

Step by Step process of installing Android Studio



Step 1 - System Requirements

You will be delighted, to know that you can start your Android application development on either of the following operating systems –

- Microsoft® Windows® 8/7/Vista/2003 (32 or 64-bit).
- Mac® OS X® 10.8.5 or higher, up to 10.9 (Mavericks)
- GNOME or KDE desktop

Second point is that all the required tools to develop Android applications are open source and can be downloaded from the Web. Following is the list of software's you will need before you start your Android application programming.

- Java JDK5 or later version
- Java Runtime Environment (JRE) 6
- Android Studio

Step 2 - Setup Android Studio Overview

Android Studio is the official IDE for android application development. It works based on **IntelliJ IDEA**, You can download the latest version of android studio from Android Studio Download, If you are new to installing Android Studio on windows, you will find

Step by Step process of installing Android Studio

a file, which is named as *android-studio-bundle-135.17407740-windows.exe*. So just download and run on windows machine according to android studio wizard guideline.

If you are installing Android Studio on Mac or Linux, You can download the latest version from Android Studio Mac Download, or Android Studio Linux Download, check the instructions provided along with the downloaded file for Mac OS and Linux. This tutorial will consider that you are going to setup your environment on Windows machine having Windows 8.1 operating system.

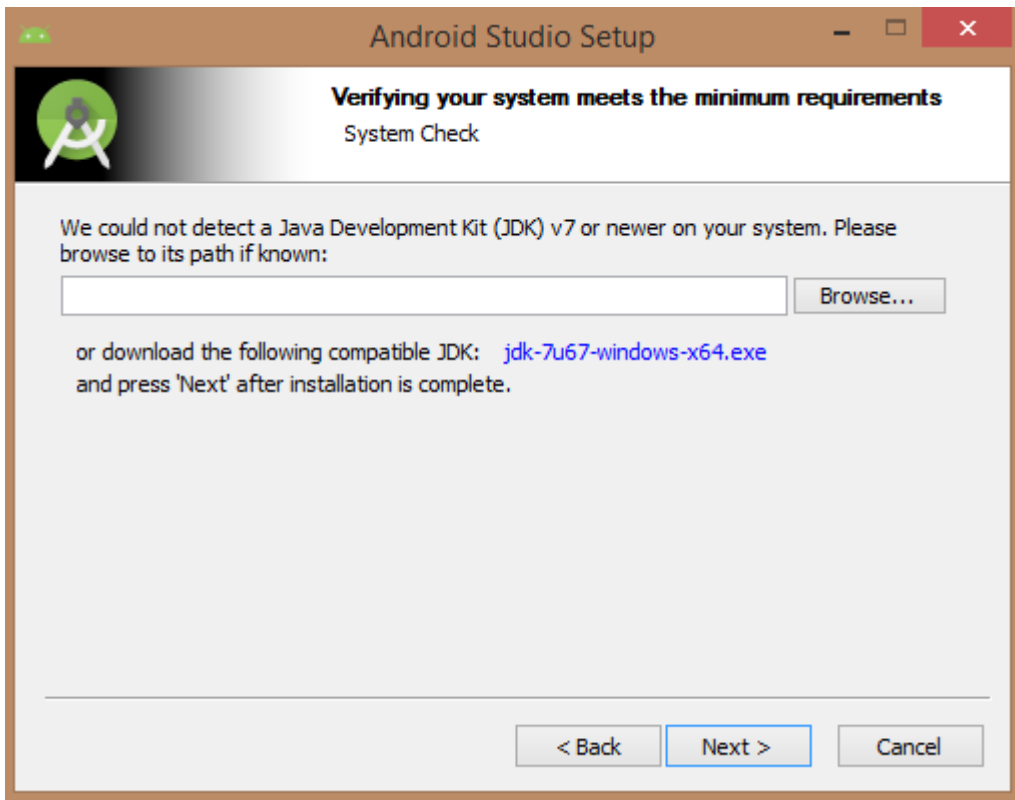
Installation

So let's launch *Android Studio.exe*, Make sure before launch Android Studio, Our Machine should required installed Java JDK. To install Java JDK, take a references of [Android environment setup](#)

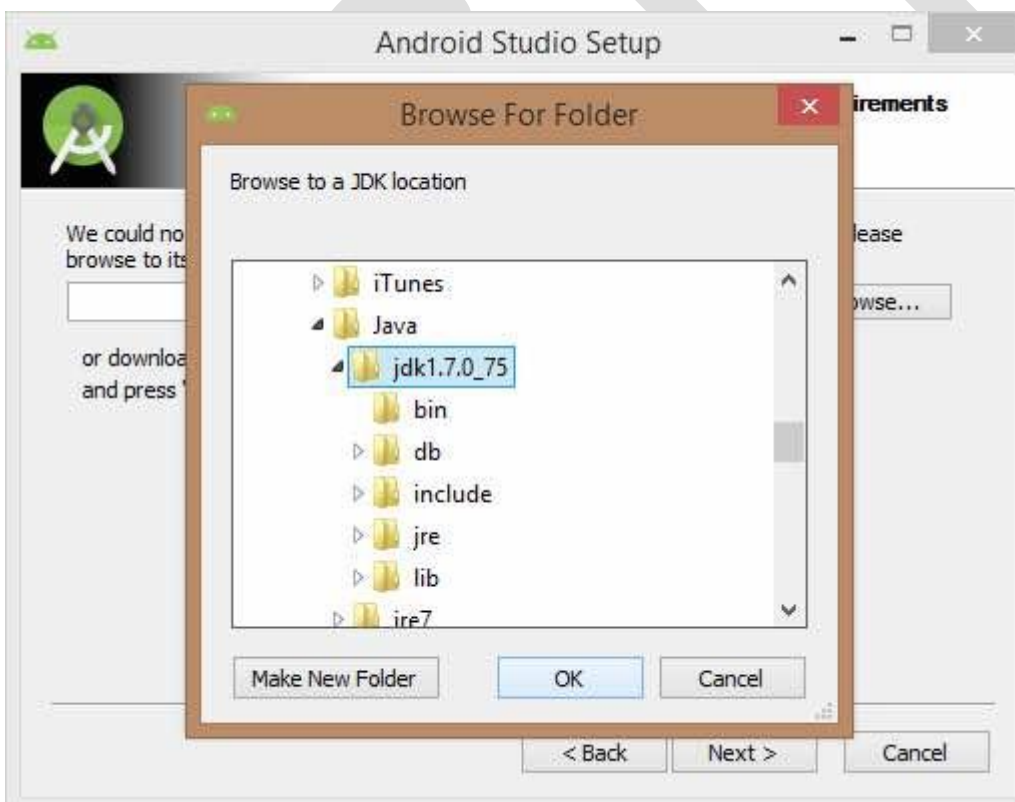


Once you launched Android Studio, its time to mention JDK5 path or later version in android studio installer.

Step by Step process of installing Android Studio

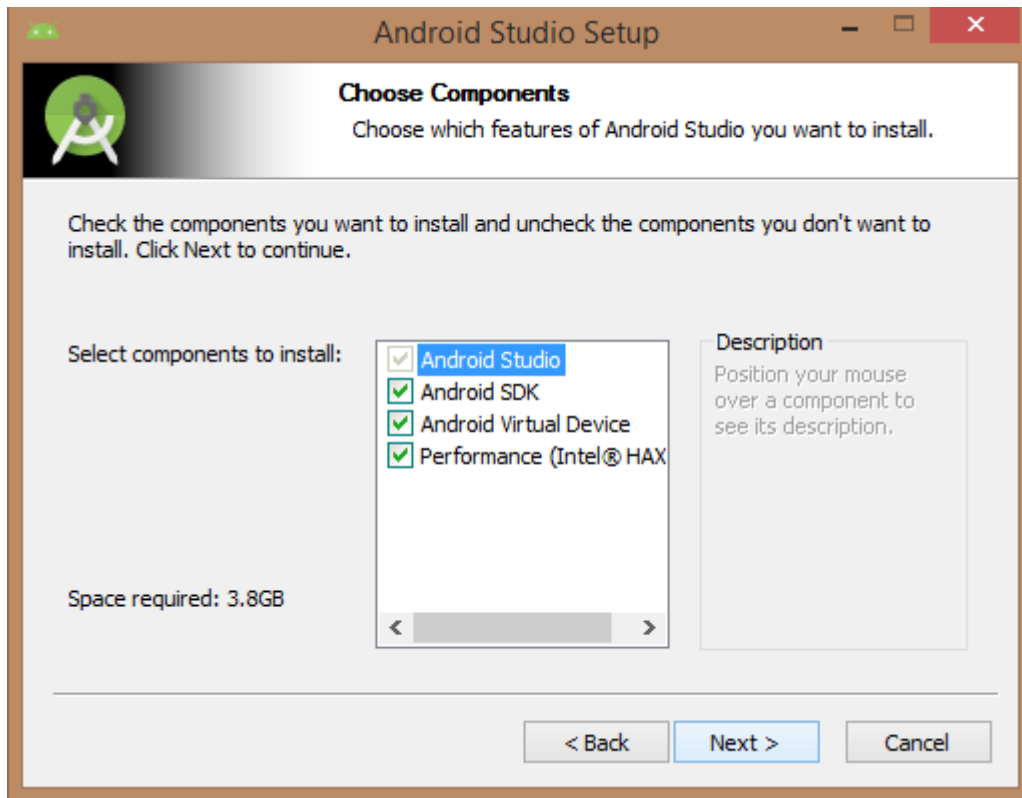


Below the image initiating JDK to android SDK

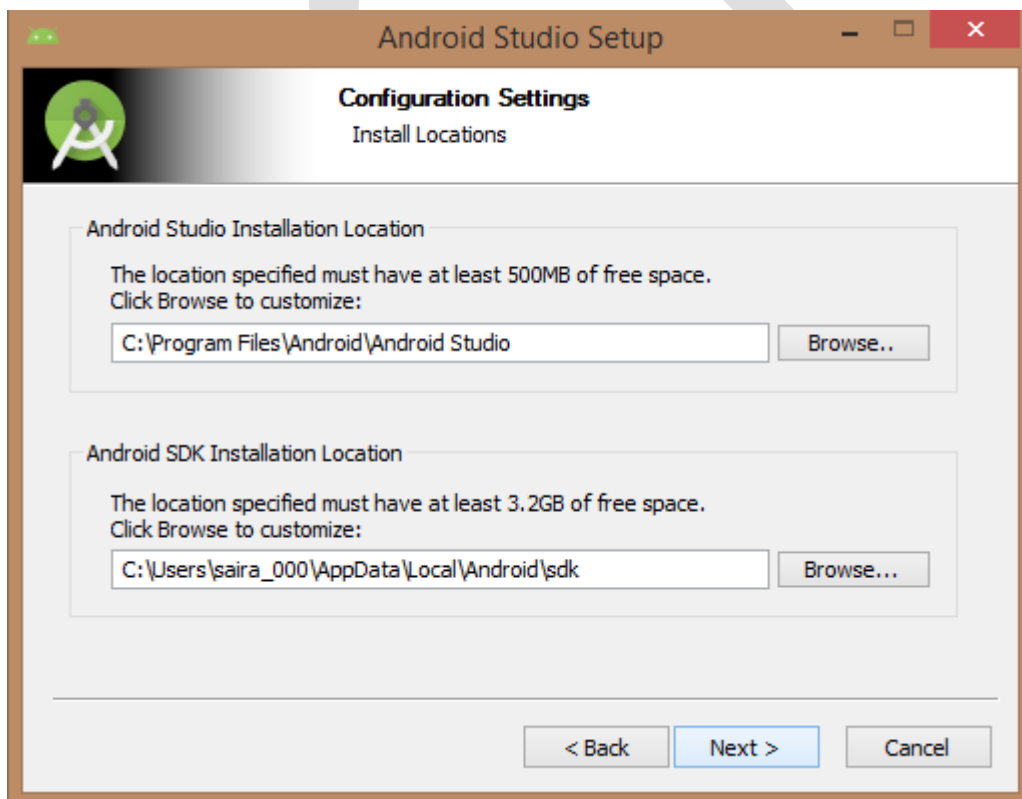


Need to check the components, which are required to create applications, below the image has selected **Android Studio, Android SDK, Android Virtual Machine** and **performance (Intel chip)**.

Step by Step process of installing Android Studio

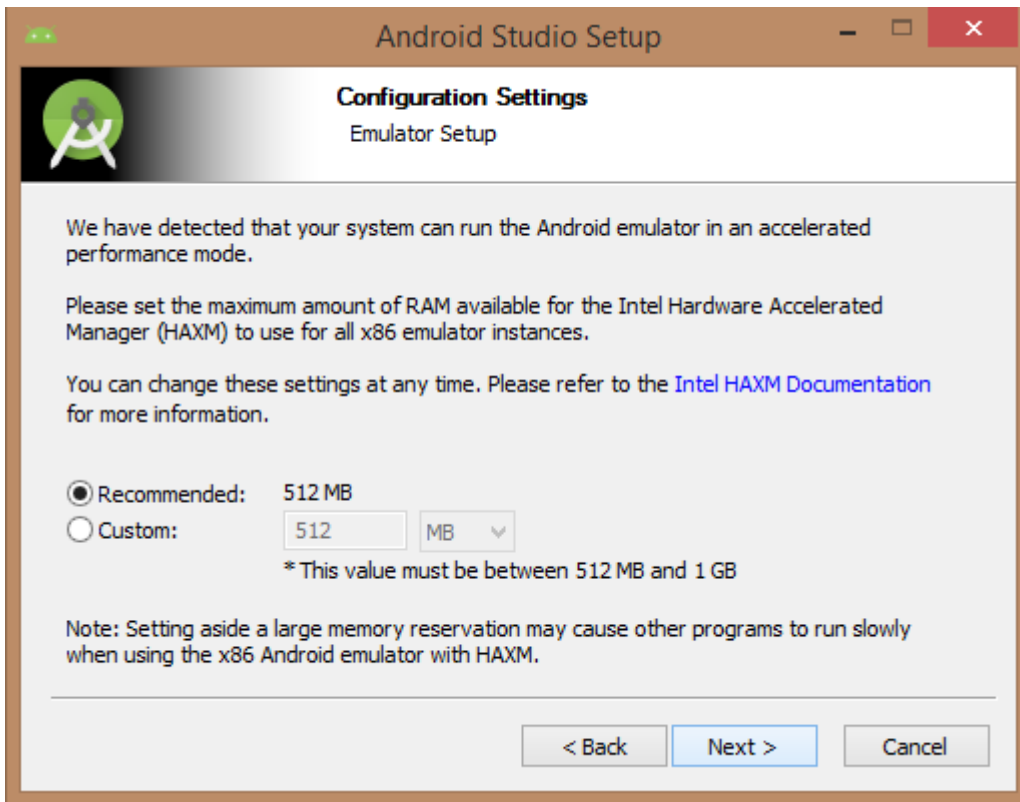


Need to specify the location of local machine path for Android studio and Android SDK, below the image has taken default location of windows 8.1 x64 bit architecture.

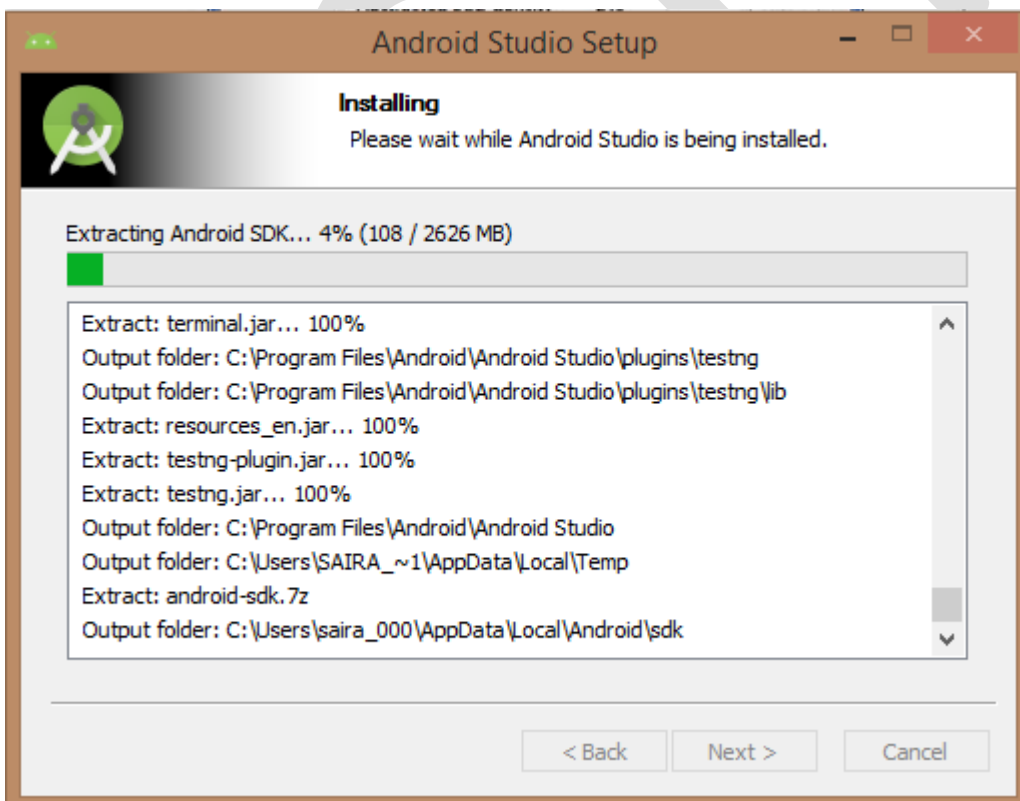


Need to specify the ram space for Android emulator by default it would take 512MB of local machine RAM

Step by Step process of installing Android Studio



At final stage, it would extract SDK packages into our local machine, it would take a while time to finish the task and would take 2626MB of Hard disk space.



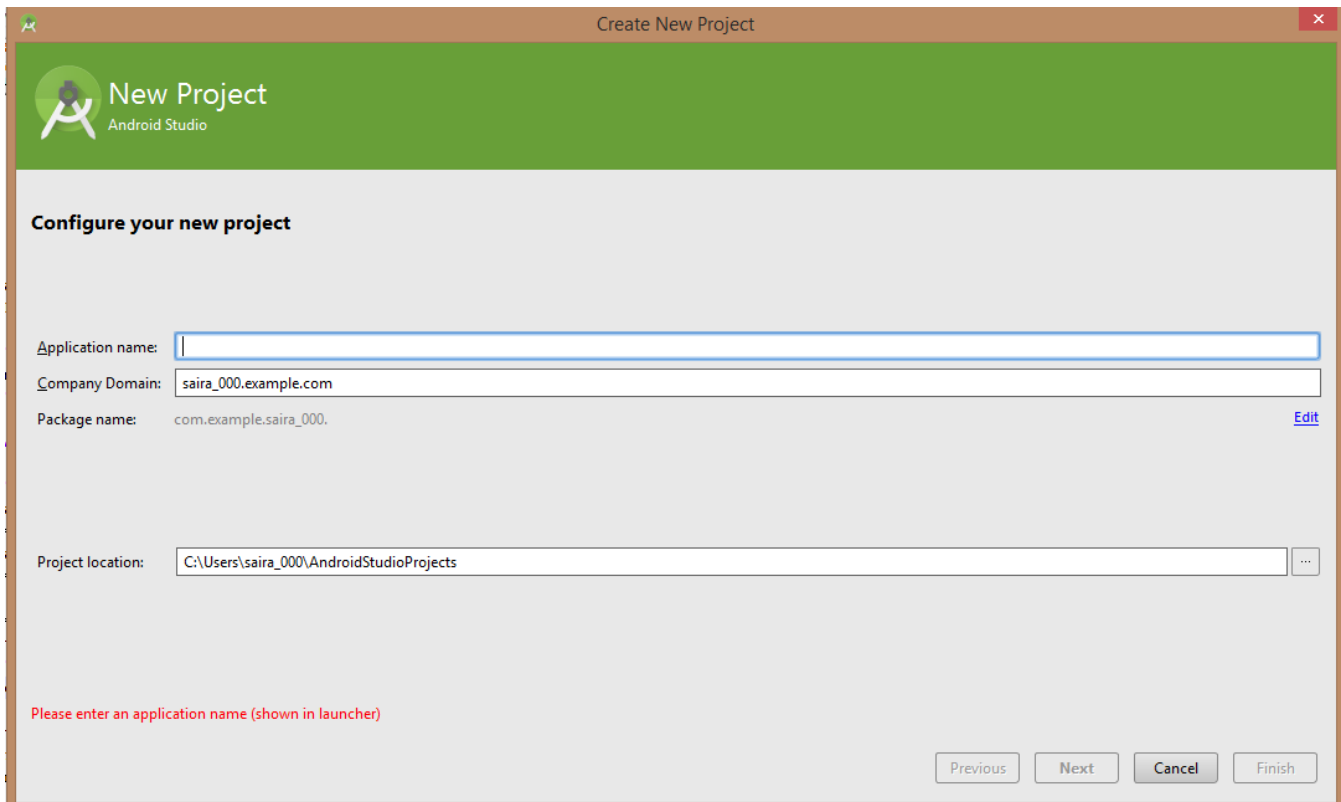
After done all above steps perfectly, you must get finish button and it gonna be open android studio project with Welcome to android studio message as shown below

Step by Step process of installing Android Studio



You can start your application development by calling start a new android studio project. in a new installation frame should ask Application name, package information and location of the project.

Step by Step process of installing Android Studio

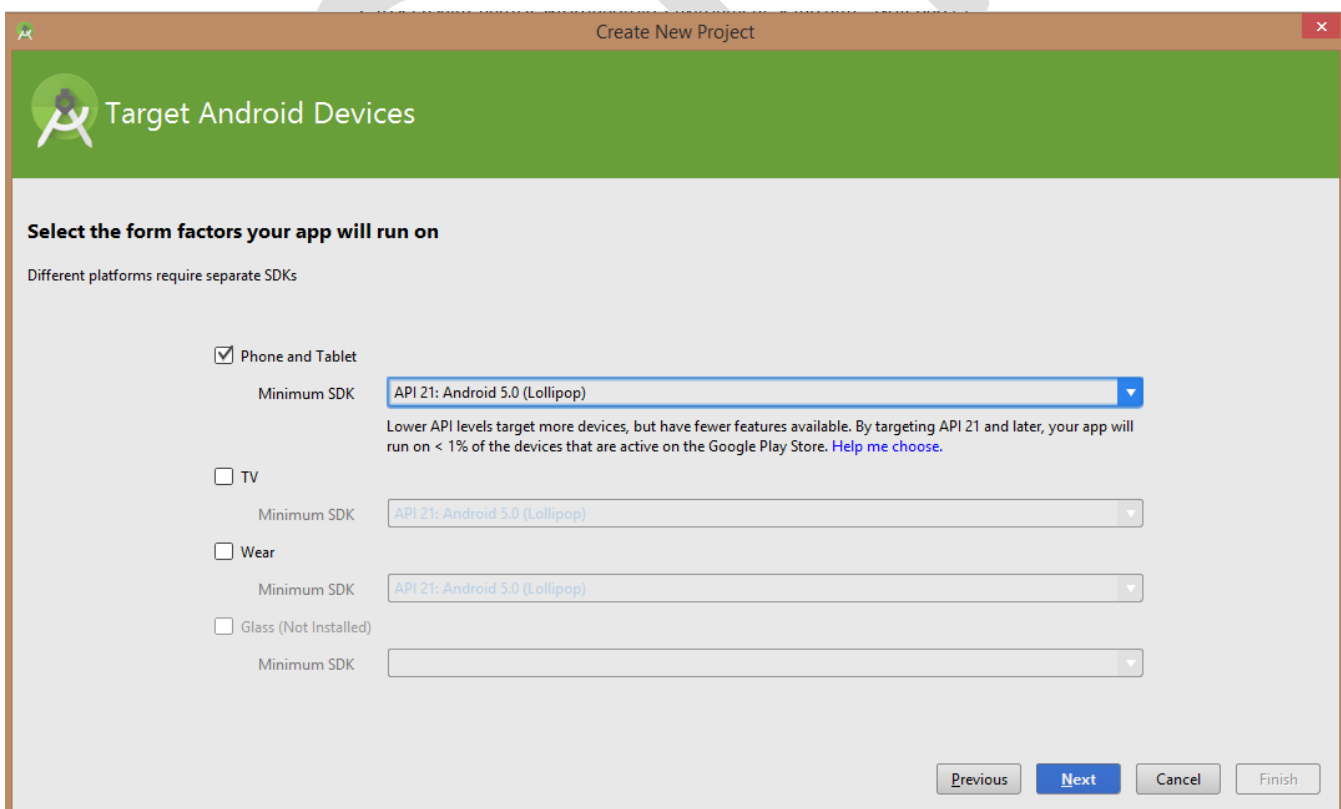


The screenshot shows the 'Create New Project' dialog in Android Studio. The title bar reads 'Create New Project'. The main heading is 'New Project' with the Android Studio logo. Below this, the instruction 'Configure your new project' is displayed. The form contains the following fields:

- Application name:** An empty text input field.
- Company Domain:** A text input field containing 'saira_000.example.com'.
- Package name:** A text input field containing 'com.example.saira_000.' with an 'Edit' link to its right.
- Project location:** A text input field containing 'C:\Users\saira_000\AndroidStudioProjects' with a browse button ('...') to its right.

At the bottom left, there is a red error message: 'Please enter an application name (shown in launcher)'. At the bottom right, there are four buttons: 'Previous', 'Next', 'Cancel', and 'Finish'.

After entering application name, specify Minimum SDK, I have declared as API21: Android 5.0(Lollipop).



The screenshot shows the 'Create New Project' dialog in Android Studio, specifically the 'Target Android Devices' step. The title bar reads 'Create New Project'. The main heading is 'Target Android Devices' with the Android Studio logo. Below this, the instruction 'Select the form factors your app will run on' is displayed, followed by the note 'Different platforms require separate SDKs'.

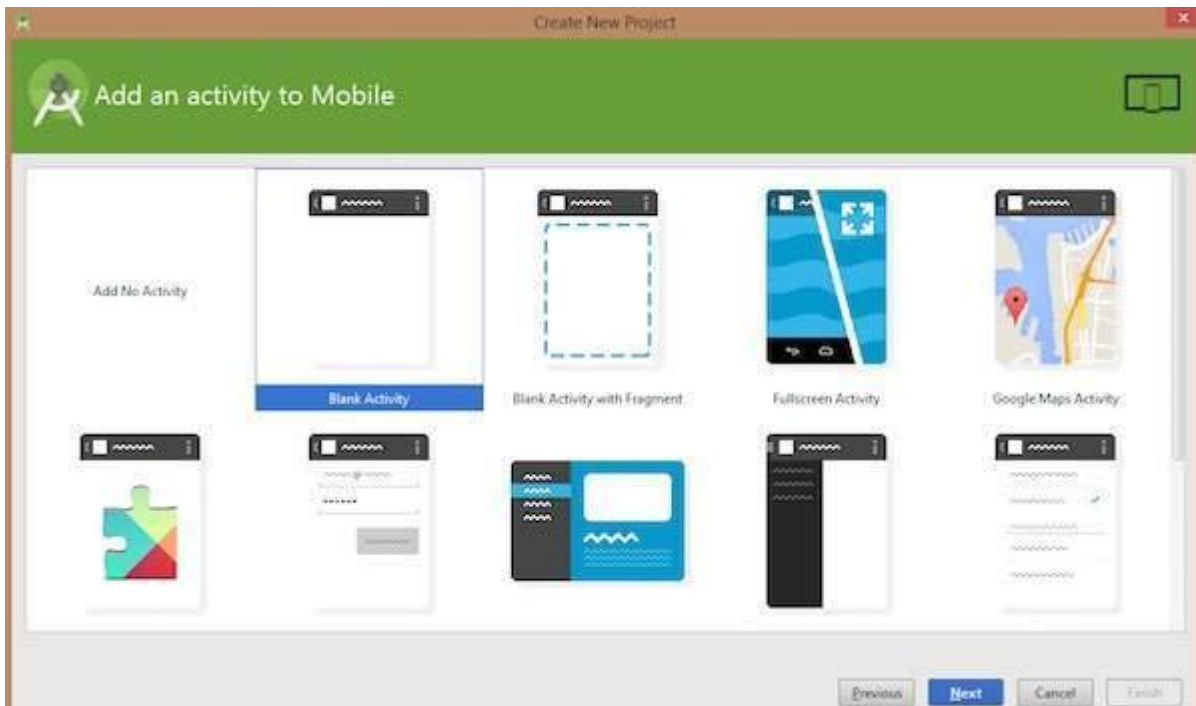
The form contains the following options:

- Phone and Tablet**
 - Minimum SDK: A dropdown menu showing 'API 21: Android 5.0 (Lollipop)'. Below this dropdown, there is a note: 'Lower API levels target more devices, but have fewer features available. By targeting API 21 and later, your app will run on < 1% of the devices that are active on the Google Play Store. [Help me choose.](#)'
- TV**
 - Minimum SDK: A dropdown menu showing 'API 21: Android 5.0 (Lollipop)'.
- Wear**
 - Minimum SDK: A dropdown menu showing 'API 21: Android 5.0 (Lollipop)'.
- Glass (Not Installed)**
 - Minimum SDK: An empty dropdown menu.

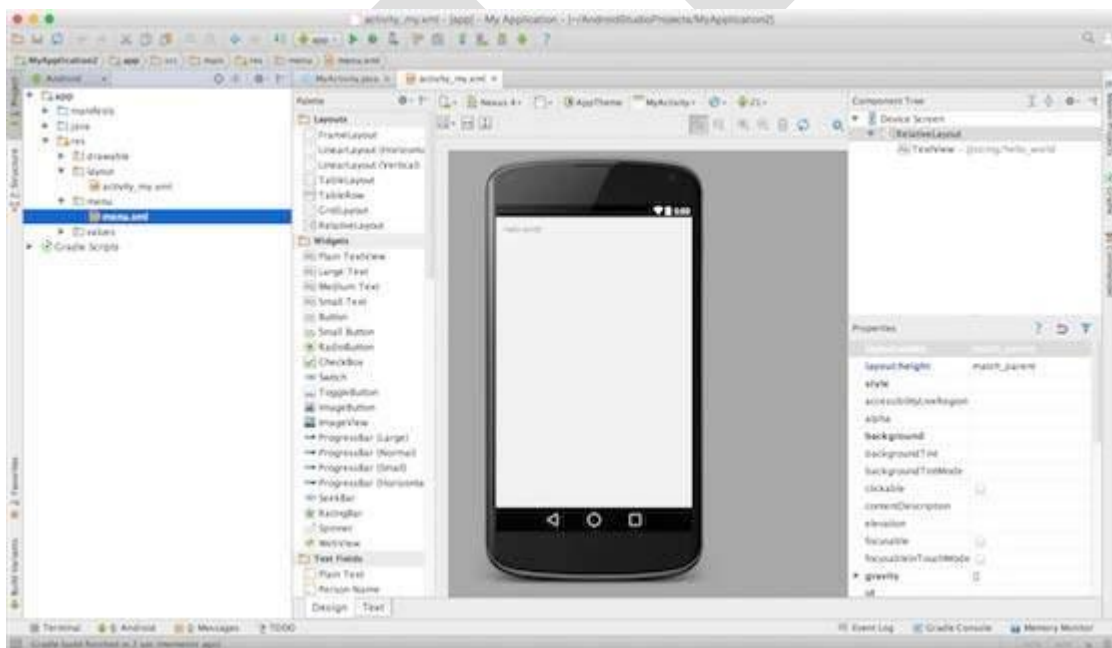
At the bottom right, there are four buttons: 'Previous', 'Next', 'Cancel', and 'Finish'.

The next level of installation should contain selecting the activity to mobile, it specifies the default layout for Applications

Step by Step process of installing Android Studio



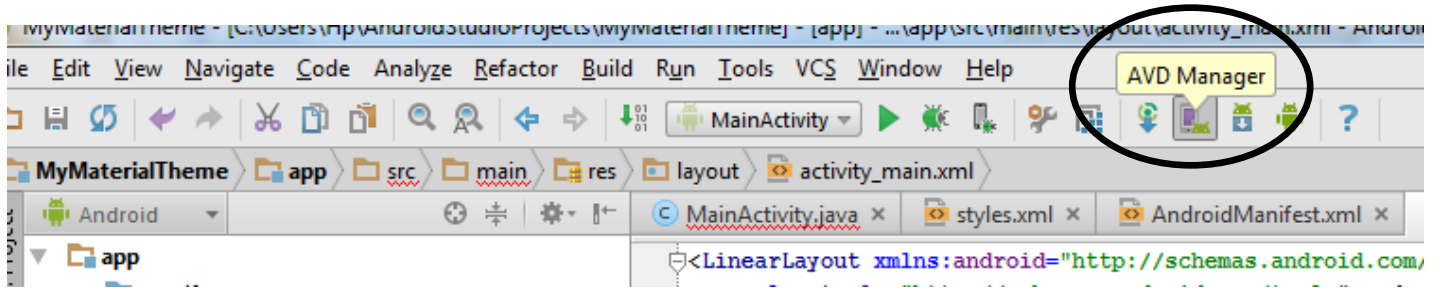
At the final stage it going to be open development tool to write the application code.



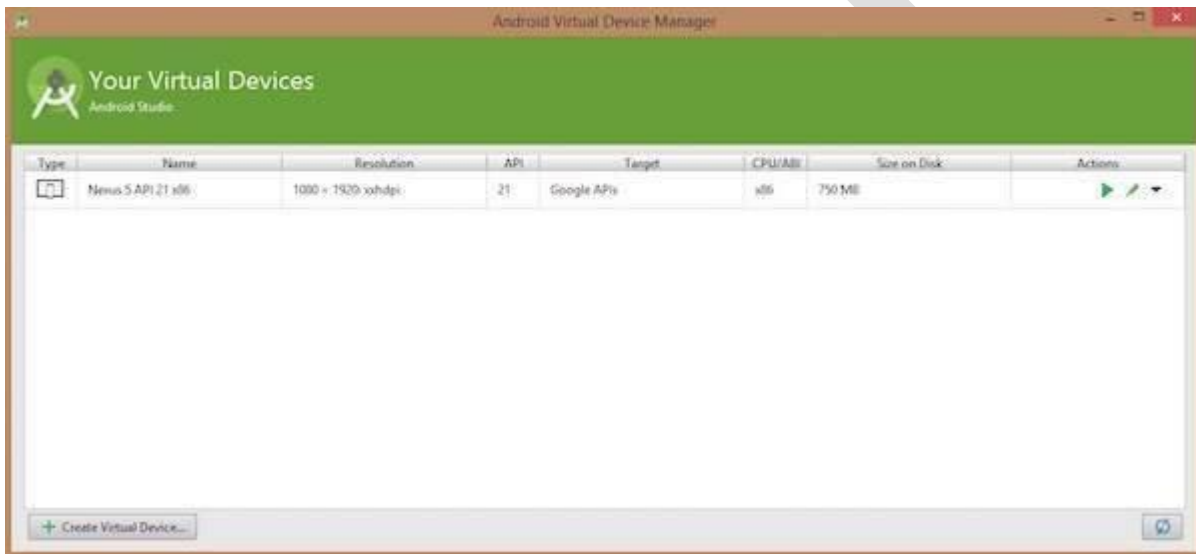
Step 3 - Create Android Virtual Device

To test your Android applications, you will need a virtual Android device. So before we start writing our code, let us create an Android virtual device. Launch Android AVD Manager Clicking AVD_Manager icon as shown below

Step by Step process of installing Android Studio



After Click on a virtual device icon, it going to be shown by default virtual devices which are present on your SDK, or else need to create a virtual device by clicking **Create new Virtual device** button

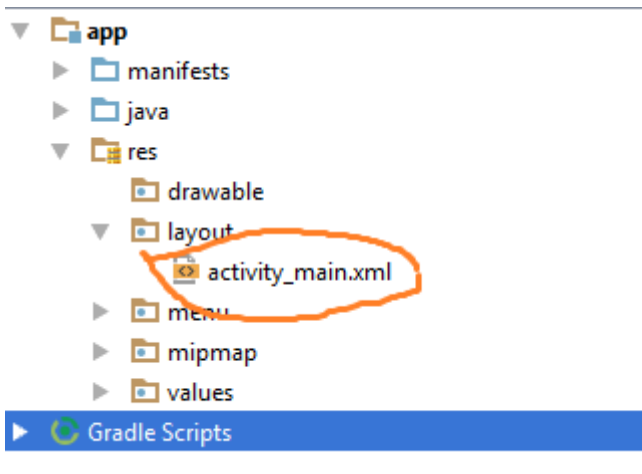


If your AVD is created successfully it means your environment is ready for Android application development. If you like, you can close this window using top-right cross button. Better you re-start your machine and once you are done with this last step, you are ready to proceed for your first Android example but before that we will see few more important concepts related to Android Application Development.

Hello Word Example

Before Writing a Hello word code, you must know about XML tags.To write hello word code, you should redirect to **App>res>layout>Activity_main.xml**

Step by Step process of installing Android Studio



To show hello word, we need to call text view with layout (about text view and layout, you must take references at Relative Layout and Text View).

```
RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin" tools:context=".MainActivity">

    <TextView android:text="@string/hello_world"
        android:layout_width="550dp"
        android:layout_height="wrap_content" />
</RelativeLayout>
```

Need to run the program by clicking **Run>Run App** or else need to call **shift+f10** key. Finally, result should be placed at Virtual devices as shown below

What is Gradle in Android Studio?

Short Answer

Gradle is a build system.

Long Answer

Before Android Studio you were using Eclipse for your development purposes (*Even today Eclipse is very popular among Java and Android Developers*), and, chances are, you didn't know how to build your Android APK without Eclipse.

You can do this on the command line, but you have to learn what each tool (dx, aapt) does in the SDK. Eclipse saved us all from these low level but important, fundamental details by giving us their own build system.

This is where the build system enters the picture. The build system automatically takes all the source files (.java or .xml), then applies the appropriate tool (e.g. takes java class files and converts them to dex files), and groups all of them into one compressed file, our beloved APK.

This build system uses some conventions: an example of one is to specify the directory containing the source files (in Eclipse it is \src folder) or resources files (in Eclipse it is \res folder). Now, in order to automate all these tasks, there has to be a script; you can write your own build system using shell scripting in linux or batch files syntax in windows. Got it?

Gradle is another build system that takes the best features from other build systems and combines them into one. It is improved based off of their shortcomings. It is a **JVM based build system**, what that means is that you can write your own script in Java, which Android Studio makes use of.

One cool thing about gradle is that it is a **plugin based system**. This means if you have your own programming language and you want to automate the task of building some package (output like a JAR for Java) from sources then you can write a complete plugin in Java or Groovy, and distribute it to rest of world.

Why did Google use it?

Google saw one of the most advanced build systems on the market and realized that you could write scripts of your own with little to no learning curve, and without learning Groovy or any other new language. So they wrote the Android plugin for Gradle.

You must have seen `build.gradle` file(s) in your project. That is where you can write scripts to automate your tasks. The code you saw in these files is Groovy code. If you write `System.out.println("Hello Gradle!");` then it will print on your console.