

Application Development for Mobile Devices

MOB3000R

If you don't know me ..yet!

- Ph.D. Software Engineering- NTNU, Trondheim
- M.Sc. Engineering & Management of Information Systems- KTH, Stockholm
- B.Sc. Computer Science- University of Dhaka, Dhaka

Contact

salah.ahmed@usn.no

Room: A318

Topics in this Lecture

1. Assignment (Canvas)
2. Final Project Assignment (Canvas)
3. Resources

A. Android Platform

Old – Deprecated!

Android platform – SDK, tools, libraries, versions of Android Studio, Gradle, plugin, development frameworks, concepts all changes very fast.

So many numbers, versions in the lecture may be old or deprecated.

And there is no meaning to change them for 3 months!

DON'T COMPLAIN of Being Deprecated!

Accept it...

That's life in Android World !

Always follow google developer's website to get the updated view of the current developments.

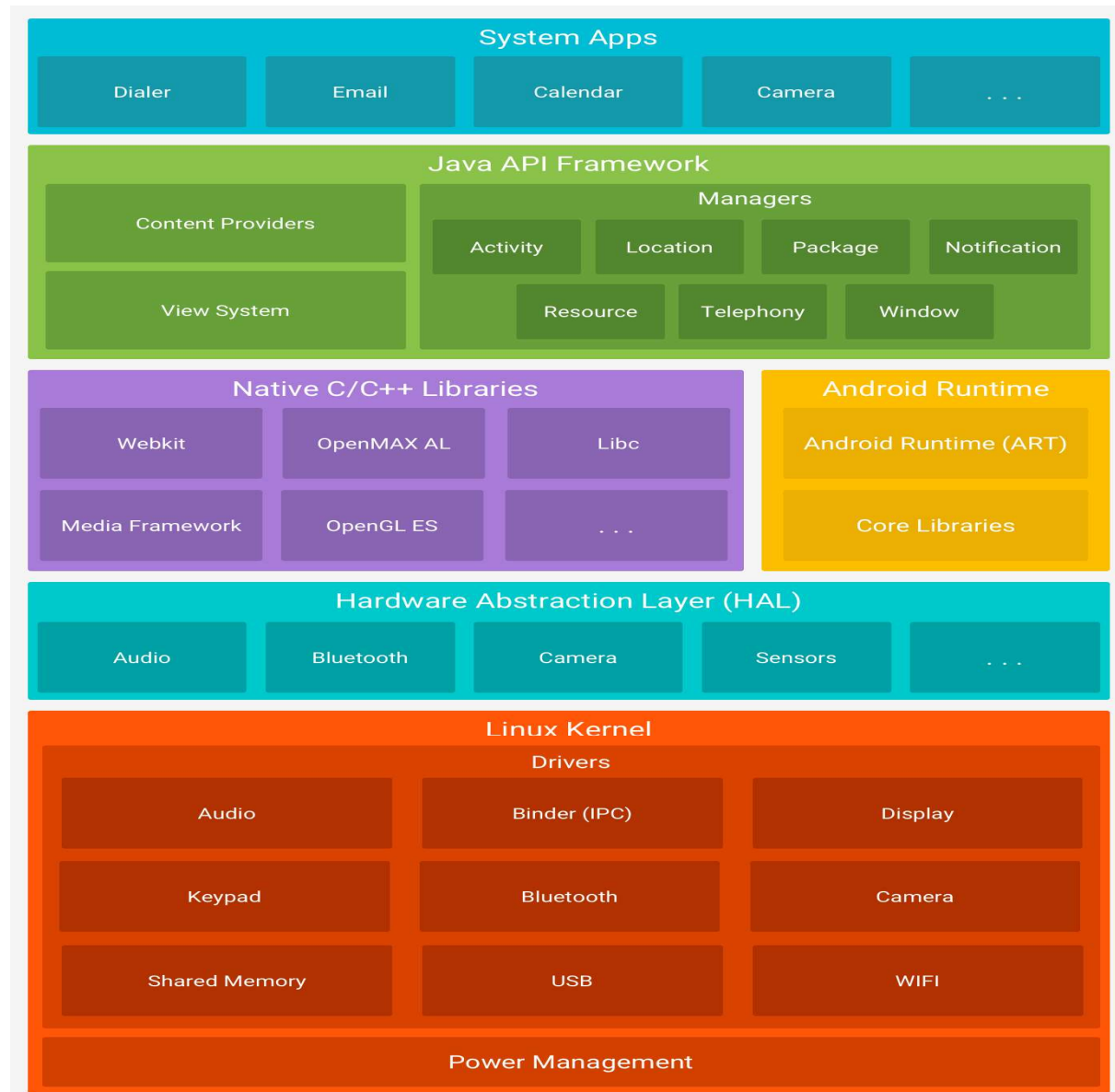
Deprecated!

Android Platform/Architecture

- Is a software stack
- Has several layers
- SDK- Software Development Kit
- Tons of documentations, blogs, tutorials

* <https://developer.android.com/guide/platform>

Software Stack

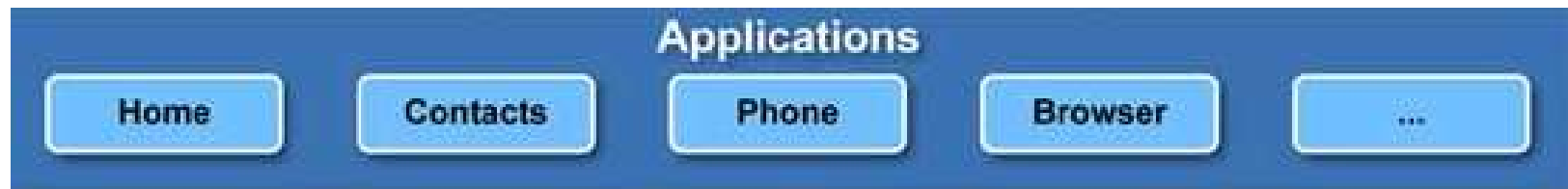
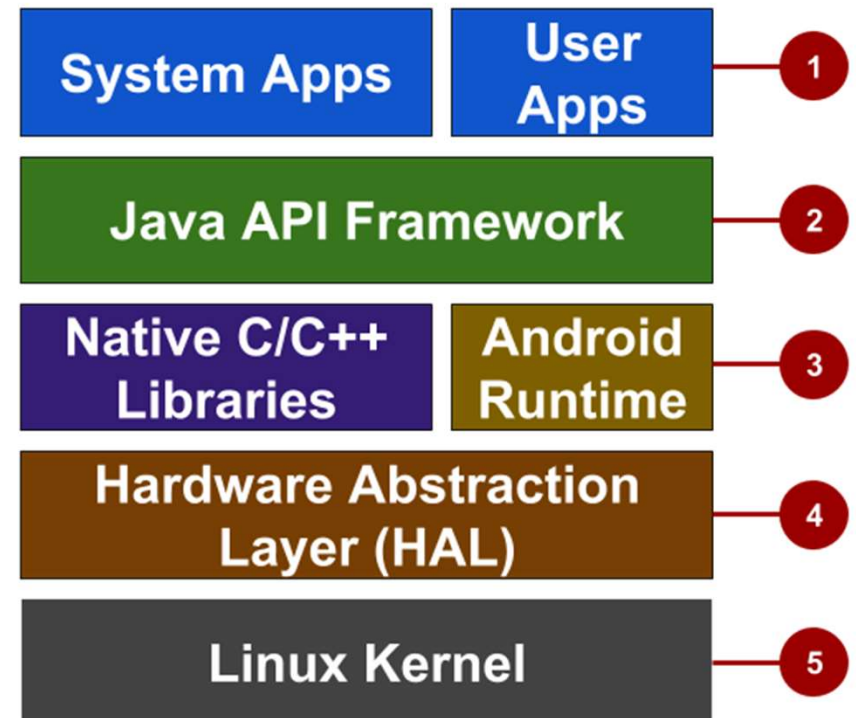


<https://developer.android.com/guide/platform>

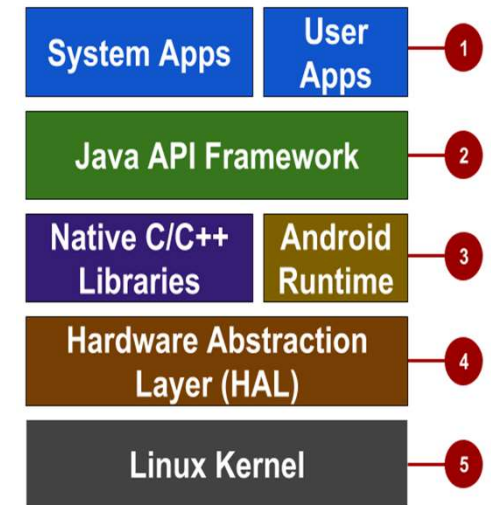
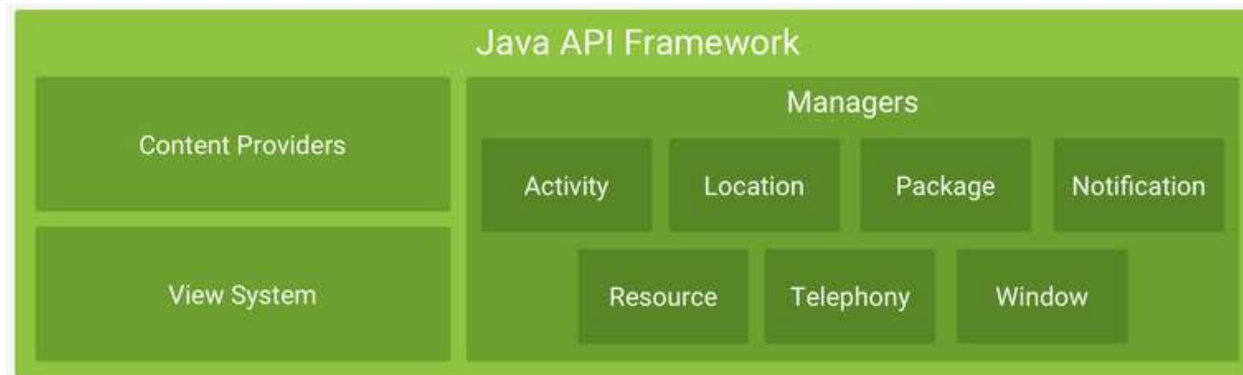
Application Layer

Apps

- System Apps + User Apps
- Examples: Contacts, Phone, Angry Birds
- Applications are not hard coded they can be changed



API Framework



These APIs form the building blocks you need to create Android apps by simplifying the reuse of core, modular system components and services

Contains reusable software that applications need –

View System – common graphical elements, Buttons, Labels, Icons

Package Manager – a database that keeps track of all the application installed on your device

Window Manager – manages the many windows that comprises an app

Resource Manager – Non Compiled Resources like- images, audio etc.

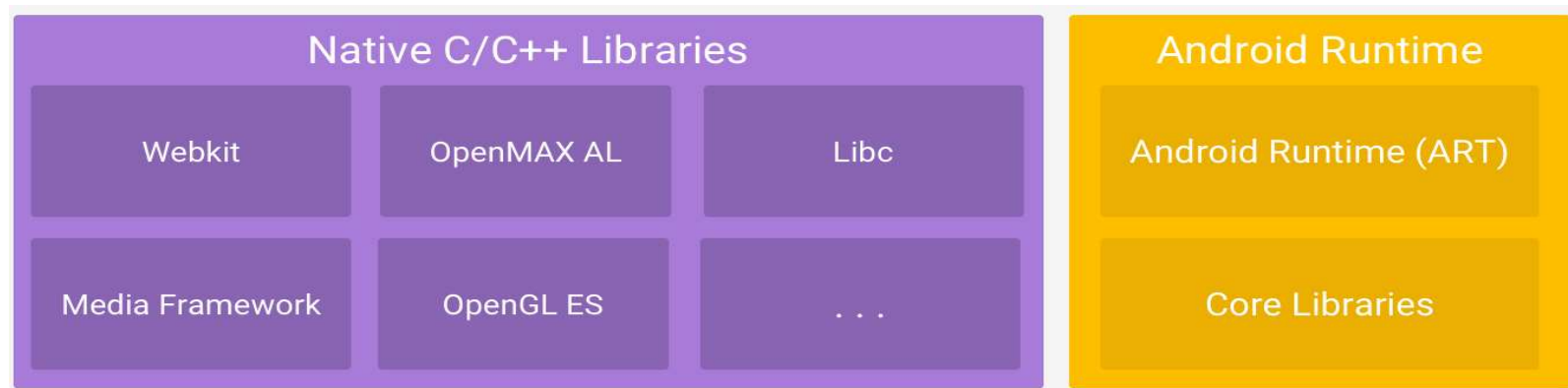
API Framework

Activity Manager

- On high level –activity corresponds to Single user interface
- Application – consists several user interfaces and navigating between them.
- Activity manager helps to coordinate and support navigations between User interfaces.

System Libraries (Native C/C++)

- Native Libraries Written in C or C++
- SQLite, SSL, OpenGL



System Libraries (Native C/C++)

- Handles lot of core performance sensitive activities
- Quickly rendering webpages, updating display
- Standard OS system calls
- Process and thread creation
- Playing back audio video files

Android Runtime

Two components

- Core Libraries
- ART Android Runtime



*Prior to API level 21 (v 5.0)

- Core Java Libraries
- Dalvik Machine



If your app runs well on ART, then it should work on Dalvik as well, but the reverse may not be true.

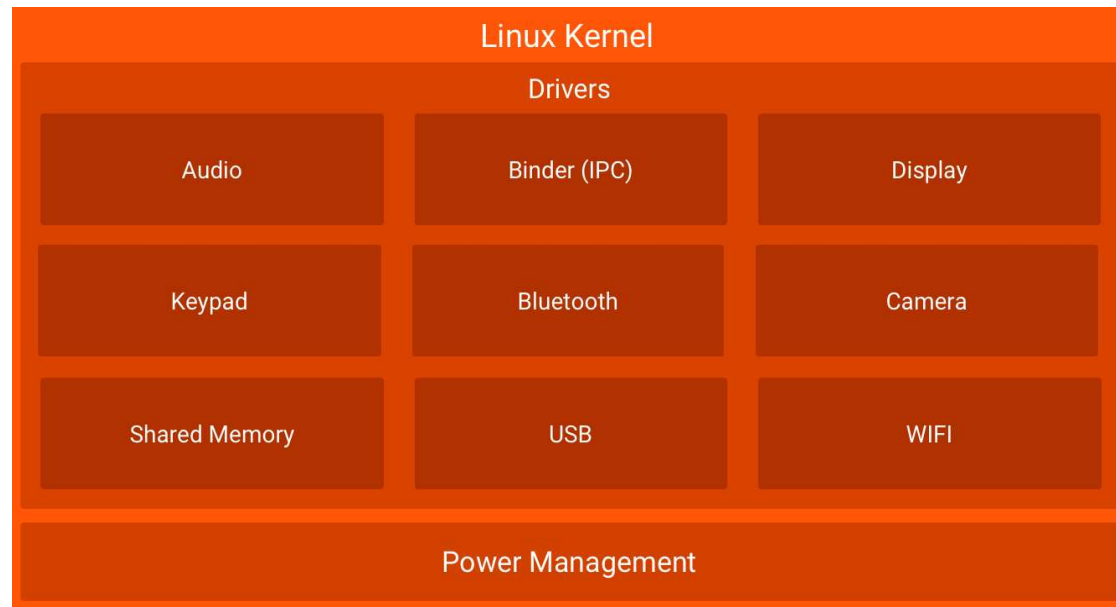
Linux Kernel Layer

- Provides Core Services for the device
- Provides generic O.S. services
- Permissions architecture
- Memory and process management
- Security

Android Specific

- Power Management
- Low Memory Killer
- Inter-Process Communication

Drivers for: Display, WIFI, Audio etc.
Power Management



Why not JVM

Because Android has specific demands

- Designed for resource-constrained environments
- Slower CPU (mobile vs PC)
- Less Memory ”
- Limited Battery Life

DEVELOPMENT ENVIRONMENT

DEVELOPMENT ENVIRONMENT

Android Studio Bundle

- Android Platform
- Android Studio/ IDE
- Development tools
- System Image for Emulator

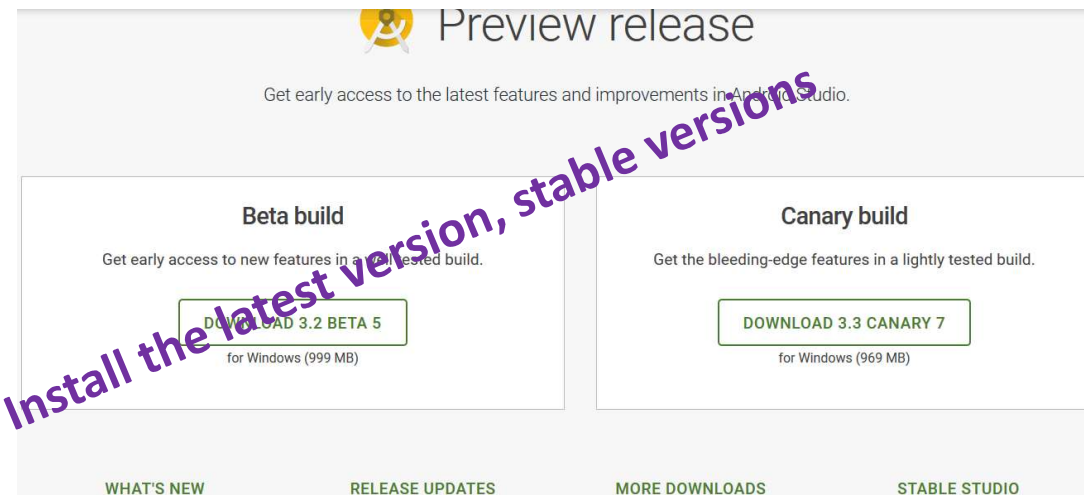
Android Studio

Beta / Canary / Stable ??

<https://developer.android.com/studio/preview/>

-- Canary build- latest features

Stable Versions



Current stable: ??

3.1 (March 2018)
3.0 (October 2017)
2.3 (March 2017)
2.2 (September 2016)
2.1 (April 2016)
2.0 (April 2016)

Stable Version

<https://developer.android.com/studio>

Android Studio

New version numbering

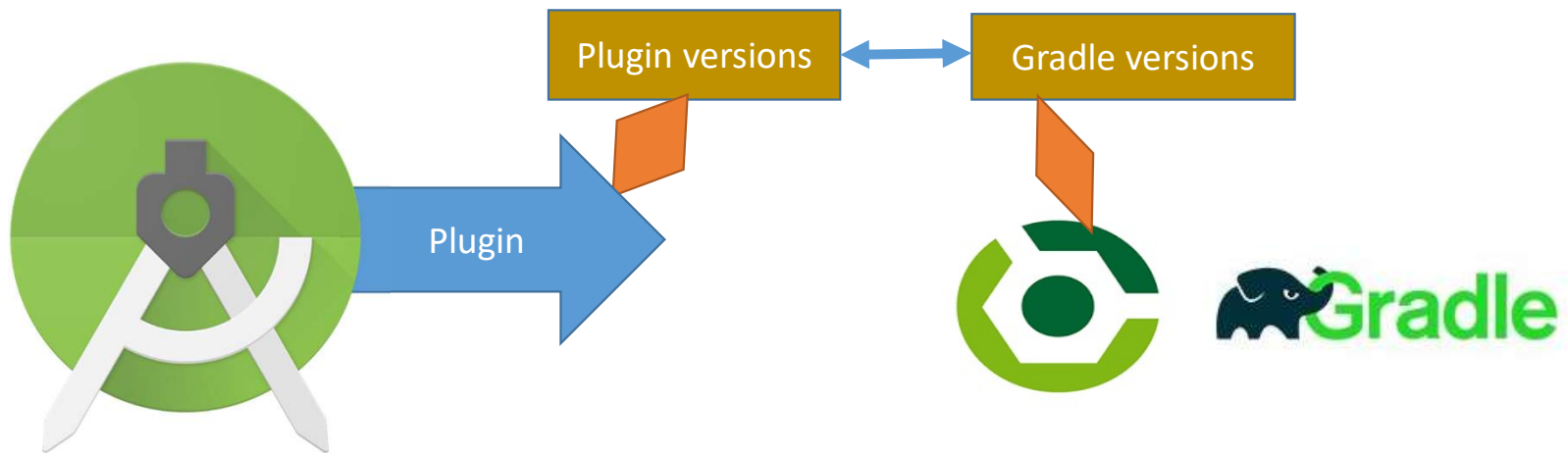
IntelliJ Version	Old Name	Old - Number System	New - Year System	New Version Name
2020.3	4.3	4.3.0	2020.3.1	Arctic Fox 2020.3.1

Install a Version of Android Studio that supports Java
Update plugin and Gradle
Install SDK Platform Version

- Choose a stable version
 - <https://developer.android.com/studio/index.html>
- Read the release notes
 - <https://developer.android.com/studio/releases/>
- See all the versions here
 - <https://developer.android.com/studio/archive>

Update Plugin version and Gradle version

Gradle and the Android plugin run independent of Android Studio.



Gradle is an open-source build automation tool focused on flexibility and performance. Gradle build scripts are written using a [Groovy](#) or [Kotlin](#) DSL.



Android Studio

Plugin version



Gradle

Plugin version	Required Gradle
1.0.0 - 1.1.3	2.2.1 - 2.3
1.2.0 - 1.3.1	2.2.1 - 2.9
1.5.0	2.2.1 - 2.13
2.0.0 - 2.1.2	2.10 - 2.13
2.1.3 - 2.2.3	2.14.1 - 3.5
2.3.0+	3.3+
3.0.0+	4.1+
3.1.0+	4.4+
3.2.0 - 3.2.1	4.6+
3.3.0 - 3.3.3	4.10.1+
3.4.0 - 3.4.3	5.1.1+
3.5.0 - 3.5.4	5.4.1+
3.6.0 - 3.6.4	5.6.4+
4.0.0+	6.1.1+
4.1.0+	6.5+
4.2.0+	6.7.1+

Distribution dashboard

- This information may help you prioritize efforts for supporting different devices by revealing which devices are active in the Android and Google Play ecosystem.
- Check!
- <https://developer.android.com/about/dashboards/index.html>
- See the distribution of different screen/resolution size.

Android codename, version number, API level

Code name	Version number	Initial release date	API level	Security patches ^[1]
(No codename) ^[2]	1.0	September 23, 2008	1	Unsupported
(Internally known as "Petit Four") ^[2]	1.1	February 9, 2009	2	Unsupported
Cupcake	1.5	April 27, 2009	3	Unsupported
Donut ^[3]	1.6	September 15, 2009	4	Unsupported
Eclair ^[4]	2.0 – 2.1	October 26, 2009	5 – 7	Unsupported
Froyo ^[5]	2.2 – 2.2.3	May 20, 2010	8	Unsupported
Gingerbread ^[6]	2.3 – 2.3.7	December 6, 2010	9 – 10	Unsupported
Honeycomb ^[7]	3.0 – 3.2.6	February 22, 2011	11 – 13	Unsupported

*wikipedia

**Check the latest versions
on WIKI**

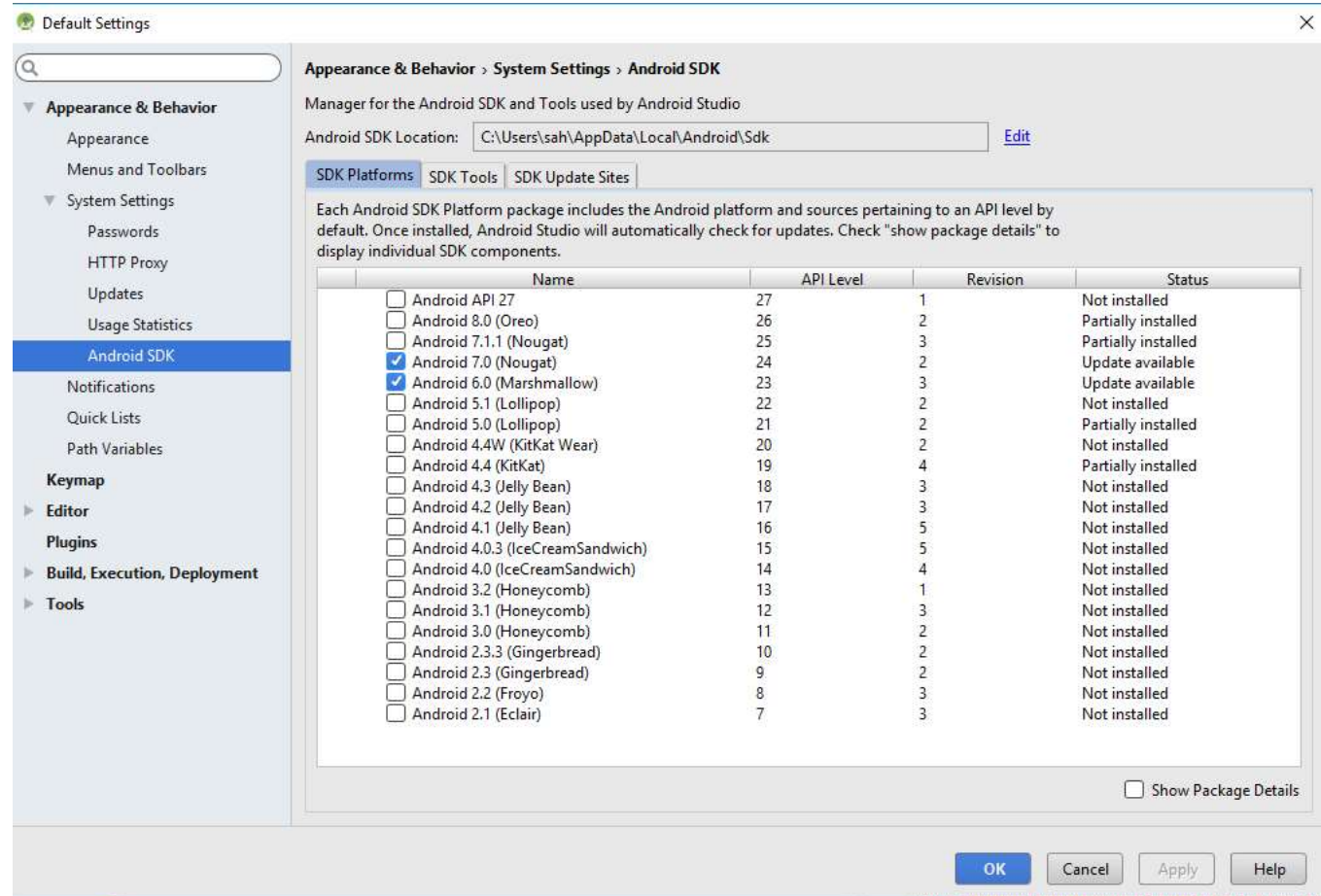
Ice Cream Sandwich ^[8]	4.0	Android Oreo	Oatmeal Cookie	8.0	26	August 21, 2017	January 2021
Jelly Bean ^[9]	4.1				8.1	27	December 5, 2017
KitKat ^[10]	4.4	Android Pie	Pistachio Ice Cream ^[20]	9	28	August 6, 2018	January 2022
Lollipop ^[12]	5.0	Android 10	Quince Tart ^[21]	10	29	September 3, 2019	August 2022
Marshmallow ^[13]	6.0	Android 11	Red Velvet Cake ^[21]	11	30	September 8, 2020	
Nougat ^[14]	7.0	Android 12	Snow Cone	12	31	October 4, 2021	
Oreo ^[15]	8.0	Android 12L	Snow Cone v2	12.1 ^[a]	32	March 7, 2022	
Legend: Old version Older		Android 13	Tiramisu ^[23]	13 ^[b]	33	Q3 2022	
Legend: Old version Older version, still maintained Latest version Latest preview version							

Which android version to use?

- Android Studio - versions?
 - Android Gradle plugin- versions?
 - Gradle- versions?
 - SDK platform version?
-
- GDT- Developers Training
 - Android Developers Fundamentals- Which versions they used?

Install SDKs

See which versions of SDKs are installed



Running Application

- Emulator
- Android Device

Pros and Cons of Using Emulator

Can you name some?

Pros and Cons

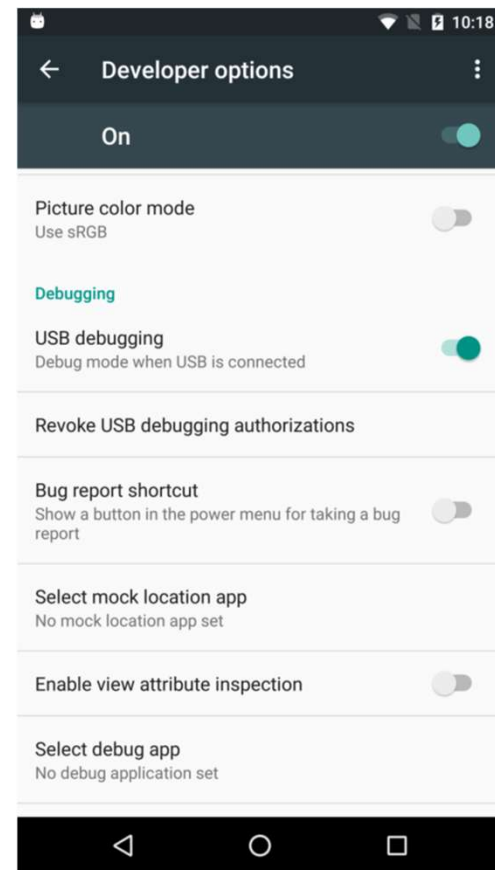
- Pros
 - No need of a device
 - Can
 - Configure hardware easily
 - Modifications is not destructive, does not destroy phone data
- Cons
 - Not the real device
 - You can't get all the features

Make Your Device Ready

The first step to make your device ready for use with development is to go into the Settings application on the device. What happens now depends a bit on your Android version:

- On Android 1.x/2.x, go into Applications, then into Development
- On Android 3.0 through 4.1, go into “Developer options” from the main Settings screen
- On Android 4.2 and higher, go into About, tap on the build number **seven times**, then press BACK, and go into “Developer options” (which was formerly hidden)

Ref: Mark Murphy, Busy Coders Guide



Project Structure

Learn about build configuration file

<https://developer.android.com/studio/build/index.html>

❑ Gradle File

- Top Level Build File
- The Module-level Build File

Top-level build file where you can add configuration options common to all sub-projects/modules.

All modules have a specific gradle file. Whatever is included in this gradle file, it will only affect the module that is included on

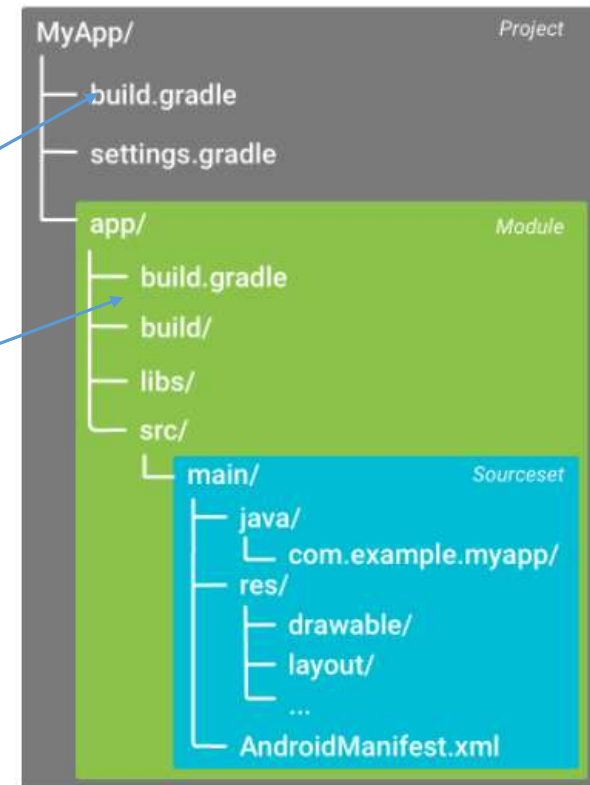


Figure The default project structure for an Android app module.

```
buildscript {  
  
    repositories {  
        google()  
        jcenter()  
    }  
    dependencies {  
        classpath 'com.android.tools.build:gradle:3.0.1'  
  
        // NOTE: Do not place your application dependencies here; they belong  
        // in the individual module build.gradle files  
    }  
}  
  
allprojects {  
    repositories {  
        google()  
        jcenter()  
    }  
}  
  
task clean(type: Delete) {  
    delete rootProject.buildDir  
}
```

For gradle itself

For all projects

Build Process

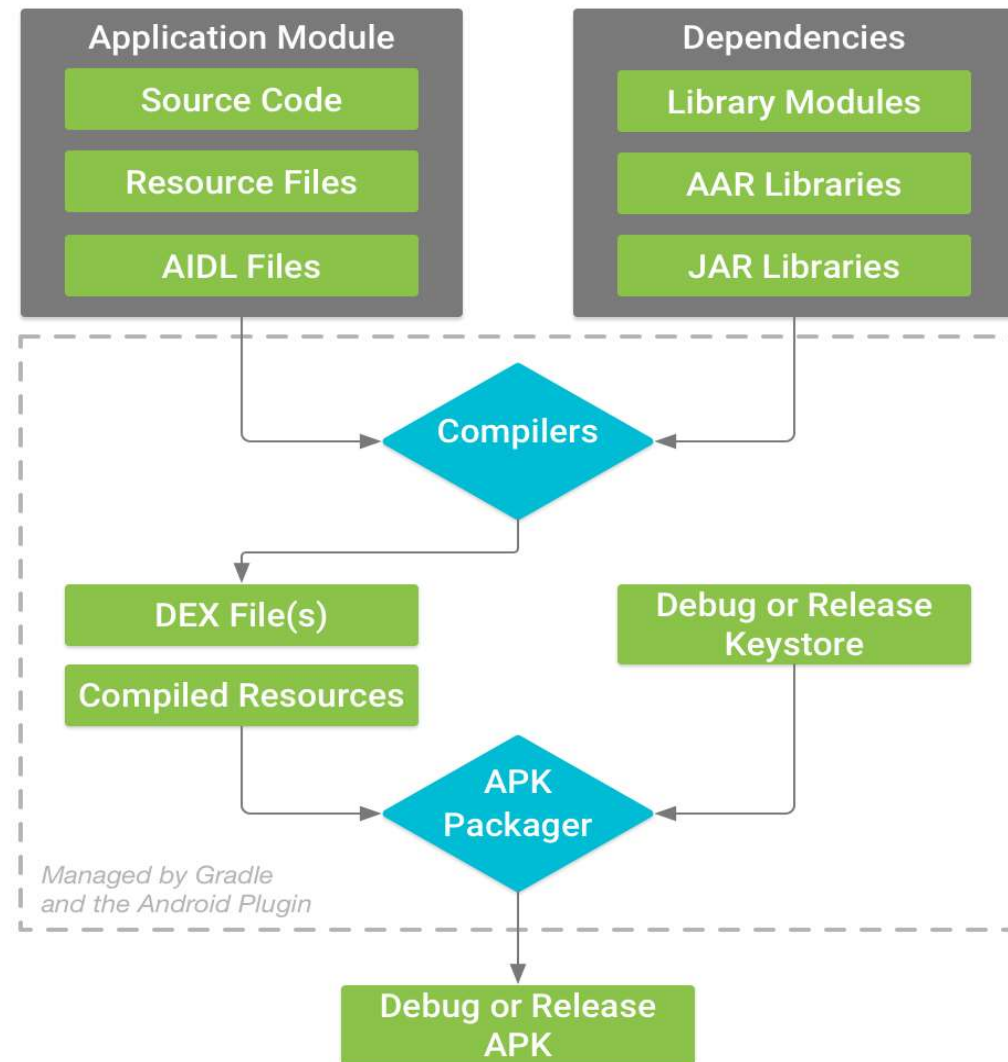


Figure 1. The build process of a typical Android app module

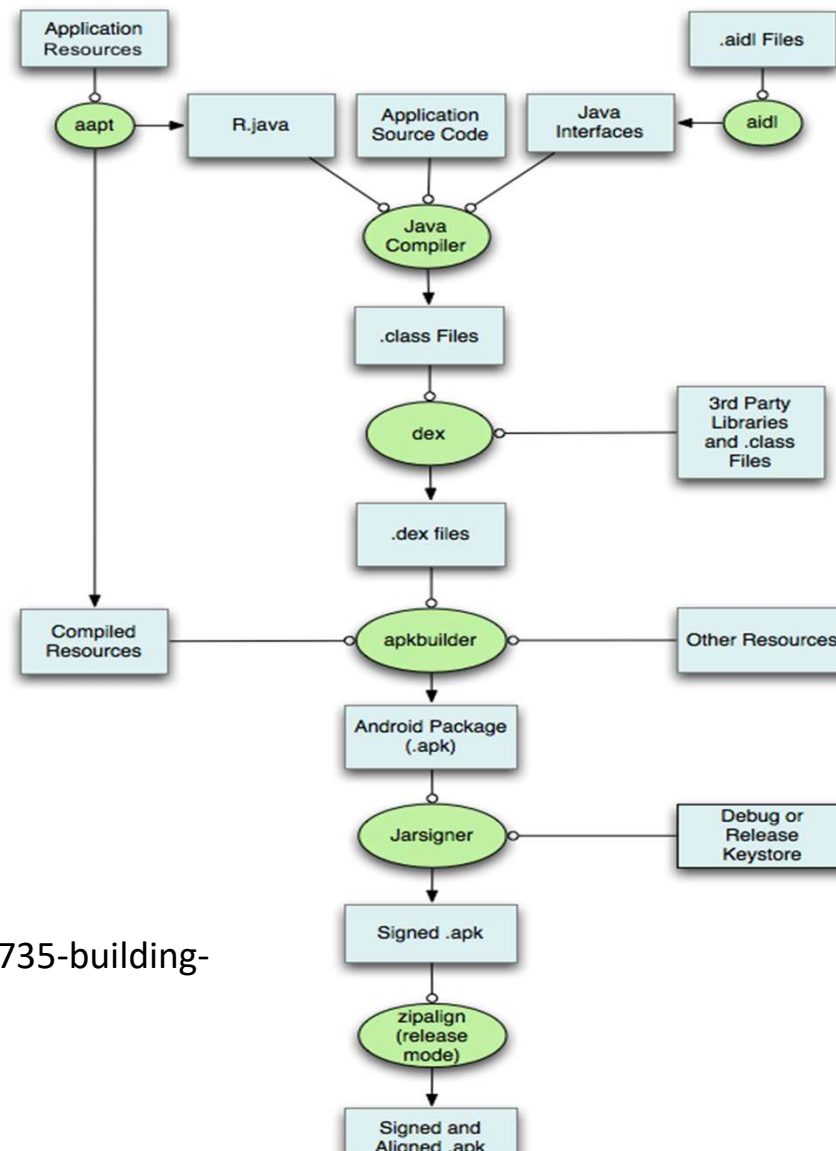
AAR = Android Archive Library

AIDL = [Android Interface Definition Language](#)

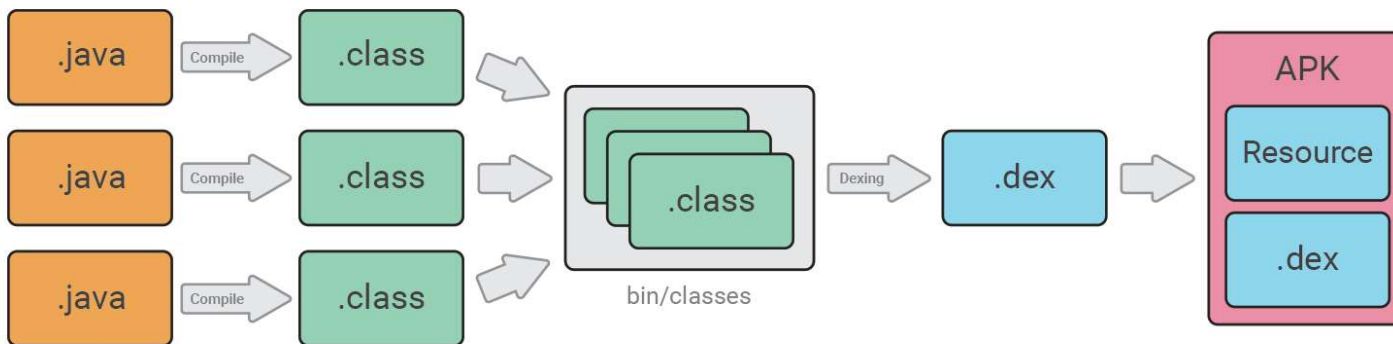
Build Process

- A good blog article to understand the gradle build process

<http://michael-huang.logdown.com/posts/293735-building-android-with-gradle>



.dex and APK



Application ID

Every Android app has a unique application ID that looks like a Java package name, such as *com.example.myapp*.

This ID uniquely identifies your app on the device and in Google Play Store. If you want to upload a new version of your app, the application ID (and the [certificate you sign it with](#)) must be the same as the original APK—if you change the application ID, Google Play Store treats the APK as a completely different app. So once you publish your app, **you should never change the application ID.**

Manifest File

Every Android app must include a file called `AndroidManifest.xml` at its root. The manifest file contains essential information about the app, such as what components it contains, required libraries, and other declarations

Resources for learning

1. Best Resource and most updated and accurate is to consult

<http://developer.android.com/>

But some parts of it can be hard to understand for some beginners

So try also other resources like blogs, YouTube, and websites

2. Tutorials Point

<http://www.tutorialspoint.com/android/index.htm>

3. Finally, the textbook HeadFirst is good for beginners
written with an approach of developing an app

Tasks to do

1. Form a group – with maximum allowed members