

OS X

Distributing outside the App Store

Distributing Outside the App Store

Gatekeeper



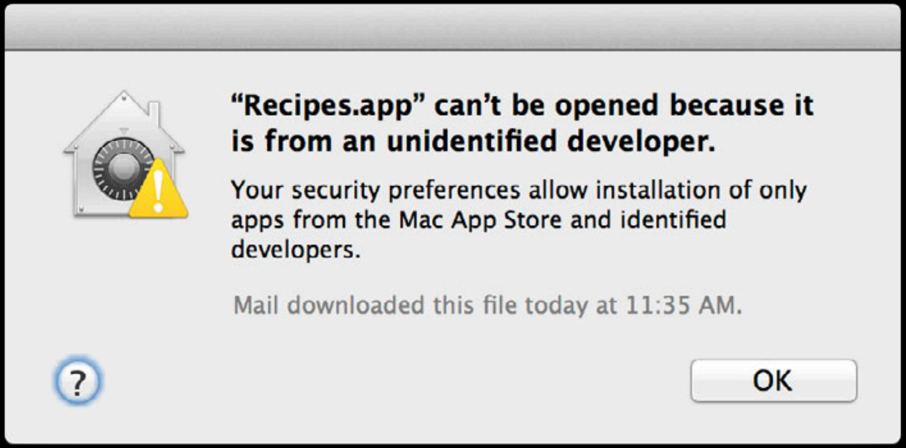
Distributing Outside the App Store

Gatekeeper

- Turned on by default in Mountain Lion
- Protect users from malicious applications
- Mac App Store and Developer ID

Distributing Outside the App Store

Gatekeeper



“Recipes.app” can’t be opened because it is from an unidentified developer.

Your security preferences allow installation of only apps from the Mac App Store and identified developers.

Mail downloaded this file today at 11:35 AM.

Distributing Outside the App Store

Developer ID

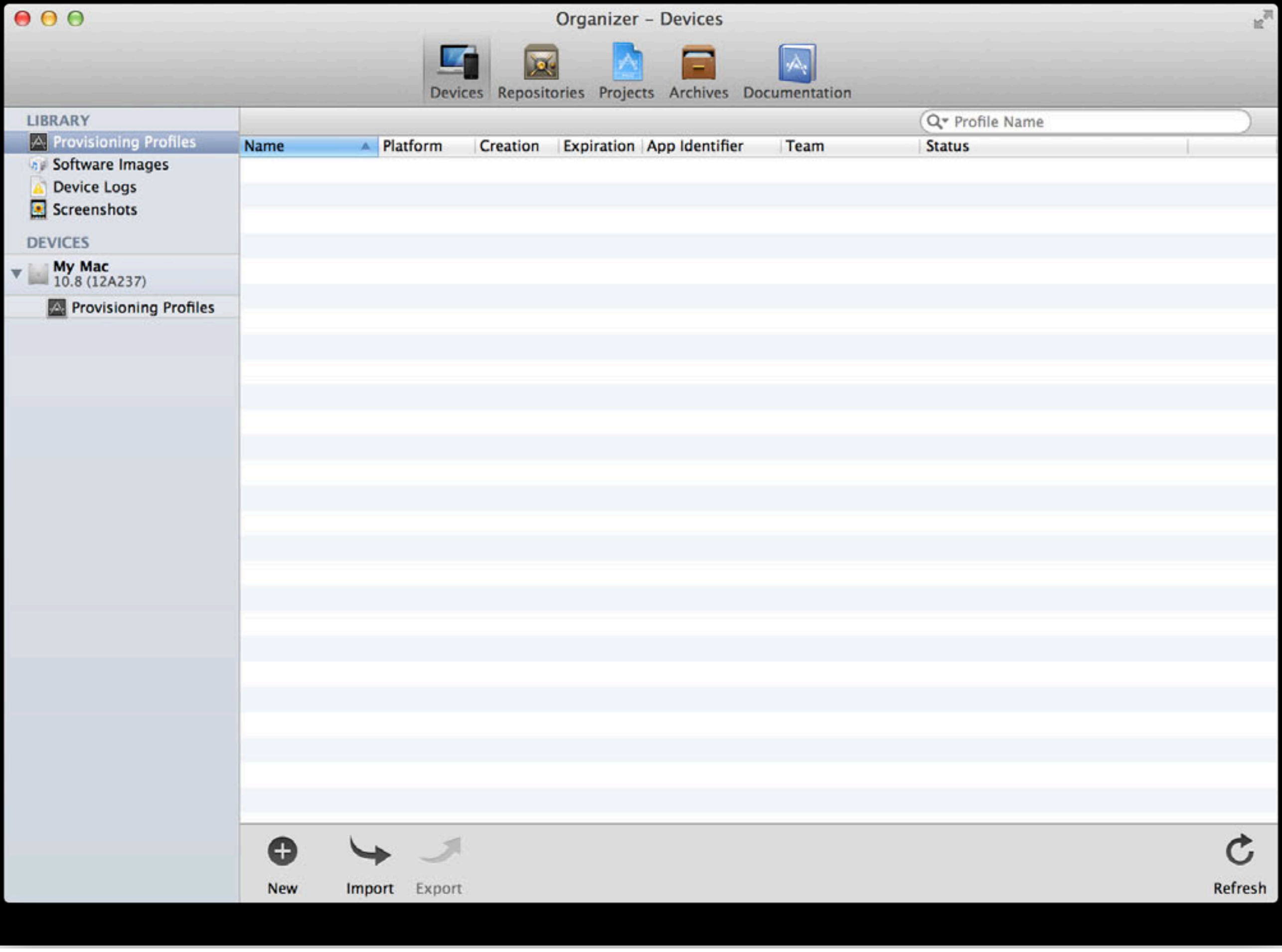
Distributing Outside the App Store

Developer ID

- Identifies you as being trusted by Apple
- OS ensures that executable content comes only from you
- Built-in support in Xcode

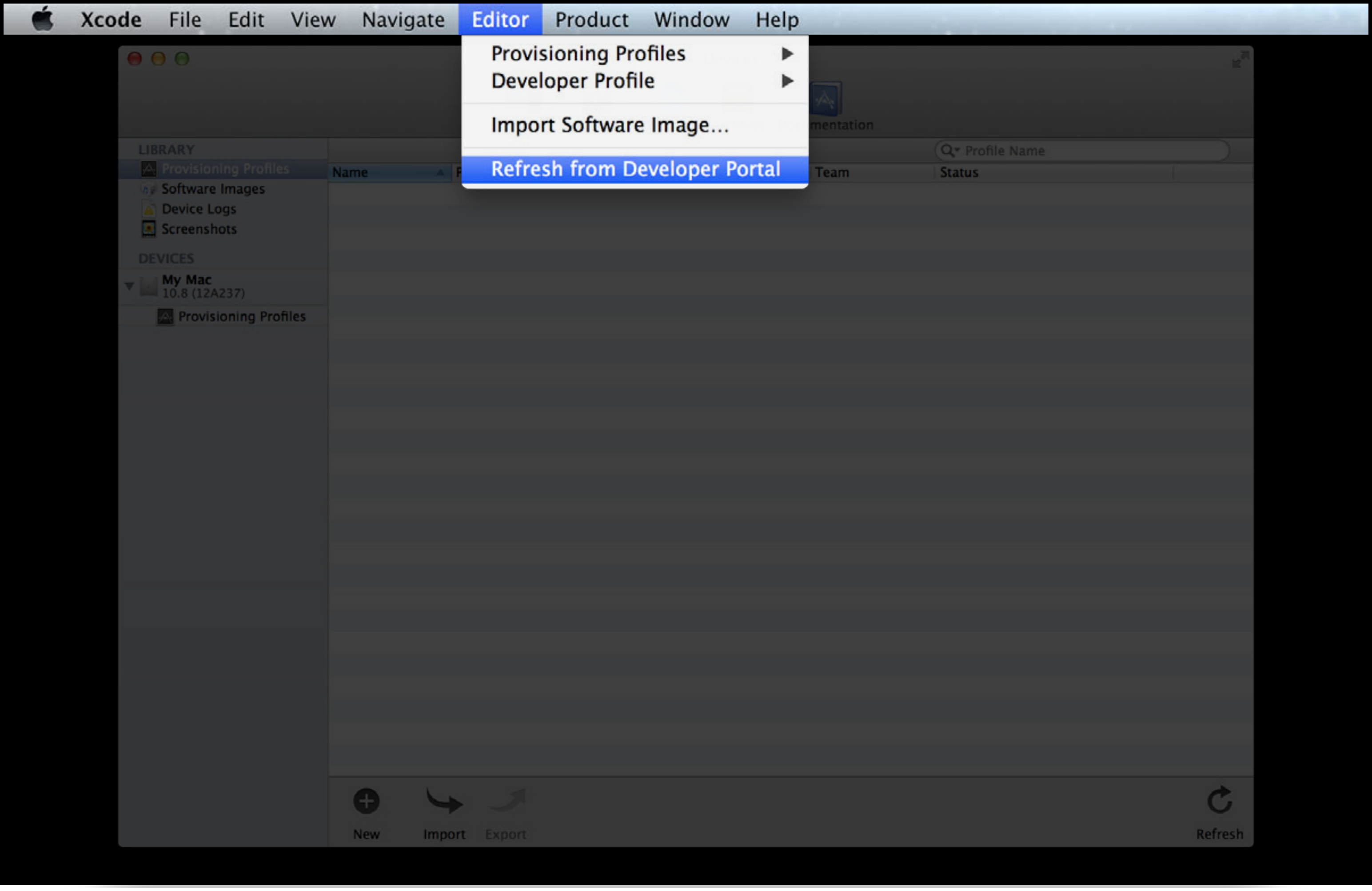
Distributing Outside the App Store

Obtaining a Developer ID



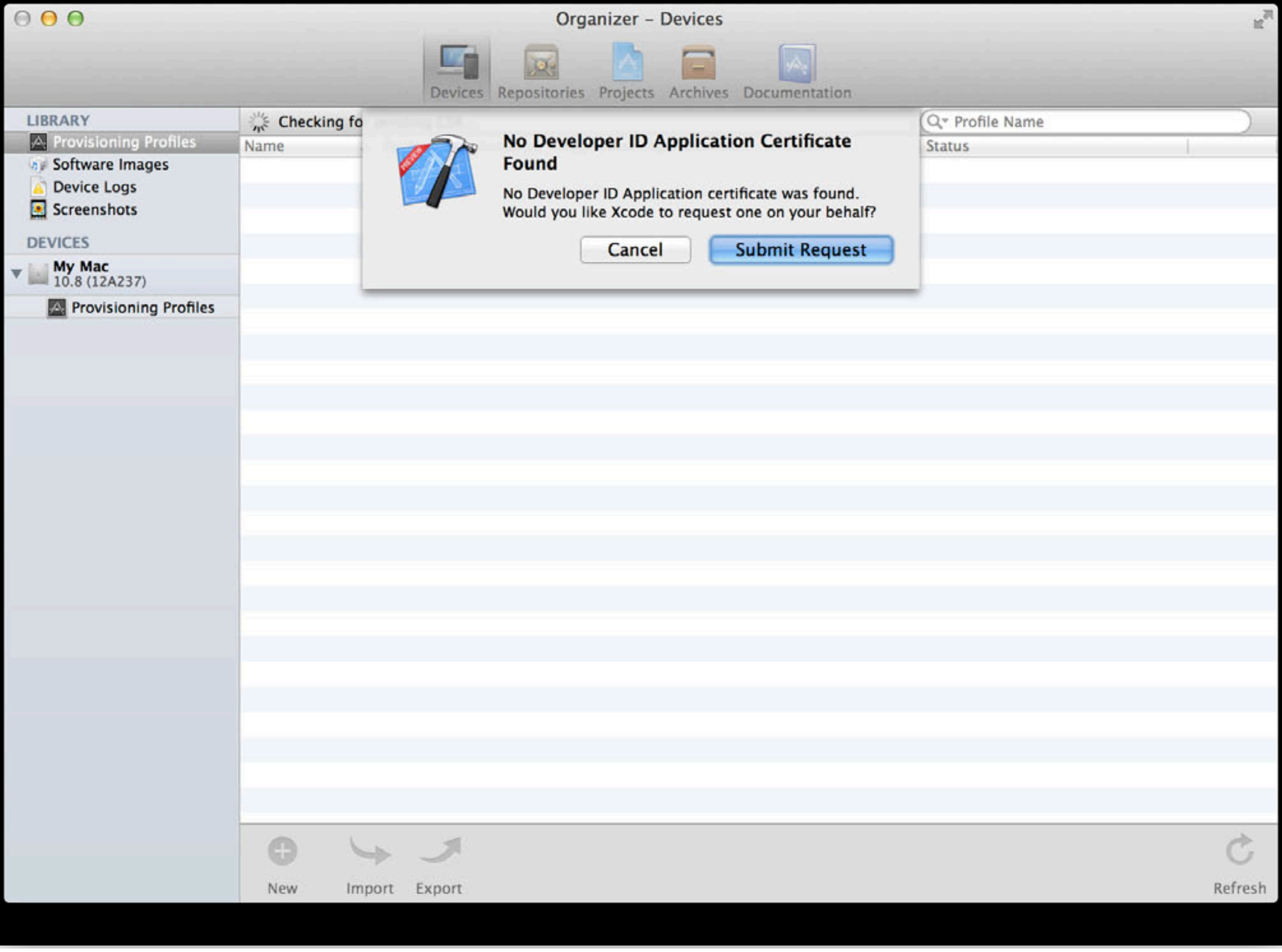
Distributing Outside the App Store

Obtaining a Developer ID



Distributing Outside the App Store

Obtaining a Developer ID

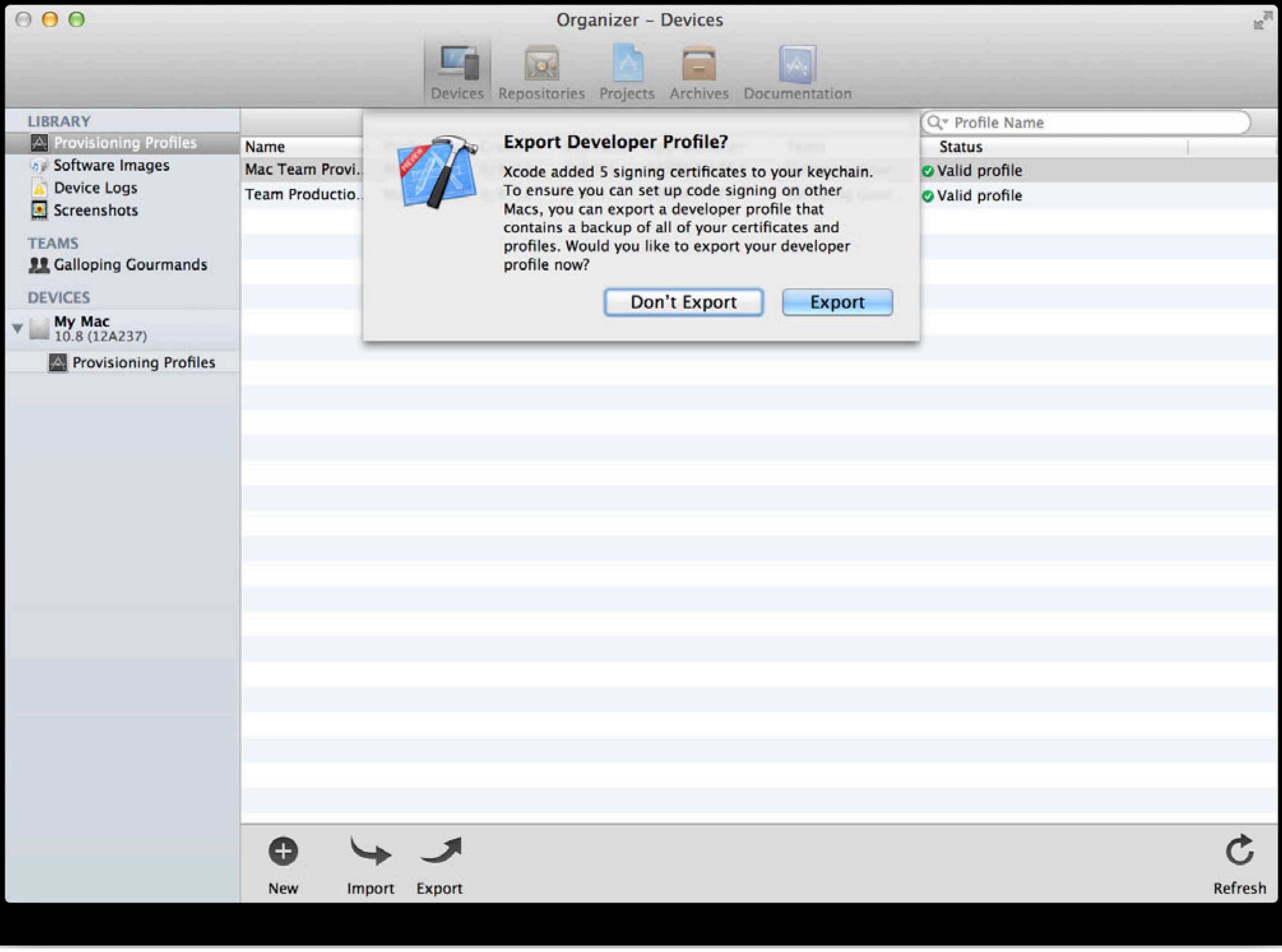


No Developer ID Application Certificate Found

No Developer ID Application certificate was found.
Would you like Xcode to request one on your behalf?

Distributing Outside the App Store

Obtaining a Developer ID



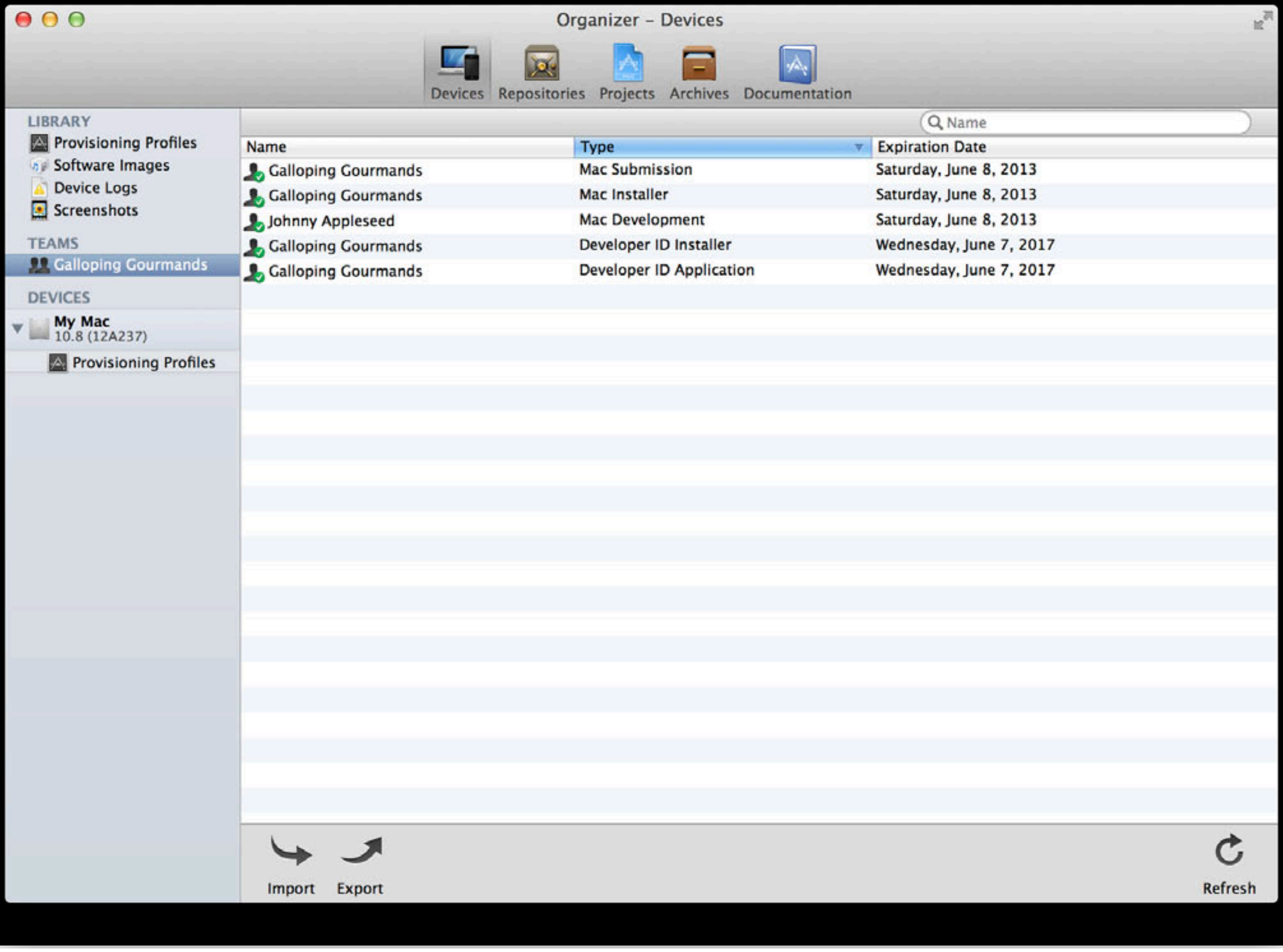
Name	Status
Mac Team Provi...	Valid profile
Team Productio...	Valid profile

Export Developer Profile?

Xcode added 5 signing certificates to your keychain. To ensure you can set up code signing on other Macs, you can export a developer profile that contains a backup of all of your certificates and profiles. Would you like to export your developer profile now?

Distributing Outside the App Store

Obtaining a Developer ID



Name	Type	Expiration Date
Galloping Gourmands	Mac Submission	Saturday, June 8, 2013
Galloping Gourmands	Mac Installer	Saturday, June 8, 2013
Johnny Appleseed	Mac Development	Saturday, June 8, 2013
Galloping Gourmands	Developer ID Installer	Wednesday, June 7, 2017
Galloping Gourmands	Developer ID Application	Wednesday, June 7, 2017

Distributing Outside the App Store

Types of IDs

- Developer ID Application
  - For signing .app bundles
- Developer ID Installer
  - For signing installer packages containing .app bundles

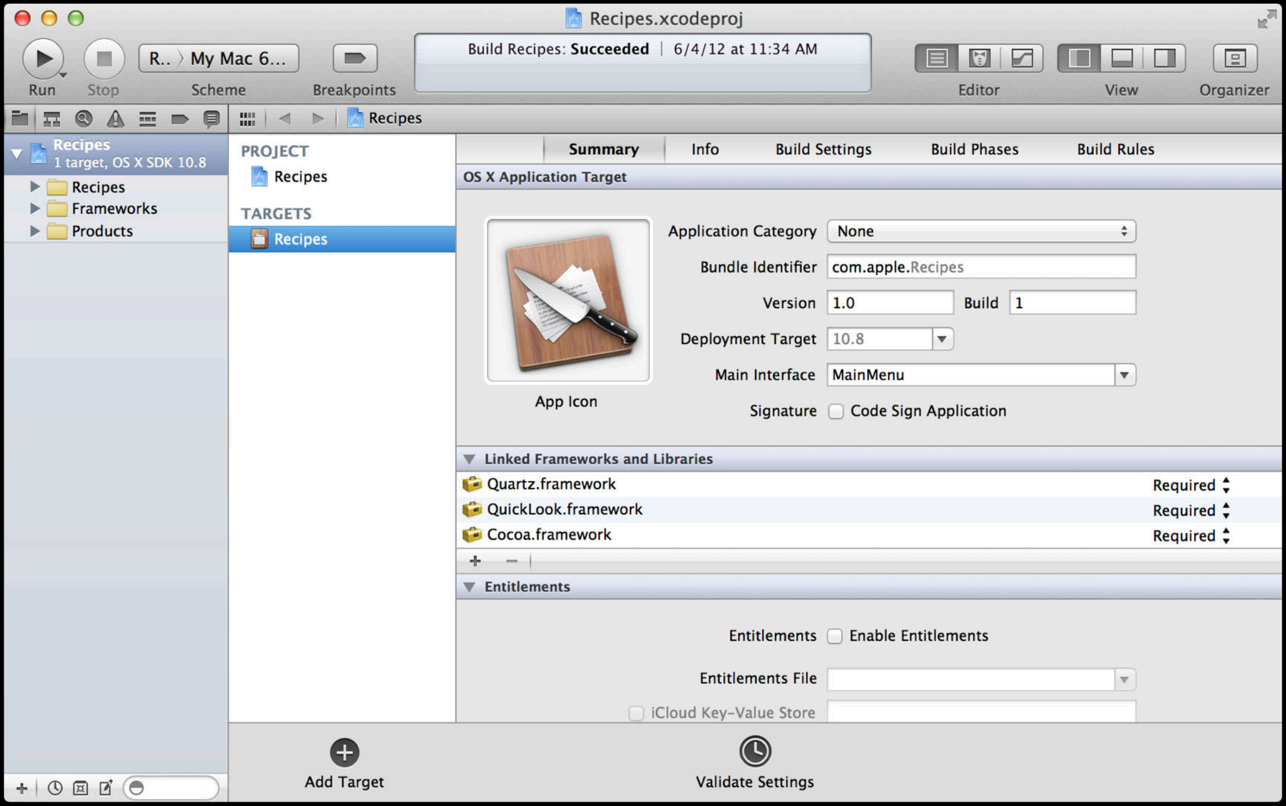
Distributing Outside the App Store

Signing an application

- Test code signed application under the debugger
- Sign your application to profile launch times

Distributing Outside the App Store

Signing an application



Build Recipes: **Succeeded** | 6/4/12 at 11:34 AM

Application Category: None

Bundle Identifier: com.apple.Recipes

Version: 1.0 Build: 1

Deployment Target: 10.8

Main Interface: MainMenu

Signature: □ Code Sign Application

Quartz.framework	Required
QuickLook.framework	Required
Cocoa.framework	Required

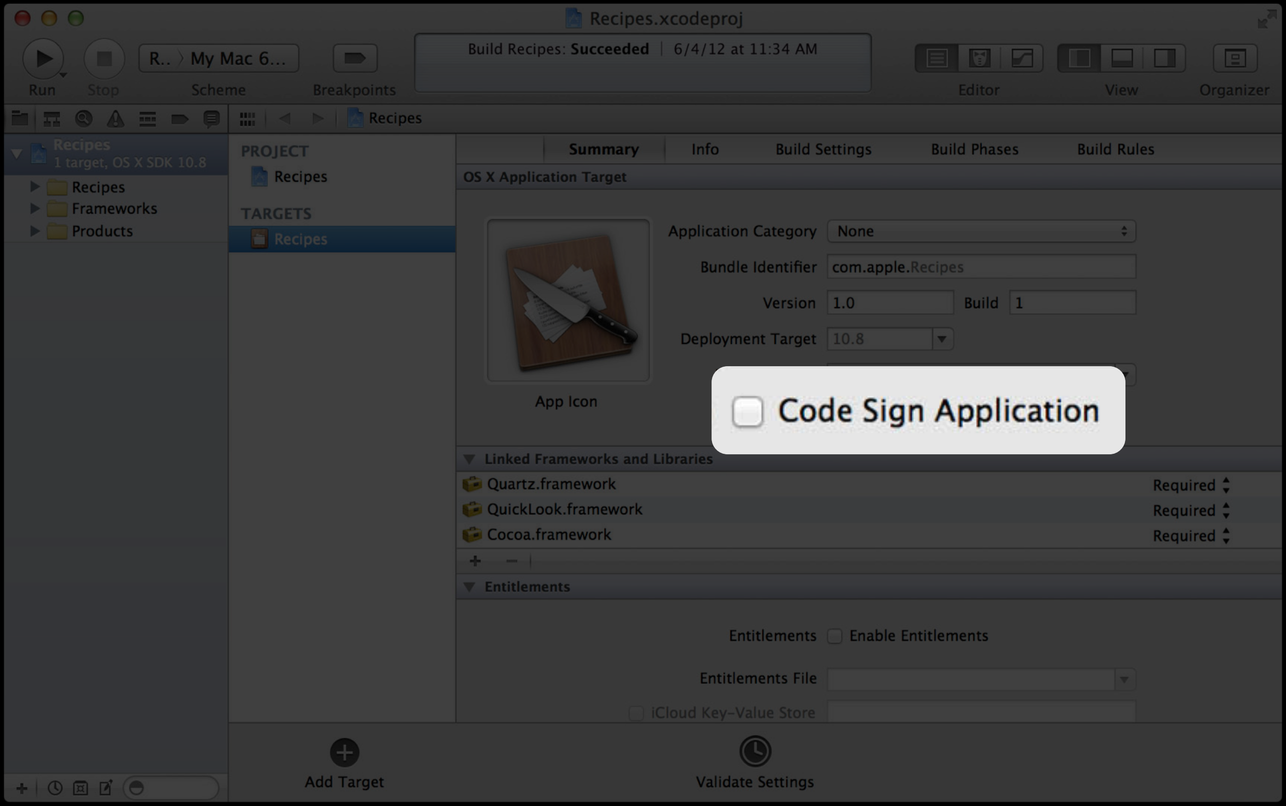
Entitlements: □ Enable Entitlements

Entitlements File:

□ iCloud Key-Value Store

Distributing Outside the App Store

Signing an application



Recipes.xcodeproj

Build Recipes: **Succeeded** | 6/4/12 at 11:34 AM

OS X Application Target

Application Category: None

Bundle Identifier: com.apple.Recipes

Version: 1.0 Build: 1

Deployment Target: 10.8

□ Code Sign Application

Linked Frameworks and Libraries

Quartz.framework	Required
QuickLook.framework	Required
Cocoa.framework	Required

Entitlements

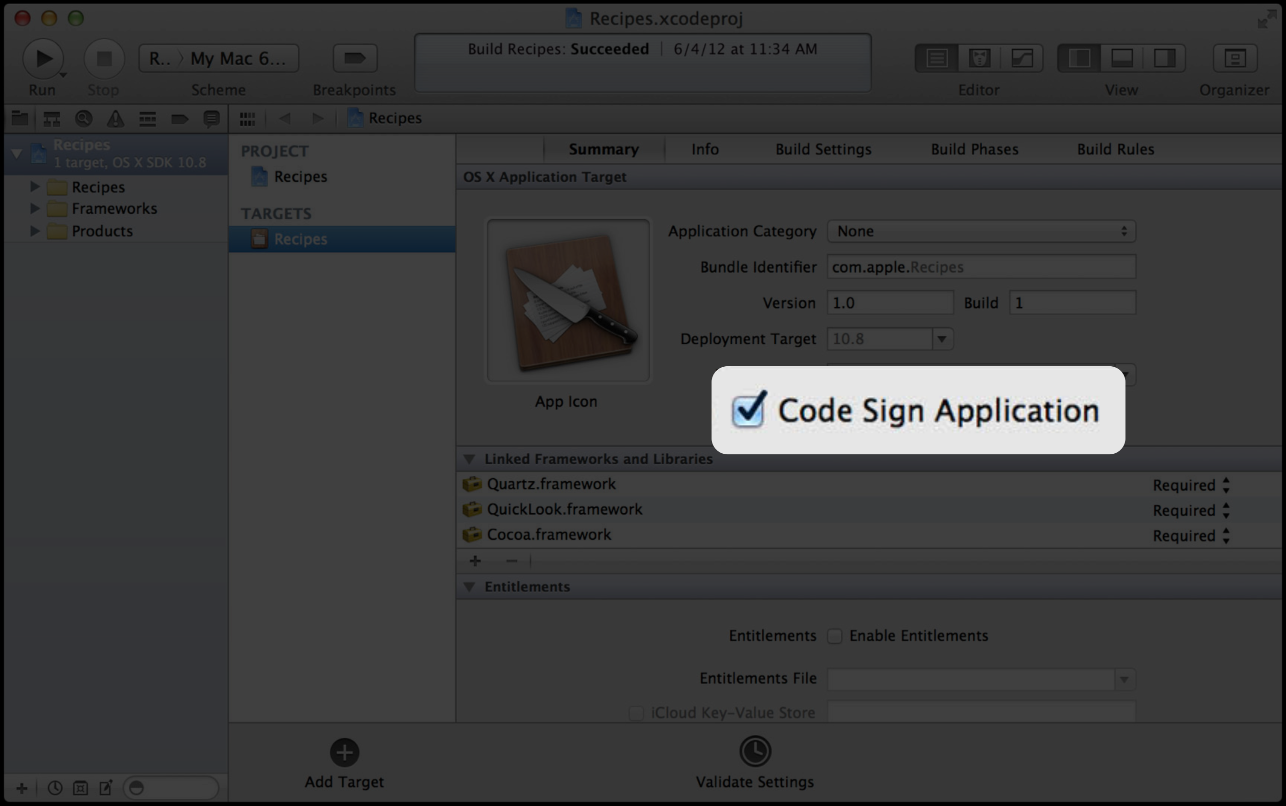
Entitlements: □ Enable Entitlements

Entitlements File

□ iCloud Key-Value Store

Distributing Outside the App Store

Signing an application



Recipes.xcodeproj

Build Recipes: **Succeeded** | 6/4/12 at 11:34 AM

OS X Application Target

Application Category: None

Bundle Identifier: com.apple.Recipes

Version: 1.0 Build: 1

Deployment Target: 10.8

☑ Code Sign Application

Linked Frameworks and Libraries

Quartz.framework	Required
QuickLook.framework	Required
Cocoa.framework	Required

Entitlements

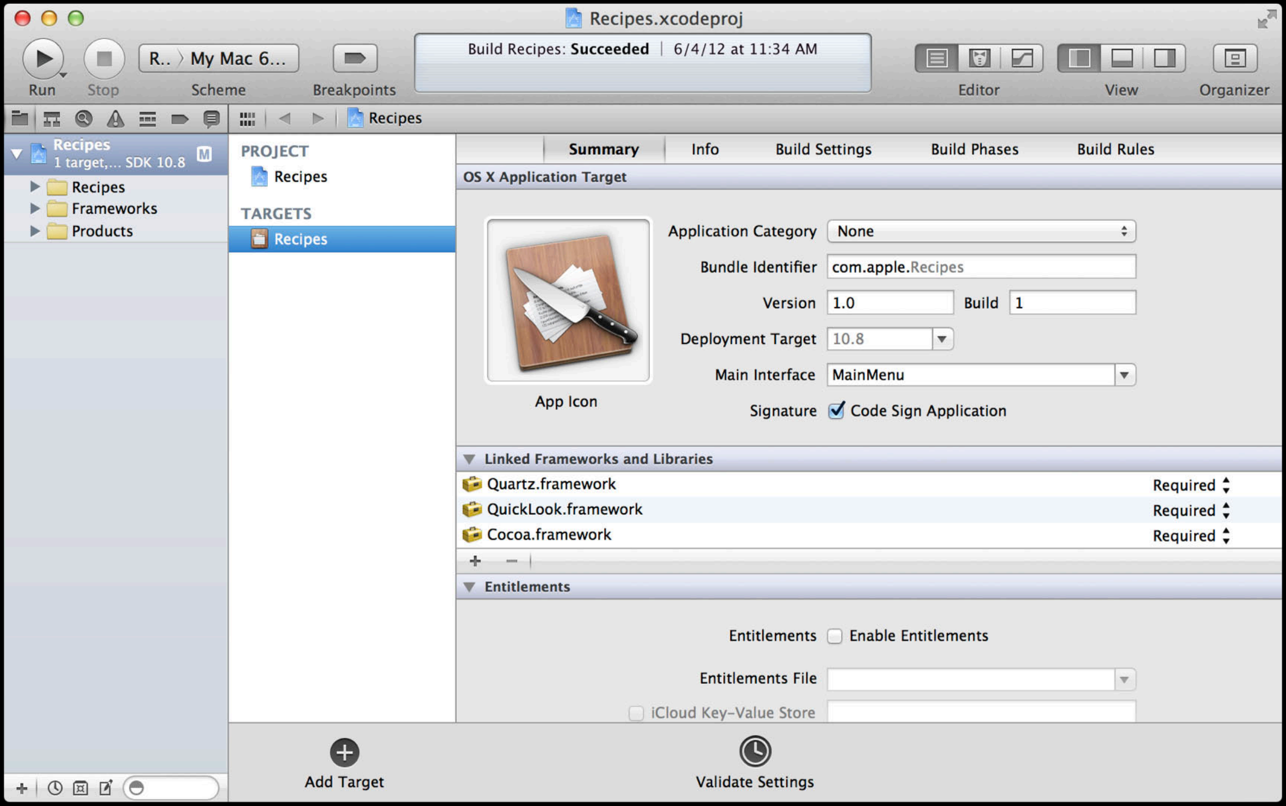
Entitlements: □ Enable Entitlements

Entitlements File

□ iCloud Key-Value Store

Distributing Outside the App Store

Signing an application



OS X Application Target

Application Category: None

Bundle Identifier: com.apple.Recipes

Version: 1.0 Build: 1

Deployment Target: 10.8

Main Interface: MainMenu

Signature: ☑ Code Sign Application

Linked Frameworks and Libraries

Quartz.framework	Required
QuickLook.framework	Required
Cocoa.framework	Required

Entitlements

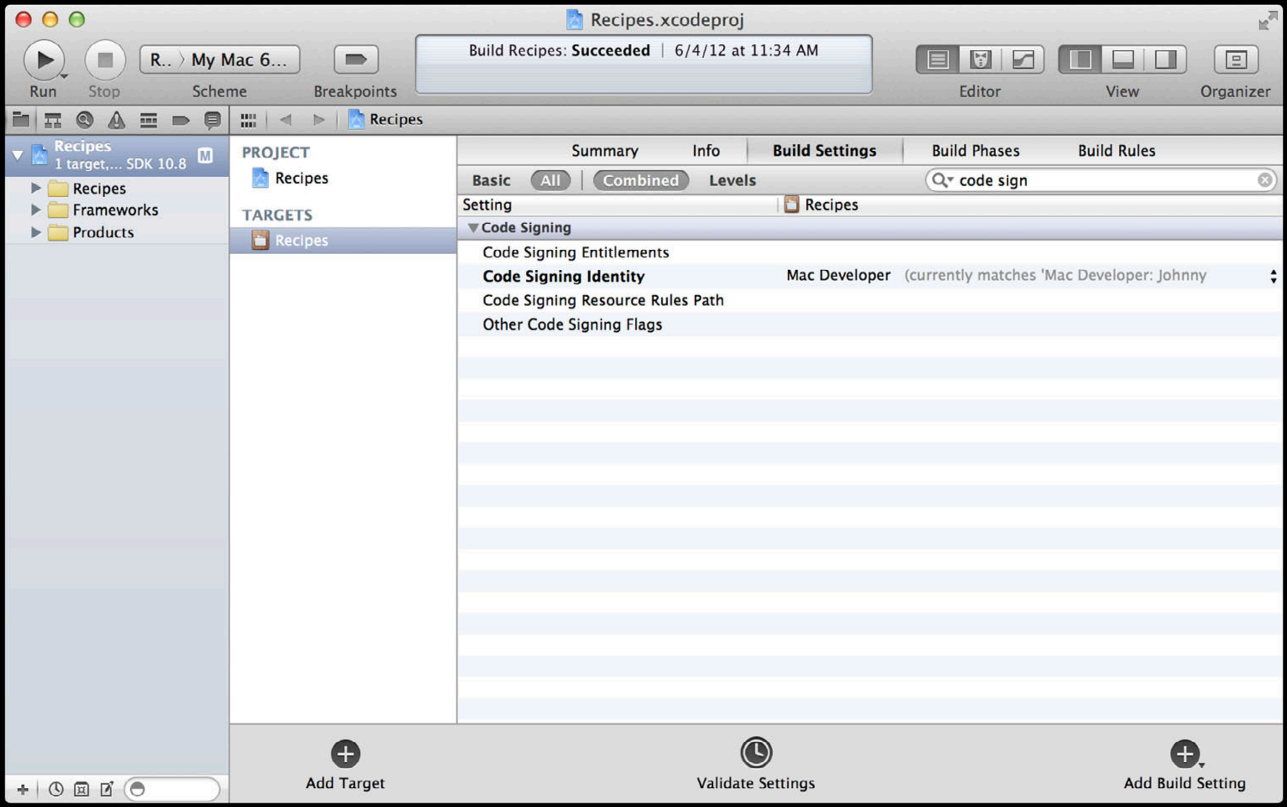
Entitlements: □ Enable Entitlements

Entitlements File:

□ iCloud Key-Value Store

Distributing Outside the App Store

Signing an application

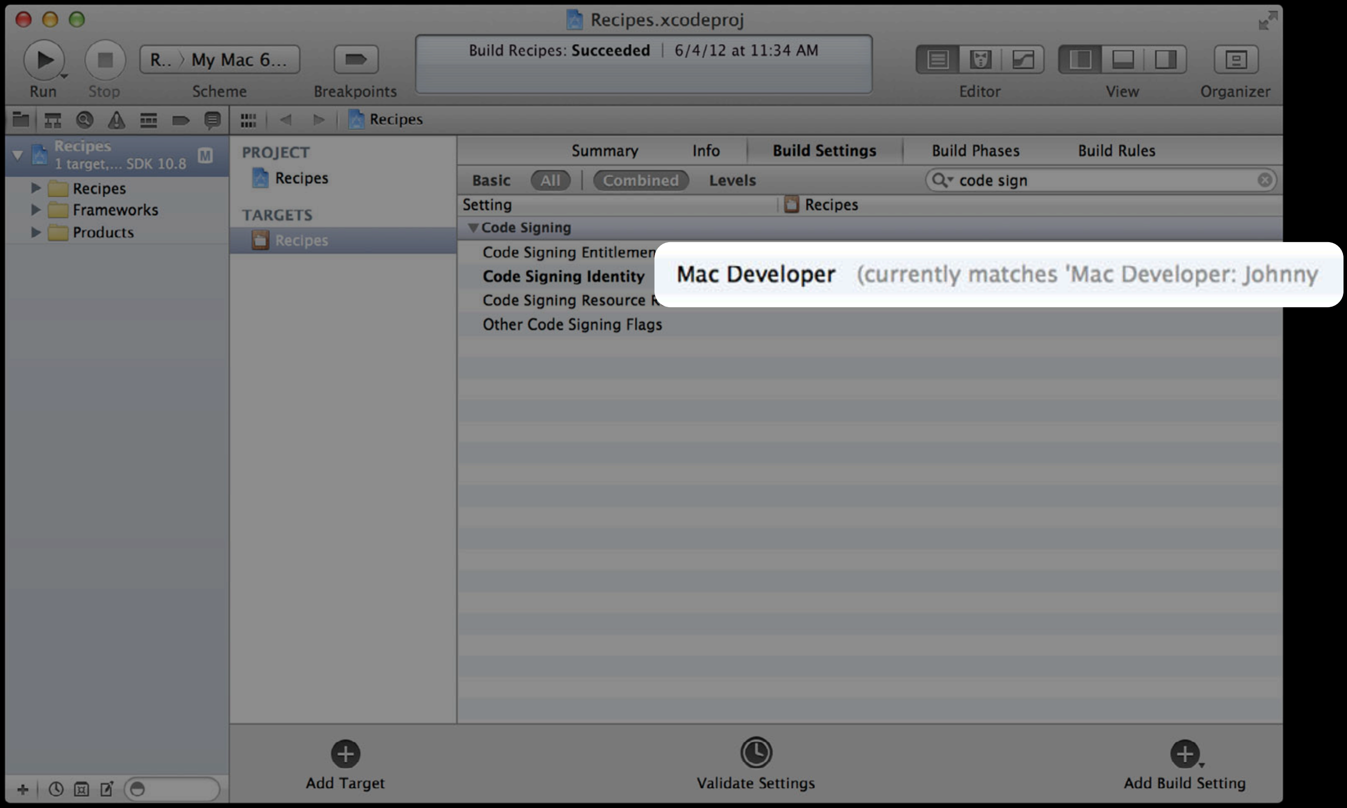


Build Recipes: **Succeeded** | 6/4/12 at 11:34 AM

Setting	Recipes
▼Code Signing	
Code Signing Entitlements	
Code Signing Identity	Mac Developer (currently matches 'Mac Developer: Johnny
Code Signing Resource Rules Path	
Other Code Signing Flags	

Distributing Outside the App Store

Signing an application

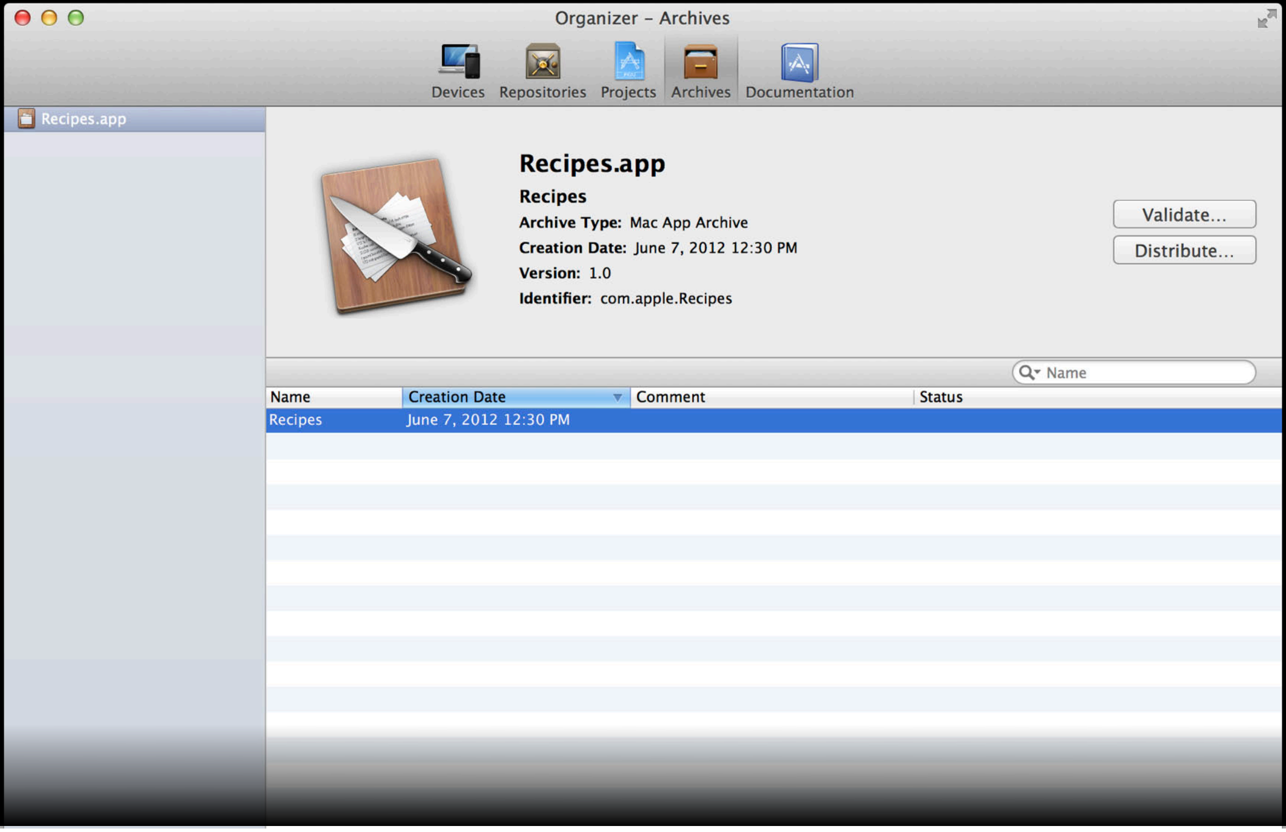


Code Signing Identity: Mac Developer (currently matches 'Mac Developer: Johnny

Distributing Outside the App Store

Exporting an application

- Sign an archive of your application with your Developer ID



Recipes.app

Recipes

Archive Type: Mac App Archive

Creation Date: June 7, 2012 12:30 PM

Version: 1.0

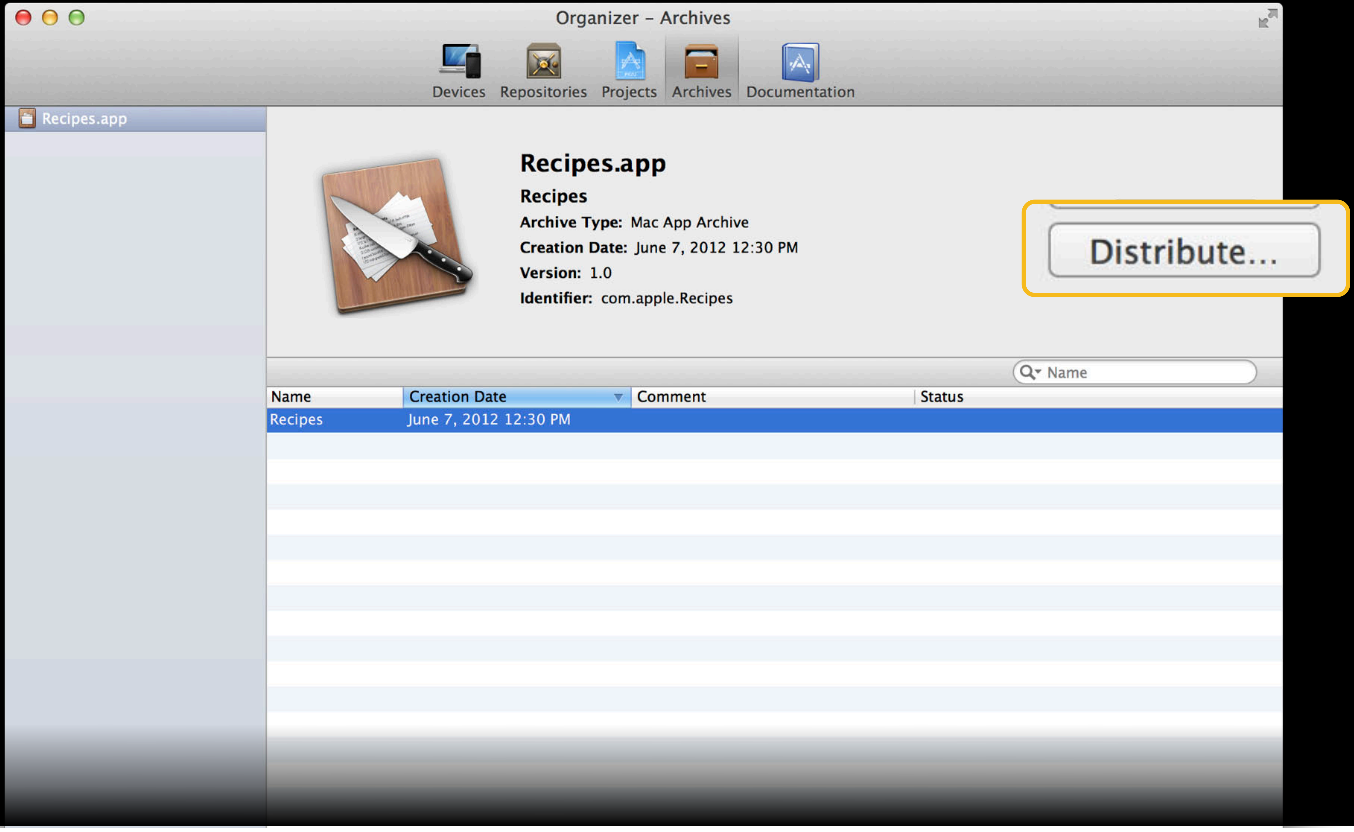
Identifier: com.apple.Recipes

Name	Creation Date	Comment	Status
Recipes	June 7, 2012 12:30 PM		

Distributing Outside the App Store

Exporting an application

- Sign an archive of your application with your Developer ID



Recipes.app

Recipes

Archive Type: Mac App Archive

Creation Date: June 7, 2012 12:30 PM

Version: 1.0

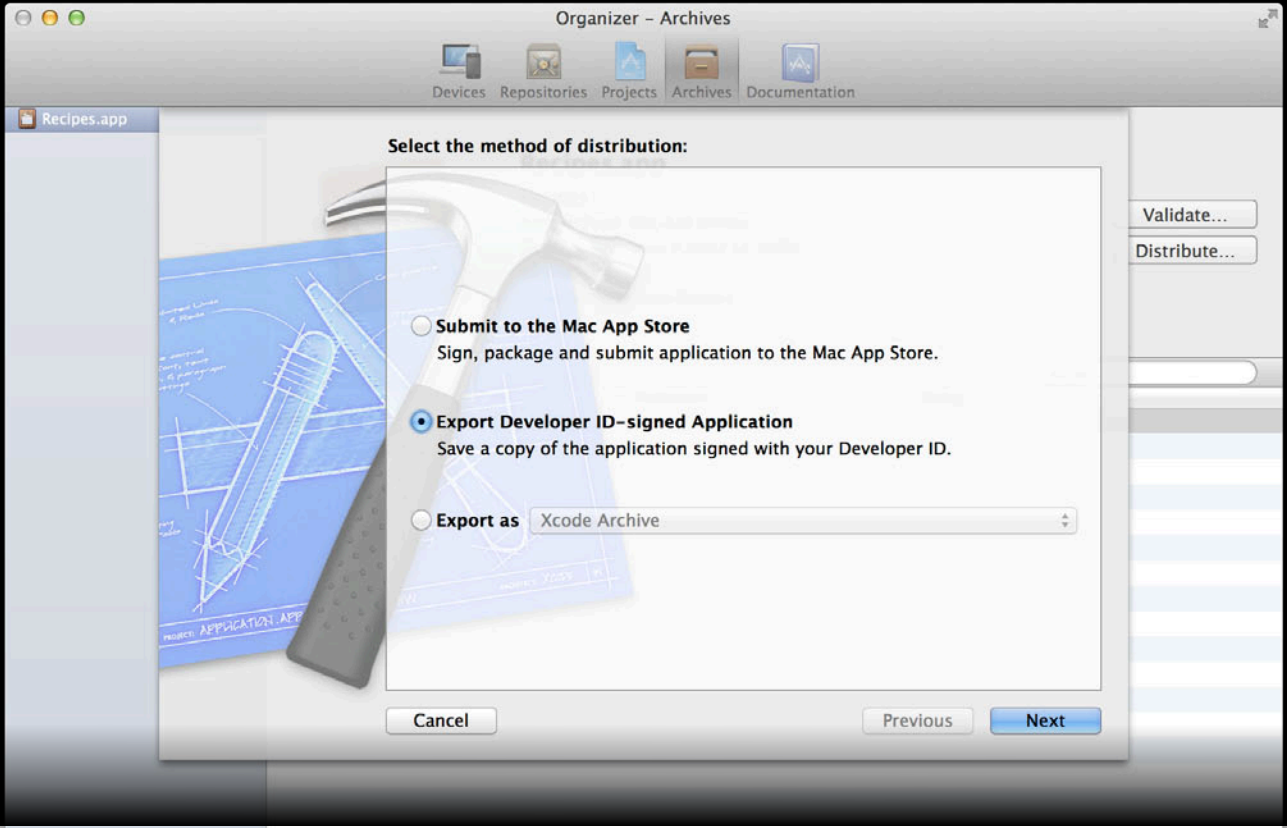
Identifier: com.apple.Recipes

Name	Creation Date	Comment	Status
Recipes	June 7, 2012 12:30 PM		

Distributing Outside the App Store

Exporting an application

- Sign an archive of your application with your Developer ID



Select the method of distribution:

○ **Submit to the Mac App Store**
Sign, package and submit application to the Mac App Store.

● **Export Developer ID-signed Application**
Save a copy of the application signed with your Developer ID.

○ **Export as** Xcode Archive

OS X

Sandboxing and entitlements

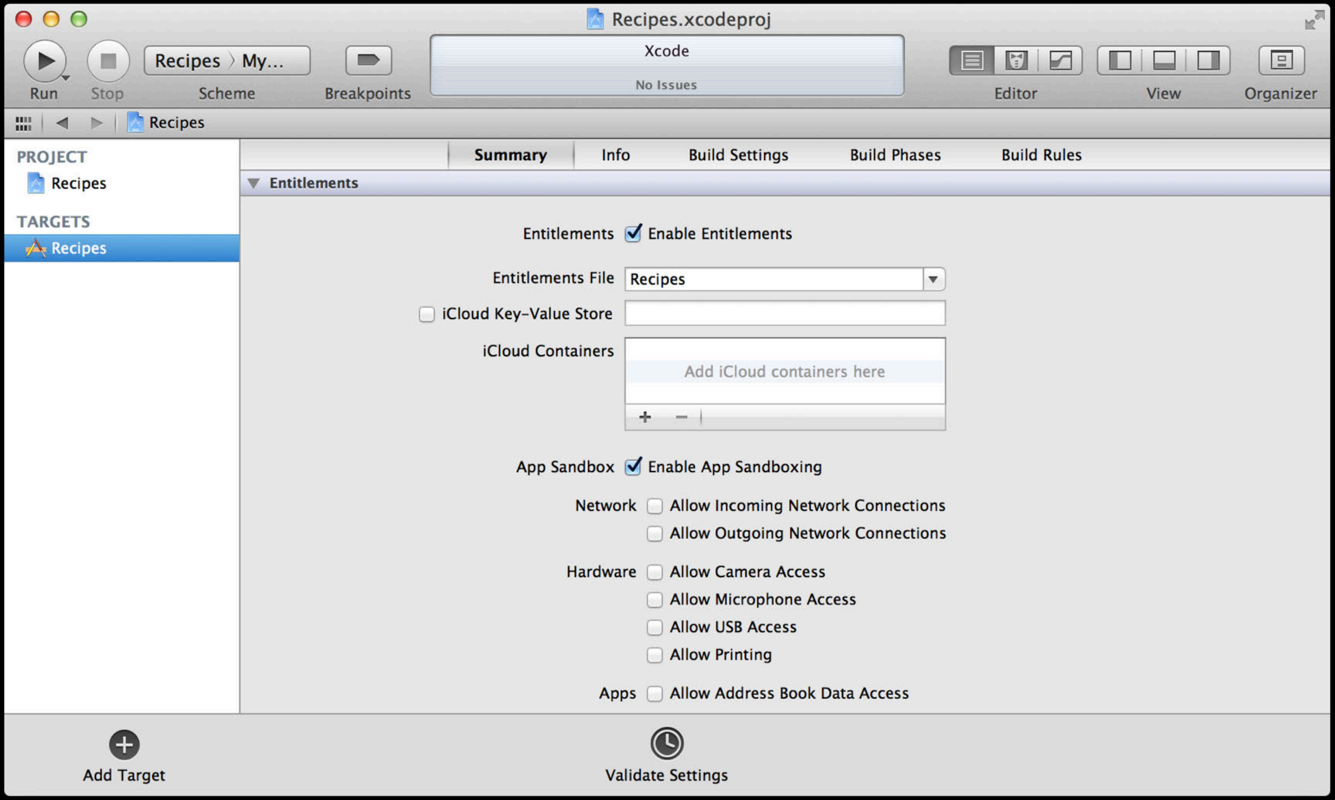
Yevgeny Binder
Xcode

Sandboxing and Entitlements

- Confine apps to a subset of system resources
- Limit direct access to hardware and personal data
- Mitigate consequences of worst-case scenario
- Extra peace of mind for users with no cost to you

Sandboxing and Entitlements

Enabling sandboxing



Entitlements

Entitlements ☑ Enable Entitlements

Entitlements File: Recipes

□ iCloud Key-Value Store

iCloud Containers: Add iCloud containers here

+ −

App Sandbox ☑ Enable App Sandboxing

Network □ Allow Incoming Network Connections

□ Allow Outgoing Network Connections

Hardware □ Allow Camera Access

□ Allow Microphone Access

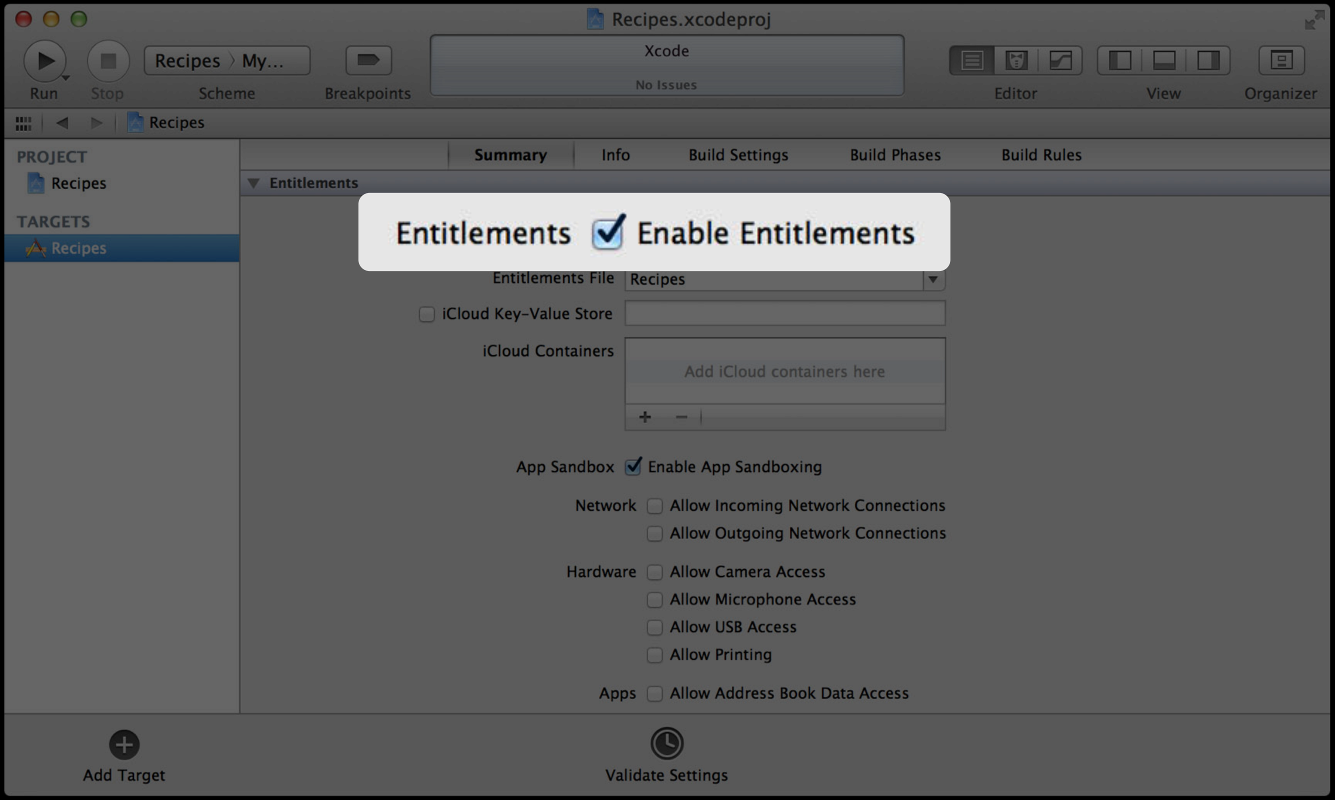
□ Allow USB Access

□ Allow Printing

Apps □ Allow Address Book Data Access

Sandboxing and Entitlements

Enabling sandboxing



Entitlements

Entitlements ☑ Enable Entitlements

Entitlements File Recipes

☐ iCloud Key-Value Store

iCloud Containers Add iCloud containers here

App Sandbox ☑ Enable App Sandboxing

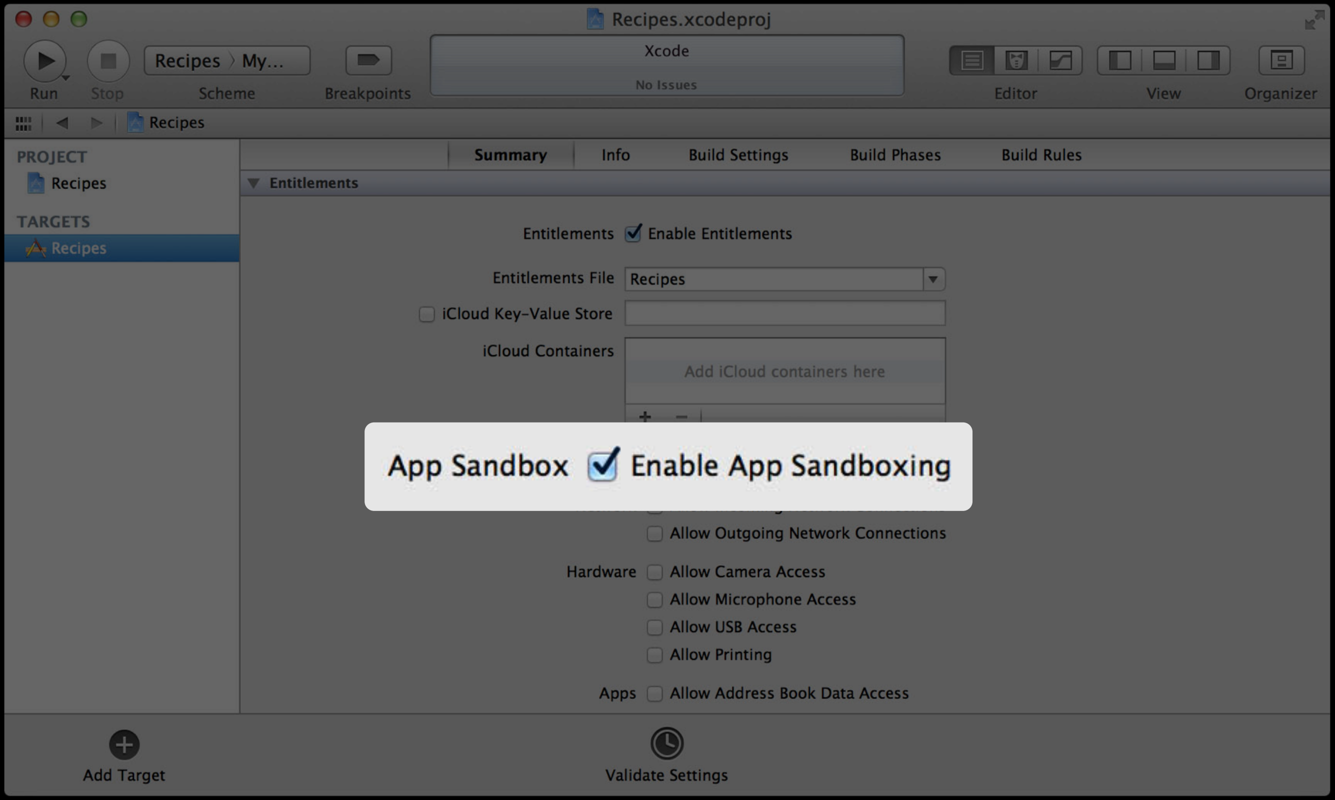
Network ☐ Allow Incoming Network Connections
☐ Allow Outgoing Network Connections

Hardware ☐ Allow Camera Access
☐ Allow Microphone Access
☐ Allow USB Access
☐ Allow Printing

Apps ☐ Allow Address Book Data Access

Sandboxing and Entitlements

Enabling sandboxing



▼ Entitlements

Entitlements ☑ Enable Entitlements

Entitlements File: Recipes

□ iCloud Key-Value Store

iCloud Containers: Add iCloud containers here

App Sandbox ☑ Enable App Sandboxing

□ Allow Outgoing Network Connections

Hardware □ Allow Camera Access
□ Allow Microphone Access
□ Allow USB Access
□ Allow Printing

Apps □ Allow Address Book Data Access

Sandboxing and Entitlements

Granular permissions for sandboxed apps

- Clear declaration of your app’s needs and intent to system APIs
- Permit controlled pinholes in security layer

Filesystem

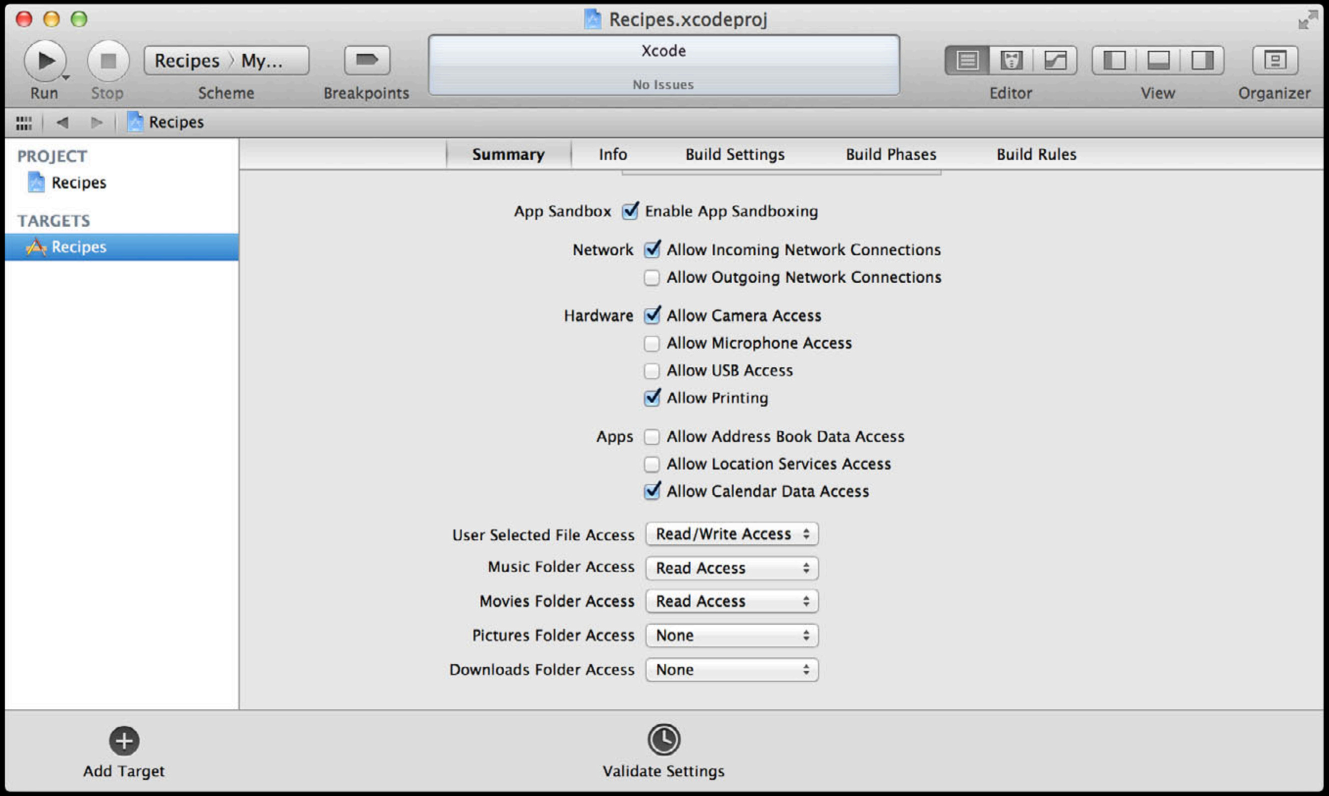
Network

Hardware

Personal Data

Sandboxing and Entitlements

Choosing entitlements



App Sandbox ☑ Enable App Sandboxing

Network ☑ Allow Incoming Network Connections
□ Allow Outgoing Network Connections

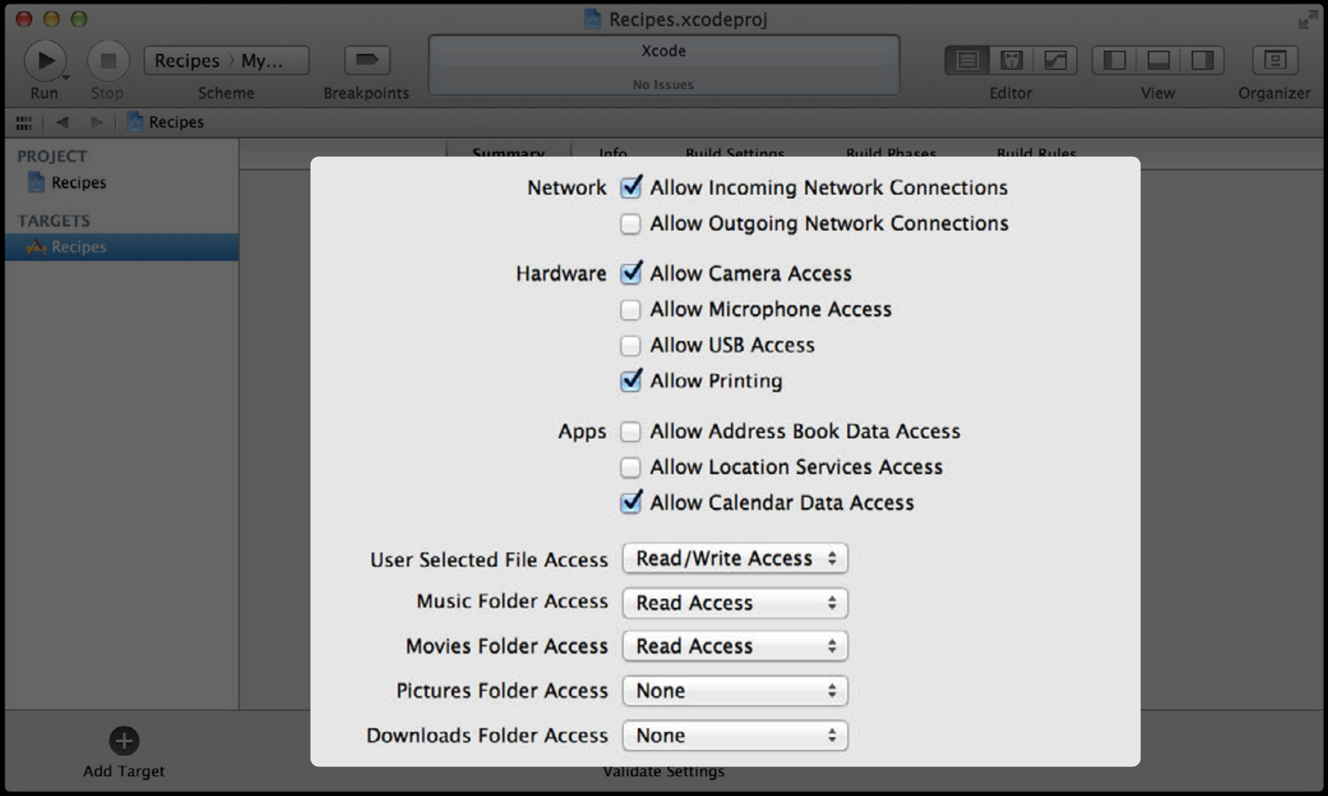
Hardware ☑ Allow Camera Access
□ Allow Microphone Access
□ Allow USB Access
☑ Allow Printing

Apps □ Allow Address Book Data Access
□ Allow Location Services Access
☑ Allow Calendar Data Access

User Selected File Access: Read/Write Access
Music Folder Access: Read Access
Movies Folder Access: Read Access
Pictures Folder Access: None
Downloads Folder Access: None

Sandboxing and Entitlements

Choosing entitlements

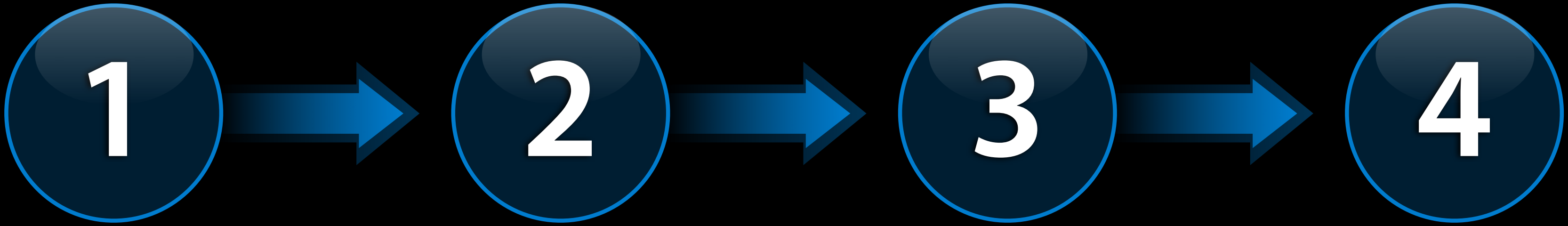


Network ☑ Allow Incoming Network Connections
☐ Allow Outgoing Network Connections

Hardware ☑ Allow Camera Access
☐ Allow Microphone Access
☐ Allow USB Access
☑ Allow Printing

Apps ☐ Allow Address Book Data Access
☐ Allow Location Services Access
☑ Allow Calendar Data Access

User Selected File Access	Read/Write Access
Music Folder Access	Read Access
Movies Folder Access	Read Access
Pictures Folder Access	None
Downloads Folder Access	None



1. Obtain a Developer ID
2. Configure your project to use code signing
3. Secure your app with sandboxing and entitlements
4. Build and archive your signed app

Demo

Distributing an app using Developer ID

Roadmap

Roadmap

OS X

Distributing in the App Store

Rob Marini
Xcode

Distributing in the App Store

- iCloud, Apple Push Notification Service, Game Center, In-App Purchase
- Distribution well integrated with Xcode
- Works with all Gatekeeper configurations

Distributing in the App Store

Configuring your app

- Bundle ID and app ID
- Sandboxing and entitlements
- Validating your app

Distributing in the App Store

CFBundleIdentifier

- Must be unique
- Must match the app ID from a provisioning profile
- Should be the same for Developer ID signed apps
    - Shared filesystem sandbox

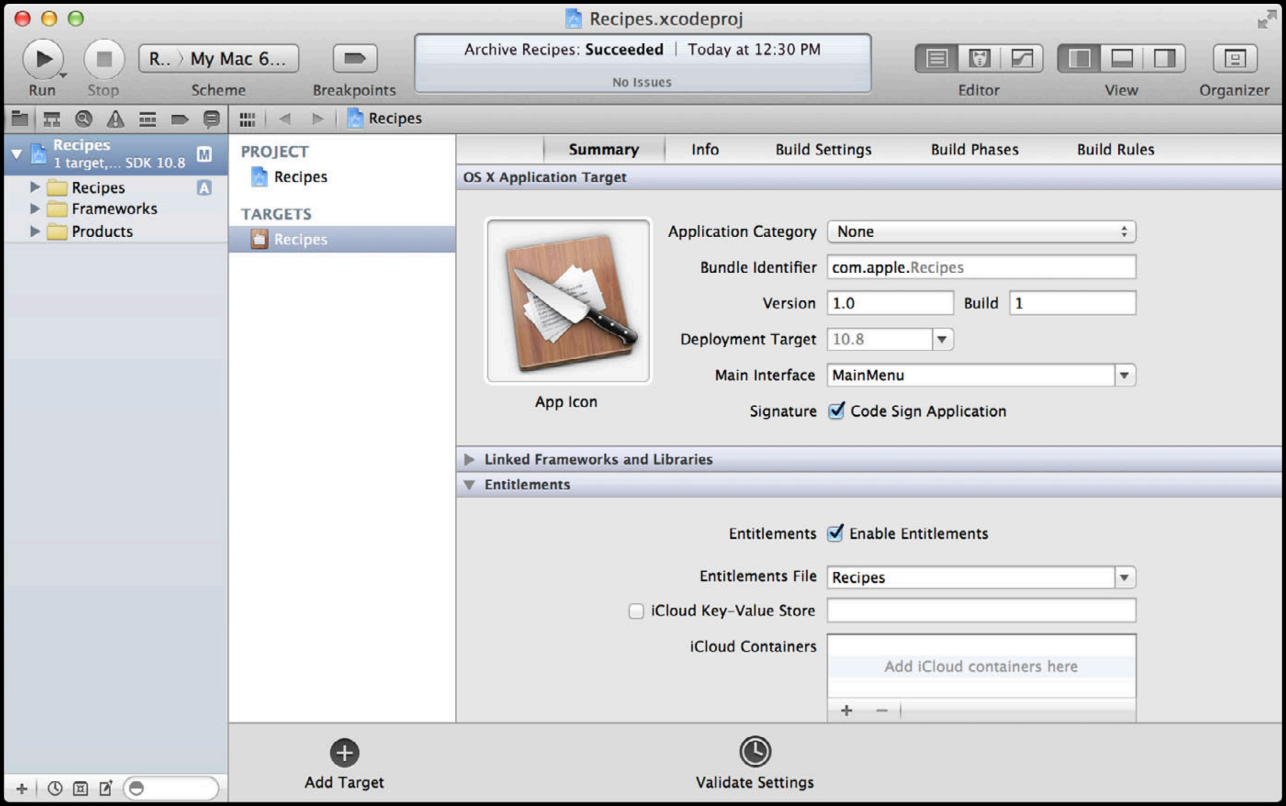
Distributing in the App Store

Sandboxing and entitlements

- All App Store apps must be sandboxed
- Entitlements for App Store-specific technologies

Distributing in the App Store

Sandboxing and entitlements



Archive Recipes: **Succeeded** | Today at 12:30 PM

No Issues

Application Category: None

Bundle Identifier: com.apple.Recipes

Version: 1.0 Build: 1

Deployment Target: 10.8

Main Interface: MainMenu

Signature: ☑ Code Sign Application

Entitlements: ☑ Enable Entitlements

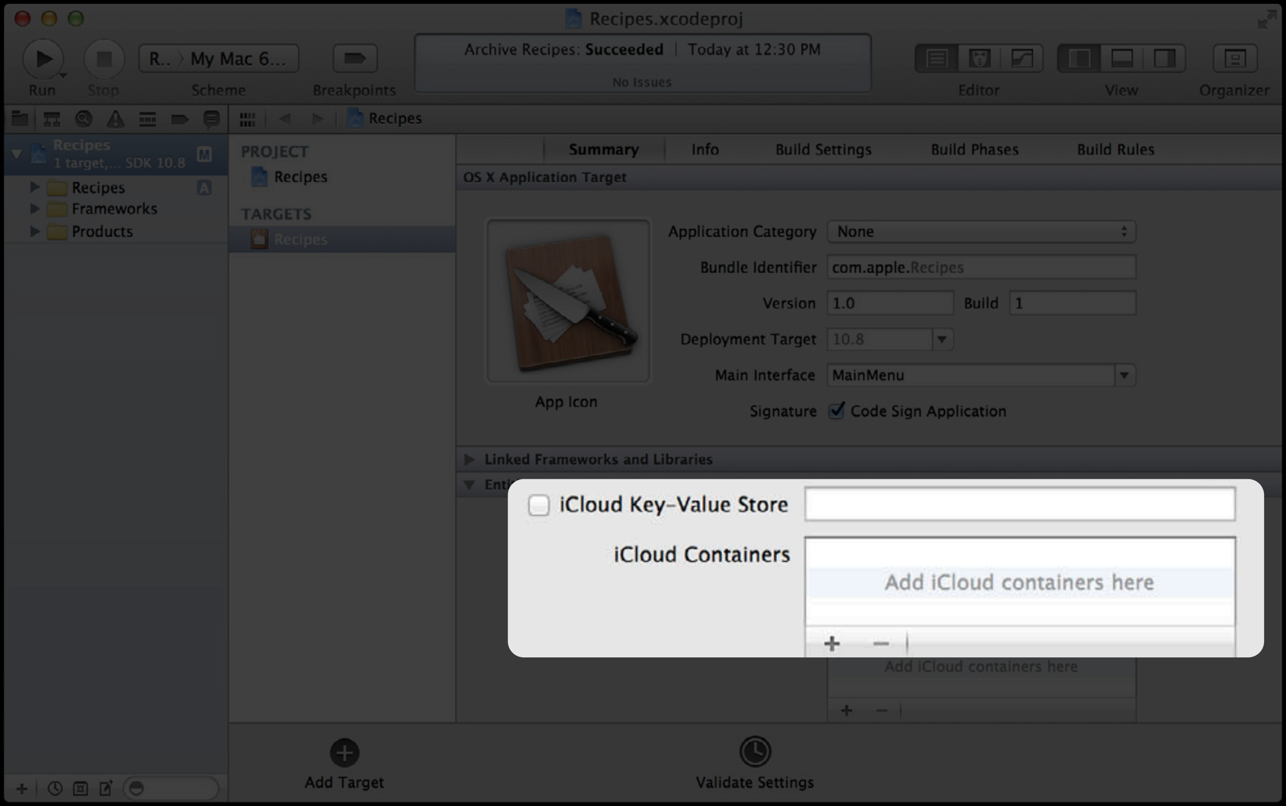
Entitlements File: Recipes

□ iCloud Key-Value Store

iCloud Containers: Add iCloud containers here

Distributing in the App Store

Sandboxing and entitlements



Application Category: None
Bundle Identifier: com.apple.Recipes
Version: 1.0 Build: 1
Deployment Target: 10.8
Main Interface: MainMenu
Signature: ☑ Code Sign Application

□ iCloud Key-Value Store

iCloud Containers: Add iCloud containers here

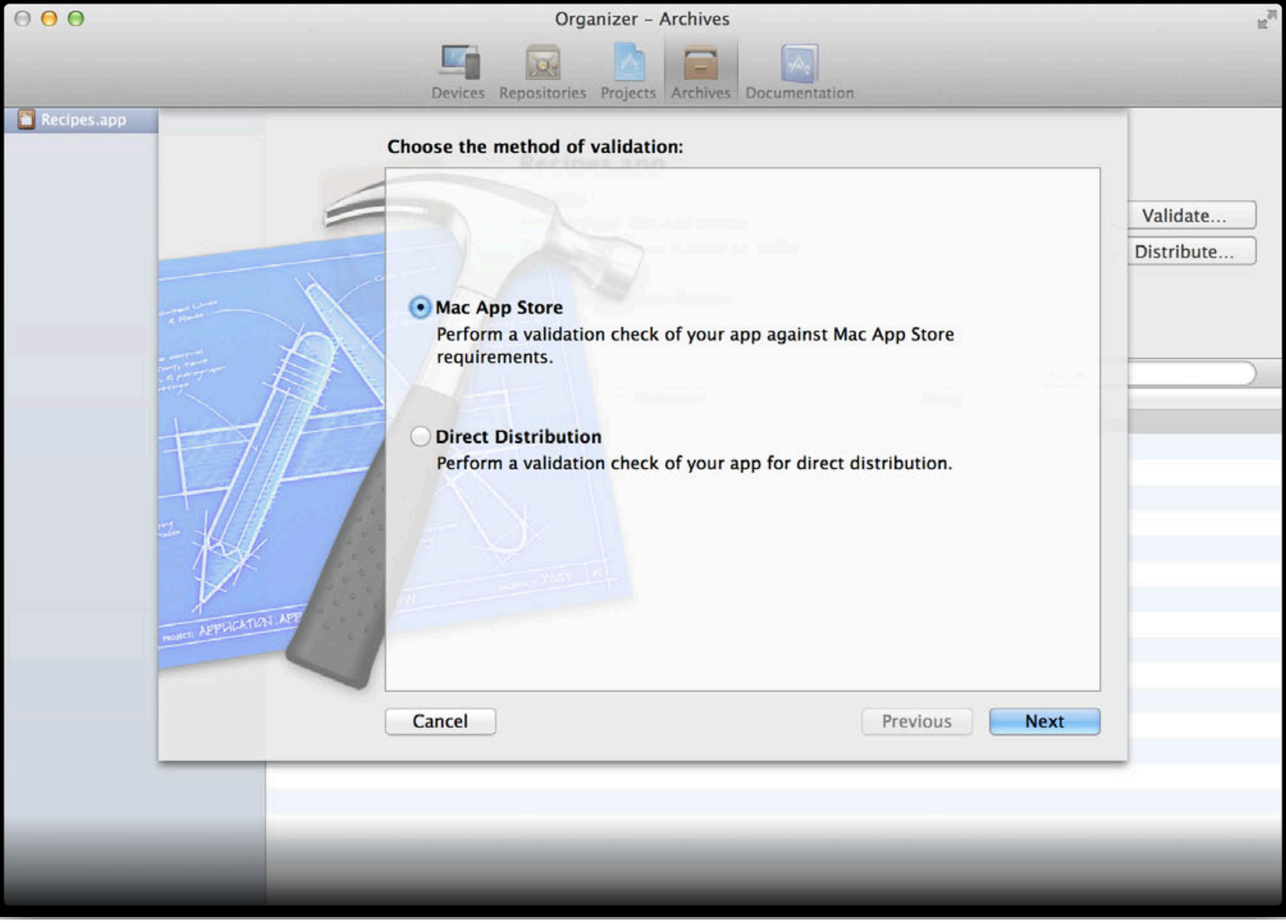
Distributing in the App Store

Validating your app

- Built application archives can be tested locally in Xcode against App Store requirements
- Online validation from Xcode uses up-to-date data from the App Store

Distributing in the App Store

Validating your app



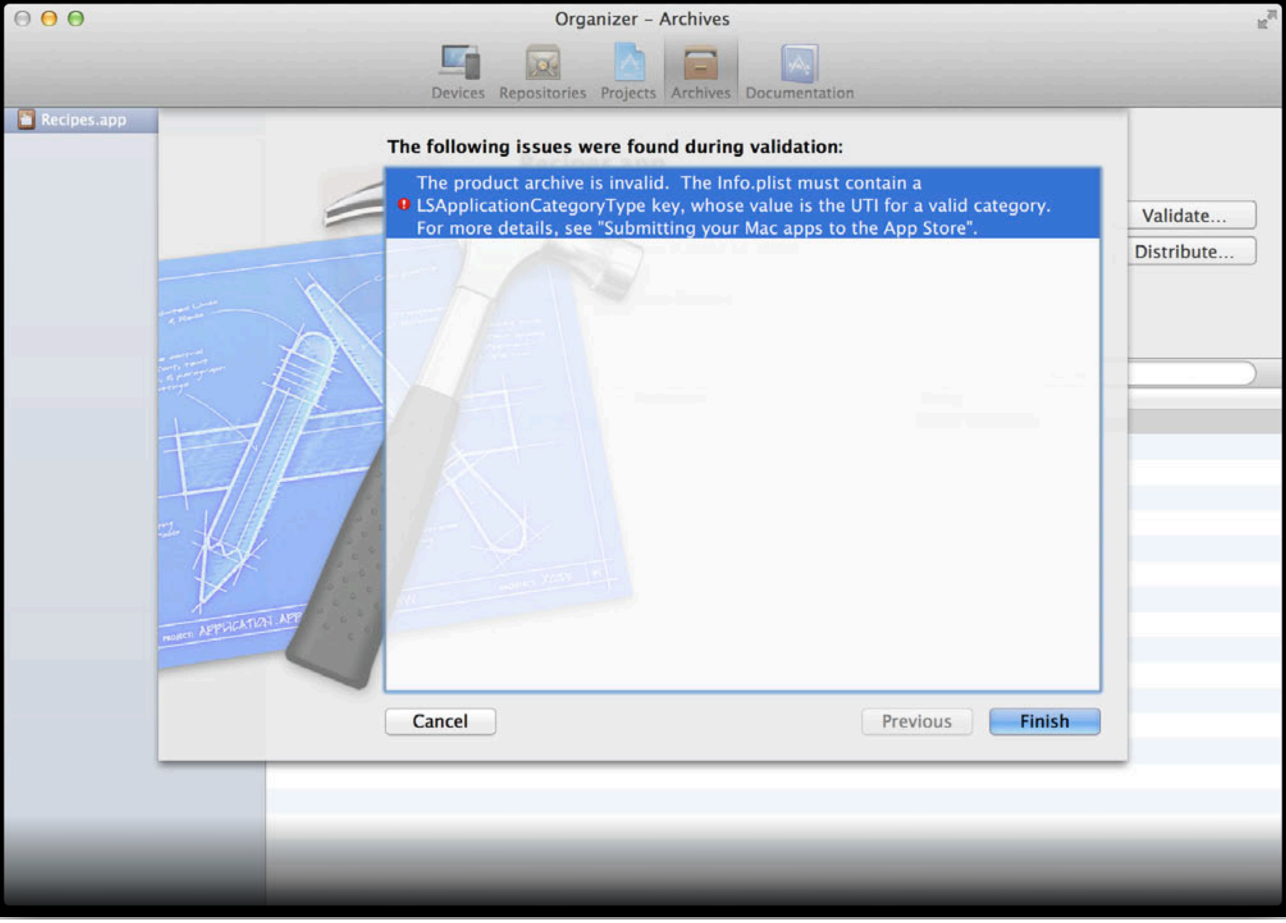
Choose the method of validation:

● Mac App Store
Perform a validation check of your app against Mac App Store requirements.

○ Direct Distribution
Perform a validation check of your app for direct distribution.

Distributing in the App Store

Validating your app



The following issues were found during validation:

The product archive is invalid. The Info.plist must contain a LSApplicationCategoryType key, whose value is the UTI for a valid category. For more details, see "Submitting your Mac apps to the App Store".

OS X

App Store-specific technologies

Itai Rom
Xcode

App Store-Specific Technologies

Overview

App Store-Specific Technologies

Overview

- Provisioning profiles are required
- In-App Purchase content can now be created and submitted to the App Store from Xcode

App Store-Specific Technologies

Team roles

App Store Team
Member

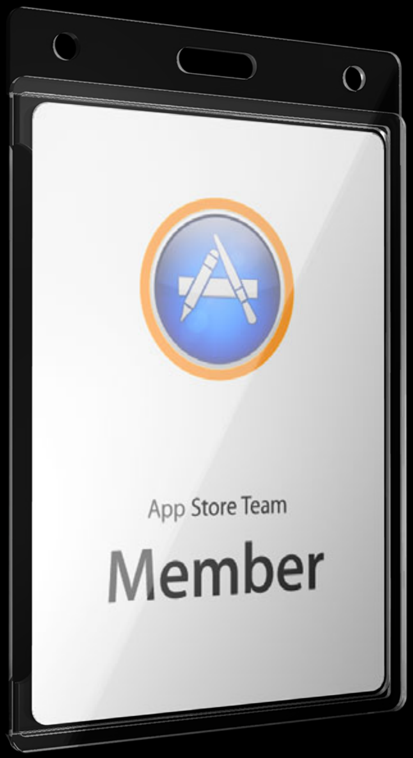
App Store Team
Admin

App Store Team
Agent

App Store-Specific Technologies

Team members

- Download provisioning profiles
- Request certificates
- Obtain software assets
- Typically engineers



App Store-Specific Technologies

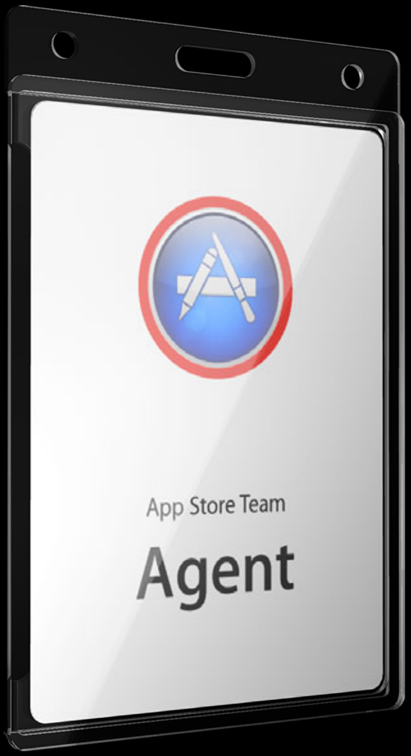
Team admins

- Manage team members
- Create and approve certificates
- Create provisioning profiles
- Register hardware and app IDs

App Store-Specific Technologies

Team agent

- Manage team membership
- Manage the developer program
- Submit applications
- Create distribution certificates



App Store-Specific Technologies

What exactly is a provisioning profile?

App Store-Specific Technologies

What exactly is a provisioning profile?

Provisioning Profile

App Store-Specific Technologies

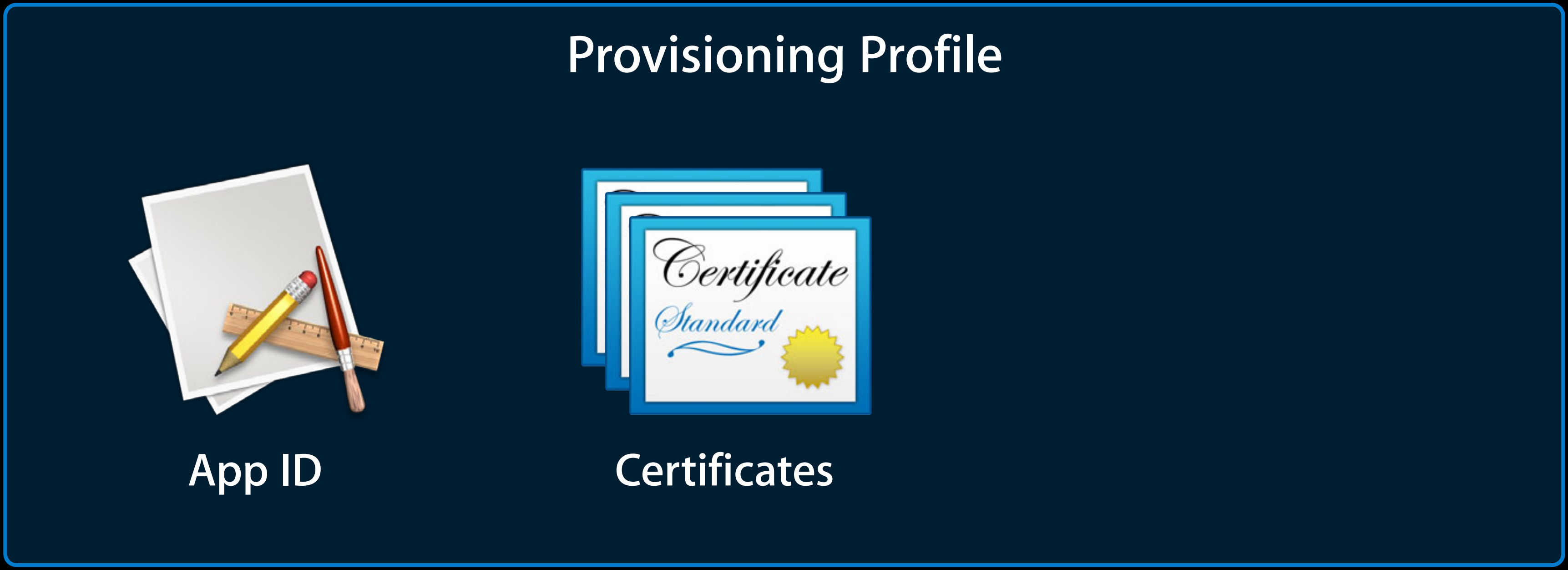
What exactly is a provisioning profile?

Provisioning Profile

App ID

App Store-Specific Technologies

What exactly is a provisioning profile?



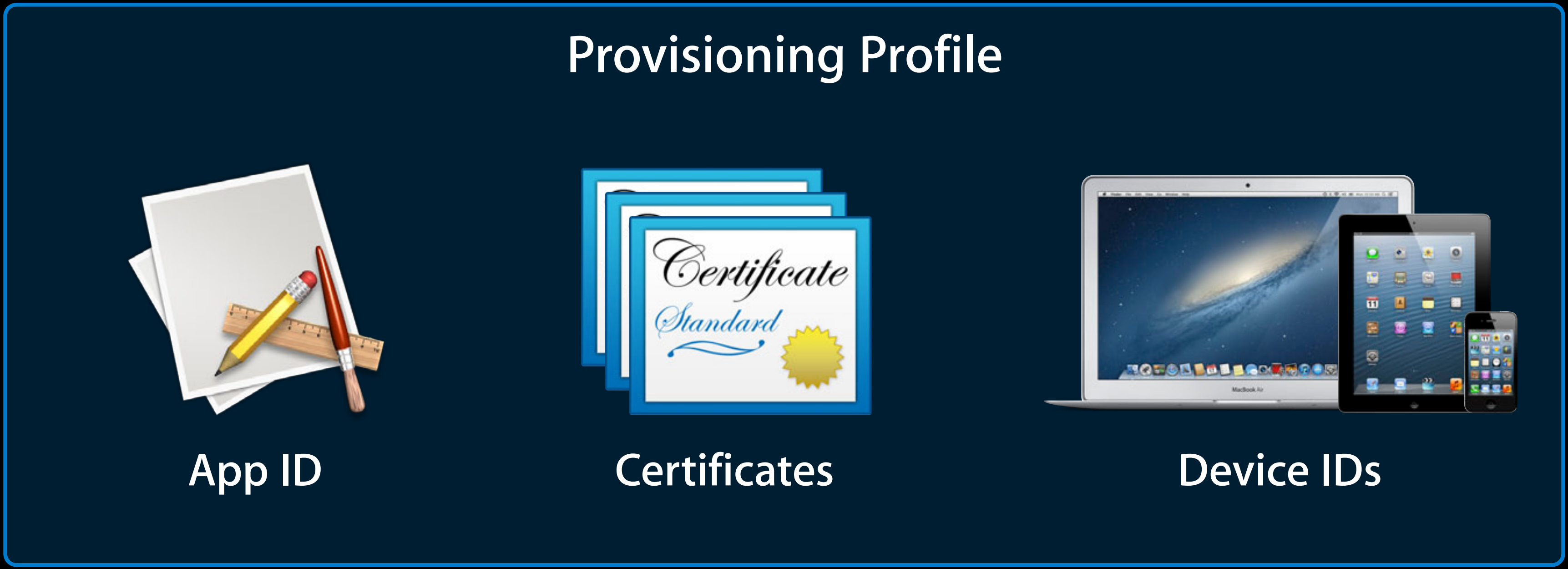
Provisioning Profile

App ID

Certificates

App Store-Specific Technologies

What exactly is a provisioning profile?



Provisioning Profile

App ID

Certificates

Device IDs

App Store-Specific Technologies

What exactly is a provisioning profile?

Provisioning Profile

An app signed by the following people
may run on the following devices

App Store-Specific Technologies

Explicit app IDs

- Explicit app IDs match exactly one app
- Created on the developer portal
- Required for In-App Purchase, Game Center, and Apple Push Notification Service

`com.apple.Recipes`

Bundle ID search string

App Store-Specific Technologies

Wildcard app IDs

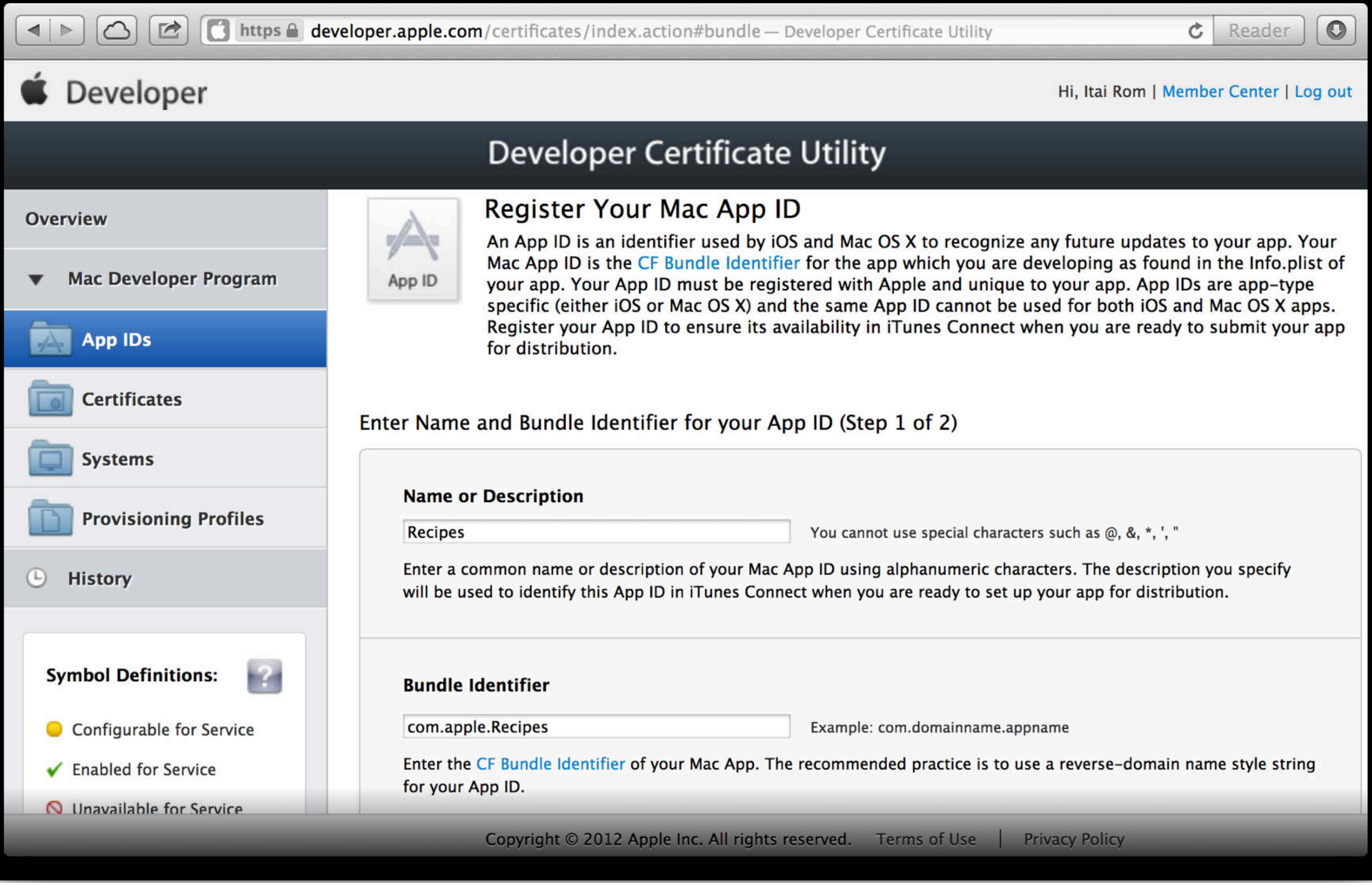
- Wildcard app IDs match multiple apps
- Xcode creates one for you automatically
- Useful when you start developing new apps
- Can be used to enable iCloud

```
com.apple.*
```

Bundle ID search string

App Store-Specific Technologies

Registering an app ID



https developer.apple.com/certificates/index.action#bundle — Developer Certificate Utility
Reader
Developer
Hi, Itai Rom | Member Center | Log out
Overview
Mac Developer Program
App IDs
Certificates
Systems
Provisioning Profiles
History
App ID

Developer Certificate Utility

Register Your Mac App ID

An App ID is an identifier used by iOS and Mac OS X to recognize any future updates to your app. Your Mac App ID is the CF Bundle Identifier for the app which you are developing as found in the Info.plist of your app. Your App ID must be registered with Apple and unique to your app. App IDs are app-type specific (either iOS or Mac OS X) and the same App ID cannot be used for both iOS and Mac OS X apps. Register your App ID to ensure its availability in iTunes Connect when you are ready to submit your app for distribution.

Enter Name and Bundle Identifier for your App ID (Step 1 of 2)

Name or Description

Recipes

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

Enter a common name or description of your Mac App ID using alphanumeric characters. The description you specify will be used to identify this App ID in iTunes Connect when you are ready to set up your app for distribution.

Bundle Identifier

com.apple.Recipes

Example: com.domainname.appname

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

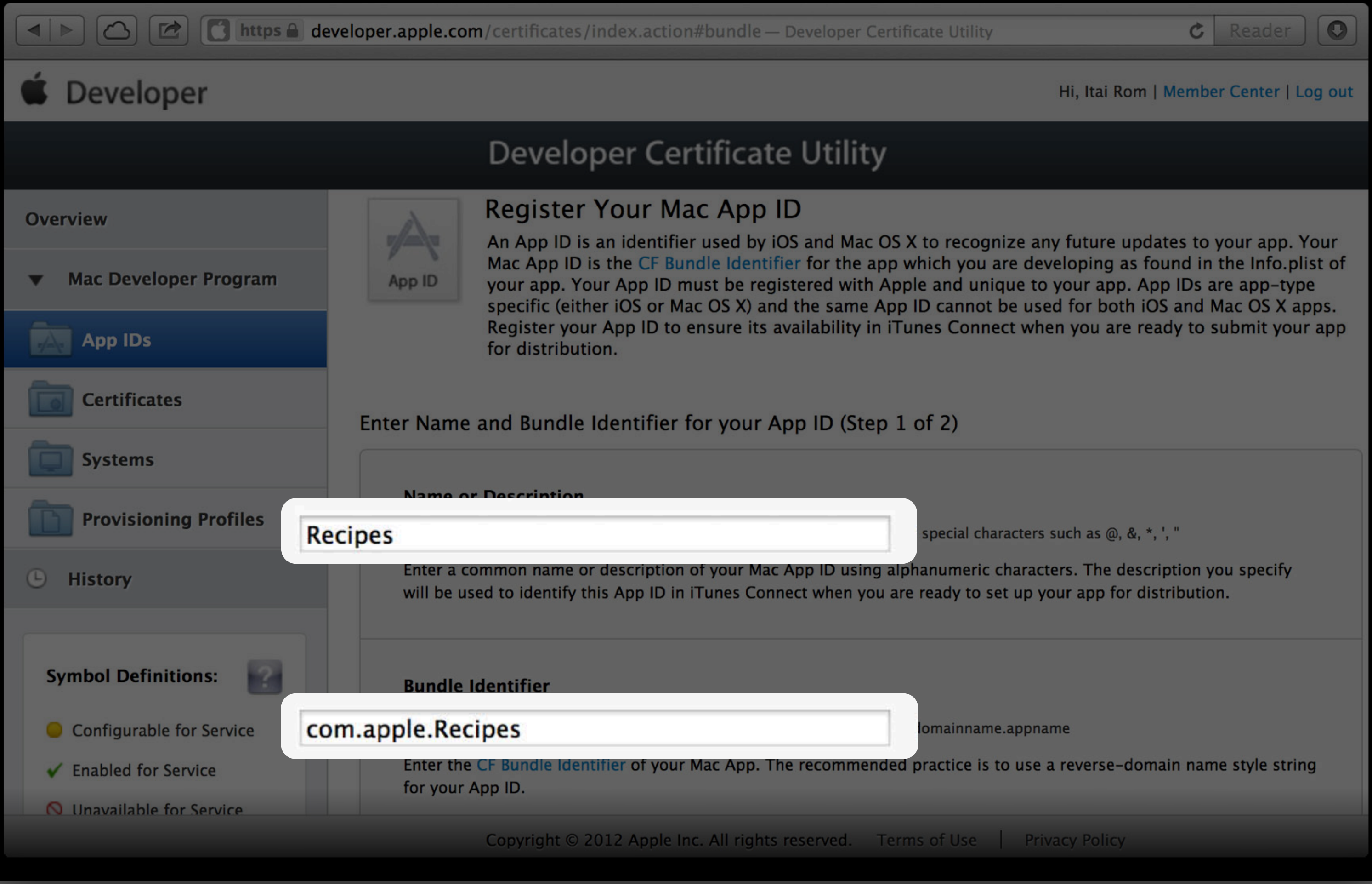
Symbol Definitions:

- ● Configurable for Service
- ✓ Enabled for Service
- ⊘ Unavailable for Service

Copyright © 2012 Apple Inc. All rights reserved. Terms of Use | Privacy Policy

App Store-Specific Technologies

Registering an app ID



Register Your Mac App ID

An App ID is an identifier used by iOS and Mac OS X to recognize any future updates to your app. Your Mac App ID is the CF Bundle Identifier for the app which you are developing as found in the Info.plist of your app. Your App ID must be registered with Apple and unique to your app. App IDs are app-type specific (either iOS or Mac OS X) and the same App ID cannot be used for both iOS and Mac OS X apps. Register your App ID to ensure its availability in iTunes Connect when you are ready to submit your app for distribution.

Enter Name and Bundle Identifier for your App ID (Step 1 of 2)

Name or Description

Recipes

special characters such as @, &, *, ', "

Enter a common name or description of your Mac App ID using alphanumeric characters. The description you specify will be used to identify this App ID in iTunes Connect when you are ready to set up your app for distribution.

Bundle Identifier

com.apple.Recipes

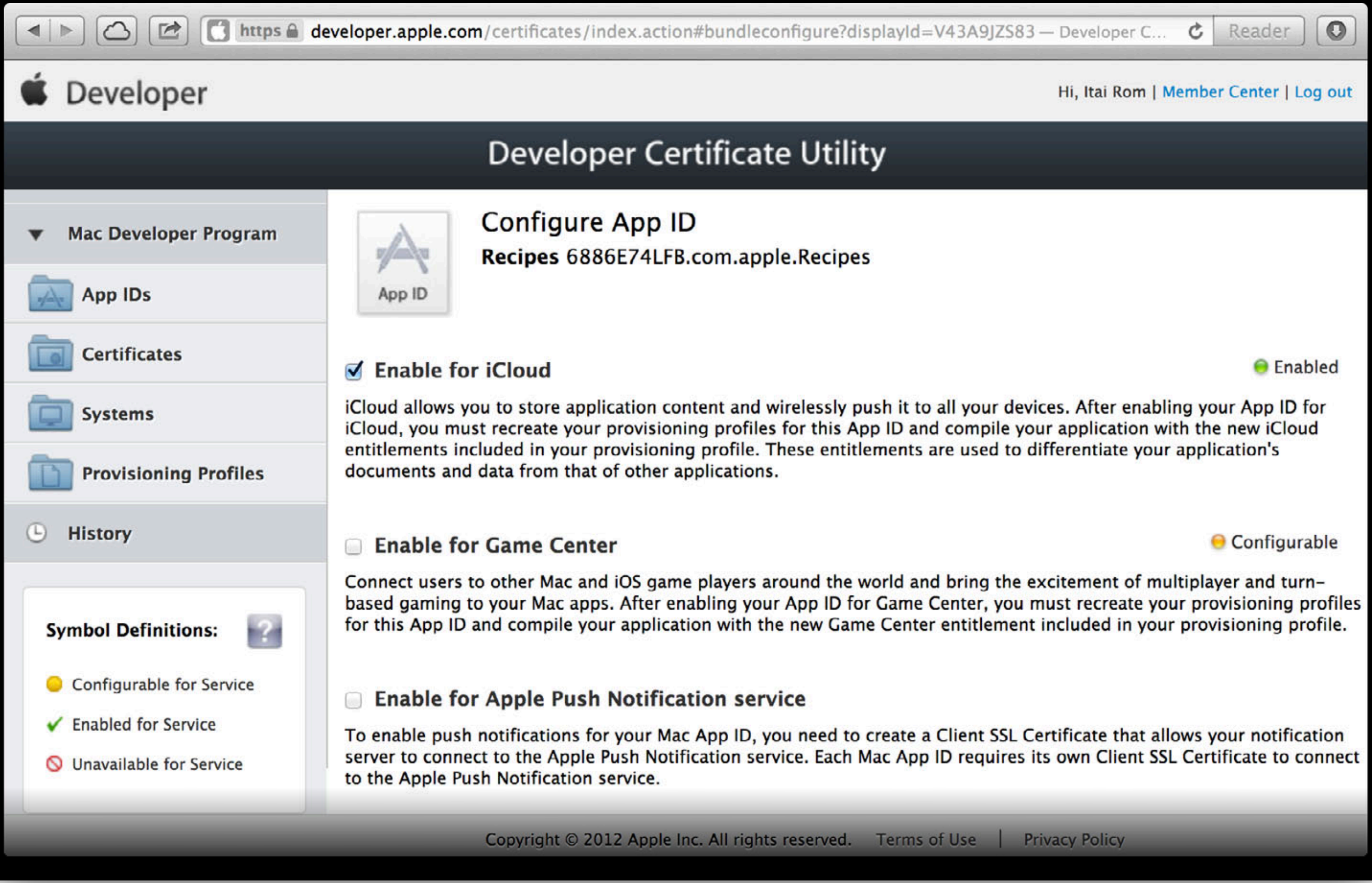
omainname.appname

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

Copyright © 2012 Apple Inc. All rights reserved. Terms of Use | Privacy Policy

App Store-Specific Technologies

Configuring an app ID for additional services


https developer.apple.com/certificates/index.action#bundleconfigure?displayId=V43A9JZS83 — Developer C...
Reader
Developer
Hi, Itai Rom | Member Center | Log out
Mac Developer Program
App IDs
Certificates
Systems
Provisioning Profiles
History
App ID

Developer Certificate Utility

Configure App ID

Recipes 6886E74LFB.com.apple.Recipes

☑ **Enable for iCloud** ● Enabled

iCloud allows you to store application content and wirelessly push it to all your devices. After enabling your App ID for iCloud, you must recreate your provisioning profiles for this App ID and compile your application with the new iCloud entitlements included in your provisioning profile. These entitlements are used to differentiate your application's documents and data from that of other applications.

□ **Enable for Game Center** ● Configurable

Connect users to other Mac and iOS game players around the world and bring the excitement of multiplayer and turn-based gaming to your Mac apps. After enabling your App ID for Game Center, you must recreate your provisioning profiles for this App ID and compile your application with the new Game Center entitlement included in your provisioning profile.

□ **Enable for Apple Push Notification service**

To enable push notifications for your Mac App ID, you need to create a Client SSL Certificate that allows your notification server to connect to the Apple Push Notification service. Each Mac App ID requires its own Client SSL Certificate to connect to the Apple Push Notification service.

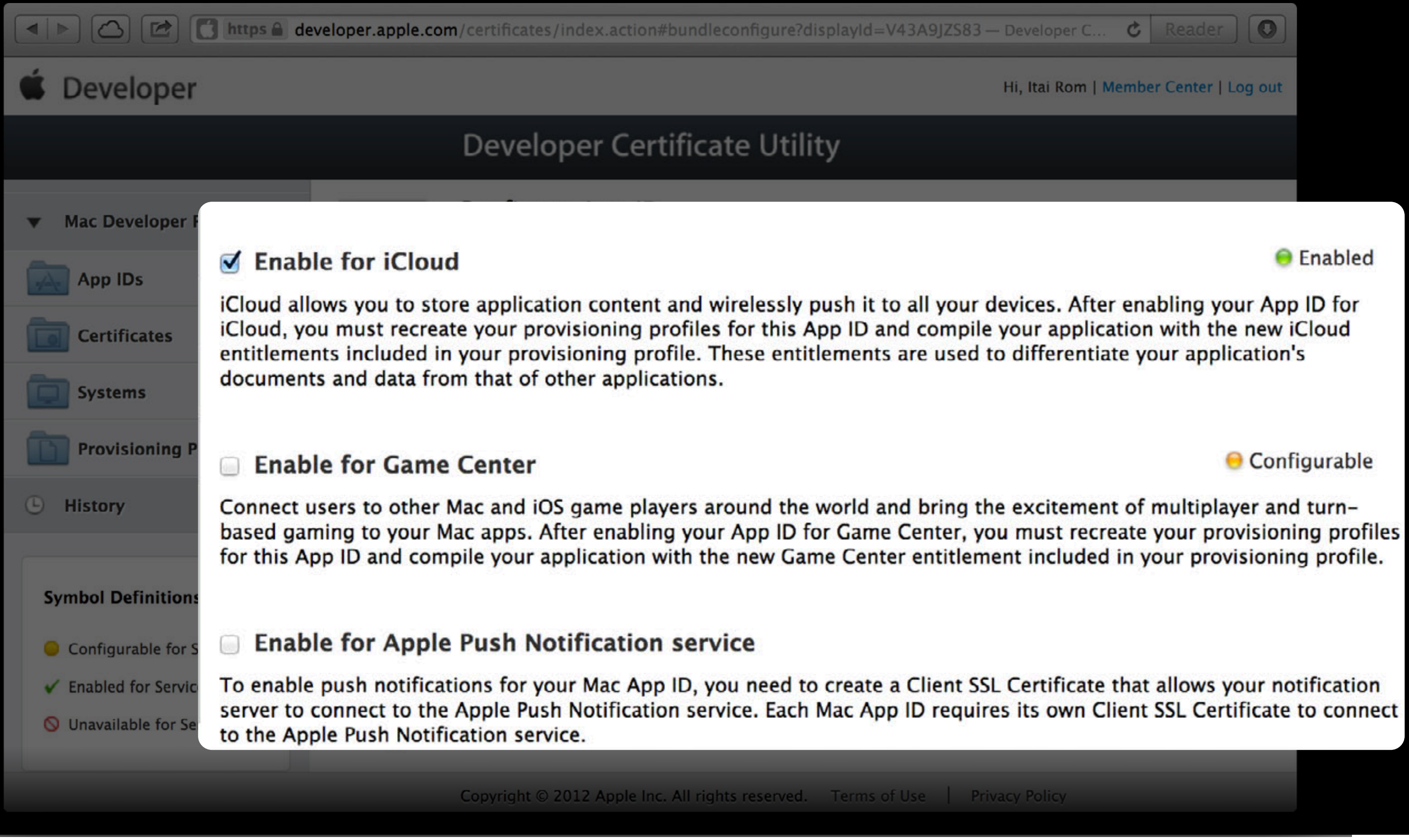
Symbol Definitions: ?

- ● Configurable for Service
- ✓ Enabled for Service
- ⊘ Unavailable for Service

Copyright © 2012 Apple Inc. All rights reserved. Terms of Use | Privacy Policy

App Store-Specific Technologies

Configuring an app ID for additional services


https developer.apple.com/certificates/index.action#bundleconfigure?displayId=V43A9JZS83 — Developer C...
Reader
Developer
Hi, Itai Rom | Member Center | Log out
Developer Certificate Utility
Mac Developer P
App IDs
Certificates
Systems
Provisioning P
History
Symbol Definitions
Configurable for S
Enabled for Servic
Unavailable for Se
Copyright © 2012 Apple Inc. All rights reserved.
Terms of Use
Privacy Policy

☑ **Enable for iCloud** ● Enabled

iCloud allows you to store application content and wirelessly push it to all your devices. After enabling your App ID for iCloud, you must recreate your provisioning profiles for this App ID and compile your application with the new iCloud entitlements included in your provisioning profile. These entitlements are used to differentiate your application's documents and data from that of other applications.

□ **Enable for Game Center** ● Configurable

Connect users to other Mac and iOS game players around the world and bring the excitement of multiplayer and turn–based gaming to your Mac apps. After enabling your App ID for Game Center, you must recreate your provisioning profiles for this App ID and compile your application with the new Game Center entitlement included in your provisioning profile.

□ **Enable for Apple Push Notification service**

To enable push notifications for your Mac App ID, you need to create a Client SSL Certificate that allows your notification server to connect to the Apple Push Notification service. Each Mac App ID requires its own Client SSL Certificate to connect to the Apple Push Notification service.

App Store-Specific Technologies

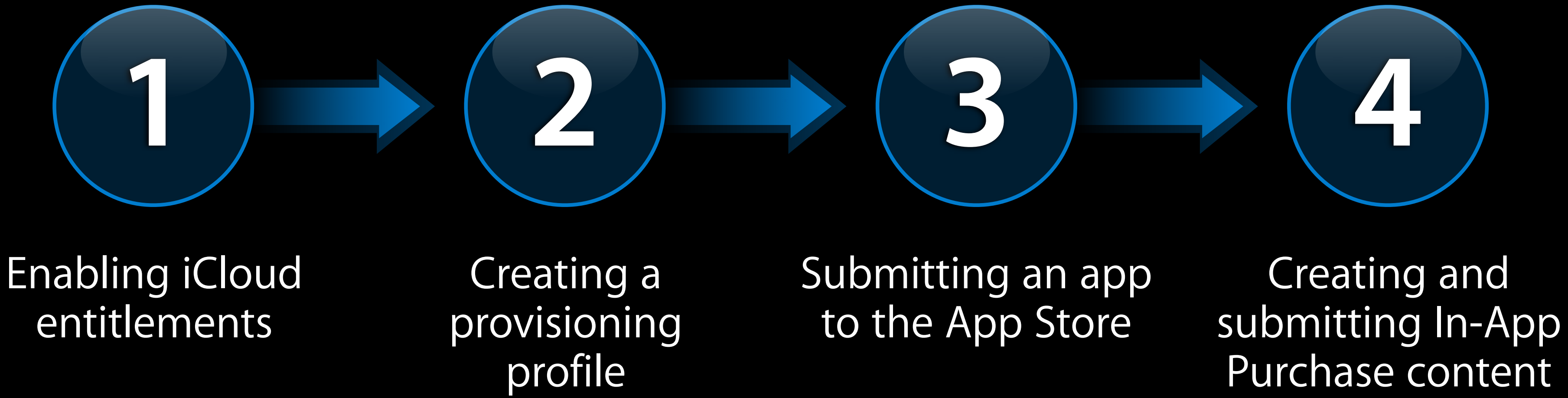
Provisioning profiles

- Team provisioning profiles:
    - Created and maintained by Xcode
    - Wildcard app ID
    - All devices and certificates
- Provisioning profiles containing explicit app IDs are created in Xcode
- Production Provisioning Profiles are used for App Store submission

App Store-Specific Technologies

In-App Purchase content

- New project templates for Mac and iOS
- Archive, package, and submit to the App Store from Xcode



1 Enabling iCloud entitlements

2 Creating a provisioning profile

3 Submitting an app to the App Store

4 Creating and submitting In-App Purchase content

Demo

Distributing an app in the App Store

Roadmap

Roadmap

iOS

iOS application best practices

Rob Marini
Xcode

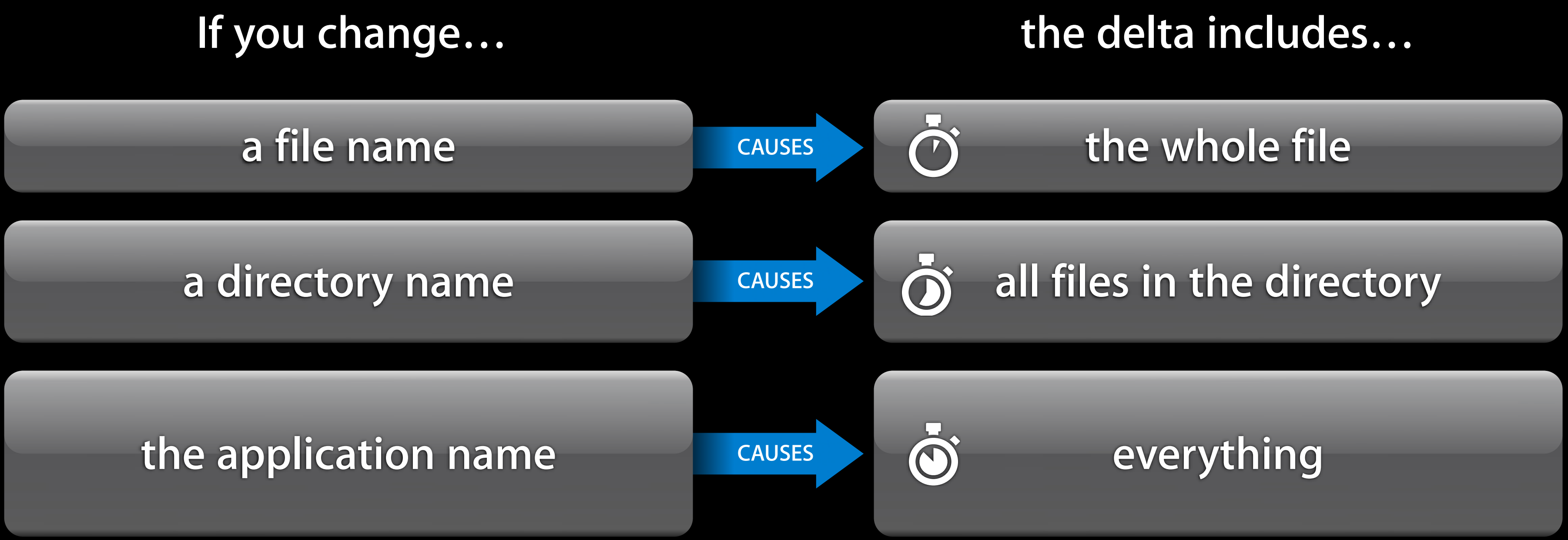
iOS Application Best Practices

App Store delta updates

- Delta generated per file from one version to the next
- Store manages which deltas are published
  - Generated by the store
  - Worst-case scenario is full app download
- Transparent at app development and submission time

iOS Application Best Practices

Small changes generate small deltas



If you change…		the delta includes…
a file name	CAUSES	the whole file
a directory name	CAUSES	all files in the directory
the application name	CAUSES	everything

iOS Application Best Practices

Advantages of beta testing

- Free quality assurance
- Opportunity to rapidly respond to customer feedback
- Direct channel for customers to influence the product
- Identify weak and strong points ahead of release

iOS Application Best Practices

Distributing application betas

- Setting up iOS application betas
- Distribution workflow

iOS Application Best Practices

Setting up iOS application betas

- Should be able to coexist with shipping version

Different bundle IDs

`com.mycompany.App`

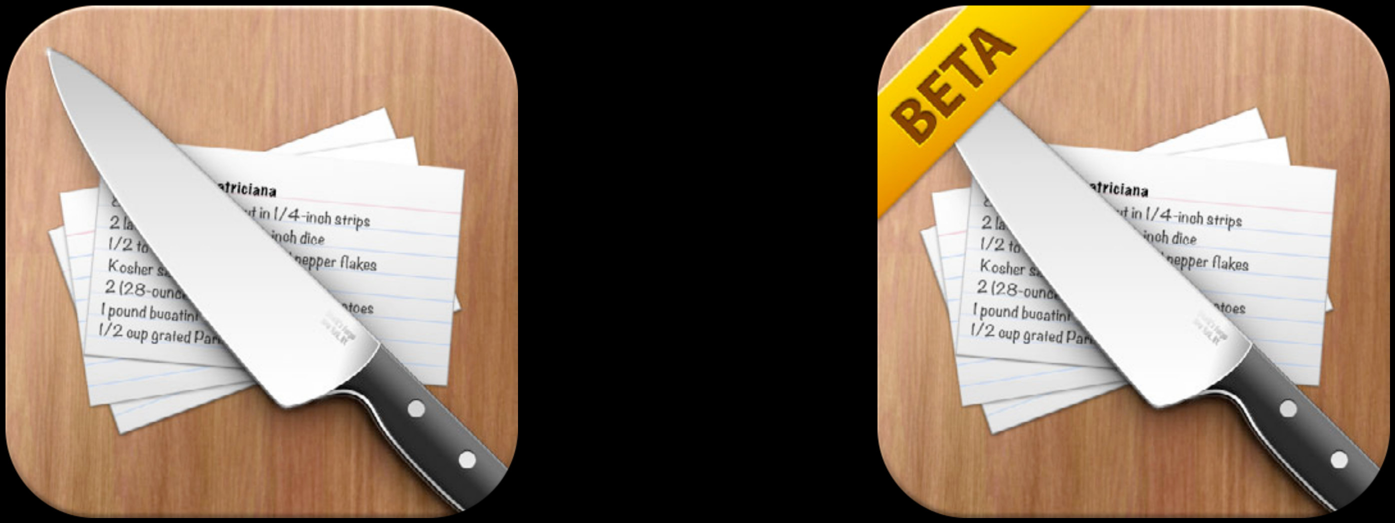
`com.mycompany.App.beta`

iOS Application Best Practices

Setting up iOS application betas

- Should be recognizable as a beta

Different Icons



iOS Application Best Practices

Distributing an iOS Application Betas

- Should be easy for testers to obtain and install

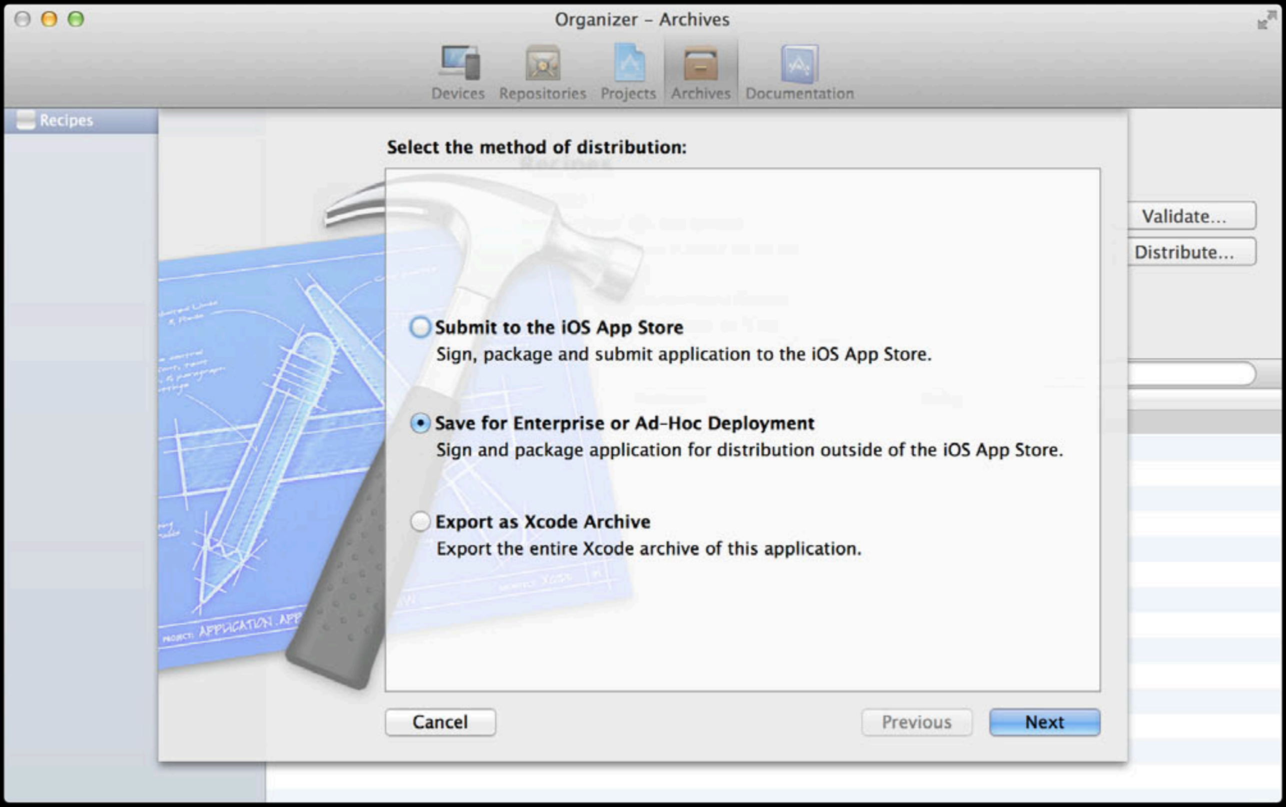
iOS Application Best Practices

Distributing an iOS Application Betas

- Should be easy for testers to obtain and install

iOS Application Best Practices

Goals in distributing a beta



Select the method of distribution:

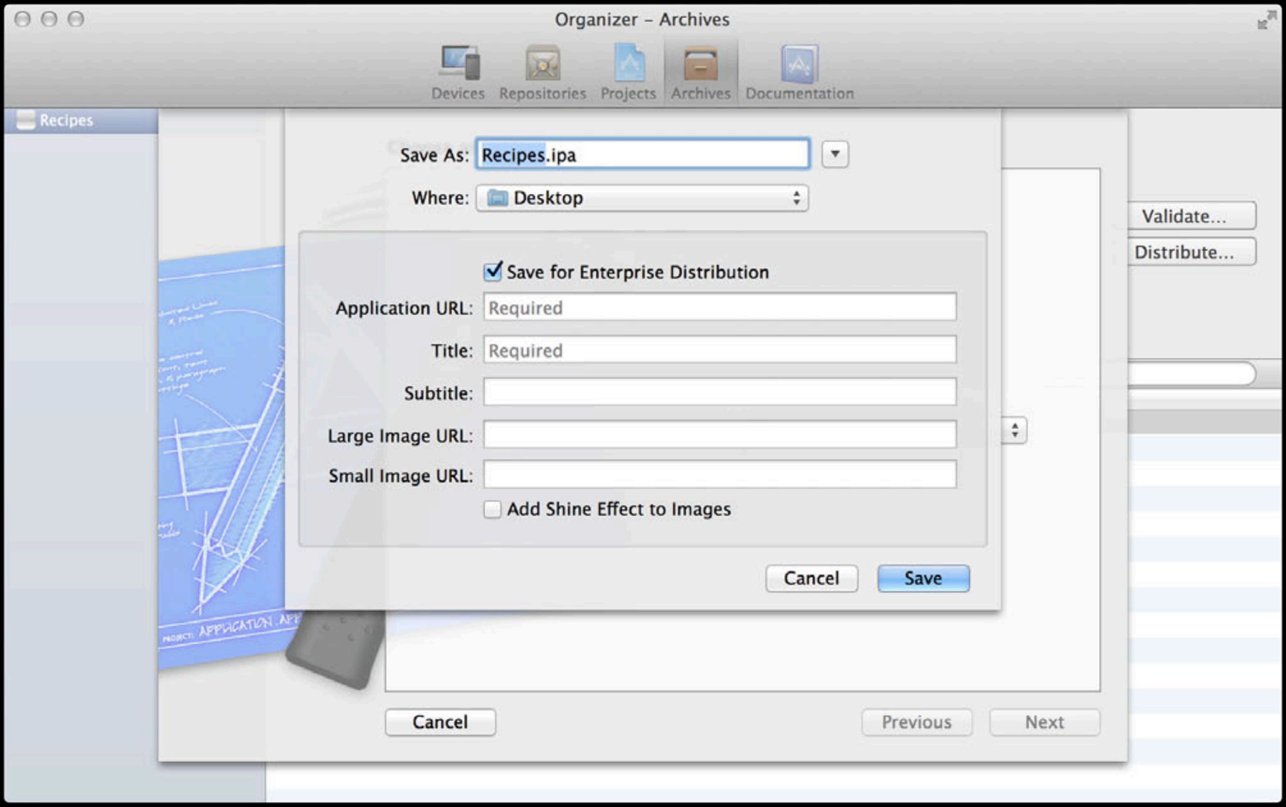
○ **Submit to the iOS App Store**
Sign, package and submit application to the iOS App Store.

● **Save for Enterprise or Ad-Hoc Deployment**
Sign and package application for distribution outside of the iOS App Store.

○ **Export as Xcode Archive**
Export the entire Xcode archive of this application.

iOS Application Best Practices

Goals in distributing a beta



Save As: Recipes.ipa

Where: Desktop

☑ Save for Enterprise Distribution

Application URL: Required

Title: Required

Subtitle:

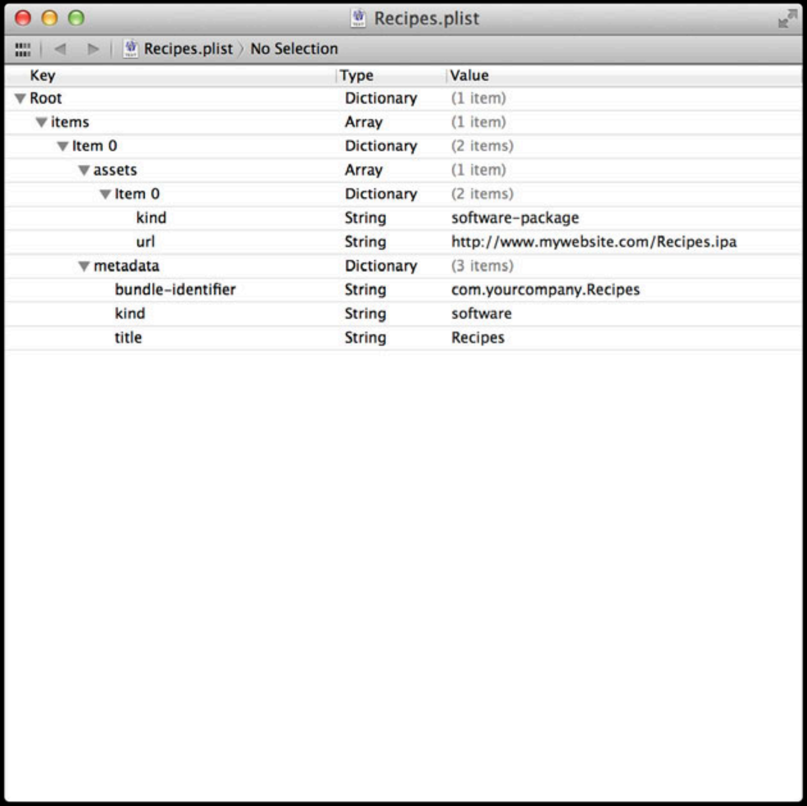
Large Image URL:

Small Image URL:

☐ Add Shine Effect to Images

iOS Application Best Practices

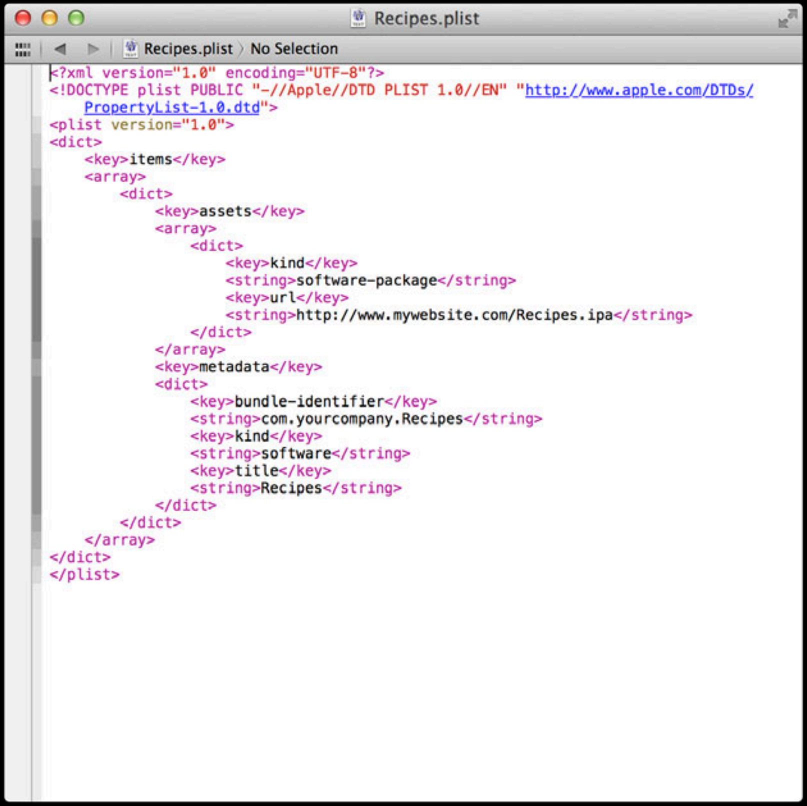
Goals in distributing a beta



Key	Type	Value
▼ Root	Dictionary	(1 item)
▼ items	Array	(1 item)
▼ Item 0	Dictionary	(2 items)
▼ assets	Array	(1 item)
▼ Item 0	Dictionary	(2 items)
kind	String	software-package
url	String	http://www.mywebsite.com/Recipes.ipa
▼ metadata	Dictionary	(3 items)
bundle-identifier	String	com.yourcompany.Recipes
kind	String	software
title	String	Recipes

iOS Application Best Practices

Goals in distributing a beta



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/
    PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>items</key>
    <array>
        <dict>
            <key>assets</key>
            <array>
                <dict>
                    <key>kind</key>
                    <string>software-package</string>
                    <key>url</key>
                    <string>http://www.mywebsite.com/Recipes.ipa</string>
                </dict>
            </array>
            <key>metadata</key>
            <dict>
                <key>bundle-identifier</key>
                <string>com.yourcompany.Recipes</string>
                <key>kind</key>
                <string>software</string>
                <key>title</key>
                <string>Recipes</string>
            </dict>
        </dict>
    </array>
</dict>
</plist>
```

Demo

Distributing an iOS beta application

Summary

- Developer ID
  - Fully Integrated Workflow in Xcode for distributing Gatekeeper compatible software
- Mac App Store
  - Distribute directly from Xcode and take advantage of App Store-specific technology
- iOS
  - Distribute test applications from Xcode
  - Minimal change yield smaller deltas

More Information

Michael Jurewitz
Developer Tools Evangelist
jury@apple.com

Documentation
Tools Workflow Guide for Mac
http://developer.apple.com/library/mac/#documentation/ToolsLanguages/Conceptual/OSXWorkflowGuide/Introduction/Introduction.html

Tools Workflow Guide for iOS
http://developer.apple.com/library/ios/DOCUMENTATION/Xcode/Conceptual/ios_development_workflow/index.html

Apple Developer Forums
http://devforums.apple.com

Related Sessions

The OS X App Sandbox	Nob Hill Tuesday 10:15AM
Gatekeeper and Developer ID	Nob Hill Tuesday 11:30AM
What’s New in iTunes Connect for App Developers	Nob Hill Thursday 9:00AM

Labs

Xcode Lab	Developer Tools Lab B Friday 9:00AM

WWDC2012