Google Android Development

Lesson #7

Publishing Your App

Google provides a centralized market place for applications known as the Android Marketplace.

You can publish your application to that market place and make it available for anyone to download via their phone.

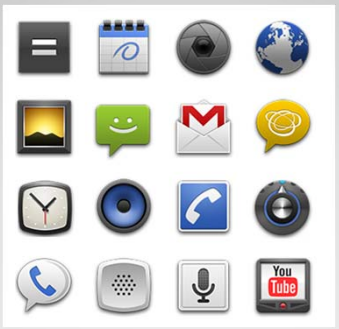
The only restrictions applied are:

- Location – Some apps are not available in different markets.
- OS Version – If you developed something using Android 2.2, phones using Android 2.1 and earlier won't see your app.

Icon

First off, you'll need a nice icon for your application. Google has made available a Android Icon Template Pack, and guidelines, available here:

http://developer.android.com/guide/practices/ui_guidelines/icon_design.html



Signing Your App

Your application must be digitally signed in order to be released into the market. By default, Eclipse creates a debug certificate for you and signs your application when you use it on your device or AVD during development. However, in order to release it to the official market place, you'll need to create your own certificate, and sign the application with it.

There are two ways to create the certificate – we can use command-line tools supplied by our Java SDK, or let Eclipse do this for us during the export process.

The certificate you create must expire after October 22nd, 2033.

AndroidManifest.XML

The AndroidManfest.xml file within your project must contain the following properties:

- **android:versionCode** – Identifies the build number of the app. You will manually increment this as you release updates to the app. Example: 1, 2, 3, 4, etc.
- **android:versionName** – This is what the user sees as far as version is concerned. For example, I tend to use 1, 2, 3, 4 as by versionCode, but will use 1.01, 1.02, 1.03, and 1.04 as my VersionName.
- **android:icon** – Identifies the location for the image used for the icon.
- **android:label** – Points to a string reference, or contains the name of your application.

These properties help the market place manage your application, and identify updates, and help users search for your app.

Example

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="packagename.goes.here"
      android:versionCode="4"
      android:versionName="1.04">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".Activity_Main"
                  android:label="@string/app_name"
                  android:screenOrientation="portrait">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

    </application>
    <uses-sdk android:minSdkVersion="3" />

</manifest>
```

Android Marketplace

You'll need to create an account at http://market.android.com/publish.

There is a $25 fee to register. Once the account is created, you can use to publish the .APK file that you get when you prepare your application for release using Eclipse.

Publishing Your App

When publishing your app, you'll need the .APK file, and I would strongly recommend two screenshots, and a promotional graphic.

Many people don't look at the description, but will buy simply by looking at the screenshot. If you do not include these, you'll miss out on many users. The promotional graphic is used on the first screen the Market Place that highlights well received apps.

You will also need to provide a title, description, promo text, choose a category for your app, and set a price (if any).

You'll also need to decide on copy protection, and location for your app. Finally, you'll enter your contact info, acknowledge the Android content guidelines, and publish your app.

If all went well, it will immediately be available in the market place.

Your App

If you have a debug version of your application installed on your phone, you'll want to remove that before attempting to download the published version from the app store. Because of the difference in keys between debug and production use, production apps won't install over debug apps, and vise versa.

The End

We've only just scratches the surface when it comes to Android. This class provides a foundation for you to build on, and I believe you are ready to tackle your first project.

As you start to develop using this platform, you'll uncover many new things, and a lot of additional functionality. Use each project you create as a way to move forward, each time pushing the code just a bit further than you did last time. Jot down different techniques and ways of doing things for later re-use. Keep a code snippet library, or create a set of projects used simply to house various classes and methods and techniques.

Overall, I hope you enjoyed the class, and I will look forward to downloading YOUR applications from the Android Market Place!