

# Learn how to Install Android Studio and Use IDE for Mobile App Development

## Contents

Introduction .....	2
Scope of the Document .....	2
Setting up the Android Studio .....	2
Step 1: Downloading the IDE .....	2
Step 2: Installing the IDE .....	2
Working with Android Studio .....	6
Step 1: Launching the IDE .....	6
Step 2: Creating a new Project.....	6
Step 3: Developing the Application.....	7
Step 4: Running the Application .....	9
References .....	10
Acronyms Used .....	9

## Introduction

Android Studio is the official IDE for Android development, and includes everything you need to build Android apps.


## Scope of the Document

This document covers the various aspects involved in the development of an Android application using Android Studio.

## Setting up the Android Studio

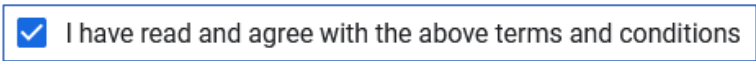
### Step 1: Downloading the IDE

- In any web browser navigate to the website -- <https://developer.android.com/studio>
- In the site navigate to the button “Download Android Studio”, shown below.



**Download Android Studio**

- On clicking this button, browser will open an agreement form, please read carefully and navigate to the bottom and click on the below checkbox to enable download.



☒ I have read and agree with the above terms and conditions

- Once above step is completed, click on the button shown below.

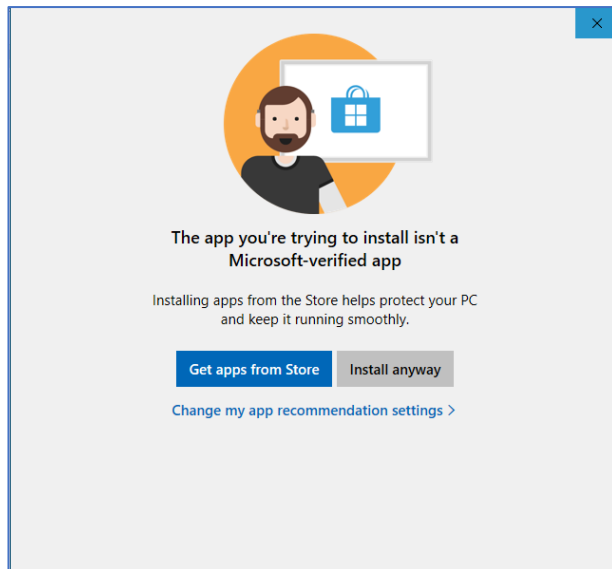


**Download Android Studio 2020.3.1 for Windows**

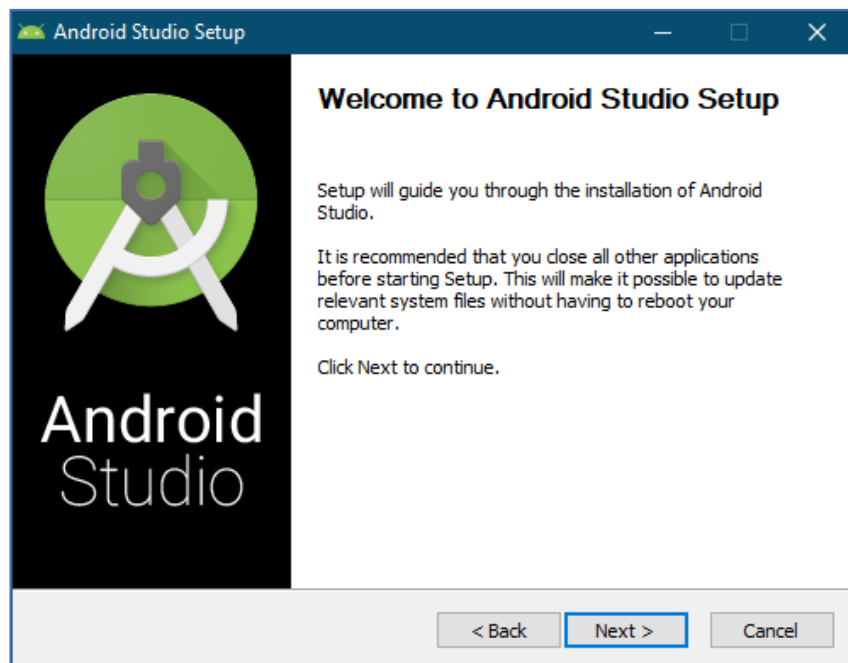
- The above step will download a file with the name “android-studio-2020.3.1.24-windows.exe” in the downloads folder of your system. This is an executable file for installing Android Studio IDE 3.1 Version for 64Bit Windows OS.

### Step 2: Installing the IDE

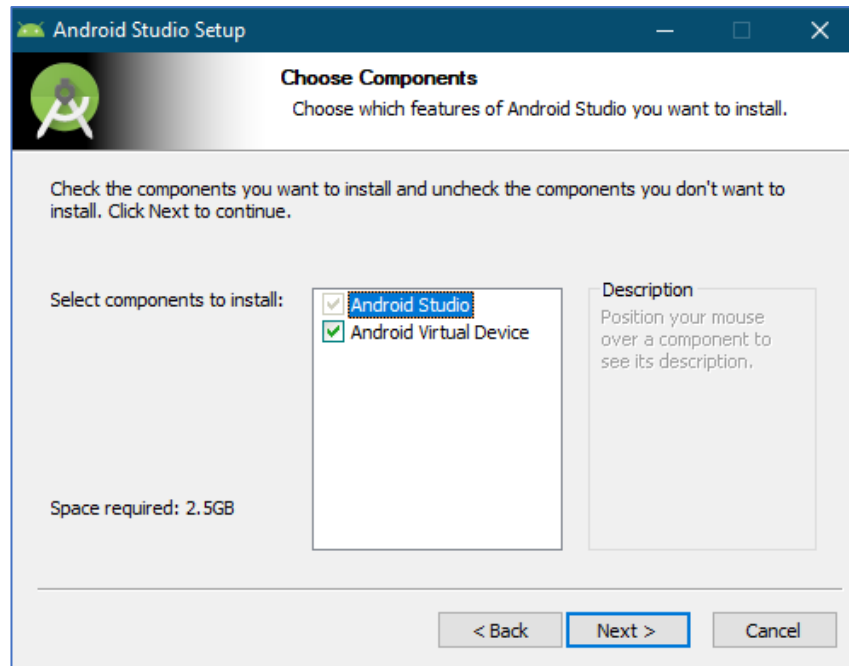
- Double click on the executable file “android-studio-2020.3.1.24-windows.exe” available in downloads folder to start the installation process. This will open a new window as below.



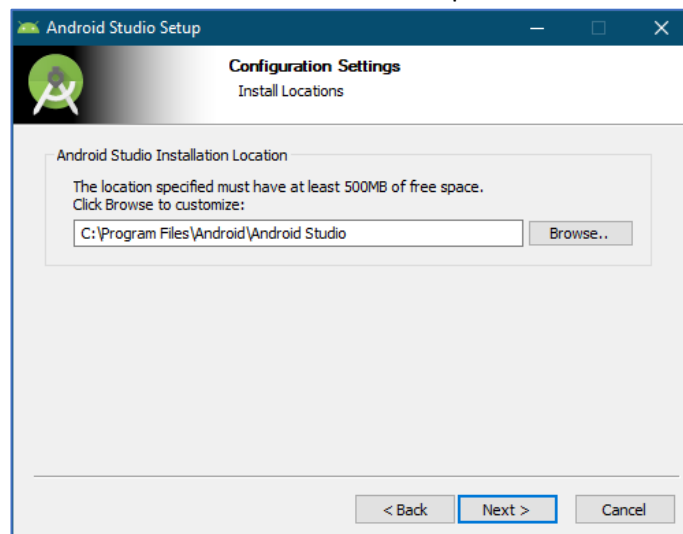
- Click on “Install Anyways”, this will lead you to Admin permission screen. After giving the requisite permission to install the application, system will open setup guide for the IDE.
- The first page will be the Welcome page in the setup guide as shown below. Click on “Next” to proceed further.



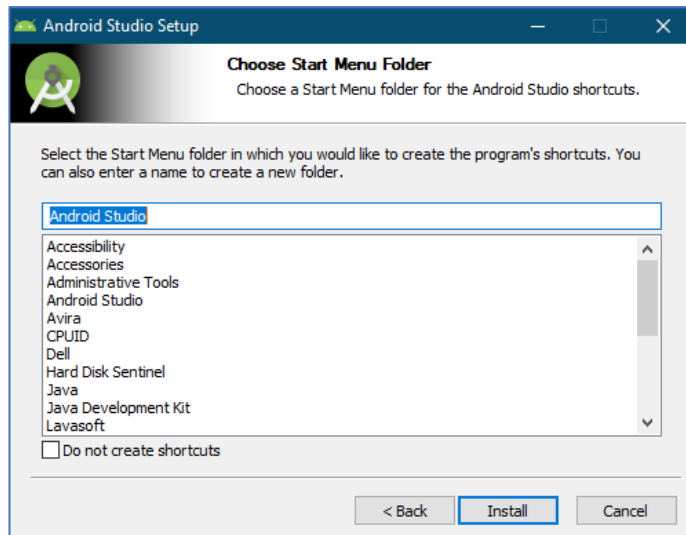
- After the above step, setup will go to next page where you have option to choose the components for installation. By default, “Android Studio” is checked and disabled. “Android Virtual Device” is optional. Keep it checked for installation. Click “Next” to proceed further.



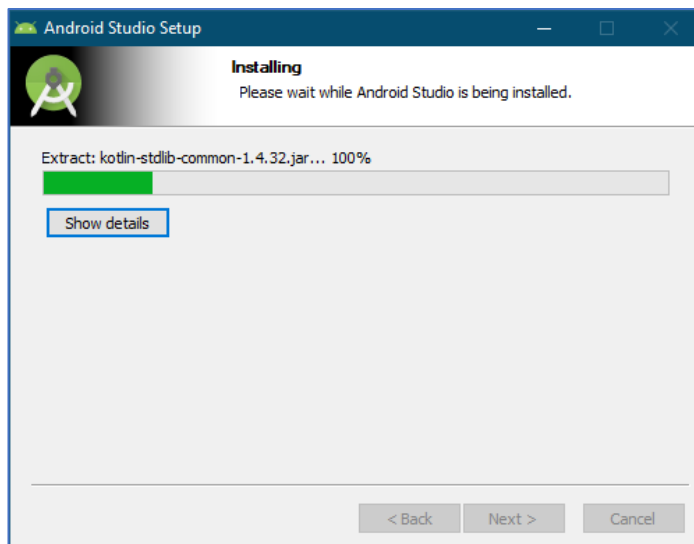
- After this setup will ask for the directory. Default directory (C:\Program Files\Android\Android Studio) can be used for installation. However, this can be changed in case you are using some specific drive for software installations. Click “Next” to proceed further.



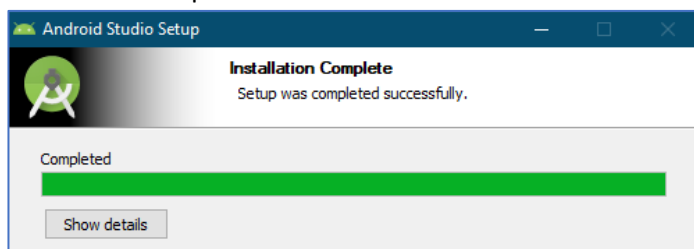
- Setup will ask for Start Menu Folder. Without changing anything click “Install” to proceed further.



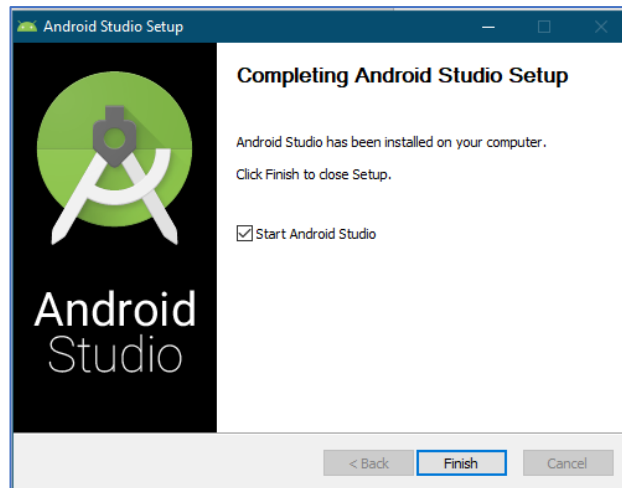
- The above step will initiate the installation procedure. Progress can be seen through the progress bar as shown below.



- Once the installation is completed, progress bar will be full with a message "Completed" as shown below. Click on "Next" to proceed further.



- Setup will show below screen. You may keep the checkbox "Start Android Studio" checked to launch the IDE immediately or Un-check (recommended) to launch later. Click on "Finish" to complete the setup.

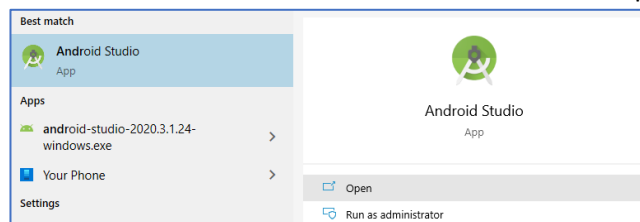


- With this Android Studio IDE installation is completed.

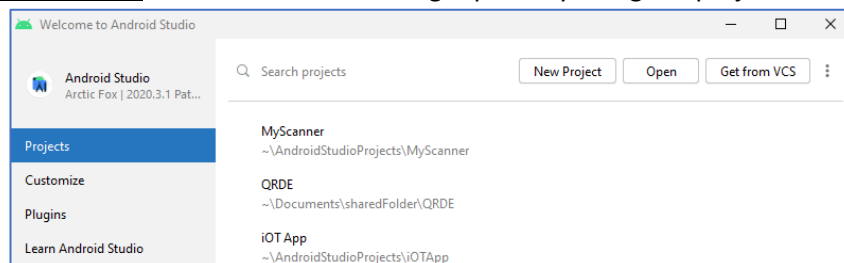
## Working with Android Studio

### Step 1: Launching the IDE

- In the start menu, search for the icon “Android Studio” and launch the application.



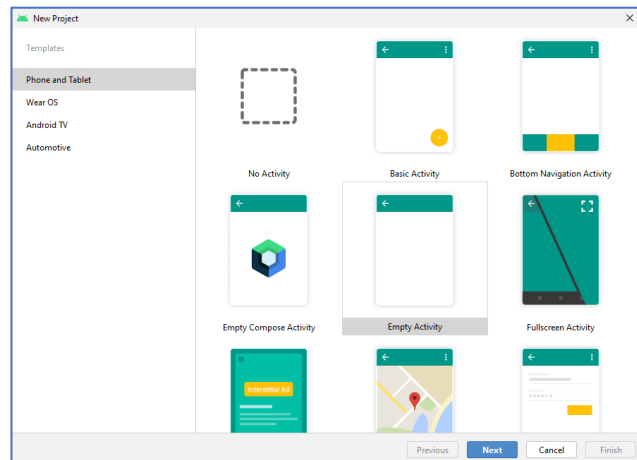
- This will launch the Android Studio welcome screen shown below.
- In this screen there are various options listed below:
  - Projects: to view and open any earlier projects built and managed in the PC.
  - Customize: to customize the IDE.
  - Plugins: to add any plugins required during the development process.
  - Learn Android Studio: a comprehensive guide on using the IDE.
  - New Project: to start a new project.
  - Open: to open an existing project.
  - Get from VCS: to connect to an existing repository and get a project.



### Step 2: Creating a new Project

- To create a new project, click on the “New Project” button.
- This will open a new window with the setup procedure for a new project.
- In this window there are a few options listed below:

- Phone and Tablet: to develop applications for Phones and Tablets.
- Wear OS: to develop applications for wearable device running on Android.
- Android TV: to develop applications for Android TV devices.
- Automotive: to develop applications for Android devices used in vehicles.



- Please select “Basic Activity” under “Phone and Tablet” section to create a basic application for phone and click on “Next” to proceed further.
- Now you will be prompted to enter the following details:
  - Name: Suitable name for the project. For now, let’s go with default “My Application”.
  - Package name: Name of the package for this application. No change required for now.
  - Save location: Location where the project has to be saved. No change for now.
  - Language: Language in which the development will take place. It can be either Java or Kotlin based on your choice. For now, let’s go with Java.
  - Minimum SDK: The minimum SDK this application has to support. For now, no change.

- Once done click on “Finish”. This will create a new Android project in a new window.
- Keep internet connected during the project creation process as IDE has to integrate some components.
- This will take a few minutes time.

### Step 3: Developing the Application

- After the project is created, there will be a project viewer in the IDE which will show all the components available. They will look something like below:

Sr.No.	Folder, File & Description
1	<b>Java:</b> This contains the <b>.java</b> source files for your project. By default, it includes an <i>MainActivity.java</i> source file having an activity class that runs when your app is launched using the app icon.



2	<b>res/drawable-hdpi:</b> This is a directory for drawable objects that are designed for high-density screens.
3	<b>res/layout:</b> This is a directory for files that define your app's user interface.
4	<b>res/values:</b> This is a directory for other various XML files that contain a collection of resources, such as strings and colours definitions.
5	<b>AndroidManifest.xml:</b> This is the manifest file which describes the fundamental characteristics of the app and defines each of its components.
6	<b>Build.gradle:</b> This is an auto generated file which contains compileSdkVersion, buildToolsVersion, applicationId, minSdkVersion, targetSdkVersion, versionCode and versionName

- Following section will give a brief overview of the important application files.

- The Main Activity File:

The main activity code is a Java file **MainActivity.java**. This is the actual application file which ultimately gets converted to a Dalvik executable and runs your application. Here, *R.layout.activity\_main* refers to the *activity\_main.xml* file located in the *res/layout* folder. The *onCreate()* method is one of many methods that are figured when an activity is loaded. We will modify it to look like below:

```
package com.example.myapplication;

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);
    }
}
```

- The Manifest File:

Whatever component you develop as a part of your application, you must declare all its components in a *manifest.xml* which resides at the root of the application project directory. This file works as an interface between Android OS and your application, so if you do not declare your component in this file, then it will not be considered by the OS. For example, a default manifest file will look like as following file:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.myapplication">
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.MyApplication">
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:label="@string/app_name"
```

```

        android:theme="@style/Theme.MyApplication.NoActionBar">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
</manifest>

```

- The Strings File:

The **strings.xml** file is located in the *res/values* folder and it contains all the text that your application uses. For example, the names of buttons, labels, default text, and similar types of strings go into this file. This file is responsible for their textual content.

We will modify the Strings to look like below:

```

<resources>
    <string name="app_name">HelloWorld</string>
    <string name="hello_world">Hello world!</string>
    <string name="menu_settings">Settings</string>
    <string name="title_activity_main">MainActivity</string>
</resources>

```

- The Layout File:


The **activity\_main.xml** is a layout file available in *res/layout* directory, that is referenced by your application when building its interface. You will modify this file very frequently to change the layout of your application. For your "Hello World!" application, this file will have following content related to default layout:

```

<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" >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:padding="@dimen/padding_medium"
        android:text="@string/hello_world"
        tools:context=".MainActivity" />
</RelativeLayout>

```

## Step 4: Running the Application

- To run the app from Android studio, open one of your project's activity files and click Run  icon from the tool bar. Android studio installs the app on your AVD and starts it and if everything is fine with your set-up and application, it will display following Emulator window.

## Acronyms Used

- IDE – Integrated Development Environment
- OS – Operating System
- AVD – Android Virtual Device
- SDK – Software Development Kit

## References

- Basics of Android & Development Tutorials:
  - <https://www.tutorialspoint.com/android/index.htm>
  - <https://developer.android.com/training/basics/firstapp>
  - YouTube link: <https://www.youtube.com/watch?v=fis26HvvDI>
  - <https://www.geeksforgeeks.org/android-tutorial/>
- Creating AVD:
  - [https://www.tutorialspoint.com/android/android\\_emulator.htm](https://www.tutorialspoint.com/android/android_emulator.htm)
- Creating Hello World application:
  - [https://www.tutorialspoint.com/android/android\\_hello\\_world\\_example.htm](https://www.tutorialspoint.com/android/android_hello_world_example.htm)
- Simple Login and Registration App development link:
  - <https://www.sourcecodester.com/android/12151/android-simple-registration-and-login-application.html>