Unit 1

Overview of Android

What is Android?

- Android is an open-source operating system based on Linux with a Java programming interface for mobile devices such as Smartphone (Touch Screen Devices who supports Android OS) as well for Tablets too.
- Android was developed by the Open Handset Alliance (OHA), which is led by Google. The Open Handset Alliance (OHA) is a consortium of multiple companies like Samsung, Sony, Intel and many more to provide services and deploy handsets using the android platform.
- In 2007, Google released a first beta version of the Android Software Development Kit (SDK) and the first commercial version of Android 1.0 (with name Alpha), was released in September 2008.
- In 2012, Google released another version of android, 4.1 Jelly Bean. It's an incremental update and it improved a lot in terms of the user interface, functionality, and performance.
- In 2014, Google announced another Latest Version, 5.0 Lollipop. In Lollipop version Google completely revamped the UI by using Material Designs, which is good for the User Interface as well for the themes related.
- All the source code for Android is available free on Git-Hub, Stack overflow, and many more websites. Google publishes most of the code under the Apache License version 2.0.

Android Features

This is a powerful open-source operating system which provides a lot of great features, those are

- It is open-source and we can customize the OS based on our requirements.
- It supports connectivity for GSM, CDMA, WIFI, NFC, Bluetooth, etc. for telephony or data transfer. It will allow us to make or receive a calls / SMS messages and we can send or retrieve data across mobile networks
- By using WIFI technology we can pair with other devices using apps
- Android has multiple APIs to support location-based services such as GPS
- We can perform all data storage related activities by using lightweight database SQLite.
- It has a wide range of media supports like AVI, MKV, FLV, MPEG4, etc. to play or record a variety of audio/video and having a different image format like JPEG, PNG, GIF, BMP, MP3, etc.
- It has extensive support for multimedia hardware control to perform playback or recording using camera and microphone
- It has an integrated open-source WebKit layout-based web browser to support HTML5, CSS3
- It supports a multi-tasking; we can move from one task window to another and multiple applications can run simultaneously
- It will give a chance to reuse the application components and the replacement of native applications.
- We can access the hardware components like Camera, GPS, and Accelerometer
- It has support for 2D/3D Graphics

Feature	Description
Beautiful UI	Android OS basic screen provides a beautiful and intuitive user interface.

Feature	Description
Connectivity	GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.
Storage	SQLite, a lightweight relational database, is used for data storage purposes
Media support	H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, and BMP
Messaging	SMS and MMS
Web browser	Based on the open-source WebKit layout engine, coupled with Chrome's V8 JavaScript engine supporting HTML5 and CSS3.
Multi-touch	Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero.
Multi-tasking	User can jump from one task to another and same time various application can run simultaneously.
Resizable widgets	Widgets are resizable, so users can expand them to show more content or shrink them to save space
Multi-Language	Support single direction and bi-directional text.
GCM	Google Cloud Messaging (GCM) is a service that let developers send short message data to their users on Android devices, without needing a proprietary sync solution.
Wi-Fi Direct	A technology that let apps discover and pair directly, over a high-bandwidth peer-to-peer connection.
Android Beam	A popular NFC-based technology that let

Android History

Initially, Google launched the first version of Android platform on Nov 5, 2007, from that onwards Google released a lot of android versions under a codename based on desserts, such as Apple Pie, Banana Bread, Cupcake, Donut, Éclair, Froyo, Gingerbread, Jellybeans, KitKat, Lollipop, marshmallow, nougat etc. and made a lot of changes and additions to the android platform.

The following table lists the different version details of android which is released by Google from 2007 to till date.

Release Date	Version	API Level	Version Name
September 23, 2008	Android 1.0	1	Apple Pie
February 9, 2009	Android 1.1	2	Banana Bread
April 30, 2009	Android 1.5	3	Cupcake
September 15, 2009	Android 1.6	4	Donut
October 26, 2009	Android 2.0	5	Eclair
December 3, 2009	Android 2.0.1	6	

Release Date	Version	API Level	Version Name
January 12, 2009	Android 2.1	7	
May 20, 2010	Android 2.2	8	Froyo
January 18, 2011	Android 2.2.1	8	
January 22, 2011	Android 2.2.2	8	
November 21, 2011	Android 2.2.3	8	
December 6, 2010	Android 2.3	9	Gingerbread
February 9, 2011	Android 2.3.1	9	
July 25, 2011	Android 2.3.3	10	
September 2, 2011	Android 2.3.4	10	
February 22, 2011	Android 3.0.x	11	Honeycomb
May 10, 2011	Android 3.1.x	12	
July 15, 2011	Android 3.2.x	13	
October 18,2011	Android 4.0	14	Ice Cream Sandwich
October 19, 2011	Android 4.0.1	14	
November 28, 2011	Android 4.0.2	14	
December 16, 2011	Android 4.0.3	15	
February 4, 2012	Android 4.0.4	15	
July 9, 2012	Android 4.1	16	Jelly Bean
July 23, 2012	Android 4.1.1	16	
October 9, 2012	Android 4.1.2	16	
November 13, 2012	Android 4.2	17	
November 27, 2012	Android 4.2.1	17	
February 11, 2013	Android 4.2.2	17	
July 24, 2013	Android 4.3	18	
October 31, 2013	Android 4.4	19	Kitkat
June 23, 2014	Android 4.4.1, 4.4.2, 4.4.3, 4.4.4	19	
October 17, 2014	Android 5.0	21	Lollipop
March 09, 2015	Android 5.1	22	

Release Date	Version	API Level	Version Name
October 5, 2015	Android 6.0	23	Marshmallow
December 7, 2015	Android 6.0.1	23	
August 22, 2016	Android 7.0	24	Nougat
October 4, 2016	Android 7.1	25	
	Android 8.0	26	O

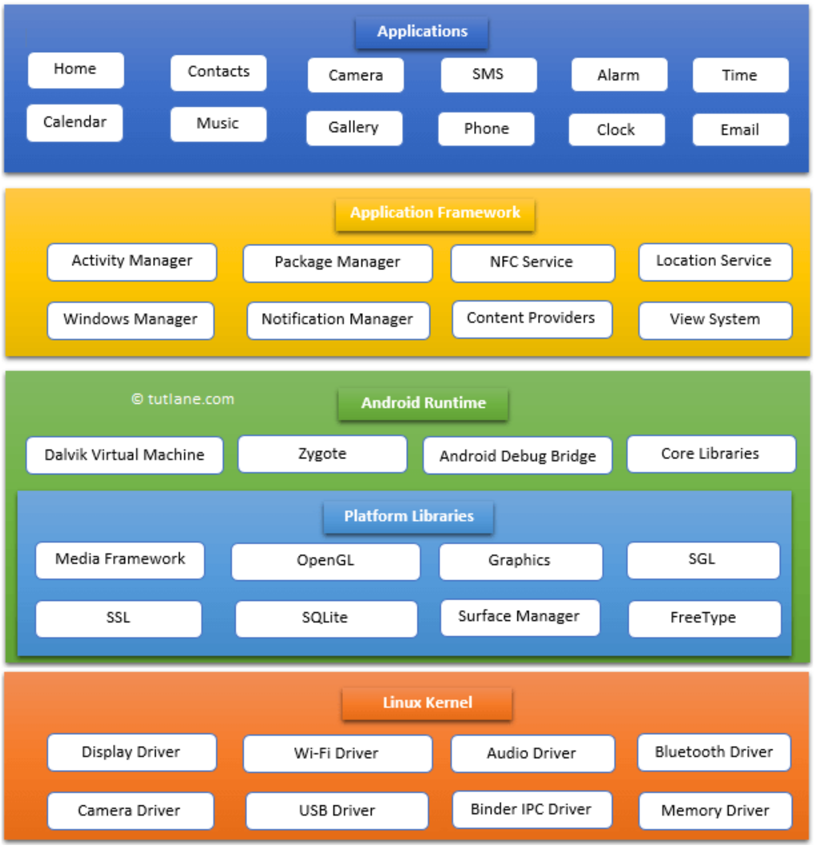
Android Architecture

Android architecture is a software stack of components to support mobile device needs. Android software stack contains a Linux Kernel, collection of c/c++ libraries which are exposed through an application framework services, runtime, and application.

Following are main components of android architecture those are

1. Applications
2. Android Framework
3. Android Runtime
4. Platform Libraries
5. Linux Kernel

In these components, the **Linux Kernel** is the main component in android to provide its operating system functions to mobile and **Dalvik Virtual Machine** (**DVM**) which is responsible for running a mobile application.



Applications

The top layer of the android architecture is Applications. The native and third-party applications like contacts, email, music, gallery, clock, games, etc. whatever we will build those will be installed on this layer only.

The application layer runs within the Android run time using the classes and services made available from the application framework.

Application Framework

The Application Framework provides the classes used to create Android applications. It also provides a generic abstraction for hardware access and manages the user interface and application resources. It basically provides the services through which we can create a particular class and make that class helpful for the Application creation.

The application framework includes services like telephony service, location services, notification manager, NFC service, view system, etc. which we can use for application development as per our requirements.

Android Runtime

Android Runtime environment is an important part of Android rather than an internal part and it contains components like core libraries and the Dalvik virtual machine. The Android run time is the engine that powers our applications along with the libraries and it forms the basis for the application framework.

Dalvik Virtual Machine (**DVM**) is a register-based virtual machine-like Java Virtual Machine (JVM). It is specially designed and optimized for android to ensure that a device can run multiple instances efficiently. It relies on the Linux kernel for threading and low-level memory management.

The **core libraries** in android runtime will enable us to implement android applications using standard JAVA programming language.

Platform Libraries

The Platform Libraries includes various C/C++ core libraries and Java-based libraries such as SSL, libc, Graphics, SQLite, Webkit, Media, Surface Manger, OpenGL, etc. to provide support for Android development.

The following are the summary details of some core android libraries available for android development.

- Media library for playing and recording audio and video formats
- The Surface manager library to provide a display management
- SGL and OpenGL Graphics libraries for 2D and 3D graphics
- SQLite is for database support and FreeType for font support
- Web-Kit for web browser support and SSL for Internet security.

Linux Kernel

Linux Kernel is a bottom layer and heart of the android architecture. It manages all the drivers such as display drivers, camera drivers, Bluetooth drivers, audio drivers, memory drivers, etc. which are mainly required for the android device during the runtime.

The Linux Kernel will provide an abstraction layer between the device hardware and the remainder of the stack. It is responsible for memory management, power management, device management, resource access, etc.

Android Studio Installation for Development Environment

To set up an Android development environment in a system we need to install the following components.

1. Eclipse IDE
2. Eclipse Plugin
3. Android SDK

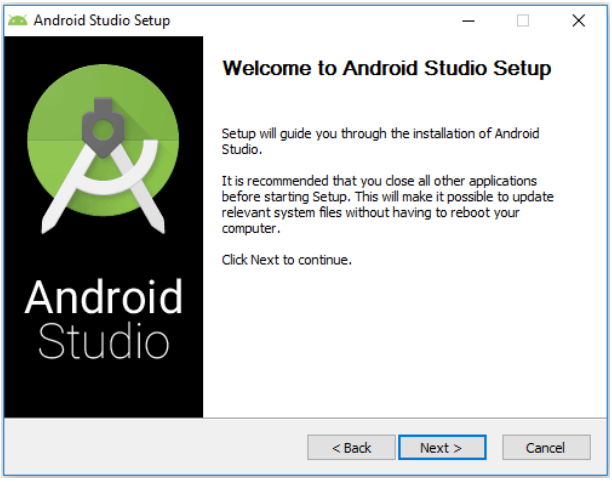
To make the android development environment setup process simple Google introduced a new android IDE called Android Studio. The Android Studio will contain all the required components like Eclipse IDE, Eclipse Plugin and Android SDK so we do not need to download the components separately.

Android Studio is the official IDE for android development and it is based on IntelliJ IDEA software. It is available for Windows, MAC, and LINUX operating systems.

We can download the latest version of Android Studio from the following URL.

Download Android Studio

Here, I am going to explain how to install android studio on a windows machine which is having windows 10 operating system.

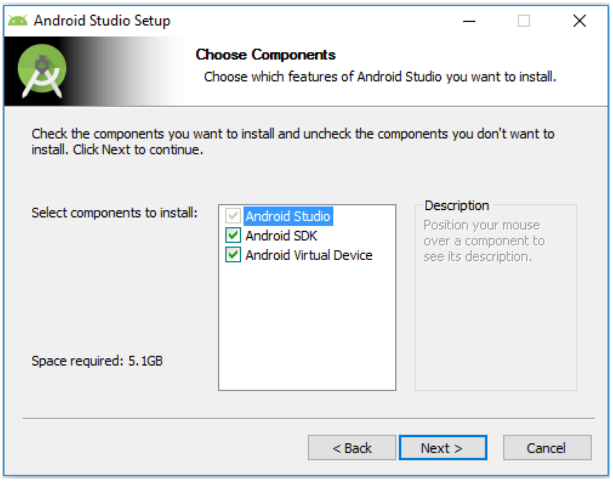


Welcome to Android Studio Setup

Setup will guide you through the installation of Android Studio.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.



Choose Components

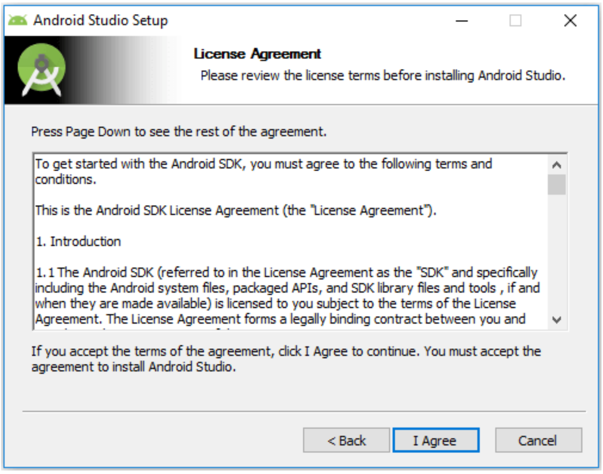
Choose which features of Android Studio you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:
- ☑ Android Studio
- ☑ Android SDK
- ☑ Android Virtual Device

Description
Position your mouse over a component to see its description.

Space required: 5.1GB



License Agreement

Please review the license terms before installing Android Studio.

Press Page Down to see the rest of the agreement.

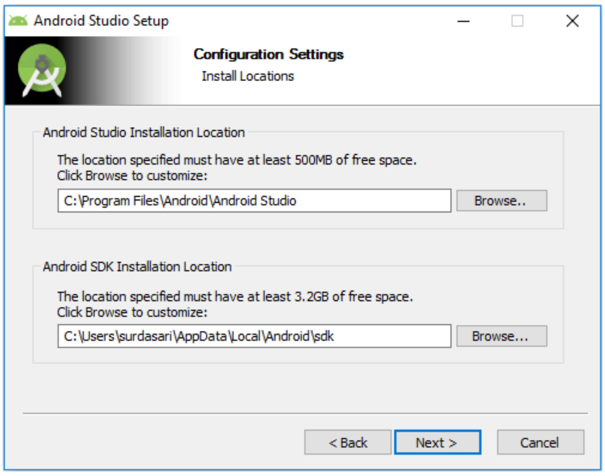
To get started with the Android SDK, you must agree to the following terms and conditions.

This is the Android SDK License Agreement (the "License Agreement").

1. Introduction

1.1 The Android SDK (referred to in the License Agreement as the "SDK" and specifically including the Android system files, packaged APIs, and SDK library files and tools , if and when they are made available) is licensed to you subject to the terms of the License Agreement. The License Agreement forms a legally binding contract between you and

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Android Studio.



Configuration Settings

Install Locations

Android Studio Installation Location

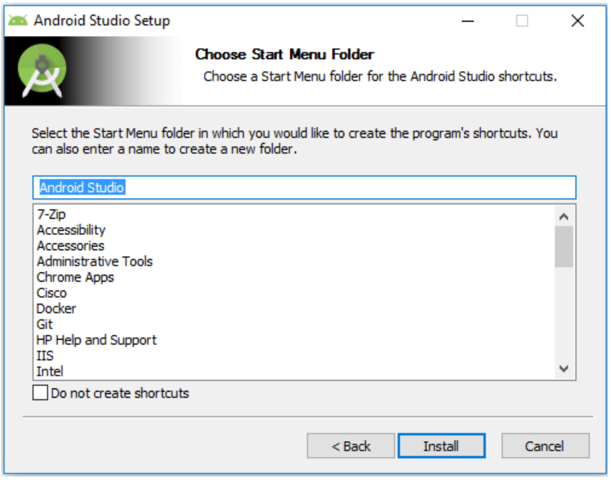
The location specified must have at least 500MB of free space.
Click Browse to customize:

C:\Program Files\Android\Android Studio

Android SDK Installation Location

The location specified must have at least 3.2GB of free space.
Click Browse to customize:

C:\Users\surdasari\AppData\Local\Android\sdk



Android Studio Setup

Choose Start Menu Folder

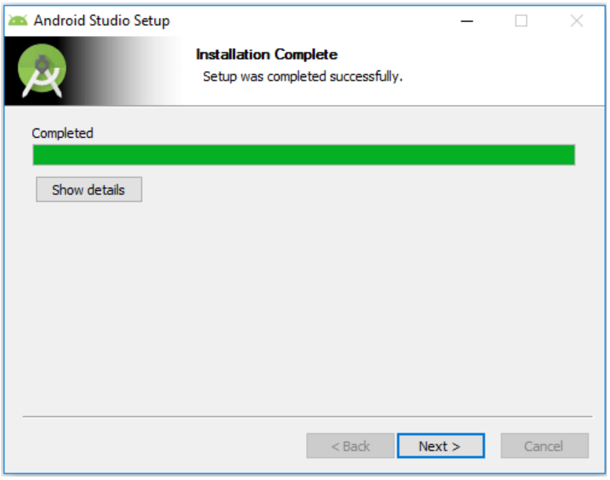
Choose a Start Menu folder for the Android Studio shortcuts.

Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

Android Studio

7-Zip
Accessibility
Accessories
Administrative Tools
Chrome Apps
Cisco
Docker
Git
HP Help and Support
IIS
Intel

□ Do not create shortcuts

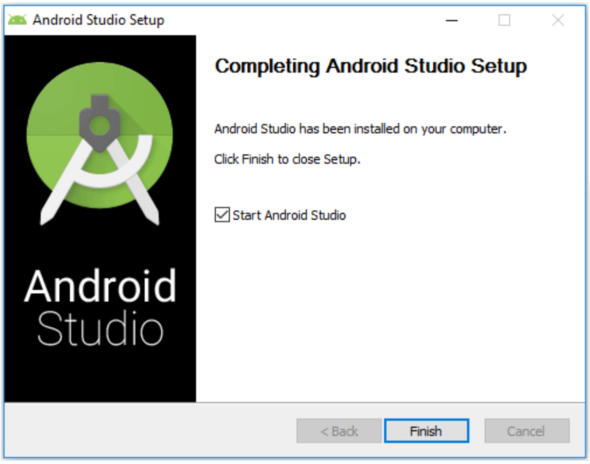


Android Studio Setup

Installation Complete

Setup was completed successfully.

Completed



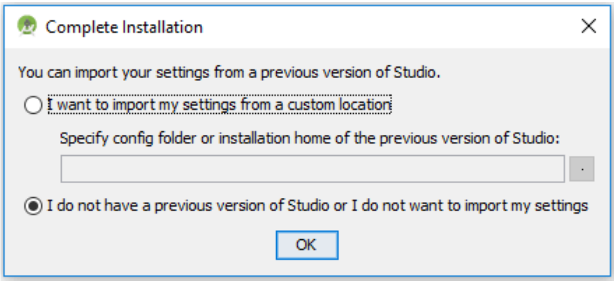
Android Studio Setup

Completing Android Studio Setup

Android Studio has been installed on your computer.

Click Finish to close Setup.

☑ Start Android Studio



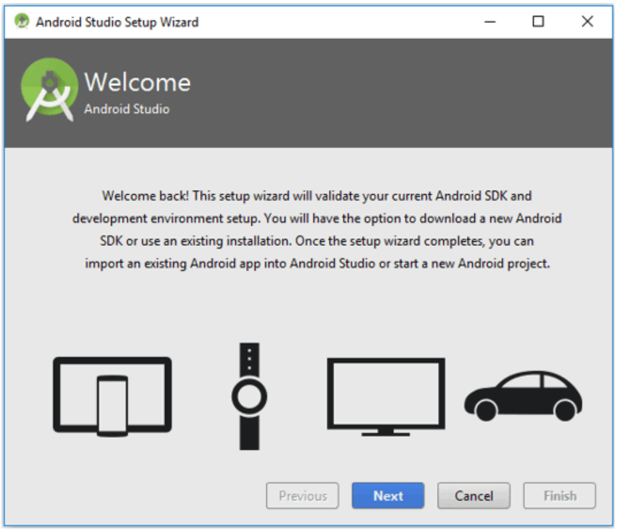
Complete Installation

You can import your settings from a previous version of Studio.

○ I want to import my settings from a custom location

Specify config folder or installation home of the previous version of Studio:

● I do not have a previous version of Studio or I do not want to import my settings

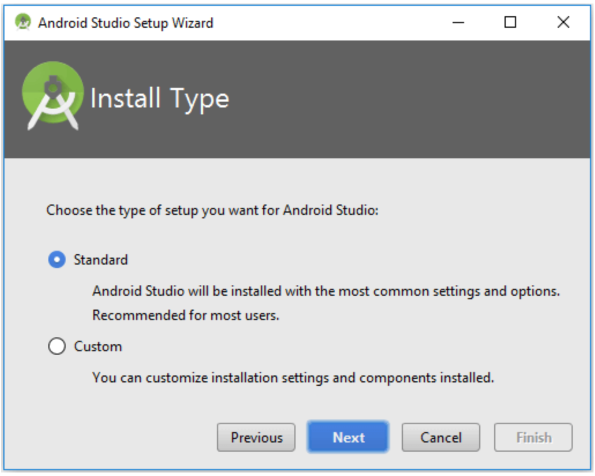


Android Studio Setup Wizard

Welcome

Android Studio

Welcome back! This setup wizard will validate your current Android SDK and development environment setup. You will have the option to download a new Android SDK or use an existing installation. Once the setup wizard completes, you can import an existing Android app into Android Studio or start a new Android project.



Android Studio Setup Wizard

Install Type

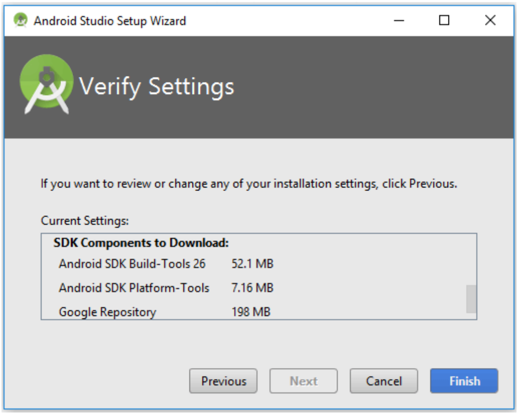
Choose the type of setup you want for Android Studio:

● Standard

Android Studio will be installed with the most common settings and options. Recommended for most users.

○ Custom

You can customize installation settings and components installed.

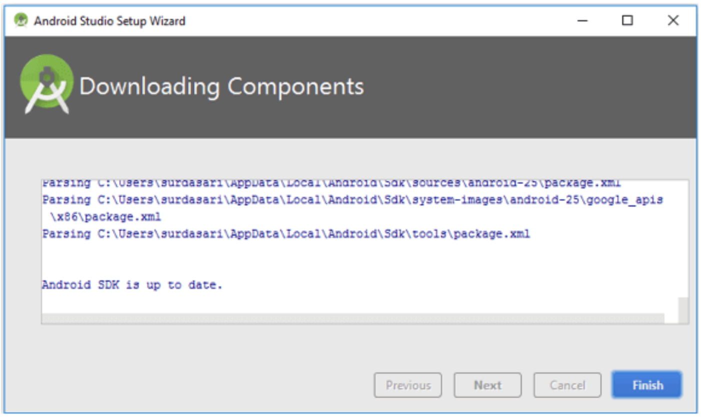


If you want to review or change any of your installation settings, click Previous.

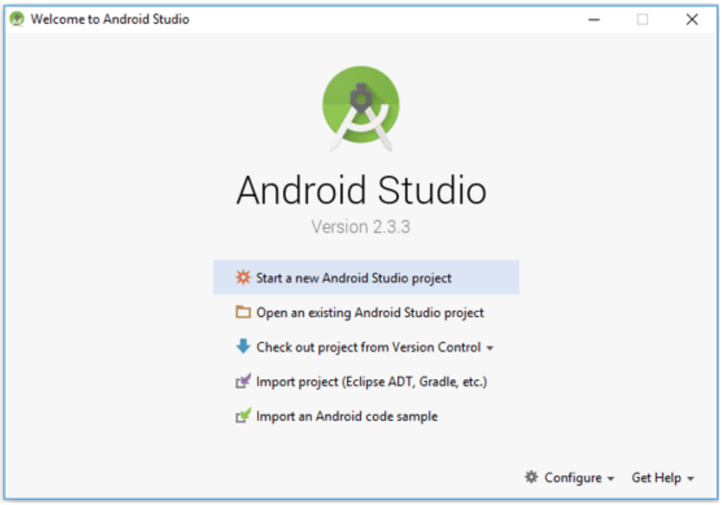
Current Settings:

SDK Components to Download:

Android SDK Build-Tools 26	52.1 MB
Android SDK Platform-Tools	7.16 MB
Google Repository	198 MB



```
Parsing C:\Users\surdasari\AppData\Local\Android\Sdk\sources\android-25\package.xml
Parsing C:\Users\surdasari\AppData\Local\Android\Sdk\system-images\android-25\google_apis
 \x86\package.xml
Parsing C:\Users\surdasari\AppData\Local\Android\Sdk\tools\package.xml


Android SDK is up to date.
```

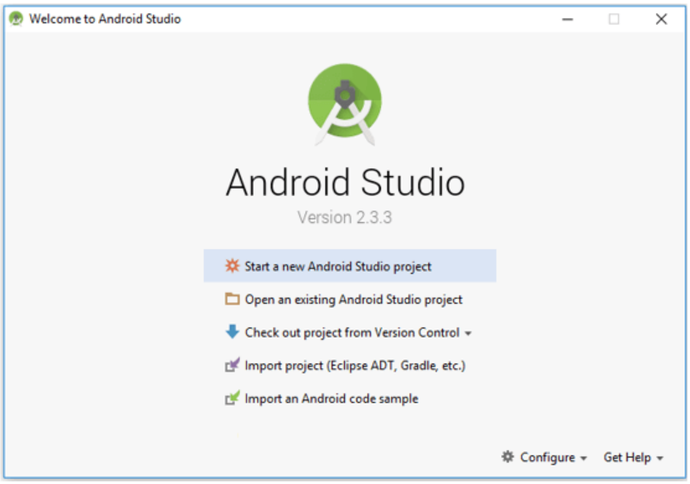

Welcome to Android Studio
Android Studio
Version 2.3.3
Start a new Android Studio project
Open an existing Android Studio project
Check out project from Version Control
Import project (Eclipse ADT, Gradle, etc.)
Import an Android code sample
Configure
Get Help

This is how we can set up an Android development environment on a windows machine which is having windows 10 operating system using android studio IDE.

*Note: This is the 2.3.3 version of Android Studio. Newer version may be different look.

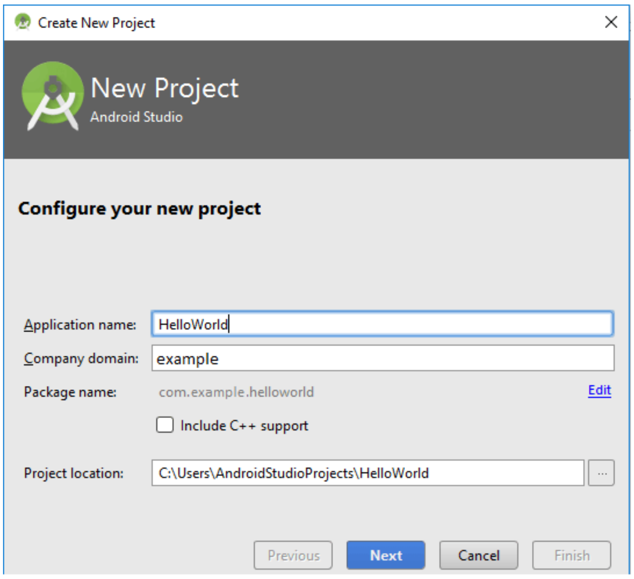
Developing First Android Application

To implement the android hello world app first we need to set up a development environment using android studio IDE which is freely provided by Google for Android developers.


Welcome to Android Studio
Android Studio
Version 2.3.3
Start a new Android Studio project
Open an existing Android Studio project
Check out project from Version Control
Import project (Eclipse ADT, Gradle, etc.)
Import an Android code sample
Configure
Get Help

Select the New Project for the first-time project creation.

- However, we can choose Import Project if we would like to import a project from any source.
- Open Project from the list is for open projects created with either Android Studio.
- Check out from Version Control, we can check out a copy of a project that is under version control. This is a great way to quickly get up to speed with an existing project.
- For the selection "New Project" from the above option, then the next screen will be open like this, where we must mention our Project's name, Company domain and Project location (we called it the main path where this application will be saved) because the Package name will be created automatically as we create the project in Android Studio.



Configure your new project

Application name: HelloWorld

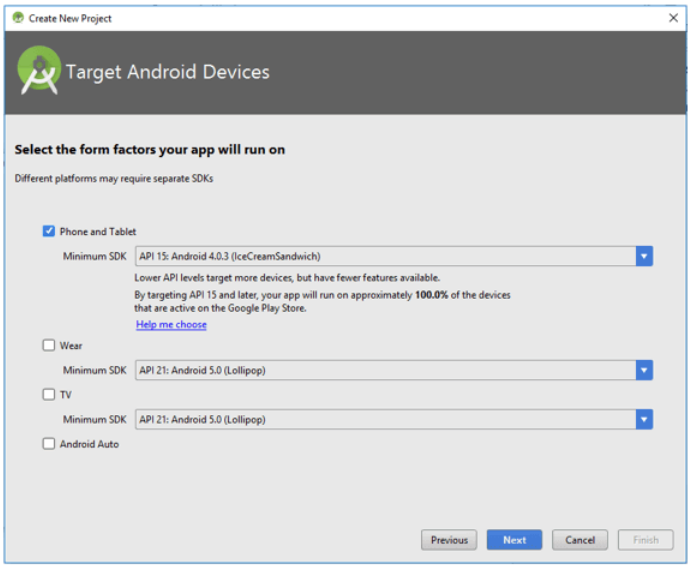
Company domain: example

Package name: com.example.helloworld

□ Include C++ support

Project location: C:\Users\AndroidStudioProjects\HelloWorld

- Select the different platforms and SDK targets like as shown below based on our requirements.



Select the form factors your app will run on

Different platforms may require separate SDKs

☑ Phone and Tablet

Minimum SDK API 15: Android 4.0.3 (IceCreamSandwich)

Lower API levels target more devices, but have fewer features available.

By targeting API 15 and later, your app will run on approximately **100.0%** of the devices that are active on the Google Play Store.

□ Wear

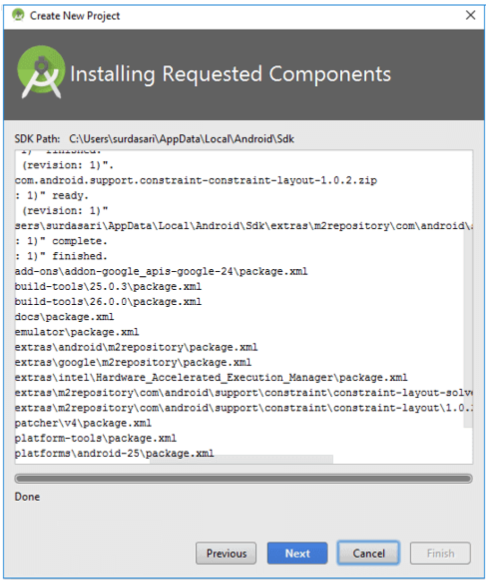
Minimum SDK API 21: Android 5.0 (Lollipop)

□ TV

Minimum SDK API 21: Android 5.0 (Lollipop)

□ Android Auto

- Wear: We use this option for Android Watches which we can wear to our hand and use the same functionality as we do with the Android devices. You can call, set the alarm, capture images, and many more things easily.
- TV: We use this option for SmartIPTV which is very common these days. We can see our favorite channels like we see in our Home Televisions and make the changes in the channel easily.



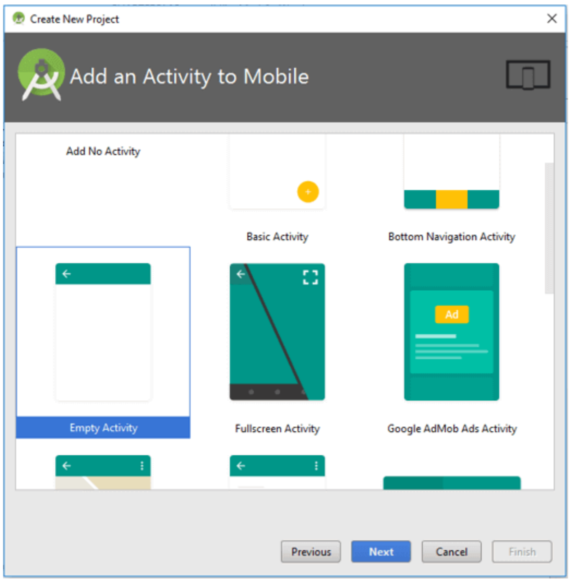
Create New Project

Installing Requested Components

SDK Path: C:\Users\surdasari\AppData\Local\Android\Sdk

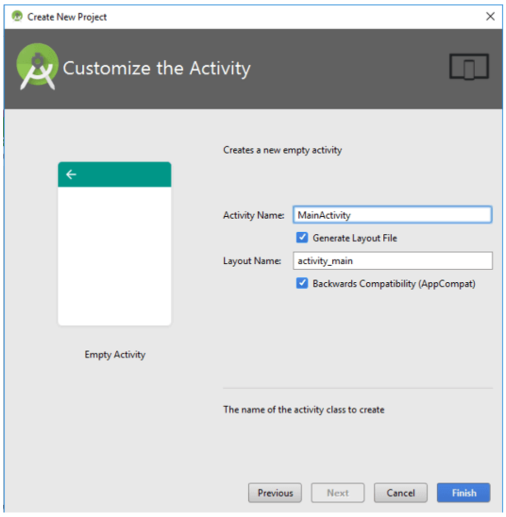
```
 (revision: 1)".
com.android.support.constraint-constraint-layout-1.0.2.zip
: 1)" ready.
 (revision: 1)"
sers\surdasari\AppData\Local\Android\Sdk\extras\m2repository\com\android\
: 1)" complete.
: 1)" finished.
add-ons\addon-google_apis-google-24\package.xml
build-tools\25.0.3\package.xml
build-tools\26.0.0\package.xml
docs\package.xml
emulator\package.xml
extras\android\m2repository\package.xml
extras\google\m2repository\package.xml
extras\intel\Hardware_Accelerated_Execution_Manager\package.xml
extras\m2repository\com\android\support\constraint\constraint-layout-solv
extras\m2repository\com\android\support\constraint\constraint-layout\1.0.
patcher\v4\package.xml
platform-tools\package.xml
platforms\android-25\package.xml
```

Done



Create New Project

Add an Activity to Mobile



Create New Project

Customize the Activity

Creates a new empty activity

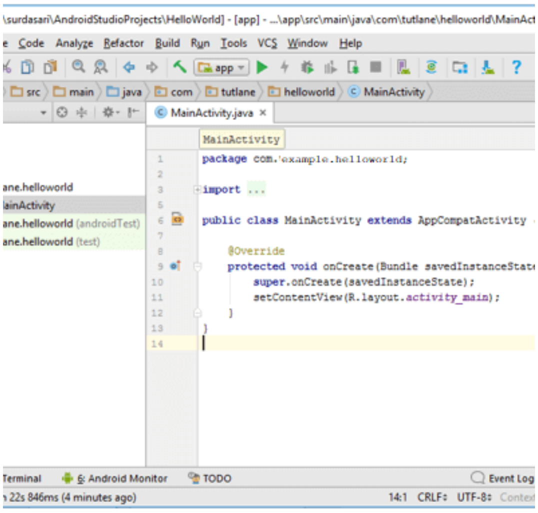
Activity Name: MainActivity

☑ Generate Layout File

Layout Name: activity_main

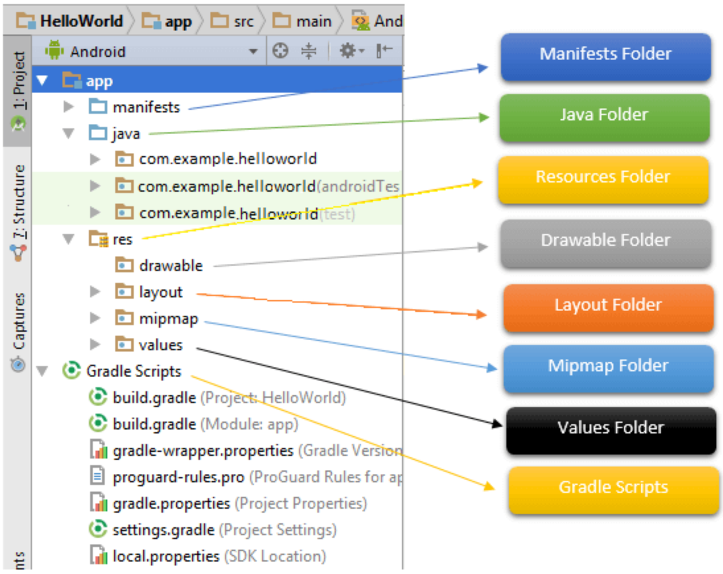
☑ Backwards Compatibility (AppCompat)

The name of the activity class to create

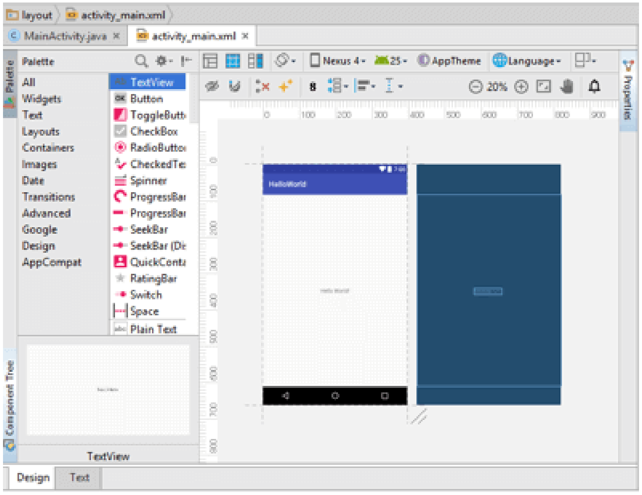


\surdasari\AndroidStudioProjects\HelloWorld] - [app] - ...\app\src\main\java\com\tutlane\helloworld\MainAct

```
package com.example.helloworld;

import ...

public class MainActivity extends AppCompatActivity

    @Override
    protected void onCreate(Bundle savedInstanceStat
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```



HelloWorld
app
src
main
And
Android
1: Project
7: Structure
Captures
app
manifests
java
com.example.helloworld
com.example.helloworld (androidTest)
com.example.helloworld (test)
res
drawable
layout
mipmap
values
Gradle Scripts
build.gradle (Project: HelloWorld)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version
proguard-rules.pro (ProGuard Rules for ap
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
Manifests Folder
Java Folder
Resources Folder
Drawable Folder
Layout Folder
Mipmap Folder
Values Folder
Gradle Scripts



Android Layout File (activity_main.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schem
as.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.helloworld.MainActivity">

      <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
```

Android Main Activity File (MainActivity.java)

```
package com.example.helloworld;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Android Manifest File (AndroidManifest.xml)

The AndroidManifest.xml file contains information of your package, including components of the application such as activities, services, broadcast receivers, content providers etc.
It performs some other tasks like:

- It is responsible to protect the application to access any protected parts by providing the permissions.
- It also declares the android Api that the application is going to use.
- It lists the instrumentation classes. The instrumentation classes provide profiling and other information's. These information's are removed just before the application is published etc.
- This is the required xml file for all the android application and located inside the root directory.

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.helloworld" >
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme" >
        <activity android:name=".MainActivity" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER
" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Elements of the AndroidManifest.xml file

The elements used in the above xml file are described below.

<manifest>: manifest is the root element of the AndroidManifest.xml file. It has package attribute that describes the package name of the activity class.

<application>: application is the sub element of the manifest. It includes the namespace declaration. This element contains several sub elements that declares the application component such as activity etc.

The commonly used attributes are of this element are icon, label, theme etc.

- android:icon represents the icon for all the android application components.
- android:label works as the default label for all the application components.
- android:theme represents a common theme for all the android activities.

<activity>: activity is the sub element of application and represents an activity that must be defined in the AndroidManifest.xml file. It has many attributes such as label, name, theme, launchMode etc.

android:label represents a label i.e. displayed on the screen.

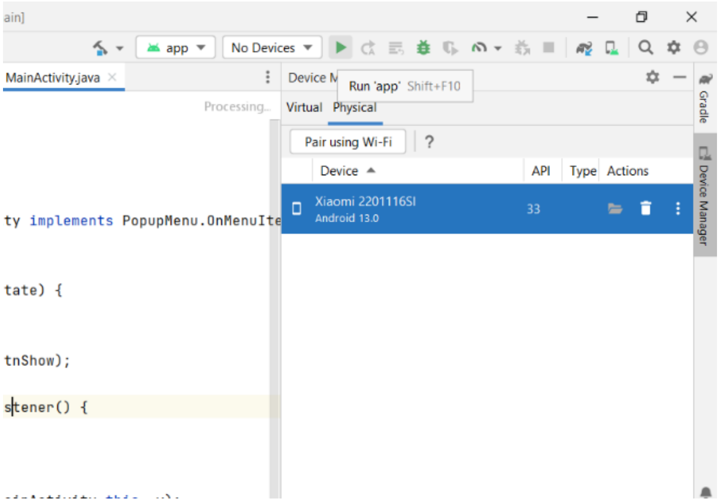
android:name represents a name for the activity class. It is required attribute.

<intent-filter>: intent-filter is the sub-element of activity that describes the type of intent to which activity, service or broadcast receiver can respond to.

<action>: It adds an action for the intent-filter. The intent-filter must have at least one action element.

Run Android Hello World App

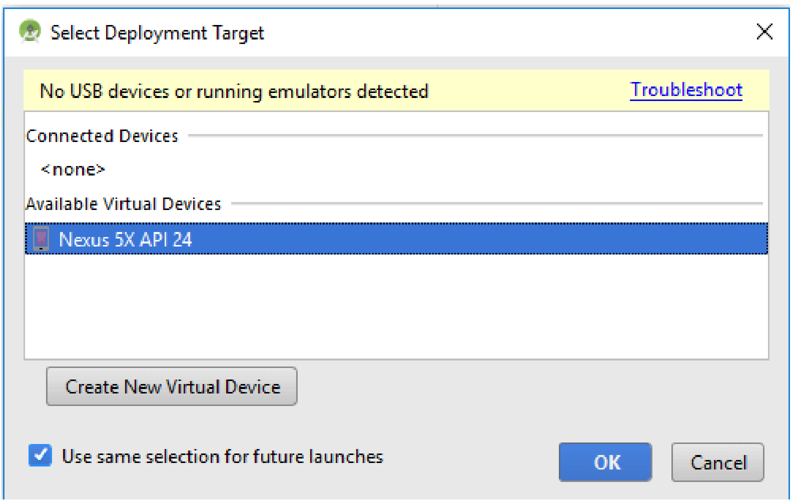
To run android applications, we need to click on **Run** button or press Shift + F10 like as shown below



```
ty implements PopupMenu.OnMenuIte

tate) {

tnShow);

stener() {
```

Device	API	Type	Actions
Xiaomi 2201116SI Android 13.0	33		

If you want to run on VM you can follow below steps.


Select Deployment Target

No USB devices or running emulators detected

Connected Devices

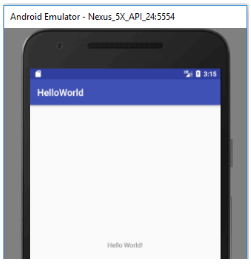
<none>

Available Virtual Devices

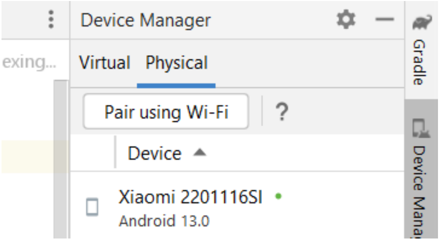
Nexus 5X API 24

☑ Use same selection for future launches

Now our android hello world application will show the result like as shown below



Here, I am going to use my android phone for the testing purpose, for using phone into a android studio, first turn on developer options as well as USB debugging from settings of your phone. For that you can follow these steps.

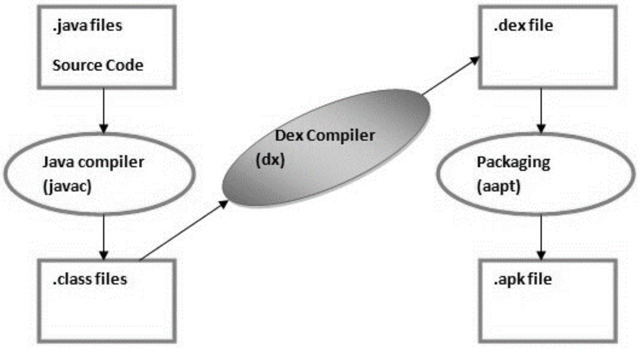


You can select, USB or Wi-Fi as well. Here just select name of your phone and then click on run to test app on your phone.

Dalvik Virtual Machine | DVM

- As we know the modern JVM is high performance and provides excellent memory management. But it needs to be optimized for low-powered handheld devices as well.
- The Dalvik Virtual Machine (DVM) is an android virtual machine optimized for mobile devices. It optimizes the virtual machine *for* memory*,* battery life *and* performance*.*
- Dalvik is a name of a town in Iceland. The Dalvik VM was written by Dan Bornstein.
- The Dex compiler converts the class files into the .dex file that run on the Dalvik VM. Multiple class files are converted into one dex file.

Below is the compiling and packaging process from the source file:



The javac tool compiles the java source file into the class file.

The dx tool takes all the class files of your application and generates a single .dex file. It is a platform-specific tool.

The Android Assets Packaging Tool (aapt) handles the packaging process.