Android Licensing

Sang Shin
Michèle Garoche
www.javapassion.com
"Learn with Passion!"

Disclaimer

- Portions of this presentation are modifications based on work created and shared by the Android Open Source Project
  - http://code.google.com/policies.html
- They are used according to terms described in the Creative Commons 2.5 Attribution License
  - http://creativecommons.org/licenses/by/2.5/

Topics

- Android Market & Licensing service
- Steps of implementing licensing service

Android Market & Licensing Service

Licensing Service From Android Market

- Android Market offers a licensing service (running on the server) that lets you enforce licensing policies for paid applications that you publish through Android Market.
- Your application can query Android Market at run time to obtain its licensing status for the current user, then allows or disallows further use as appropriate.

App-specific Licensing Policy

- Using the licensing service (running on the server), you can apply a flexible licensing policy on an application-by-application basis — each application can enforce licensing in the way most appropriate for it.
- If necessary, an application can apply custom constraints based on the licensing status obtained from Android Market.
  - For example, an application can check the licensing status and then apply custom constraints that allow the user to run it unlicensed for a specific number of times, or for a specific validity period.
  - An application can also restrict use of the application to a specific device, in addition to any other constraints.

Secure Communication with Licensing Service

- The licensing service is a secure means of controlling access to your applications.
- When an application checks the licensing status, the Market server signs the licensing status response using private key of key pair that is uniquely associated with the publisher account.
- Your application stores the public key in its compiled .apk file and uses it to verify the licensing status response.
  - You will have to acquire a public key from your publisher account account page and then provide it for each of your application

Steps of Implementing Licensing in Your Application

Steps

- Step #1 - Get public key of your publisher account (one-time task)
- Step #2 - Set up licensing-enabled runtime environment (one-time task)
- Step #3 - Download and install License Verification Library (LVL) (one-time task)
- Step #4 - Sign in to an authorized account in the runtime environment (one-time task)
- Step #5 - Add licensing support code to an application (for each application)
- Step #6 - Test licensing using either publisher account or test account (for each application)

Step #1 - Get Public Key of Your Publisher Account

How Public Key is used in Licensing (1)

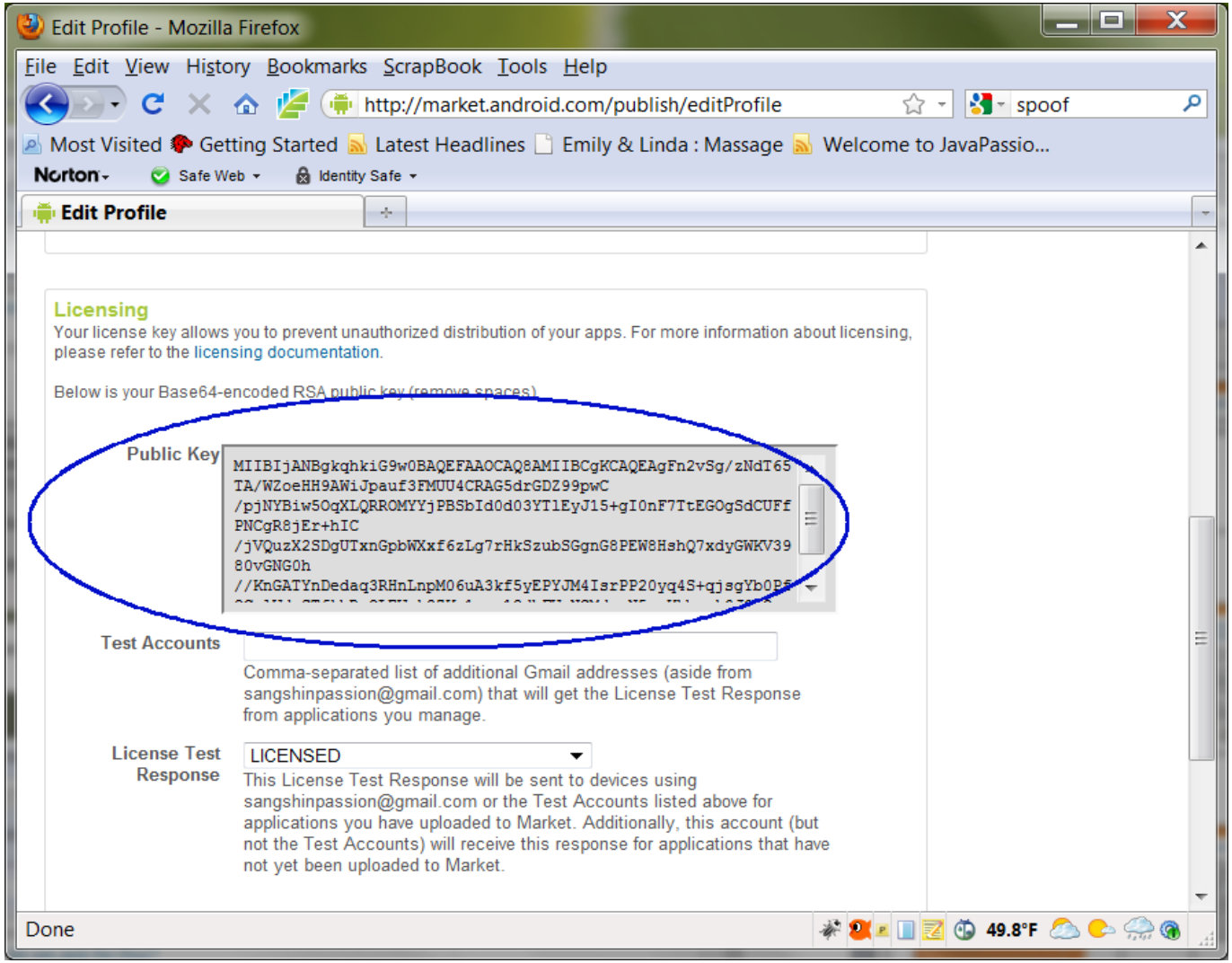
- The licensing service (running on the server) generates a single licensing public key and private key pair for each publisher account and exposes the public key in the account's profile page
- The publisher copies the public key and embeds it in the application source code, then compiles and publishes the .apk.
- The licensing server retains the private key internally and uses it to sign license responses for applications published on that account.

How Public Key is used in Licensing (2)

- When the application receives a signed response, it uses the embedded public key to verify the data.
- The use of public key cryptography in the licensing service makes it possible for the application to detect responses that have been tampered with or that are spoofed.
  - Data integrity (addresses tampering) – making sure the data has not been changed
  - Authentication (addressed spoofing) – making sure the identity of the licensing service (running on the server)

Note: spoofing attack is a situation in which one person or program successfully masquerades as another by falsifying data

Step #1: Get Public Key



Step #2 - Set up Licensing-enabled Runtime Environment

Why Licensing Runtime Environment?

- Applications check licensing status not by contacting the licensing server directly, but by binding to a service provided by the Android Market application (running on the device – we will call it Android Market service) and initiating a license check request.
- The Android Market service then handles the direct communication with the licensing server (running on the server) and finally routes the response back to your application.
- To debug and test licensing in your application, you need to set up a runtime environment that includes the necessary Android Market service, so that your application is able to send license check requests to the licensing server.

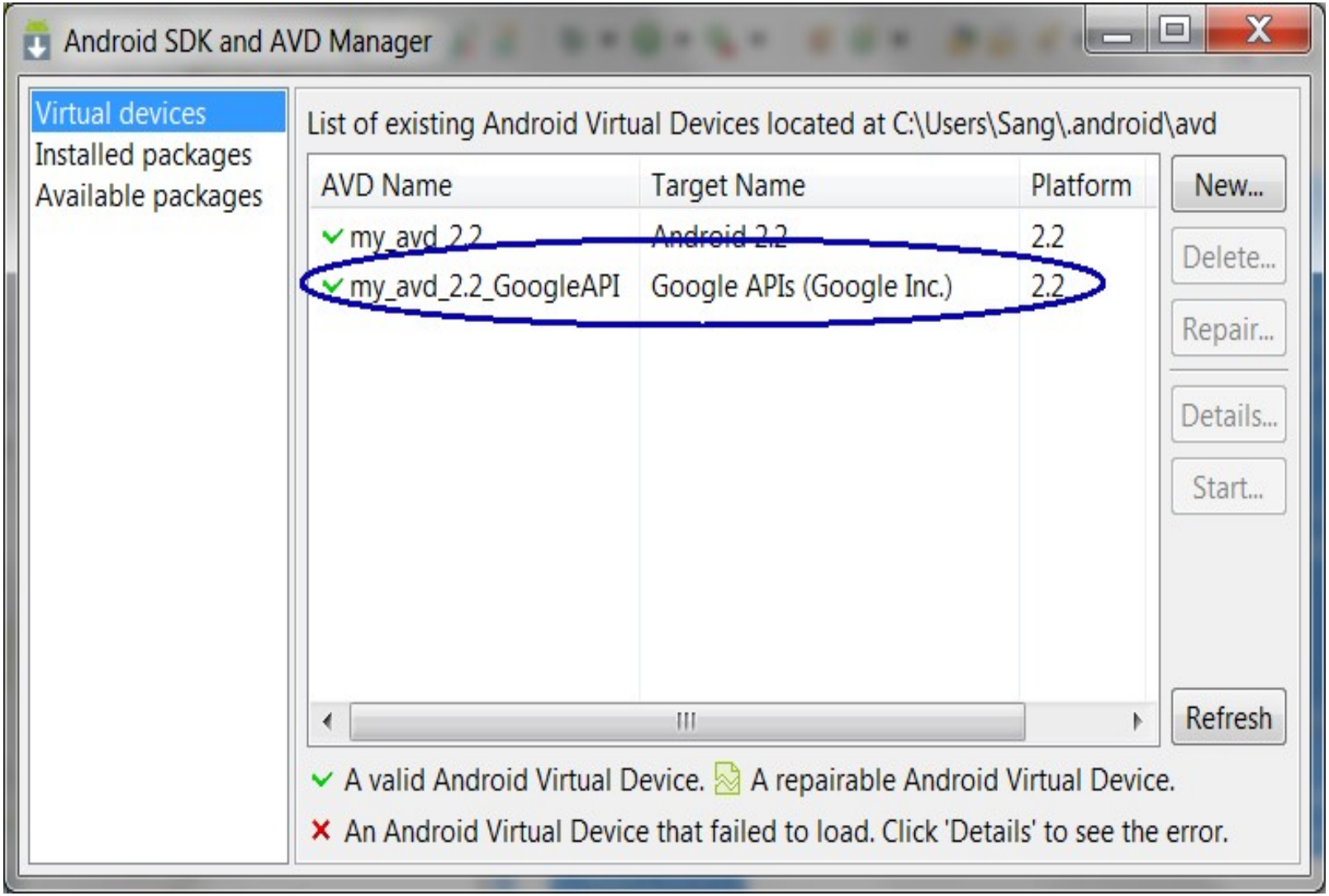
Two Licensing-enabled Runtime Env's

- An Android-powered device that includes the Android Market application, or
- An Android emulator running the Google APIs Add-on, API level 8 (release 2) or higher
  - Standard Android platforms provided in the Android SDK do not include Android Market

Google APIs Add-on, API Level 8

- The Google APIs Add-On does not include the full Android Market client. However, it does provide:
  - An Android Market background service that implements the *ILicensingService* remote interface, so that your application can send license checks over the network to the licensing server.
  - A set of underlying account services that let you add an a Google account on the AVD and sign in using your publisher account or test account credentials. Signing in using your publisher (your Google account) or test account enables you to debug and test your application without having publish it.

Licensing-enabled Virtual Device (AVD)

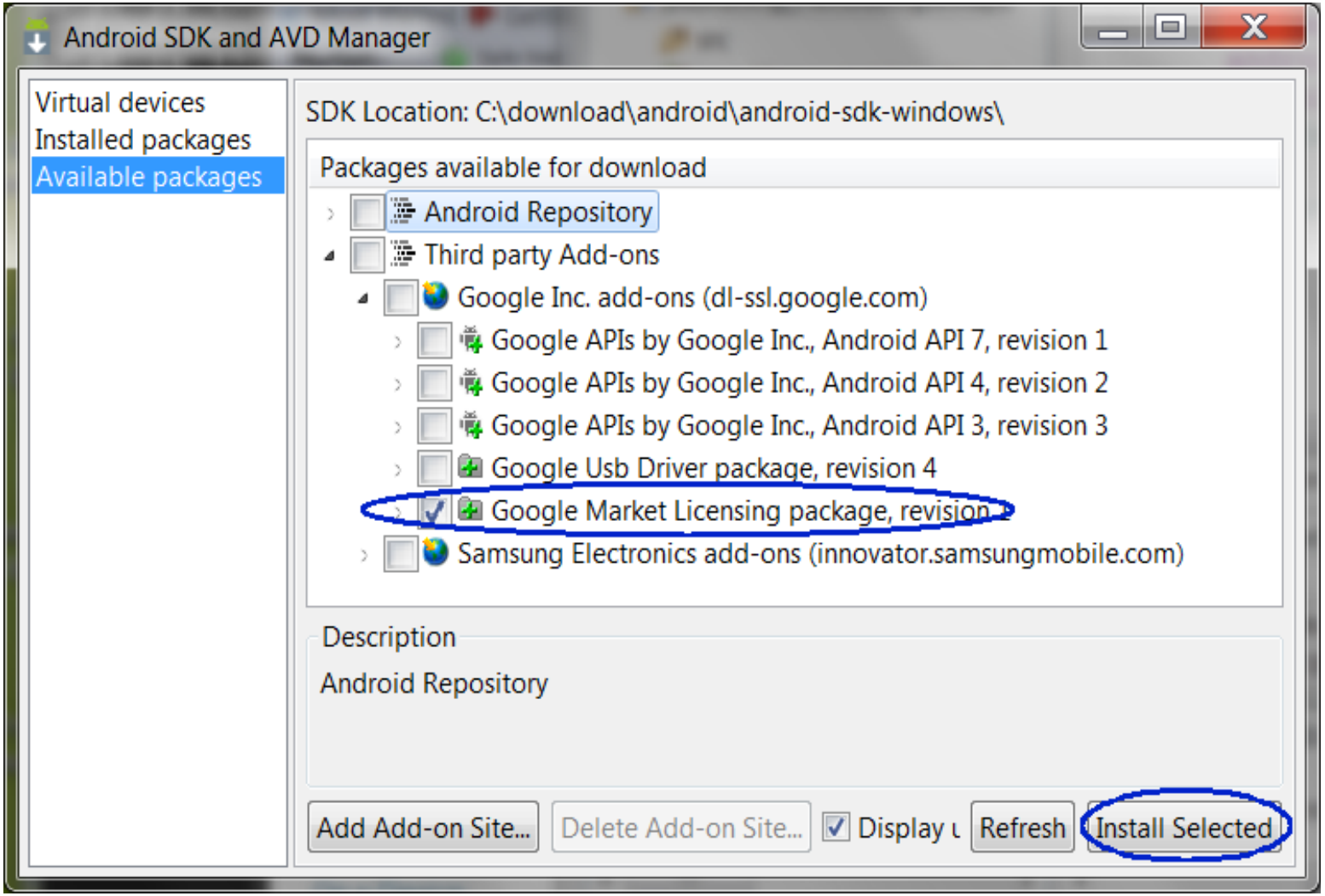


Step #3 - Download and Install License Verificaiton Library (LVL)

What is License Verification Lib (LVL)?

- A collection of helper classes that greatly simplify the work that you need to do to add licensing to your application
- Provided as part of Google Market Licensing Package

Google Market Licensing Package

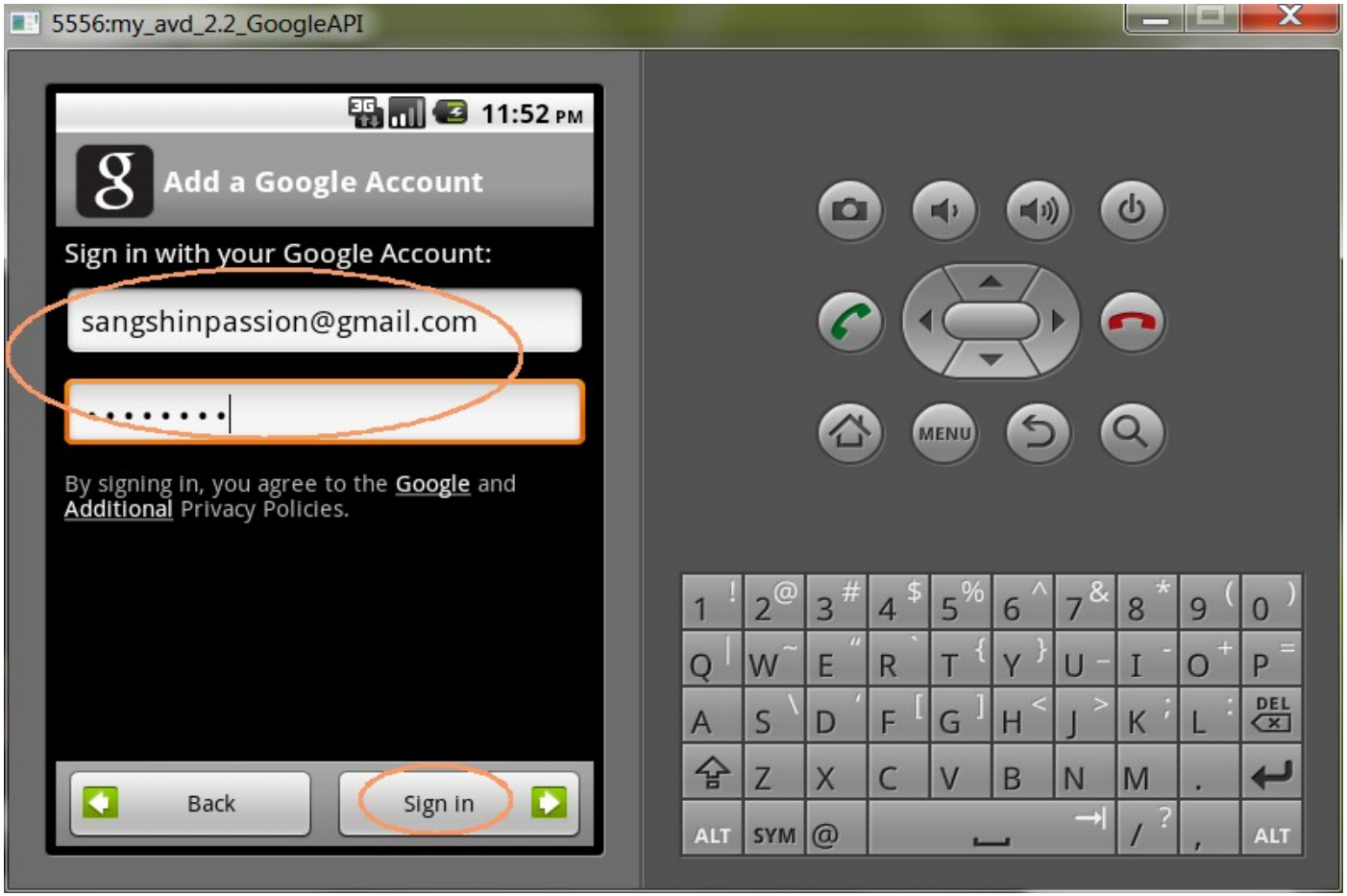


Step #4 - Sign in to an Authorized Account in the Licensing-enabled Runtime Environment

Why User Account?

- The licensing service is designed to determine whether a given user is licensed to use a given application — during a license check, the Android Market application (running on a device or emulator) gathers the user ID from the primary account on the system and sends it to the server, together with the package name of the application and other information.
- If there is no user information available, the license check cannot succeed, so the Android Market application terminates the request and returns an error to the application.

Signing with Publisher Account as a User

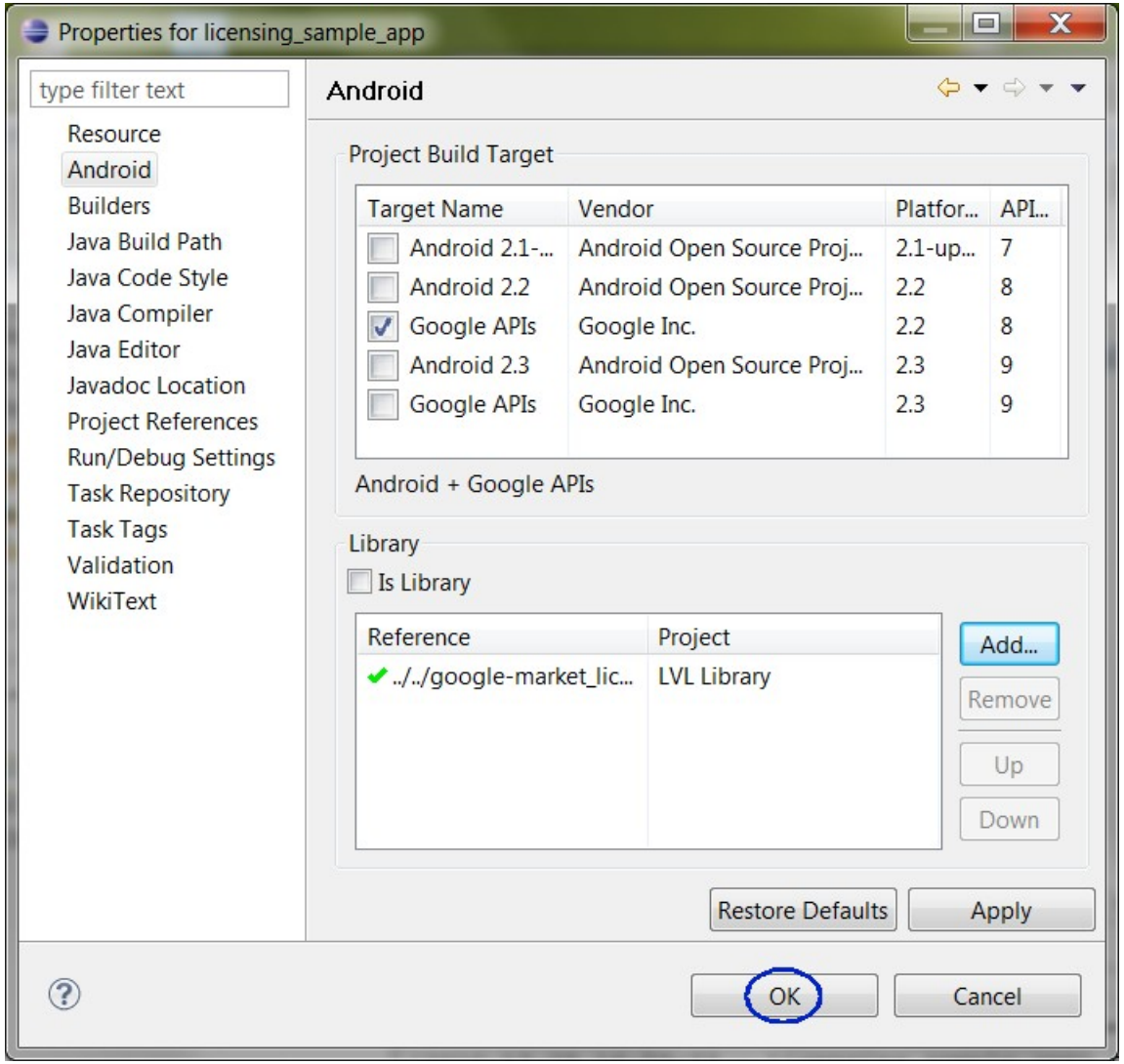


Step #5 - Add License Support Code to an Application

Things to be done

- Add License Verification Library (LVL) to the project
- Add licensing permission to the manifest
- Add licensing support code to an activity
  - Implement a policy
  - Implement an obfuscator
  - Check the license
  - Implement a DeviceLimiter

Add LVL to the project



Add licensing to the manifest

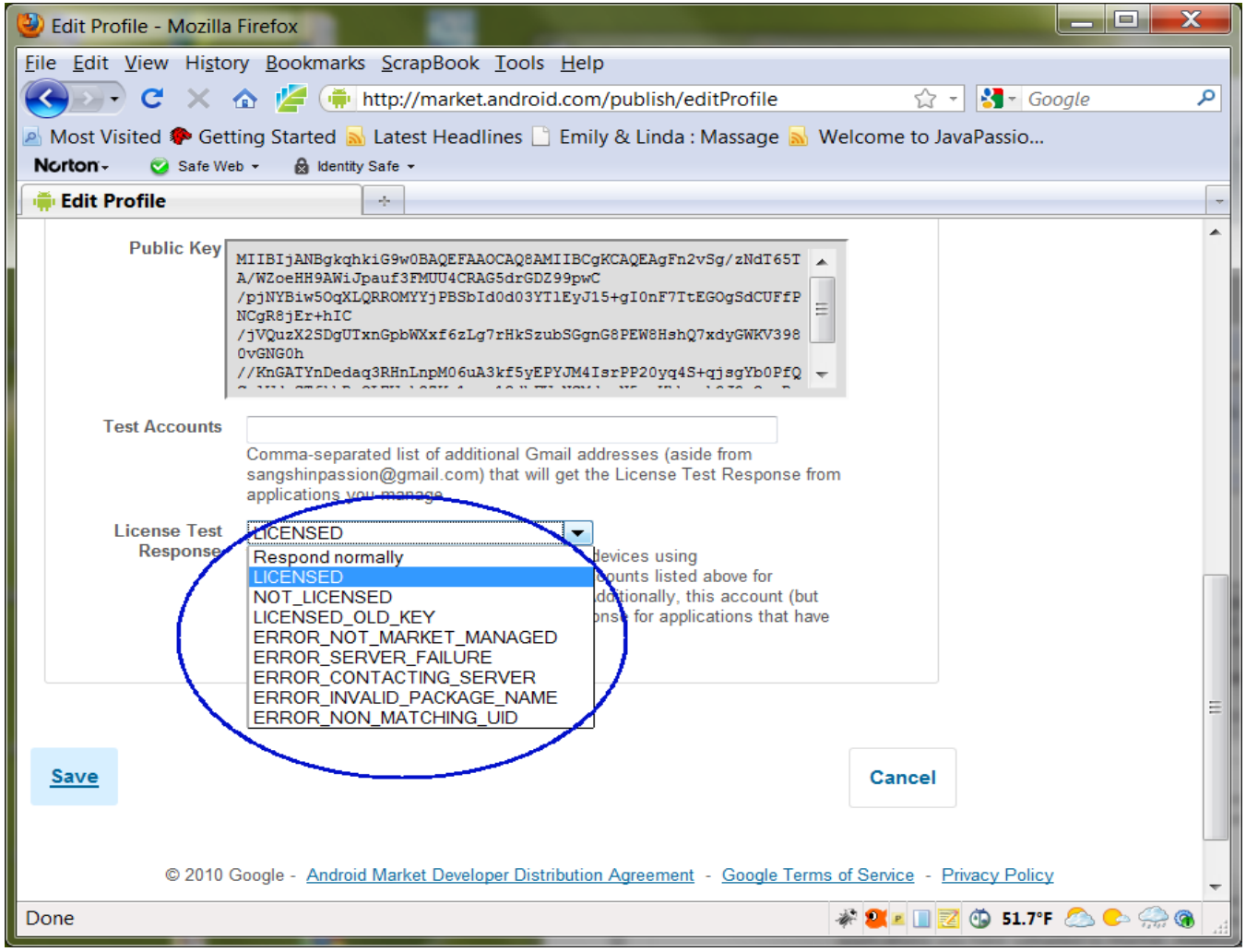
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.javapassion"
    android:versionCode="2"
    android:versionName="1.1">
    <application android:icon="@drawable/icon"
                 android:label="@string/app_name">
        <activity android:name=".MainActivity" android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

    </application>
    <uses-sdk android:minSdkVersion="8" />
    <!-- Required permission to check licensing. -->
    <uses-permission android:name="com.android.vending.CHECK_LICENSE" />
</manifest>
```

Add Licensing Support Code

- Choose and implement a policy among the following
  - ServerManagedPolicy (recommended)
  - StrictPolicy
  - Custom policy
- Implement an obfuscator
- Check the license
- Implement a DeviceLimiter

Step #6 - Test Licensing

Setting Test License Response



Thank you!

Sang Shin
Michèle Garoche
http://www.javapassion.com
“Learn with Passion!”