

# Android ROM Developers' Manual

Aniruddha Adhikary

Published  
with GitBook



# Table of Contents

Introduction	0
Breaking the Surface	1
Shifting the Angle	1.1
Glossary	

---

# Boring Stuff

Android's open source nature made it the most popular mobile operating ever created. Anyone could just pick up the source code, port it to their own device and become a smartphone manufacturer. Android revolutionized the smartphone industry and established a common platform for devices.

Wait a minute. That means, someone working at your smartphone manufacturer actually ported Android to your device. If they can could do it, we can port and build Android for devices ourselves! We could build the **latest Android** without waiting for manufactures. Maybe we could **add new features** like gesture support or improved performance! What if we customized the interface to make devices completely **unique**?

If you read this book, you will be on the track for learning the bits and bytes of doing all these crazy stuff! You will (hopefully) learn about the kernel, HALs, drivers, configuration files, libraries and beyond.

## Prerequisites

To follow along this book, you should have the following things under your belt (or on your desk).

- **A computer** running a **Linux**-based operating system (preferably Ubuntu)
- **A smartphone** to experiment on
- **A backup phone** for staying connected with your family

And most importantly, I cannot emphasize this more. **PATIENCE**. Most of you would be tempted to skim along the book and jump right into building stuff. **Don't do it**. Calm down, be patient. Every single word written in the book is written for a purpose.

```
import human.mind.Patience
import adhikary.aniruddha.Book

Patience.activate()
Book.follow()
```

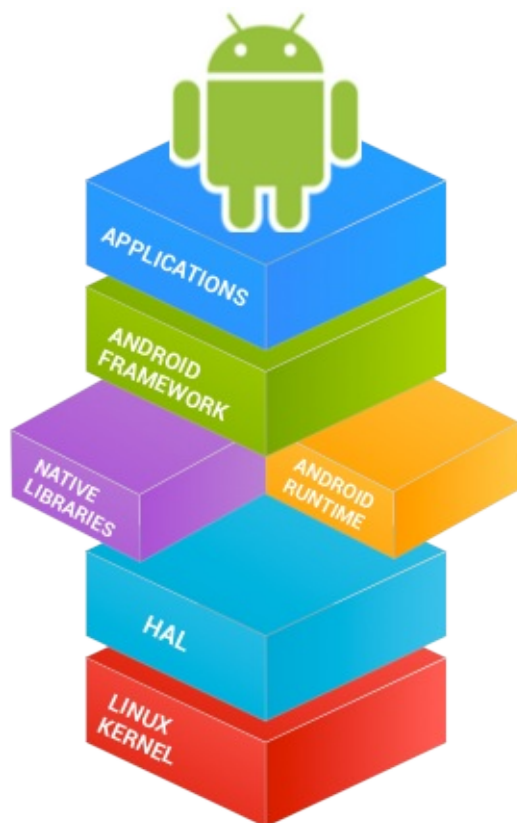
Wishing you a happy Android expedition. By the way, your donations would help to continue write this book, and more books like this one!

## About the author

Aniruddha Adhikary is a Firmware Engineer working at GOLDBERG Limited. He has contributed to CyanogenMod, Firefox OS, Ubuntu Touch and other open source projects. He has done extensive work on the MediaTek platform. He loves riding his bike and playing badminton.

# Breaking the Surface

I assume all readers know the basics of Android and how it works. Yet still let us have a review. The following blocks should give you an idea of the architecture of Android.



## Apps

Native Android apps are written in **Java** using the [Android SDK](#). Android apps carry the extension **".apk"**, although it's just a **zip** file! Which means you can **make changes** to any Android app including system apps like the Dialer or Gallery! (More about this later)

## Framework

Android apps run **on top of** Android frameworks. Frameworks provide the **foundation** required (e.g. APIs) for apps to run on.

## Runtime and Native Libraries

Java programs cannot be executed directly by the CPUs (in most cases). The [Java bytecode](#) is usually executed by a JVM (Java Virtual Machine). Android uses it's own JVM named **Dalvik ART (Android Runtime)**. The Framework and Apps run atop the Runtime.

The native libraries work in par with the runtime to provide extra functionality. For example, camera access, sensor access, media playback or media encoding.

## HAL

As the name suggests, the layer creates an **abstraction**. Since device vendors use different hardware, it would be a **nightmare** to maintain compatibility. Developers would have to change lots of libraries for porting Android. The abstractions ensure that only the HALs need to be changed from device to device. (Very long story, short)

HALs differ from manufacturer to manufacturer and from device to device.

## Kernel

The kernel is the **lowest level** Android component with the **highest priority**. It talks directly to the hardware, scales CPU cores up and down, manages memory. It does everything to keep your Android device up and running. The kernel makes sure you don't burn your CPU or apps don't overflow data. It is the transatlantic tunnel from your hardware to your system. Android uses **Linux** as it's Kernel. Understandably, kernels are very device specific!

Without the kernel, your device won't work!

## Bootloaders

Strictly speaking, bootloaders are not considered a part of Android. Yet the bootloader **starts** the kernel correctly, so that the system can boot successfully. It usually **maps partitions** way before the kernel even finds out whats going on. The kernel often receives various boot parameters or **configuration** from it.

Different vendors use different bootloaders. For example, MediaTek uses U-BOOT.

Now that we have revised our understanding of these terms, we are good to go ahead! If you are still confused, do ask Uncle Google.

# Shifting the Angle

# Glossary

## Android SDK

The Android SDK (Software Development Kit) is a suite of tools required to compile Android apps.

[1. Breaking the Surface](#)

## Java bytecode

The "compiled" form of java that can run on any JVM.

[1. Breaking the Surface](#)

## SELinux

A security enhancement module of Linux, created by the NSA.