18 – Code Signing and Distribution

Computer Science Department
California State University, Sacramento

Overview

- Common Terms in Signing and Distribution
  - Certificate, Provisioning Profile, Development, Distribution, Signing Authority, UDID
- iOS vs Android Signing and Distribution
- iOS/Android Signing and Distribution Details
- Building/Downloading/Installing Native Apps using CN1

Common Terms

- Code Signing and Distribution requires:
  - **Certificate**
  - **Provisioning profile**
- **Development** (debug) is testing your app on your own device whereas **distribution** (release) involves publishing it on a marketplace (e.g., Apple's App Store or Google's Play Store)
- Certificates are issued by a **signing authority**
  - A body that certifies that you are who you say you are

Certificate and Provisioning Profile

- Certificate is like a company stamp or your signature
  - You use it to sign an app so the users know who it is from
  - You can use the same certificate for all your apps
- Provisioning profile gives hints/guidelines for the application installation
  - You choose which devices can run your app and which app services your app can access
  - Gives details about the application and who is allowed to execute it

UDID

- UDID (Universal Device Identifier) identifies mobile devices uniquely
- It is used in creating provisioning profiles
- Some operating systems (e.g., iOS) block access to this value due to privacy concerns:
  - Don't use an iOS app to get the UDID since most return the wrong value. The official way to get the UDID is through iTunes.

Signing Authority

- Apple issues certificates for iOS development
  - To generate a certificate you need to be registered to Apple Developer Program ($99/ year).
- Android uses self-signed certificates
  - Anyone can ship Android app
  - Certificate indicates that you are the same person who previously ship the app because the app can only be updated with the exact same certificate

Signing and Distribution: iOS vs Android

- iOS process is complicated compared to Android process
- For iOS, you need a separate certificate (.p12 file) and provisioning profile (.mobileprofile file) files
  - For Android, you only need a certificate file (.ks file)
- For iOS, you need certificate and provisioning file for development as well as distribution
  - For Android, you only need it for distribution

Signing and Distribution: iOS vs Android (cont)

- You should re-use the same certificate for all iOS apps you develop
  - This is recommended for all platforms for simplicity
  - But some Android developers prefer creating per-application certificates. This way they can easily transfer ownership of different apps to different developers.
- You need to be registered to the Apple Developer Program for iOS.

iOS Signing and Distribution Details

- You need a different certificate and provisioning profile files in development and distribution
  - You need to generate two pairs of files (four in total).
- Usually requires a Mac, but:
  - In CN1, You can use iOS Signing Wizard to generate certificates/provisioning files without requiring a Mac or deep understanding of the signing process for iOS
  - You can use other software running on Windows…

iOS Signing Wizard

You can reach the iOS Signing Wizard from:

Right click on the project →
CodenameOne →
CodenameOne Settings →
Build Settings →
Device Settings →
iOS →
Certificate Wizard

iOS Signing and Distribution Details (cont.)

- If you already have the certificate and provisioning profile, you do not need to use the Wizard. Simply enter them to "iOS Signing" properties panel (Right click on the project → CodenameOne → CodenameOne Settings → Build Settings → Device Settings → iOS → Signing).
- See "Advanced iOS Signing" of CN1 Developer Guide for generating certificate/provisioning profile without the Wizard.
- While creating your provisioning file, you need to list UDID of the iOS devices used during development/testing.

iOS Signing and Distribution Details (cont.)

- Apps signed with development certificate can only be installed on one of the iOS devices added to the provisioning profile (e.g., your own device).
- Apps signed with distribution certificate cannot be directly installed to your own iOS device. First, you need to upload it to iTunes. Then you can test this build and submit it to App Store.

Android Signing and Distribution Details

- CN1 provides an automated tool also for generating Android certificate (i.e., keystore file with extension .ks)
- The certificate can also be generated manually using JDK's keytool executable
  - See CN1 Developer Guide for more details

Building Native Apps in CN1

- After you populate the signing properties panel for iOS/Android with proper certificates (and provisioning profiles), you can send your app to CN1 build servers to generate native apps.
- Build servers take in CN1 code and generate a native code that can run on Android/iOS/Windows Phones, Mac/Windows Desktop environments, etc...
- Then you can upload these native apps on your devices (in development mode) or distribute them on marketplaces (in distribution mode).

Building Native Apps in CN1 (cont.)

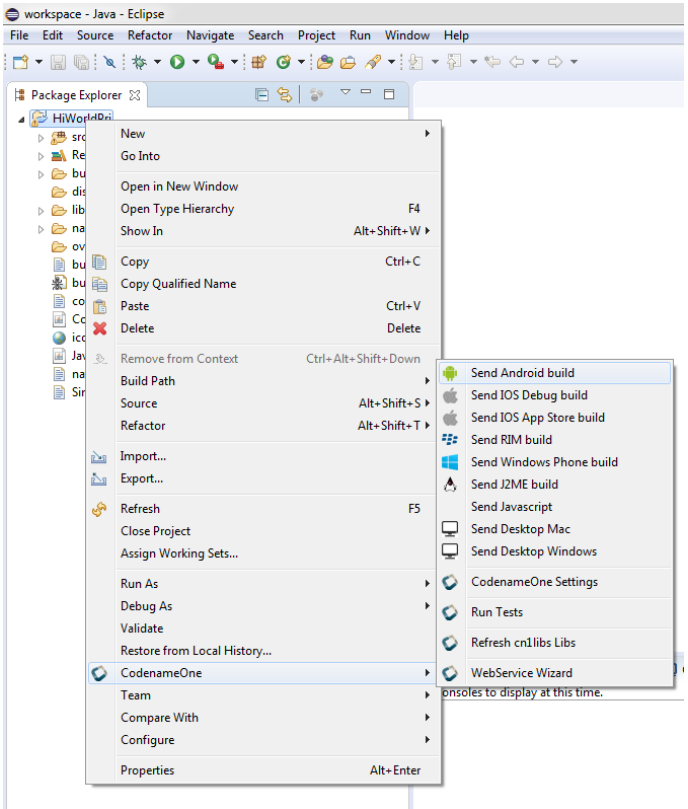
- First you need to create a Codename One user id and password.
- You need to use this id/password to send your code to build servers.
- Send your code to servers as follows:

Right Click on Project →

Codename One →

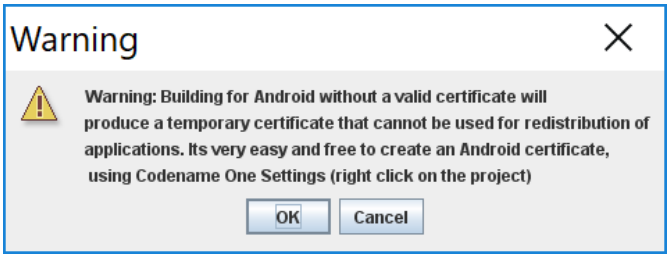
Send to Android Build (or Send to iOS Debug/App Store, etc…)

Sending CN1 Code for Android Build



Sending CN1 Code for Android Build (cont.)

- If you have NOT specified a certificate for your project, you will receive this warning:



- This means that after you download the build to your Android device, to install it, you need to change the device settings to allow installation from “Unknown Sources”…

Downloading/Installing Native Apps

- Sign in with the same id/password to the CN1 website (codenameone.com).
- Go to “Dashboard” to see the status of your build.
- When the build is done, download and install it to your device (i.e., the .apk file for Android app).
- You can also use a QR reader to download the app to your device.

Downloading the Native Android App

(Screenshot of "Dashboard" in CN1 website)