A Developer's First Look At Android

Look into the insides of Android from a developer's point of view, and you'll probably conclude that the only limitation could be your imagination.

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. Most of you may already be aware of Android, its capabilities and other aspects. (Please refer to the article, "*Linux May Put the Desktop In Your Pocket*" published in the December issue of *LFY*). In this article, we shall take a brief look at Android from the developer's point of view.

The architecture

The various components of Android are designed as a stack, with the 'Applications' forming the top layer of the stack, while the Linux kernel forms the lowest layer. "Interesting!" I hear you say, and it sure is.

Applications

Android ships with a set of core applications including an e-mail client, SMS program, calendar, maps, browser, contacts, and other features. All applications are written using the Java programming language. So, it is in this layer that you will fit in your next big mobile application.

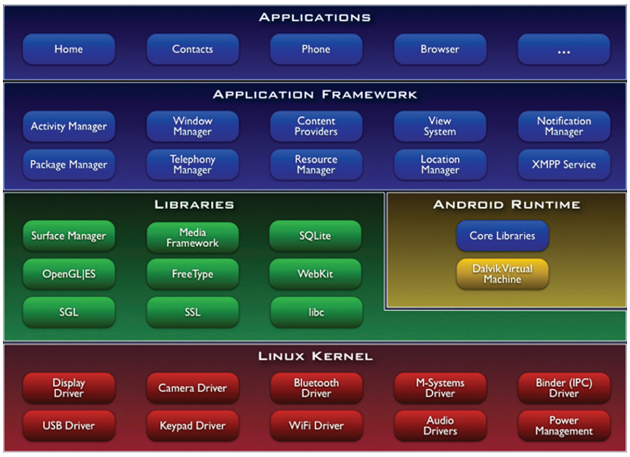
Application framework

Developers have full access to the same framework APIs used by the core applications. The application architecture is designed to simplify the reuse of components; the capabilities of any application can be published and then be made use of by any other application (subject to security constraints enforced by the framework). This same mechanism allows components to be replaced by the user.

For example, if you have a small note-taking application on your mobile and want to search for a particular location whose address you have just noted down, you might consider using the maps application directly from your note-taking application, instead of switching applications.

Libraries

Android includes a set of C/C++ libraries used by various components of the Android system. These capabilities are exposed to developers through the Android application framework.



(Image source: Official Google Android documentation)

Figure 1: The Android component stack

Android Emulator
3:32 PM
Android - Google Search
Web Images Maps News Products Gmail
Google
Android
Results 1 - 10 of about 3,080,000 for A
Android Platform
Sponsored Link
code.google.com/android Learn about the platform and get an early look at the Android SDK.
[*and... [amit... Gmail ... Terminal Andro...

Figure 2: The Android emulator

Android runtime

Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language.

Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently. The Dalvik VM executes files in the Dalvik Executable (*.dex*) format, which is optimised for minimal memory footprint. The VM is register-based, and runs classes compiled by a Java language compiler that has been transformed into the *.dex* format by the included *dx* tool. Interested in knowing more about the *.dex* file format? Please refer to http://retrodev.com/android/dexformat.html

The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

The Linux kernel

Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack.

The source code of the Linux kernel used for the Android stack is available at http://android.googlecode.com/files/linux-2.6.23-android-m3-rc20.tar.gz

The Android SDK

To jump start your Android development, download the Android SDK (for Linux) from http://code.google.com/android/download.html. It includes sample projects with source code, development tools, an emulator, and of course all the libraries you'll need to build an Android application.

Developing Android applications

You can develop Android applications with the same high-quality tools you use to develop Java applications. The Android core libraries provide the functionality needed to build some amazingly rich mobile applications, and the Android development tools make running, debugging and testing your applications a snap.

You can either use the Eclipse IDE (http://www.eclipse.org) or the command line tools available with the SDK to develop your Android app.

Android development with Eclipse

Life is made a lot simpler with the Eclipse plug-in for Android development, called the Android Development Tools (ADT). It basically brings the power of the IDE to Android development. Refer to http://code.google.com/android/adt_download.html for download and installation instructions.

Android development sans Eclipse

The Android SDK is bundled with all the tools to enable non-Eclipse users to write their Android application. In order to do so, navigate to the '*tools/*' folder in the extracted Android SDK folder. There is a Python script called *activityCreator.py.* You get the following terminal output when you run the script:

```
$ python activityCreator.py

Activity Creator Script

Usage:
  activityCreator [–out outdir] [–ide intellij] your.package.
name.ActivityName

Creates the structure of a minimal Android application.
The following will be created:
    - AndroidManifest.xml: The application manifest file.
    - build.xml: An Ant script to build/package the application.
    - res : The resource directory.
    - src : The source directory.
    - src/your/package/name/ActivityName.java the Activity java
class. packageName is a fully qualified java Package in the
format <package1>.<package2>... (with at least two components).
    - bin : The output folder for the build script.

Options:
 –out <folder>: specifies where to create the files/folders.
```

```
  —ide intellij: creates project files for IntelliJ

For Eclipse support, please use the ADT plugin.
```

Using the Android emulator

The emulator is essentially meant to test your Android applications. It resides under the *'tools/'* directory. For instructions on how you can test your just-developed Android application on the emulator, please refer to the Android SDK documentation. The emulator also ships with a plethora of applications like a browser, a cooler-than-traditional phonebook and a map application, among other features. So, even if you have not yet written an Android application, the emulator is a must checkout.

Competing for a slice of the pie!

The Android Developer Challenge announced along with the launch of the SDK, will award $10 million to developers who build great applications for Android. Check out *http://code.google.com/android/adc.html* for more details.

It is still a matter of speculation as to when Android will make it into mainstream mobile phones, but it is definitely on its way and deserves a serious look from the developers. The Android SDK documentation is extensive and is the one-stop resource for complete information. One of the very first cool applications written for Android is WamF (*http://blogoscoped.com/archive/2007-11-19-n27.html*), where the developer signs off: *"Android is an environment where the biggest limitation is your imagination."* This 'droid' is definitely cool! END

LINKS TO FURTHER INFORMATION

- Google Android (http://code.google.com/android)
- Google Android discuss (http://code.google.com/android/groups.html)
- The Android Developer Challenge (http://code.google.com/android/adc.html)
- The Open Handset Alliance (http://www.openhandsetalliance.com/)
- WamF (http://blogoscoped.com/archive/2007-11-19-n27.html)
- Google Calling, Inside Android, the gPhone SDK (http://www.onlamp.com/pub/a/onlamp/2007/11/12/google-calling-inside-the-gphone-sdk.html)

By: Amit Kumar Saha. *The author is a technical writer, a contributor to open source projects and a researcher. More on his works is available at http://amitsaha.in.googlepages.com*