A Beginner's Guide to Android Application Development

Preeti U Murthy
Python Member
National Institute of Technology Karnataka
Surathkal, India
preeti.murthy@gmail.com

***Abstract*—Android is a mobile operating system initially developed by Android Inc. Android was bought by Google in 2005. Android has a large community of developers writing application programs ("apps") that extend the functionality of the devices. There are currently over 200,000 apps available for Android. The Android software development kit (SDK) includes a comprehensive set of development tools.**

I. Introduction

Android is built on top of a modified version of the Linux kernel [1]. Canalys reported that in Q4 2010 the Android O.S. was the world's best-selling smartphone platform, dethroning Nokia's Symbian from the 10-year top position [2]. Developers write primarily in the Java language, controlling the device via Google-developed Java libraries. These include a debugger, libraries, a handset emulator, documentation, sample code and tutorials.

For the ease of development, the Android SDK includes a mobile device emulator - a virtual mobile device that runs on your computer. The emulator lets you prototype, develop, and test Android applications without using a physical device.

This document proposes a methodology to set up the android development platform and get started with building android applications. Section 2 details the proposed method to set up the development environment, Section 3 describes briefly about the mobile device emulator and the launching procedure, Section 4 identifies resources to proceed with working off the SDK, and Section 5 concisely summarizes this document's purpose and future activities the reader can pursue.

II. Setting Up

Throughout, it is assumed that the Operating system in question is Linux.The recommended IDE to develop android applications is Eclipse. Following which, the Android SDK needs to be installed.

Android offers a custom plugin for the Eclipse IDE, Android Development Tools (ADT), that is designed to give you a powerful, integrated environment in which to build Android applications. Following the installation of the android SDK, the ADT is installed.

The following section takes you through these steps. Confirm if your system has JDK 5 or JDK 6, before beginning the following steps.

A. Installing Eclipse

To install Eclipse, download it from the Eclipse Foundation's website. Make sure the version is 3.5 or higher. Several types of Eclipse packages are available for each platform. Download the package meant for Linux. For developing Android applications, it is recommended that you install one of these packages:

- Eclipse IDE for Java Developers
- Eclipse Classic (versions 3.5.1 and higher)
- Eclipse IDE for Java EE Developers

B. Installing the Android SDK

The next step is to get the Android Software Development Kit (SDK) working in Eclipse.

Get the latest version of the SDK starter package from the SDK download page. On downloading unpack the `.zip` or `.tgz` package to a safe location on your machine. By default, the SDK files are unpacked into a directory named `android-sdk-<machine-platform>...`

Make a note of the name and location of the SDK directory on your system - you will need to refer to the SDK directory later, when setting up the ADT plugin and when using the SDK tools from command line.

C. Installing the ADT

1) Start Eclipse, then select Help –> Install New Software
2) Click Add, in the top-right corner.
3) In the Add Repository dialog that appears, enter "ADT Plugin" for the Name and the following URL for the Location:

   `https://dl-ssl.google.com/android/eclipse/`[1]

   Click **OK**.
4) In the Available Software dialog, select the checkbox next to Developer Tools and click **Next**.

[1] If you have trouble acquiring the plugin, try using "http" in the Location URL, instead of "https" (https is preferred for security reasons).

5) In the next window, you'll see a list of the tools to be downloaded. Click **Next**.
6) Read and accept the license agreements, then click **Finish**.
7) When the installation completes, restart Eclipse.

Now you need to make the ADT plugin point to the SDK.

1) Select **Window** –> **Preferences** to open the Preferences panel
2) Select **Android** from the left panel.
3) For the *SDK Location* in the main panel, click **Browse...** and locate your downloaded SDK directory.
4) Click **Apply**, then **OK**.

D. Adding Components

The SDK starter package is not a full development environment - it includes only the core SDK Tools, which you can use to download the rest of the SDK components (such as the latest Android platform).

The Android SDK and AVD Manager helps you download these components.They can be launched from within Eclipse - select **Window** –> **Android SDK and AVD Manager**.

The Android SDK and AVD Manager's **Available Packages** panel shows the SDK components that are available for you to download into your environment. Android SDK tools are pre-installed with the SDK starter package. Select the following to be downloaded:

1) Android SDK platform-tools
2) Android SDK platform 2.2
3) Samples and Documentation

The Android SDK and the AVD Manager will install the above packages. If you encounter problems while installing the above, launch the manager by browsing to the /tools directory in the SDK installation directory and type 'android'.

Ok, all set to get started!

E. Creating an AVD

Before creating a new Android project, you need to create a virtual machine. An AVD has the configuration for the emulator you run your android applications on. To do so:

1) Select **Window** –> **Android SDK and AVD Manager**, or click the Android SDK and AVD Manager icon in the Eclipse toolbar.
2) In the *Virtual Devices* panel, you'll see a list of existing AVDs. Click **New** to create a new AVD.
3) Fill in the details for the AVD.
   Give it a name, a platform target, an SD card size, and a skin (HVGA is default). Mention the platform target to be Android SDK platform 2.2 and the API version you selected earlier to be downloaded.
4) Click **Create AVD**.

You can begin by creating a simple Hello World project [3]. To develop in any other IDE visit the android developer guide.

III. Android Emulator

Android emulator is a QEMU based application, which means it can run programs written for one machine on another machine. The Java programs that we write are meant to run on an ARM machine. The QEMU layers of the emulator provides binary dynamic translation of the ARM machine code to the machine code of your system.

During development and testing of your application, you can install and run your application in the Android emulator. You can launch the emulator as a standalone application, from a command line, or you can use it as part of your Eclipse development environment. If you are doing the former, you need to mention the AVD configuration to load.

In *Section III.E - Creating an AVD* - we provided details about creating a virtual machine. Using the below command line, you specify which virtual machine to load:

```
emulator -avd <avd_name>
```

To run your application in an emulator, you can simply click RUN in Eclipse. The emulator, meant for the target platform, that your application is running on, is loaded to run your application. The android home screen is loaded. Click on the menu button in the emulator to see your application running.

To stop an emulator instance, just close the emulator's window.

IV. Resources

A. Notepad tutorial

The notepad tutorial gives you a "hands-on" introduction to the Android framework and the tools you use to build applications on it. Starting from a preconfigured project file, it guides you through the process of developing a simple notepad application and provides concrete examples of how to set up the project, develop the application logic and user interface, and then compile and run the application.

The tutorial presents the application development as a set of exercises each consisting of several steps. You should follow the steps in each exercise to gradually build and refine your application. The exercises explain each step in detail and provide all the sample code you need to complete the application.

When you are finished with the tutorial, you will have created a functioning Android application and will have learned many of the most important concepts in Android development.

To begin with the tutorial, do the following:

1) Download the project exercises archive (`.zip`)
2) Unpack the archive file to a suitable location on your machine.
3) Open the `NotepadCodeLab` folder.
   Inside the `NotepadCodeLab` folder, you should see six project files: `Notepadv1`,

`Notepadv2`, `Notepadv3`, `Notepadv1Solution`, `Notepadv2Solution` and `Notepadv3Solution`. The `Notepadv#` projects are the starting points for each of the exercises, while the `Notepadv#Solution` projects are the exercise solutions.

It is advised you go through the remaining resources only after familiarising yourself with the Notepad tutorial.

B. SQLite

The SQLite database is used in Android for storing and managing application data. SQLite is an open-source lightweight transaction SQL database engine [4]. There are other means of storing application data in Android rather than database called Preferences, but those are more often used to store a very small amount of information like usernames and passwords. For sample apps that demonstrate how to use SQLite databases in Android, take a look at the NotePad and Searchable Dictionary applications.

C. Developing Web Apps

Targeting Screens from Web Apps is simple to do for Android apps. Android provides support to properly size your web app on Android-powered devices and support multiple screen densities.

If you want to deliver a web application (or just a web page) as a part of a client application, you can do it using WebView [5].

V. Conclusion

Android development tools allow 2D and 3D graphics, audio, video, maps and bluetooth to be easily integrated with your application. The developer's page on the android home page, is a very useful and extensive guide to building rich applications for android.

Many simple apps have gained remarkable popularity. Recognising what kind of application strikes the user community is getting half the work done. This document is meant to ease the work from then on. We hope this document enables you to start building applications for the Android platform.

References

[1] P. K. Roy, "The rise of the Droid", *pkr.in*, 2010.

[2] Canalys. "Googles Android becomes the worlds leading smart phone platform" *Canalys Research Release*, 2011.

[3] Google Inc. "Hello World — Android Developers", *Android Developer Resources*, 2011.

[4] M. Owens, "The Definitive Guide to SQLite",*USA: Apress*, pp341-362, 2006.

[5] Google Inc. "Web View — Android Developers", *Android Developer Resources*, 2011.