Mobile Testing Enterprise

UIAutomation Compatibilty for iOS Applications

Release 7.3

Mobile Testing Enterprise 7.3

September 2015

Copyright Notice

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT ANY EXPRESS REPRESENTATIONS OF WARRANTIES. IN ADDITION, KEYNOTE DISCLAIMS ALL IMPLIED REPRESENTATIONS AND WARRANTIES, INCLUDING ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

Information in this document is subject to change without notice and does not represent a commitment on the part of Keynote. Keynote may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents except as expressly provided in any written license agreement from Keynote.

Please forward any comments or suggestions regarding this document to Keynote Support.

Keynote LLC
777 Mariners Island Blvd.
San Mateo, CA 94404

Contents

Overview

With the latest releases of iOS and Xcode, Apple has made using Keynote's AppSign tool to change the identity of an app a lot harder to accomplish. In effect, the tool is doing steps that Apple does not want you to be able to do. There is a high possibility that Apple will eventually close all such loopholes, and we will no longer be able to sign an app for our customers.

So that you do not need to avail of Keynote's AppSign tool, follow these following steps to make your apps UIAutomation compatible and therefore able to use iOS object-level scripting Mobile Testing Enterprise.

The high-level steps below are explained further in the sections that follow. It is assumed the person performing these steps has basic-intermediate knowledge of iOS app development, specifically the inner workings of Apple iOS Certificates, Apple iOS Identifiers, and Provisioning Profiles.

1 Ensure there is a valid iOS App Development Certificate.

2 Ensure there is a valid App ID for your customer app.

3 Ensure there is a valid Development Provisioning Profile for your app.

4 Ensure the Development Provisioning Profile contains UDIDs of devices the app will be deployed to.

5 Build the customer App using the proper App Development Certificate, App ID, and Development Provisioning Profile under Xcode.

6 Make the app UIAutomation compatible.

1 iOS App Development Certificate

To create a certificate:

1 Log in to the Member Center at https://developer.apple.com/account/ios/certificate/certificateList.action.

2 Click the plus symbol at the top right to add a new certificate.

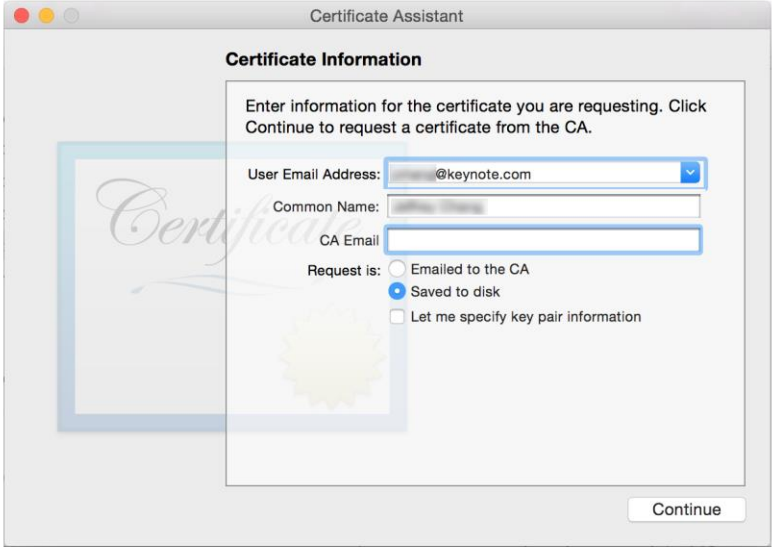
- For a development certificate, select **iOS App Development** and click **Continue**.
- For a distribution certificate, select **App Store and Ad Hoc** and click **Continue**.

3 Follow the instructions on the page for creating a Certificate Signing Request.

4 Open Keychain Access on your Mac.

5 Select **Keychain Access** > **Certificate Assistant** > **Request a Certificate from a Certificate Authority**.

6 Follow instructions to create a certificate:



Certificate Information

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

User Email Address: [redacted]@keynote.com

Common Name: [redacted]

CA Email:

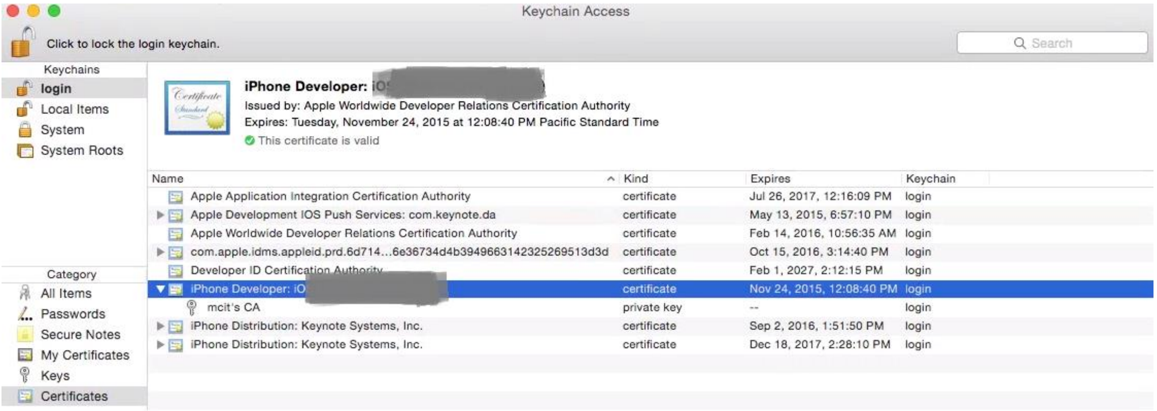
Request is: ○ Emailed to the CA
● Saved to disk
□ Let me specify key pair information

7 Once you have the CSR file, continue with the web page for creating a Certificate. The page will ask you to select the CSR to generate the certificate.

8 Download the certificate.

9 Locate the certificate and double click it to add it to your Keychain.

10 If everything is done correctly, you will see a new Certificate along with a Private key in the Keychain Access:



Click to lock the login keychain.

iPhone Developer: iO[redacted]
Issued by: Apple Worldwide Developer Relations Certification Authority
Expires: Tuesday, November 24, 2015 at 12:08:40 PM Pacific Standard Time
✓ This certificate is valid

Name	Kind	Expires	Keychain
Apple Application Integration Certification Authority	certificate	Jul 26, 2017, 12:16:09 PM	login
▶ Apple Development IOS Push Services: com.keynote.da	certificate	May 13, 2015, 6:57:10 PM	login
Apple Worldwide Developer Relations Certification Authority	certificate	Feb 14, 2016, 10:56:35 AM	login
▶ com.apple.idms.appleid.prd.6d714...6e36734d4b394966314232526951 3d3d	certificate	Oct 15, 2016, 3:14:40 PM	login
Developer ID Certification Authority	certificate	Feb 1, 2027, 2:12:15 PM	login
▼ iPhone Developer: iO[redacted]	certificate	Nov 24, 2015, 12:08:40 PM	login
mcit's CA	private key	--	login
▶ iPhone Distribution: Keynote Systems, Inc.	certificate	Sep 2, 2016, 1:51:50 PM	login
▶ iPhone Distribution: Keynote Systems, Inc.	certificate	Dec 18, 2017, 2:28:10 PM	login

2 Valid Customer App ID

Steps to create an app ID:

1 Download the certificate.

2 In the Member Center, under **Certificates, Identifiers, & Profiles**, click on **App IDs under Identifiers**.

3 Click the plus symbol at the top right of the page.

4 For **App ID Description**, type the desired app ID

5 Select all the services and click **Continue**.

6 Click **Submit**.

3 Development Provisioning Profile

To ensure there is a Development Provisioning Profile for your app:

1 In the Member Center, under **Certificates, Identifiers, & Profiles**, select **Provisioning Profiles** > **Development**.

2 Click the plus symbol at the top right of the page.

3 For a development profile, click on **iOS App Development**.

4 When asked to select an App ID, choose the ID for your app.

5 Select the certificate used with the profile.

6 Then select the devices that will be included in the provisioning profile.

7 Name the profile.

8 Download the profile. It will need to be added to devices in order to run apps that have been signed.

4 UDIDs in the Development Provisioning Profile

Ensure that the Development Provisioning Profile contains all the UDIDs of devices the app will be deployed to.

To add a device UDID to the Provisioning Profile:

1 Log in to the Apple Developer Portal.

2 Click on **Provisioning Profiles**.

3 Select the appropriate profile.

4 Click on the **Edit** button.

5 Select the desired device that you would like the app to be deployed to.

6 Click **Generate** to generate a new provisioning profile if there are any changes made.

5 Build the App

Build the app using the proper App Development Certificate, App ID, and Development Provisioning Profile under Xcode:

1 Launch Xcode and open your project.

2 Under the **Build Settings** menu, select the appropriate Development Code Signing Identity and Development Provisioning Profile.

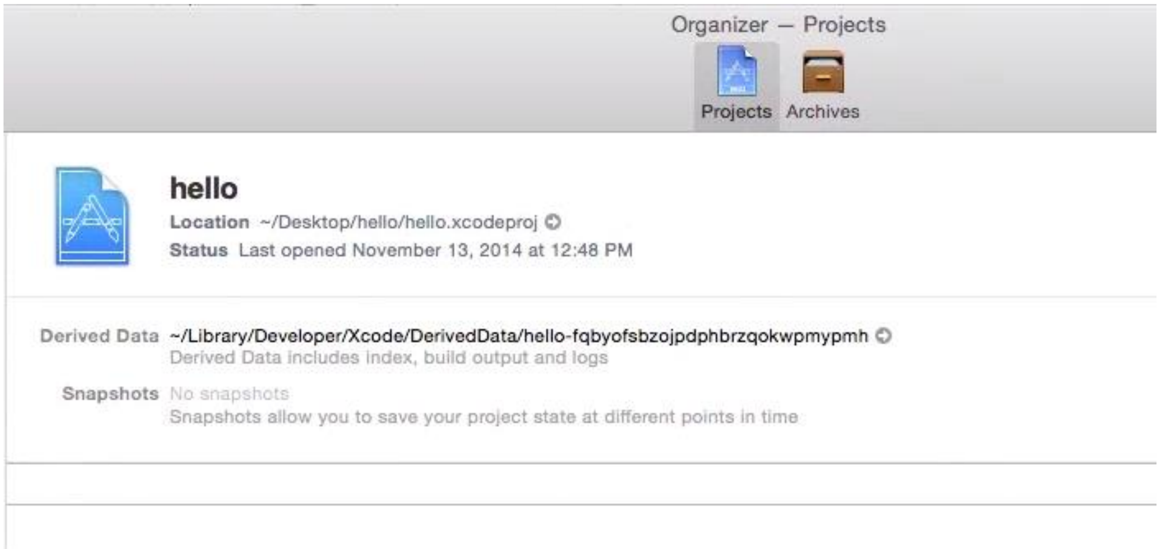
NOTE You might have to exit Xcode and reopen the project for the build to see changes.

3 Ensure the build is successful.

6 UIAutomation Compatibility

To create a UIAutomation compatible app:

1 Locate your app's derived data. This can be located by going to Xcode Organizer and clicking on the project. You will see a Derived Data main folder like this:



hello

Location ~/Desktop/hello/hello.xcodeproj

Status Last opened November 13, 2014 at 12:48 PM

Derived Data ~/Library/Developer/Xcode/DerivedData/hello-fqbyofsbzojpdphbrzqokwpmypmh
Derived Data includes index, build output and logs

Snapshots No snapshots
Snapshots allow you to save your project state at different points in time

2 From the Derived Data main folder, navigate to the Debug derived data folder. Typically, they are located under a folder structured like this:

```
/Users/<UserName>/Library/Developer/Xcode/DerivedData/<AppName>/Build/Products/Debug-iphoneos/<AppName>.app
```

3 Copy this folder into a folder named `Payload`.

4 Zip the `Payload` folder and name it with an `.ipa` file extension, e.g., `hello.ipa`.

Now you have a UIAutomation compatible app.

NOTE Navigating the file system using OS X's Finder App to perform the above steps will add a hidden indexing file called `.DS_Store` and that may create problems if included. Performing the above steps under the command line tool will not have these side effects:

Starting from the `Debug-iphoneos` directory,

```
cp -aXv <APPNAME> Payload
```

For example, if you are in

```
/Users/<USERNAME>/Library/Developer/Xcode/DerivedData/<APPDATA>/Build/Products/Debug-iphoneos
```

issue the command:

```
cp -aXv hello.app Payload
# zip -r hello.ipa Payload
```