

# Mobile Application Development

## Unit- 1 & 2





# Prerequisites

- Object Oriented Programming using Java

## Compulsory Reading Material

- Android ( Vikas Publications) by Prasannakumar Dixit
- The Android™ Developer’s Cookbook-Building Applications with the Android SDK,  
by Ronan Schwarz, Phil Dutson, James Steele, Nelson To (2nd Edition)

## Assessment & Evaluation Criteria

• Quizzes	10 %
• Assignments	10 %
• Mid Term	20 %
• Project	10 %
• Final Term	50 %
• <b>Total:</b>	<b>100 %</b>



# Mobile Operating Systems

- A mobile OS is an operating system for smartphones, tablets, PDAs, or other mobile devices.
- Mobile OSs combine features of a personal computer OS with other features useful for mobile or handheld use; usually including, and most of the following considered essential in modern mobile systems;
  - touchscreen, cellular, Bluetooth, Wi-Fi, GPS mobile navigation, camera, video camera, speech recognition, voice recorder, music player, etc.

## Some Current software platforms

- **Android** (based on the Linux Kernel) is from Google Inc.
- **CyanogenMod** and **Cyanogen OS** are based on the open source Android Open Source Project(AOSP).
- **Fire OS** is an operating system launched by Amazon based on Google's AOSP.
- **iOS** (previously known as iPhone OS) is from Apple Inc.
- **Windows Phone** (Soon to be Windows 10 Mobile) is from Microsoft.
- **BlackBerry 10** (based on the QNX OS) is from BlackBerry.
- **Firefox OS** is from Mozilla.



# Introduction to Android

- Android is an OS based on Linux with a Java programming interface. It is a comprehensive open source platform designed for mobile devices.
- First beta version of Android Software Development Kit (SDK) was released by Google in 2007 where as first commercial version, Android 1.0, was released in September 2008.

## Features of Android

- Beautiful UI, Connectivity, Storage, Media support, Messaging, Web browser, Multi-touch, Multi-tasking, Resizable widgets, Multi-Language, GCM, Wi-Fi Direct, Android Beam

## Android Applications

- Android applications are usually developed in the Java language using the Android Software Development Kit.
- Once developed, Android applications can be packaged easily and sold out either through a store such as Google Play, SlideME, Opera Mobile Store, Mobango, F-droid and the Amazon Appstore.

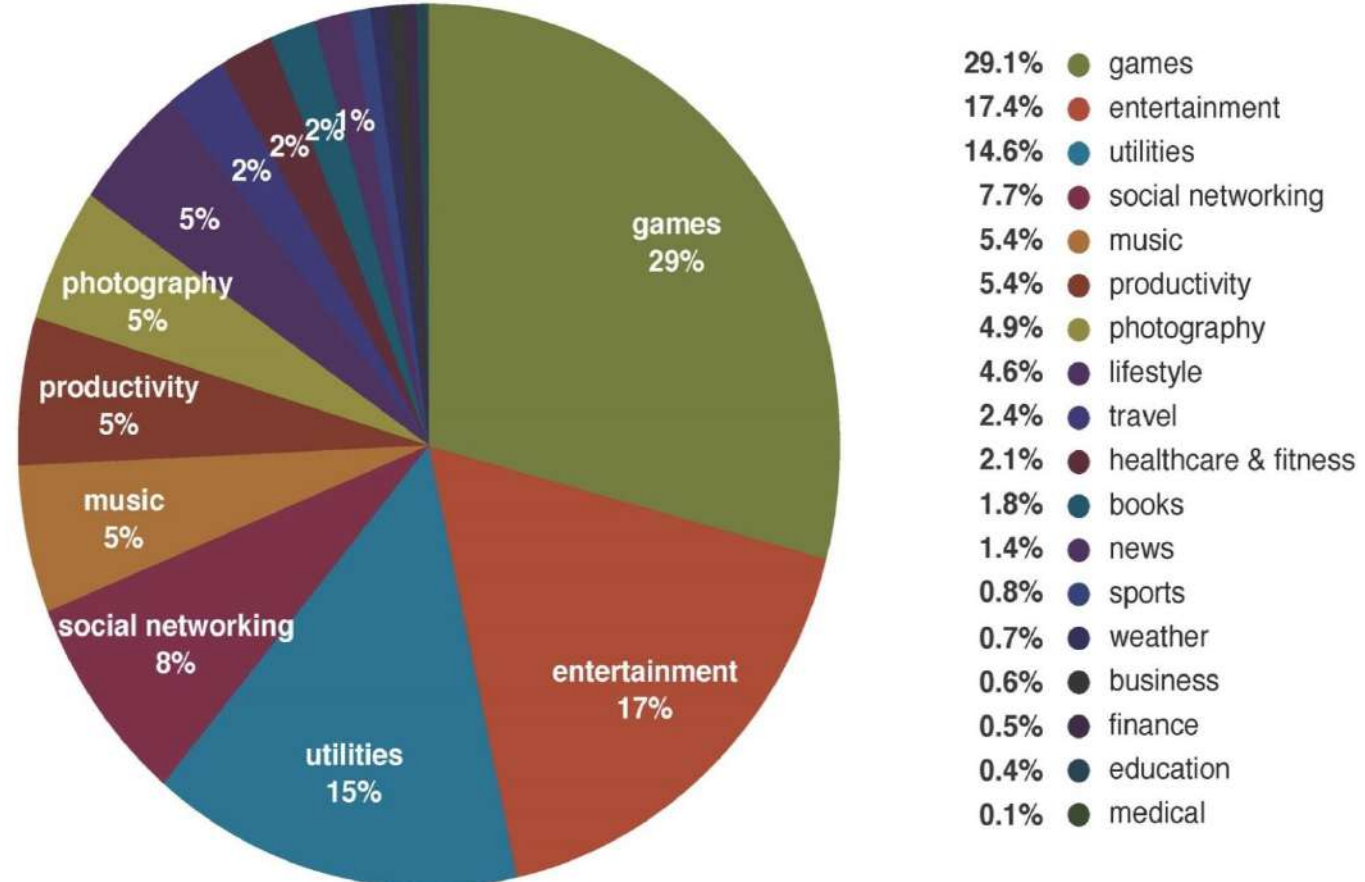
# Introduction to Android

## History of Android

- The code names of android ranges from A to L currently, such as Aestro, Blender, etc...



## Categories of Android applications



# Requirements for Android



## OS

- Android application development on either of the following operating systems:
  - Microsoft Windows.
  - Mac OS X 10.5.8 or later version with Intel chip.
  - Linux including GNU C Library 2.7 or later.

## Tools

- All the required tools to develop Android applications are freely available and can be downloaded from the Web. Following is the list of software's you will need before you start your Android application programming.
  1. Java JDK6 or later version
  2. Android Studio / Android SDK and Eclipse IDE for Java Developers (optional) and Android Development Tools (ADT) Eclipse Plug-in (optional)



# Requirements for Android

## Download Android Studio

- From <http://developer.android.com/sdk/installing/index.html>

## System Requirements for Windows

- Microsoft Windows 7/8/10 (32-bit or 64-bit)
- 3 GB RAM minimum, 8 GB RAM recommended (plus 1 GB for the Android Emulator)
- 2 GB of available disk space minimum, 4 GB recommended (500 MB for IDE plus 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution
- Java Development Kit (JDK) 8



# Java JDK

- The Java Development Kit (JDK) is a software development environment used for developing Java applications and applets.
- It includes the Java Runtime Environment (JRE), an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (javadoc) and other tools needed in Java development.

## JVM

- JVM (Java Virtual Machine) is a specification that provides runtime environment in which java bytecode can be executed.
- JVMs are available for many hardware and software platforms. JVM is a part of Java Run Environment (JRE).
- The JVM performs following operation:
  - Loads code
  - Verifies code
  - Executes code
  - Provides runtime environment

## JRE

The Java Runtime Environment (JRE) is a set of software tools for development of Java applications. It combines the Java Virtual Machine (JVM), platform core classes and supporting libraries.





# Android SDK

The Android SDK (software development kit) is a set of development tools used to develop applications for Android platform. The Android SDK includes the following:

- Required libraries
- Debugger
- An emulator
- Relevant documentation for the Android application program interfaces (APIs)
- Sample source code
- Tutorials for the Android OS



# Android Development Tools (ADT)

The android developer tools let you create interactive and powerful application for android platform.

## 1. Android Studio

Developed by Google, Android Studio is an all-rounder integrated development environment. Android has Gradle-base support that has features like visual layout editor, intelligent code editor, real-time profilers and APK analyzer.

## 2. Visual Studio – Xamarin

Xamarin was launched in 2011 which is the best free IDE for delivering an enterprise-quality, cross-platform approach. Xamarin supplies add-ins to Microsoft Visual Studio that allows developers to build Android, iOS, and Windows apps within the IDE

## 3. IntelliJ IDEA

Using this IDE, you can do in-depth coding, quick navigation, and error analysis. It supports mobile app development with the help of Java, Scala, Kotlin, Groovy.

## 4. Eclipse IDE

It is one of the most popular IDEs of Android apps. The open-source software is free to use. Released under the Eclipse Public License, it holds a large community having plenty of plugins and configurations. Highly customizable offers full support for Java programming language and XML.

# Android Virtual Devices (AVDs)

An Android Virtual Device (AVD) is a configuration that defines the characteristics of an Android phone, tablet, Wear OS, Android TV, or Automotive OS device that you want to simulate in the Android Emulator.

The AVD Manager is an interface you can launch from Android Studio that helps you create and manage AVDs.

To open the AVD Manager, do one of the following:


Select Tools > AVD Manager.




















Click AVD Manager  in the toolbar.


# Create a new AVD

1. Open the AVD Manager by clicking **Tools > AVD Manager**.


Android Virtual Device Manager

Your Virtual Devices  
Android Studio

Type	Name ▲	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	10.1 WXGA (Tablet) ...		800 × 1280: mdpi	28	Android 9.0 (Google APIs)	x86	513 MB	  
	Android TV (1080p) ...		1920 × 1080: xhdpi	28	Android 9.0 (Android TV)	x86	513 MB	  
	Android Wear Round ...		320 × 320: hdpi	28	Android 9.0 (Wear OS)	x86	513 MB	  
	Automotive (1024p la...		1024 × 768: mdpi	28	Android 9.0 (Automotive)	x86	513 MB	  
	Pixel 3 API 28		1080 × 2160: 440dpi	28	Android 9.0 (Google Play)	x86	8.1 GB	  

 ?

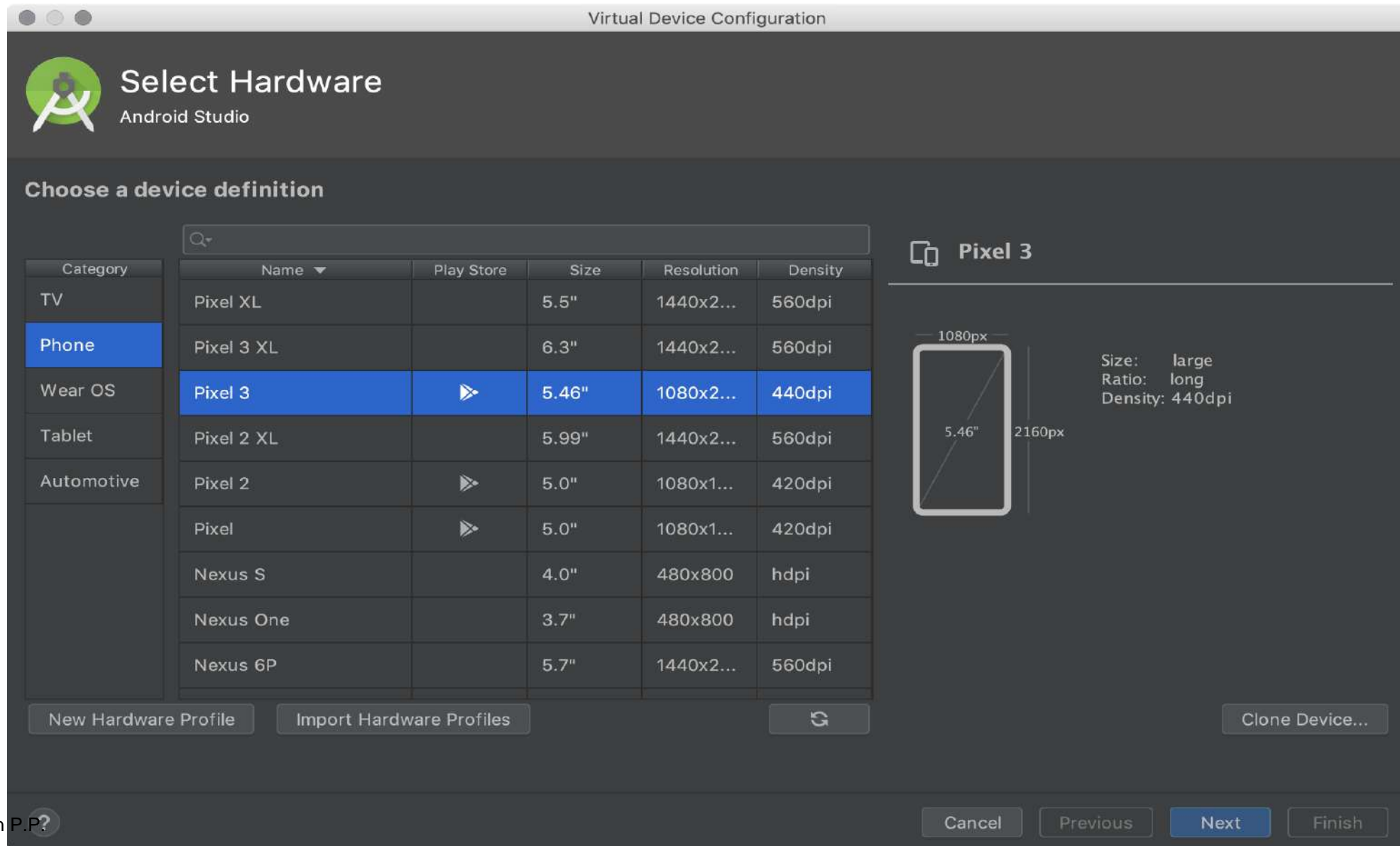
+ Create Virtual Device...



Mrs. Chavan P.P.

# Create a new AVD


2. Click Create Virtual Device, at the bottom of the AVD Manager dialog. The Select Hardware page appears.



# Create a new AVD

3. Select a hardware profile, and then click Next. If you don't see the hardware profile you want, you can [create](#) or [import](#) a hardware profile. The **System Image** page appears.

Virtual Device Configuration



System Image  
Android Studio

Select a system image


Recommended

x86 Images

Other Images

Release Name	API Level	ABI	Target
<a href="#">Q Download</a>	Q	x86	Android 9.+ (Google Play)
<b>Pie</b>	28	x86	<b>Android 9.0 (Google Play)</b>
<a href="#">Oreo Download</a>	27	x86	Android 8.1 (Google Play)
<a href="#">Oreo Download</a>	26	x86	Android 8.0 (Google Play)
<a href="#">Nougat Download</a>	25	x86	Android 7.1.1 (Google Play)
<a href="#">Nougat Download</a>	24	x86	Android 7.0 (Google Play)

Pie



API Level

28

Android

9.0

Google Inc.

System Image

x86

We recommend these Google Play images because this device is compatible with Google Play.

Questions on API level?  
[See the API level distribution chart](#)

? Mrs. Chavan P.P.

Cancel

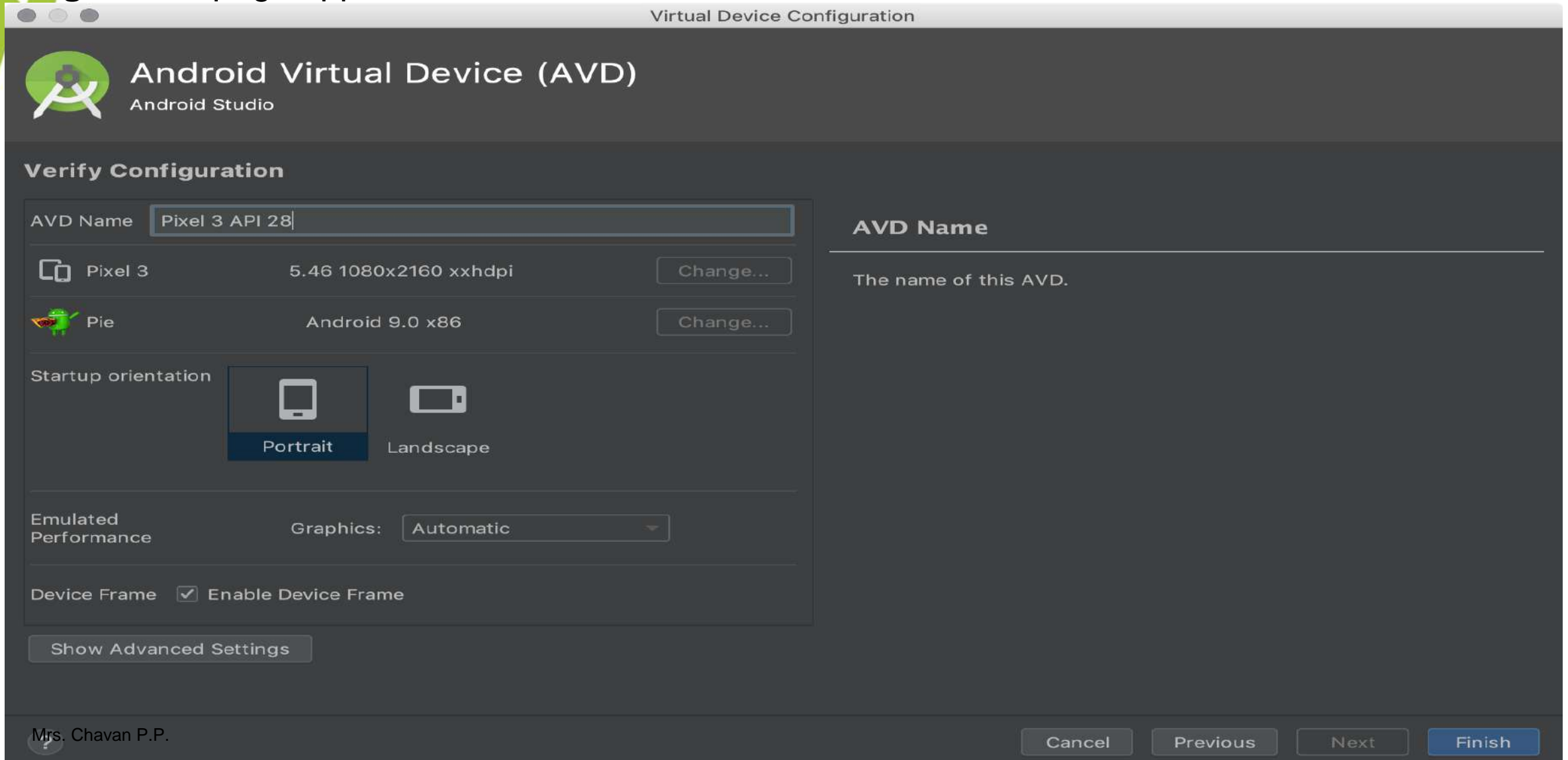
Previous

Next

Finish

# Create a new AVD

4. Select the system image for a particular API level, and then click **Next**. The **Verify Configuration** page appears.



Virtual Device Configuration

Android Virtual Device (AVD)  
Android Studio

**Verify Configuration**

AVD Name: Pixel 3 API 28

Device	API Level	Resolution	Action
Pixel 3	5.46	1080x2160 xxhdpi	Change...
Pie	Android 9.0	x86	Change...

Startup orientation: Portrait (selected), Landscape

Emulated Performance: Graphics: Automatic

Device Frame: ☒ Enable Device Frame

Show Advanced Settings

AVD Name: The name of this AVD.

Mrs. Chavan P.P.

Cancel Previous Next Finish



## Create a new AVD

5. Change AVD properties as needed, and then click **Finish**.

**Now you get a new AVD ready for launching your apps on it.**



# Emulators

- The Android Emulator simulates Android devices on your computer so that you can test your application on a variety of devices and Android API levels without needing to have each physical device.
- The emulator provides almost all of the capabilities of a real Android device. You can simulate incoming phone calls and text messages, specify the location of the device, simulate different network speeds, simulate rotation and other hardware sensors, access the Google Play Store, and much more.
- The emulator comes with predefined configurations for various Android phone, tablet, Wear OS, and Android TV devices.
- In short, An Android emulator is an Android Virtual Device (AVD) that represents a specific Android device. You can use an Android emulator as a target platform to run and test your Android applications on your PC. Using Android emulators is optional.

## To start the emulator:

1. Open the AVD Manager.
2. Double-click an AVD, or click Run



The Android Emulator loads.

# Dalvik Virtual Machine (DVM)

- 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*.
- 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.

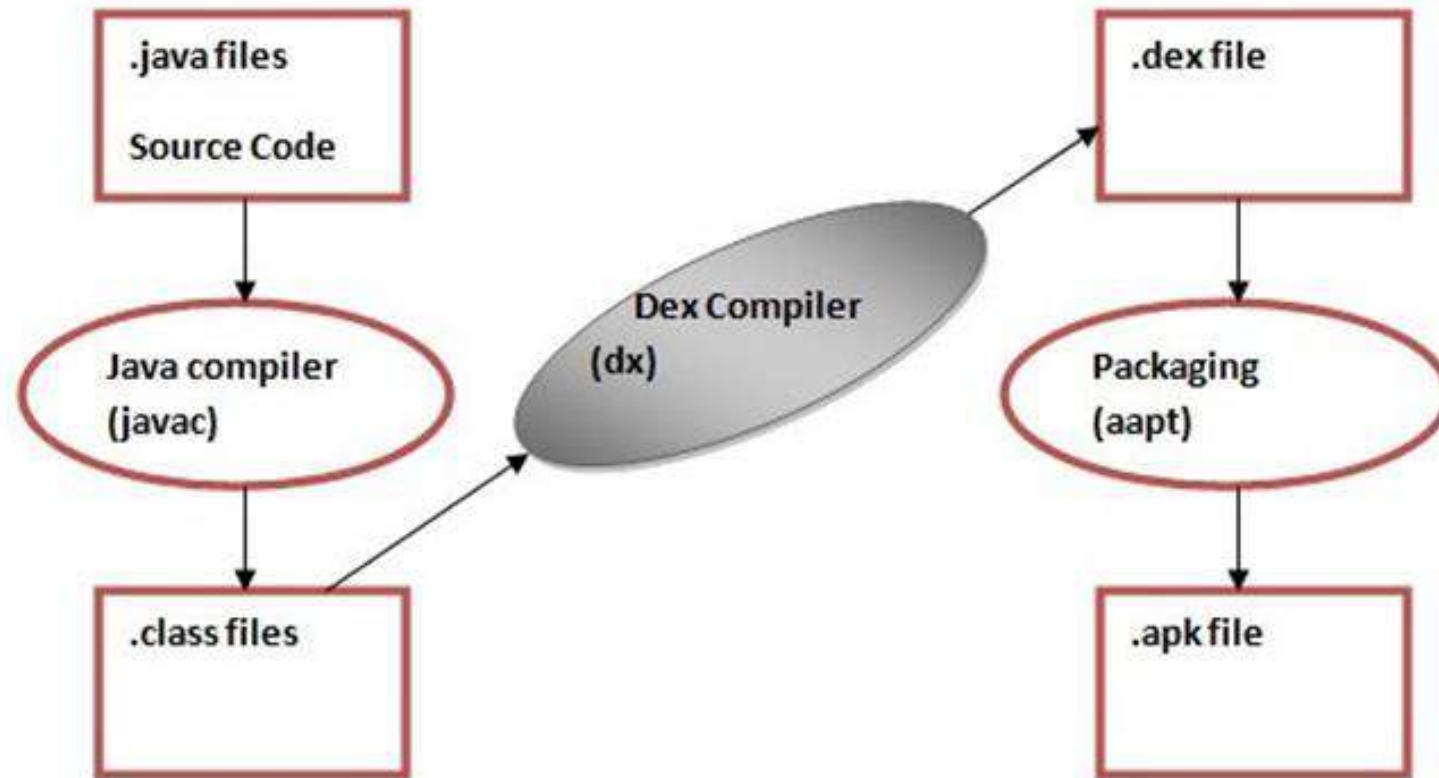


Fig.: The compiling and packaging process from the source file



# **Dalvik Virtual Machine (DVM)**

- 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.

# Difference between JVM and DVM

DVM (Dalvik Virtual Machine)	JVM (Java Virtual Machine)
It is Register based which is designed to run on low memory.	It is Stack based.
DVM uses its own byte code and runs “.Dex” file. From Android 2.2 SDK Dalvik has got a Just in Time compiler	JVM uses java byte code and runs “.class” file having JIT (Just In Time).
DVM has been designed so that a device can run multiple instances of the VM efficiently. Applications are given their own instance.	Single instance of JVM is shared with multiple applications.
DVM supports Android operating system only.	JVM supports multiple operating systems.
For DVM very few Re-tools are available.	For JVM many Re-tools are available.
There is constant pool for every application.	It has constant pool for every class.
Here the executable is APK.	Here the executable is JAR.

# Installation



## Java

1. Visit <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
2. Install it.

## Android Studio

3. Visit <http://developer.android.com/sdk/index.html>
4. click the button *Download Android Studio*.
5. Accept terms, and click *Download*.
6. Run executable file of setup.
7. Follow the setup wizard to install Android Studio and any necessary SDK tools.
8. On some Windows systems, the launcher script does not find where Java is installed. If you encounter this problem, you need to set an environment variable indicating the correct location.
9. Select Start menu > Computer > System Properties > Advanced System Properties. Then open Advanced tab > Environment Variables and add a new system variable JAVA\_HOME that points to your JDK folder, for example C:\Program Files\Java\jdk1.7.0\_45



# Installation

10. The individual tools and other SDK packages are saved outside the Android Studio application directory. If you need to access the tools directly, use a terminal to navigate to the location where they are installed. For example:
11. `\Users\<user>\sdk\`
12. Android Studio is now ready and loaded with the Android developer tools, but there are still a couple packages you should add to make your Android SDK complete.

## Run

1. Run Android Studio as Administrator.
2. Before you create new project, click *Configure* from splash screen. Click *SDK Manager*.
3. Don't select all. In bottom, in *Extra* section, select *Intel x86 Emulator Accelerator*.
4. Click *Install* button.

# Installation

launched android-studio-ide-181.5056338-windows.exe to start the installation process. The installer responded by presenting the **Android Studio Setup** dialog box shown in Figure 1.

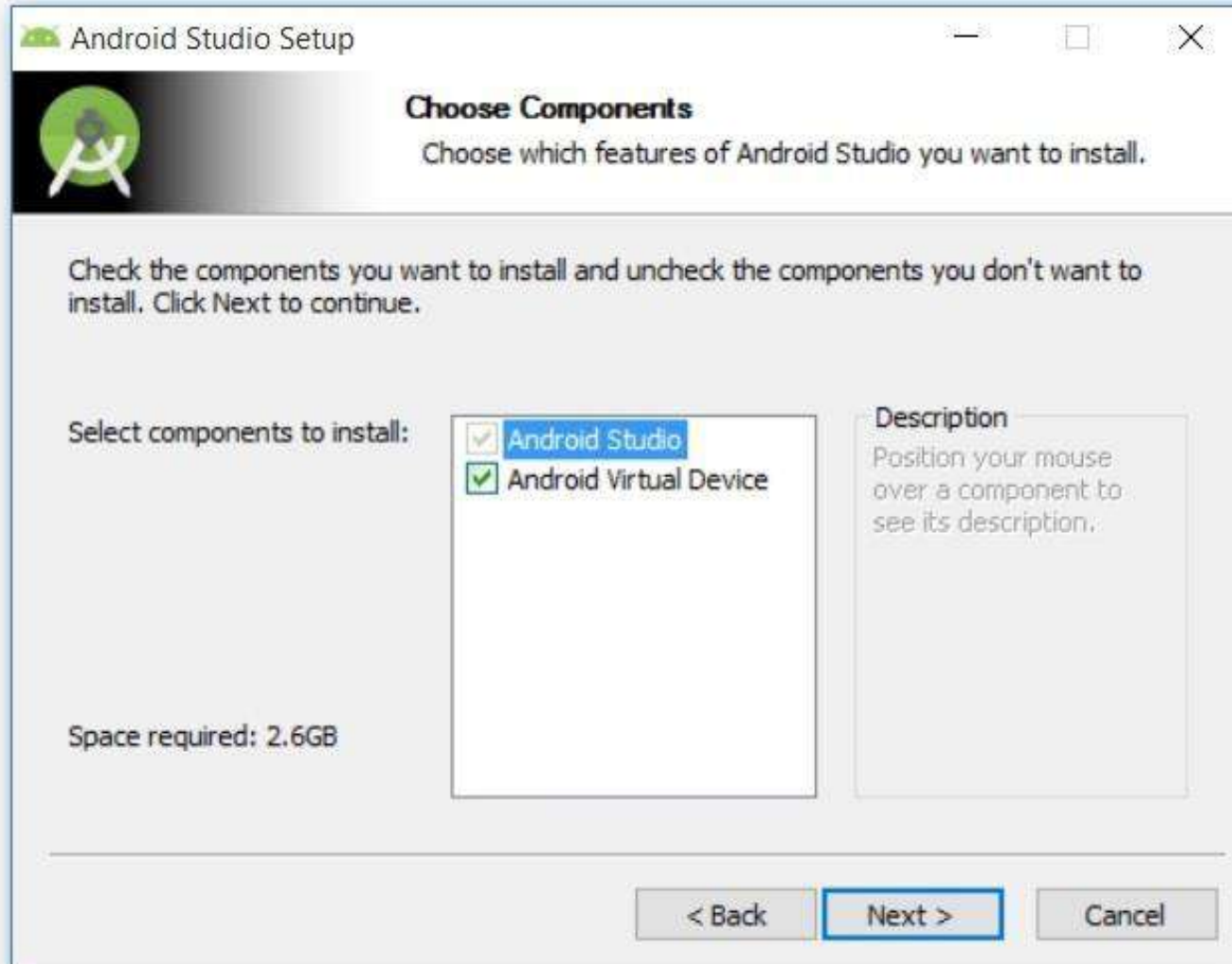


## 1. Set up Android Studio

Clicking **Next** will take you to the following panel, which provides the option to decline installing an Android Virtual Device (AVD).

# Installation

chose to keep the default settings.

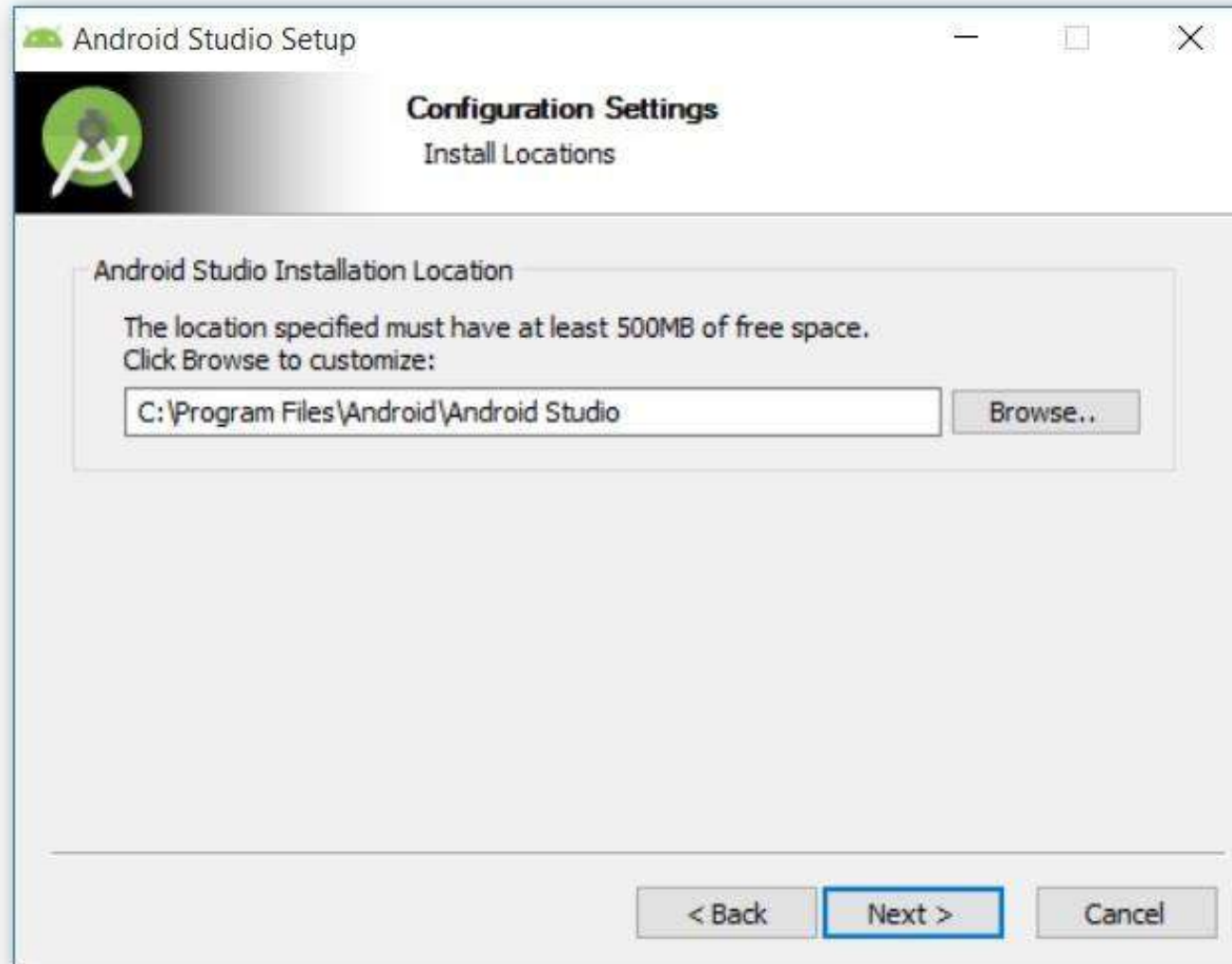


After clicking **Next**, the **Configuration Settings** panel will be shown,



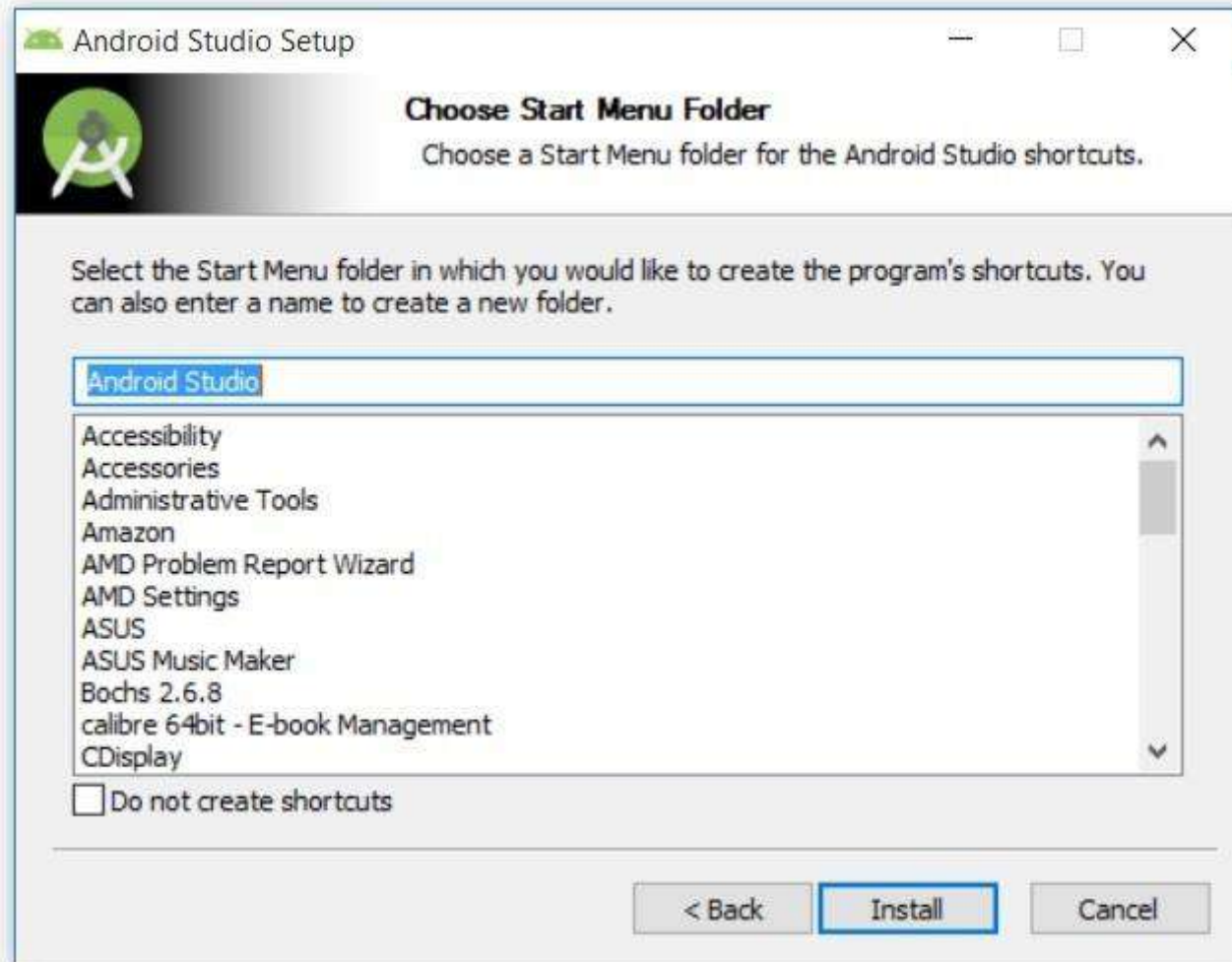
# Installation

where you will be asked to choose where to install Android Studio.



# Installation

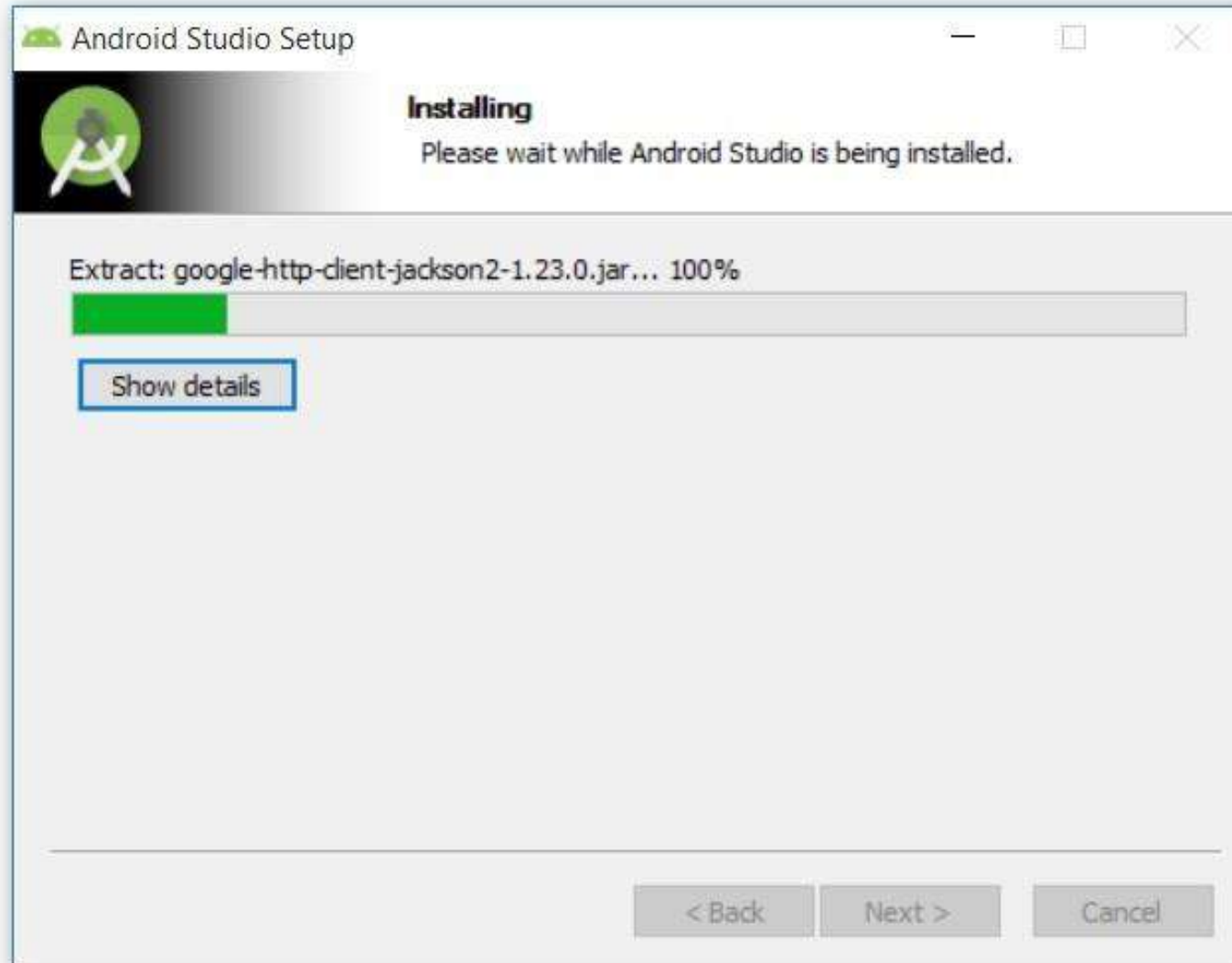
clicking **Next** will open **Choose Start Menu Folder** panel.



keep the default setting and click **Install**

# Installation

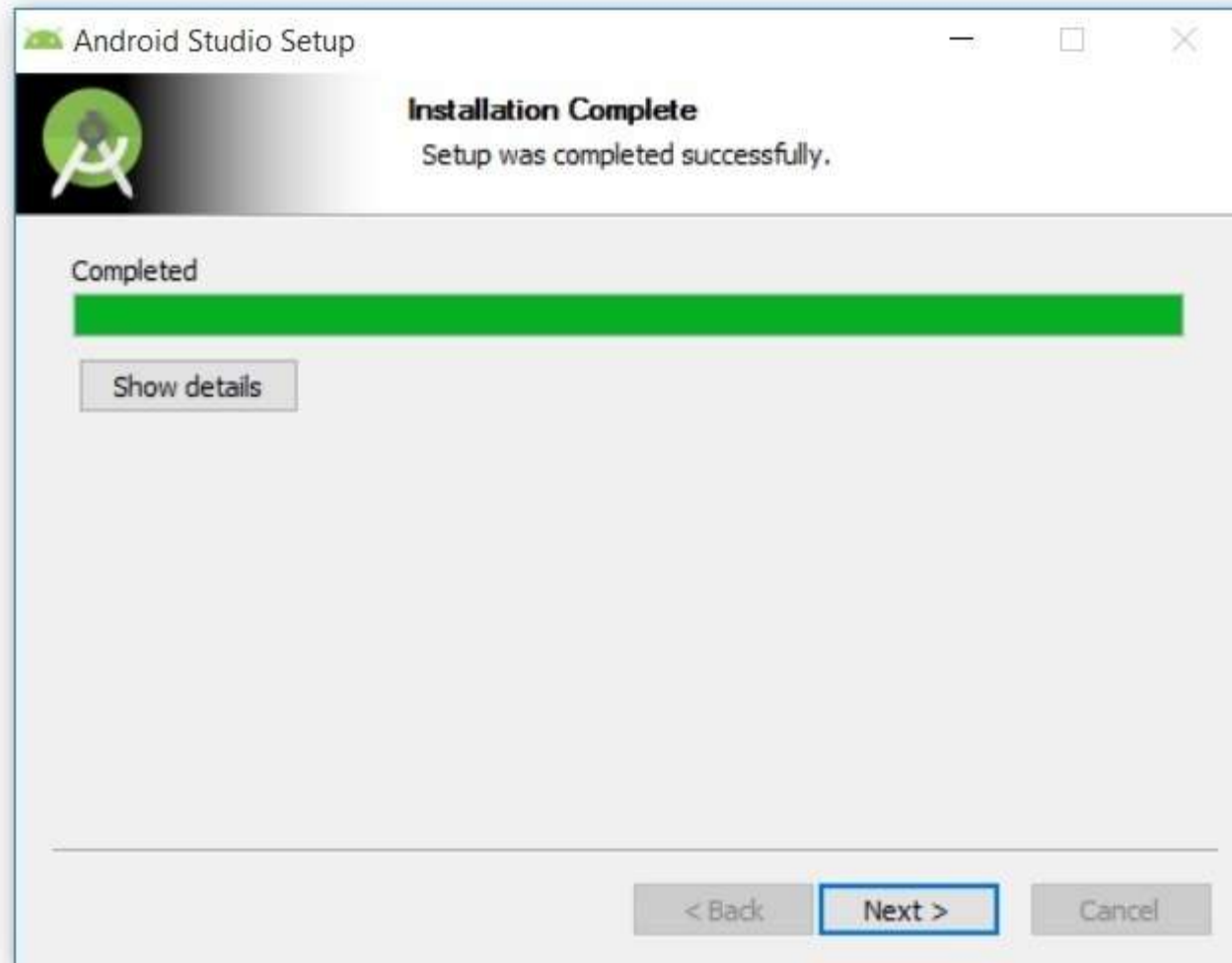
The following **Installing** panel will appear:



When installation finished, the **Installation Complete** panel appeared.

# Installation

When installation finished, the **Installation Complete** panel appeared.



# Installation

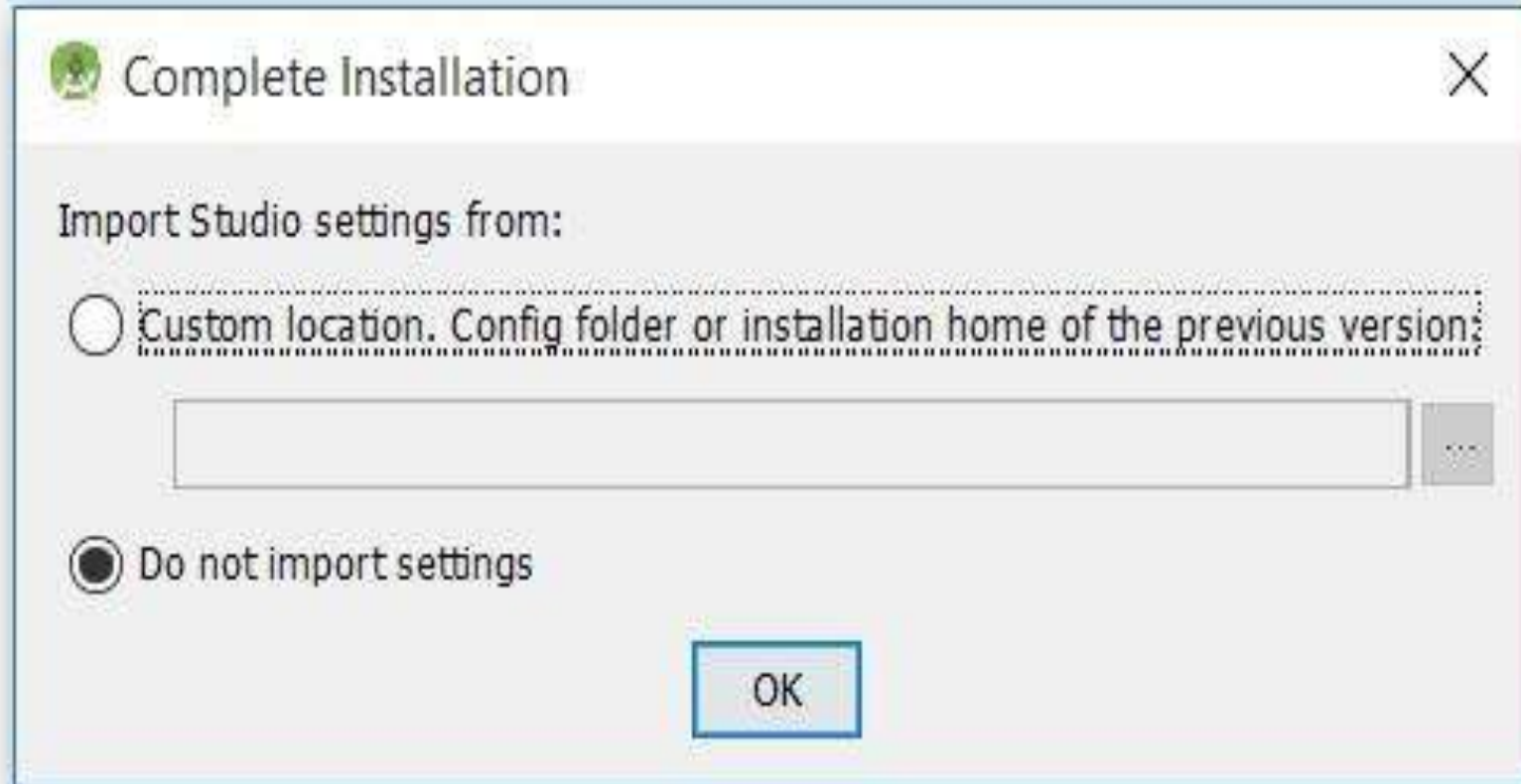
After clicking **Next**, the installer presented the **Completing Android Studio Setup** panel.



To complete the installation, leave **Start Android Studio** box checked and click **Finish**.

# Running Android Studio

The first time Android Studio runs, it presents a **Complete Installation** dialog box that offers the option of importing settings from a previous installation.



choose not to import settings (the default selection) and click **OK**, the following splash screen will appear:



# Running Android Studio

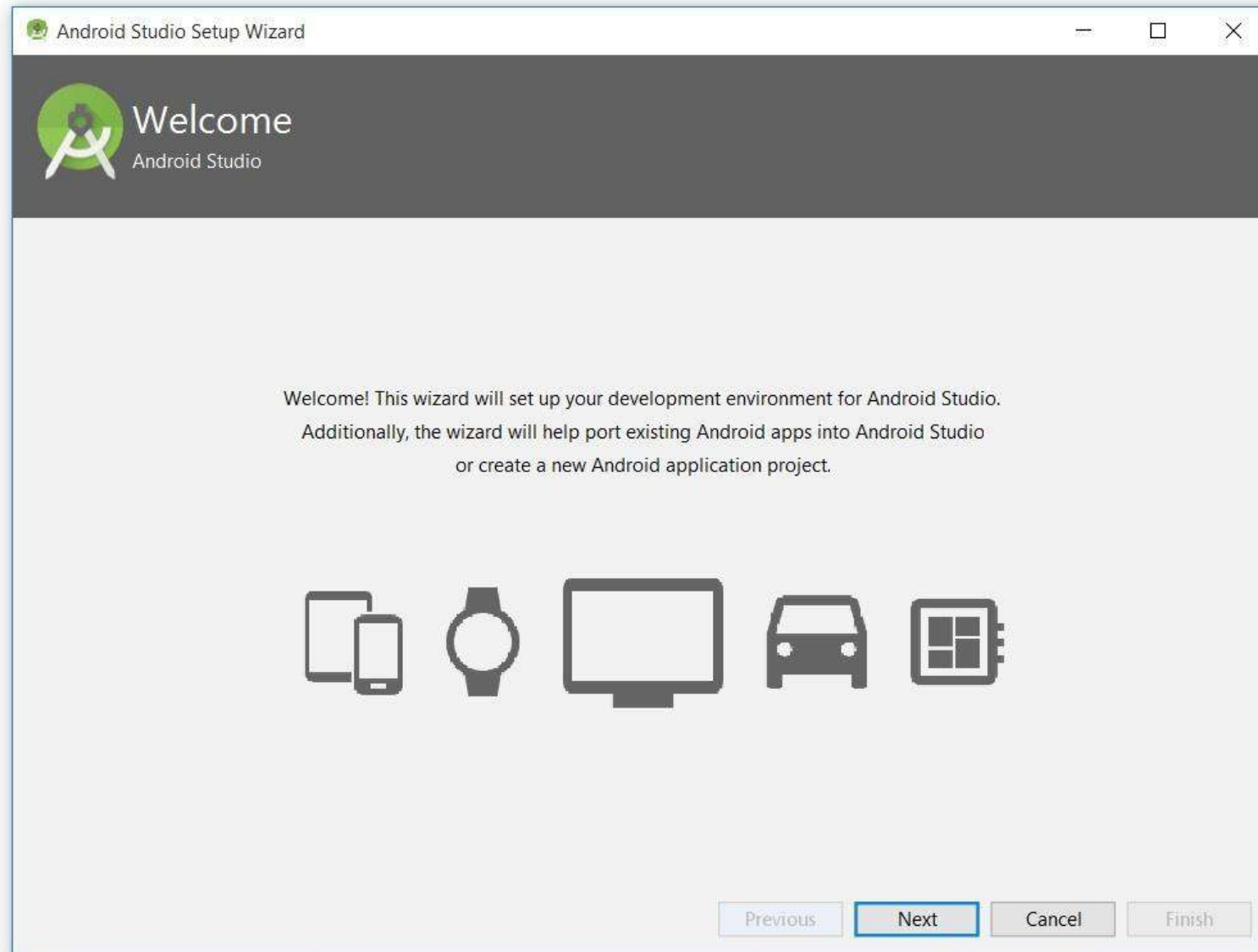


You will observe the following **Finding Available SDK Components** message box.



# Running Android Studio

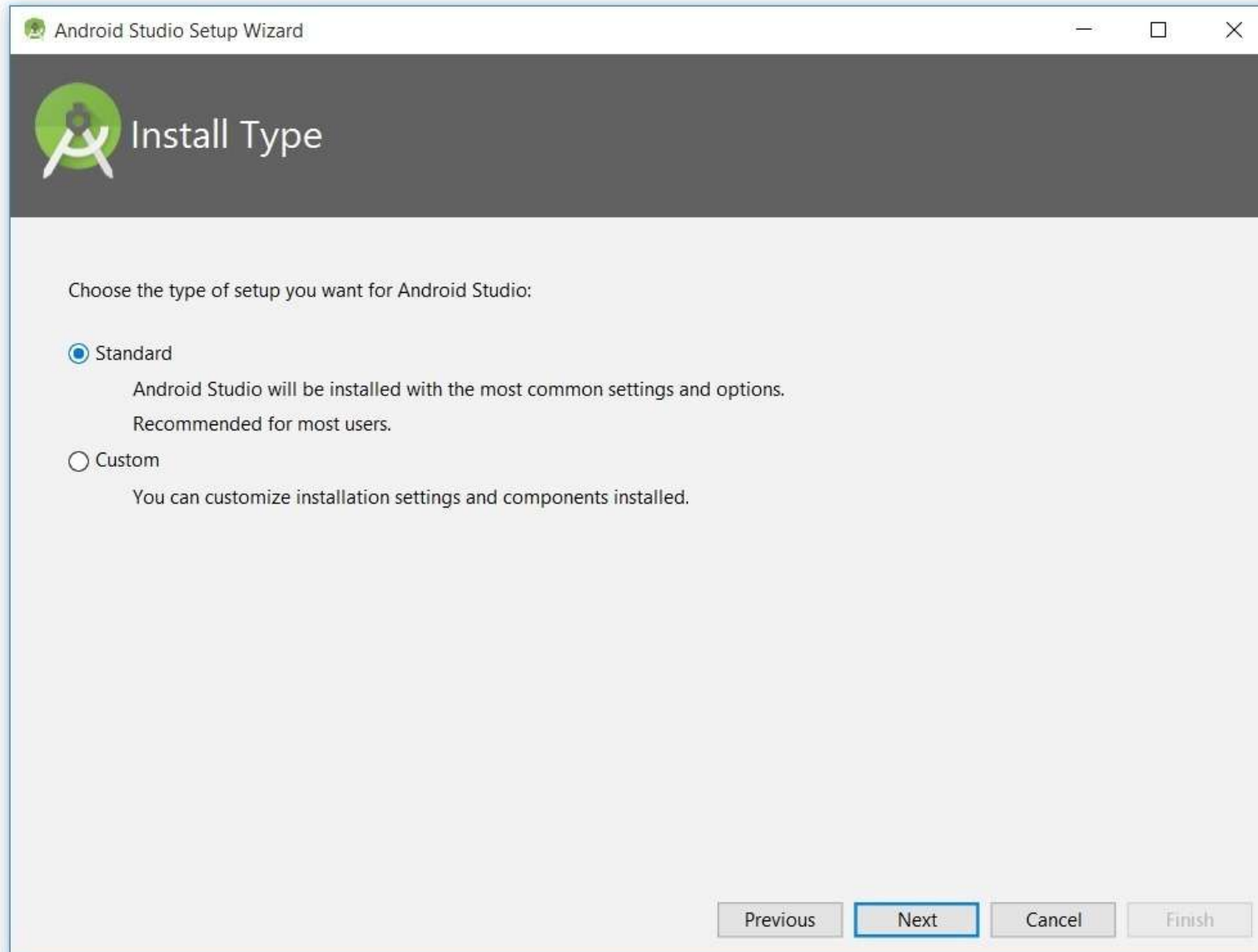
Android Studio will present the following **Android Studio Setup Wizard** dialog box:





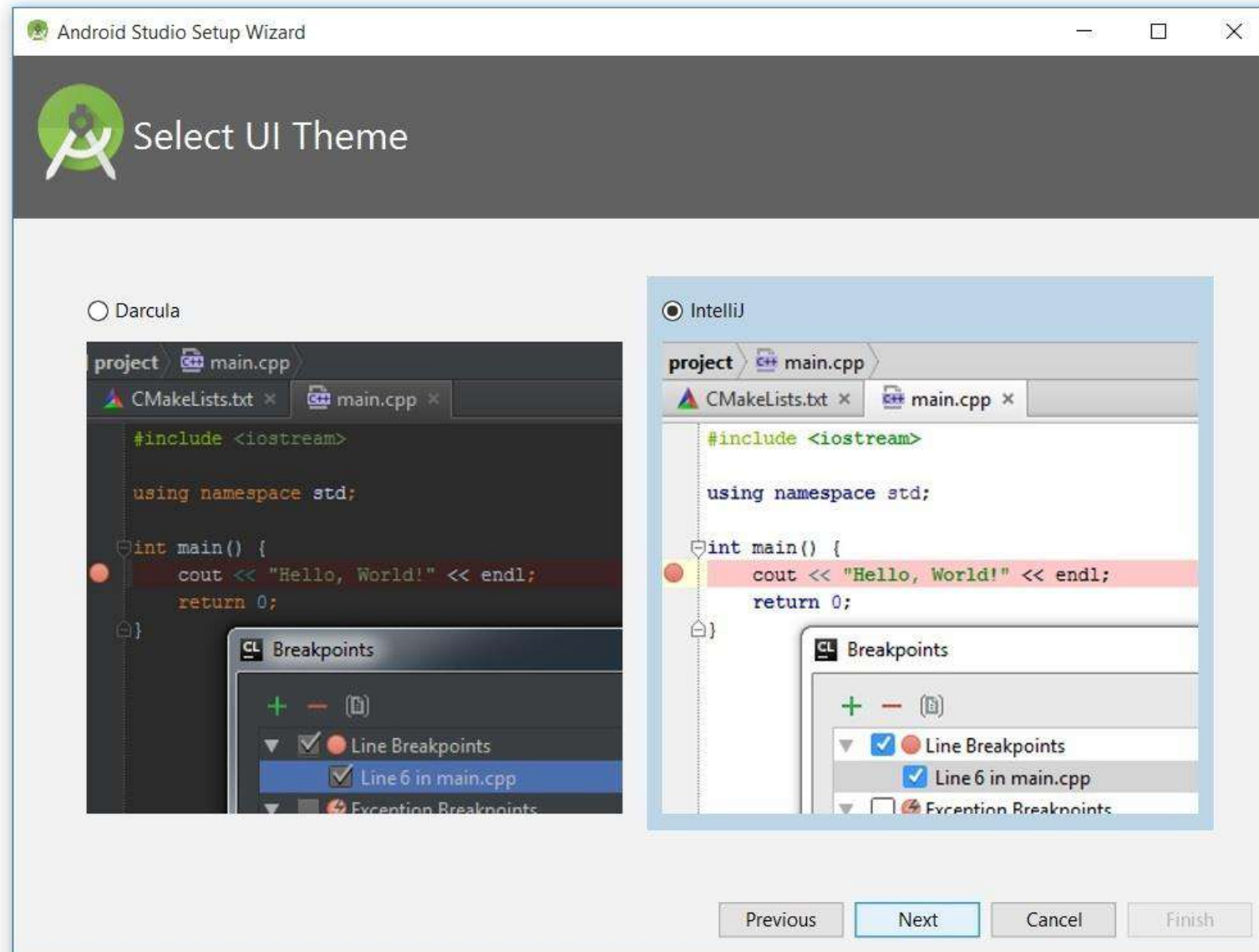
# Running Android Studio

Click **Next**, and the wizard will invite you to select an installation type. Keep the default standard setting.



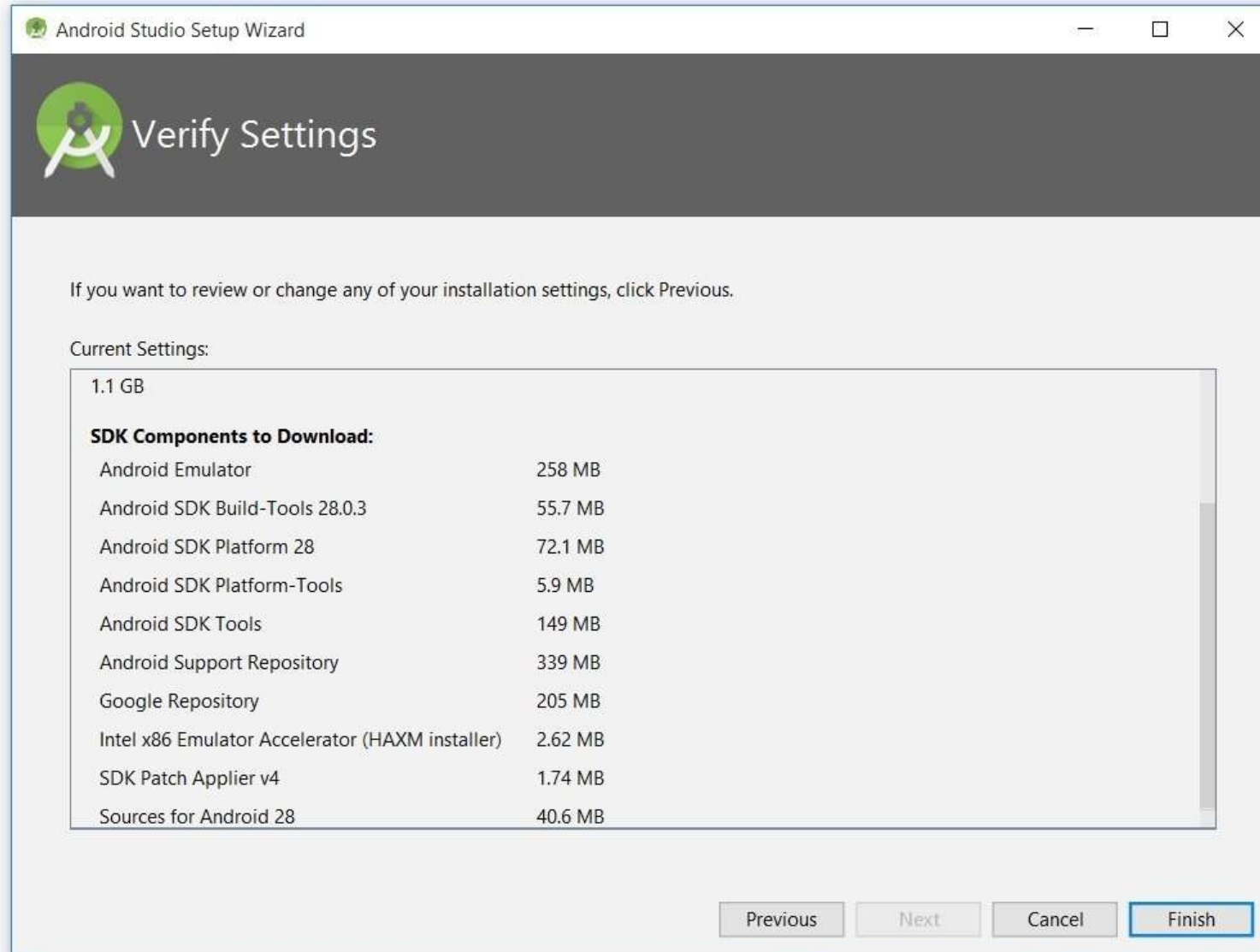
# Running Android Studio

Then you can choose a user interface theme.



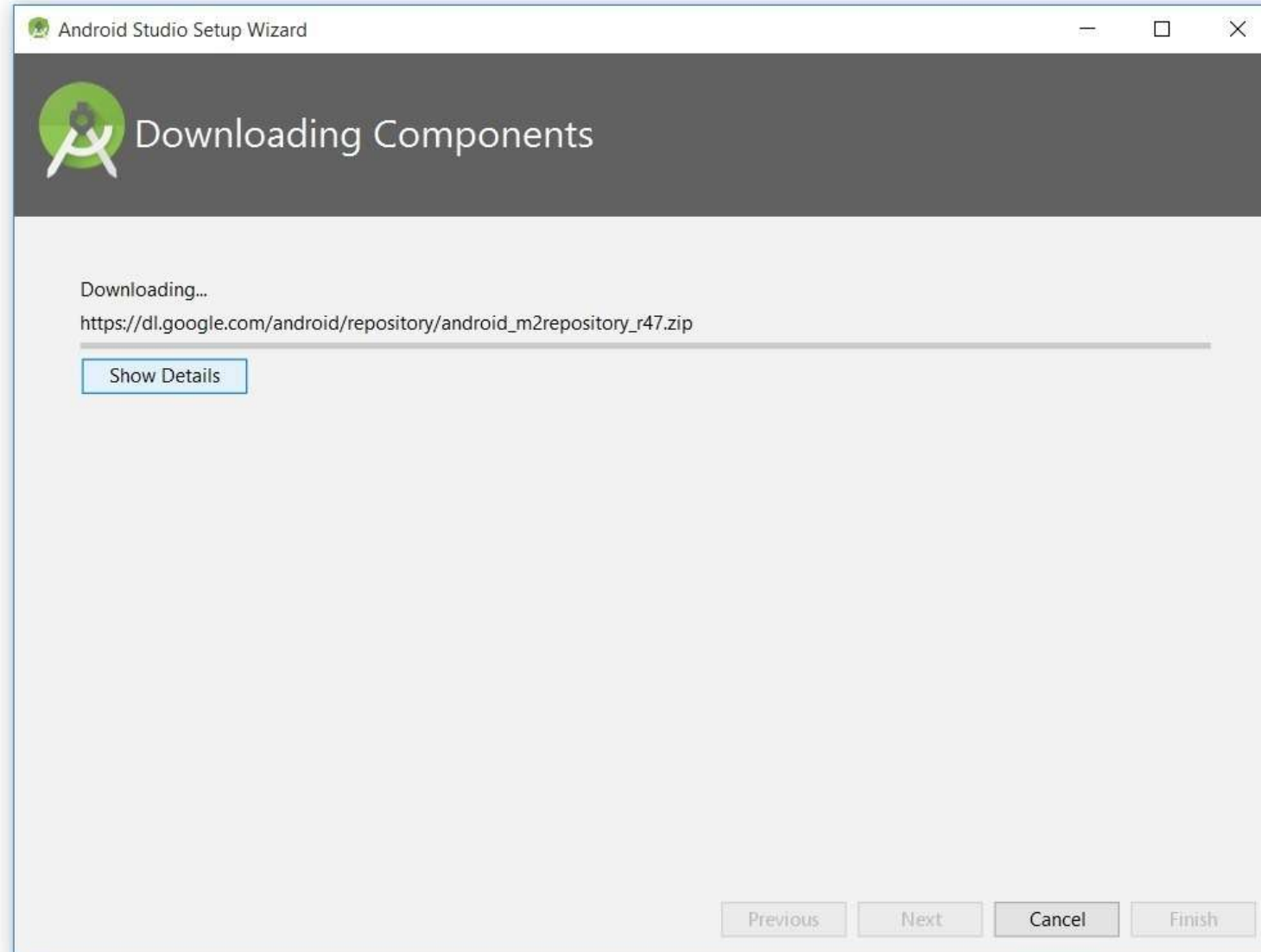
# Running Android Studio

You can choose whichever you want and click **Next**. Android Studio will next provide the opportunity to verify settings.



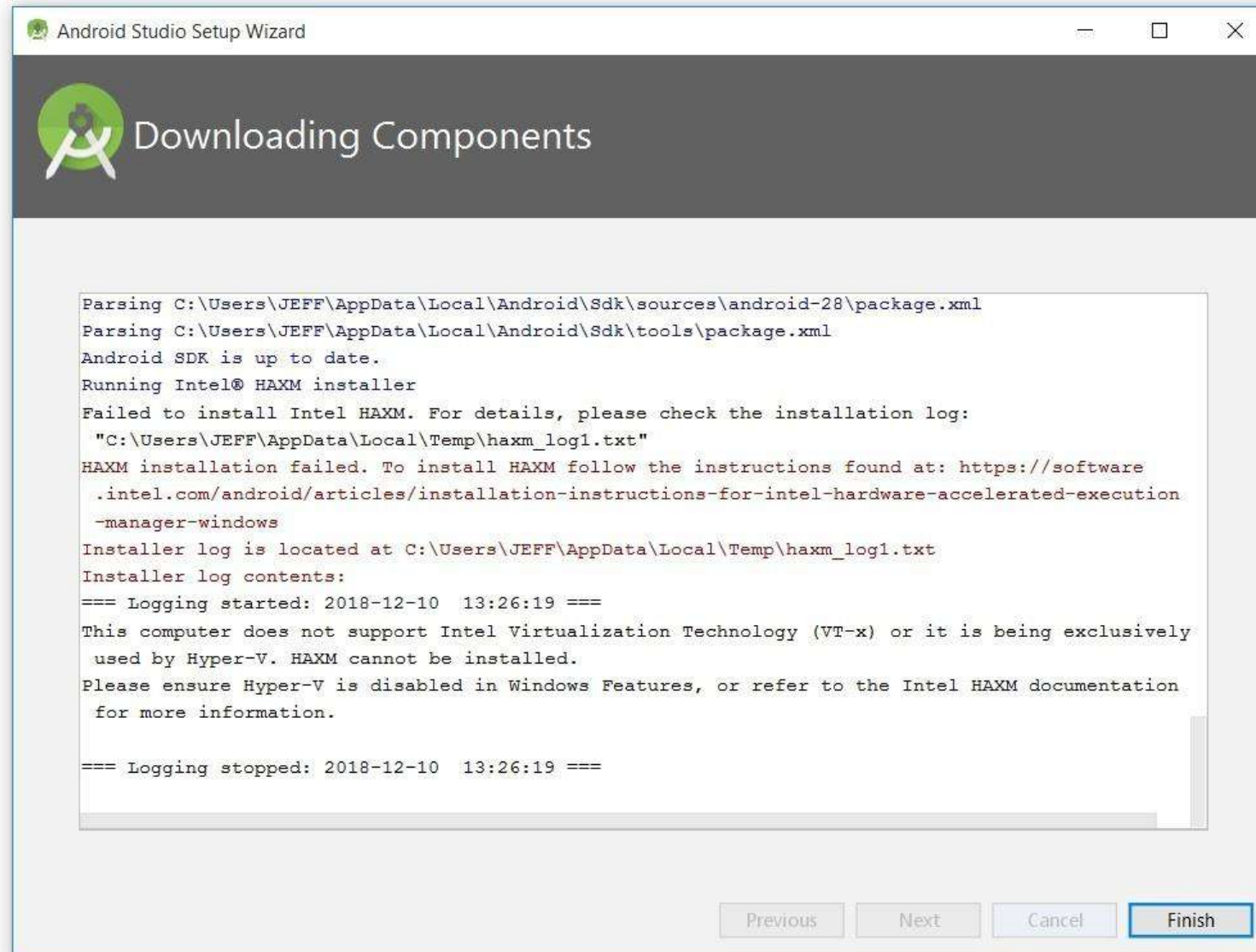
# Running Android Studio

click **Finish** and Android Studio will begin the process of downloading SDK components.



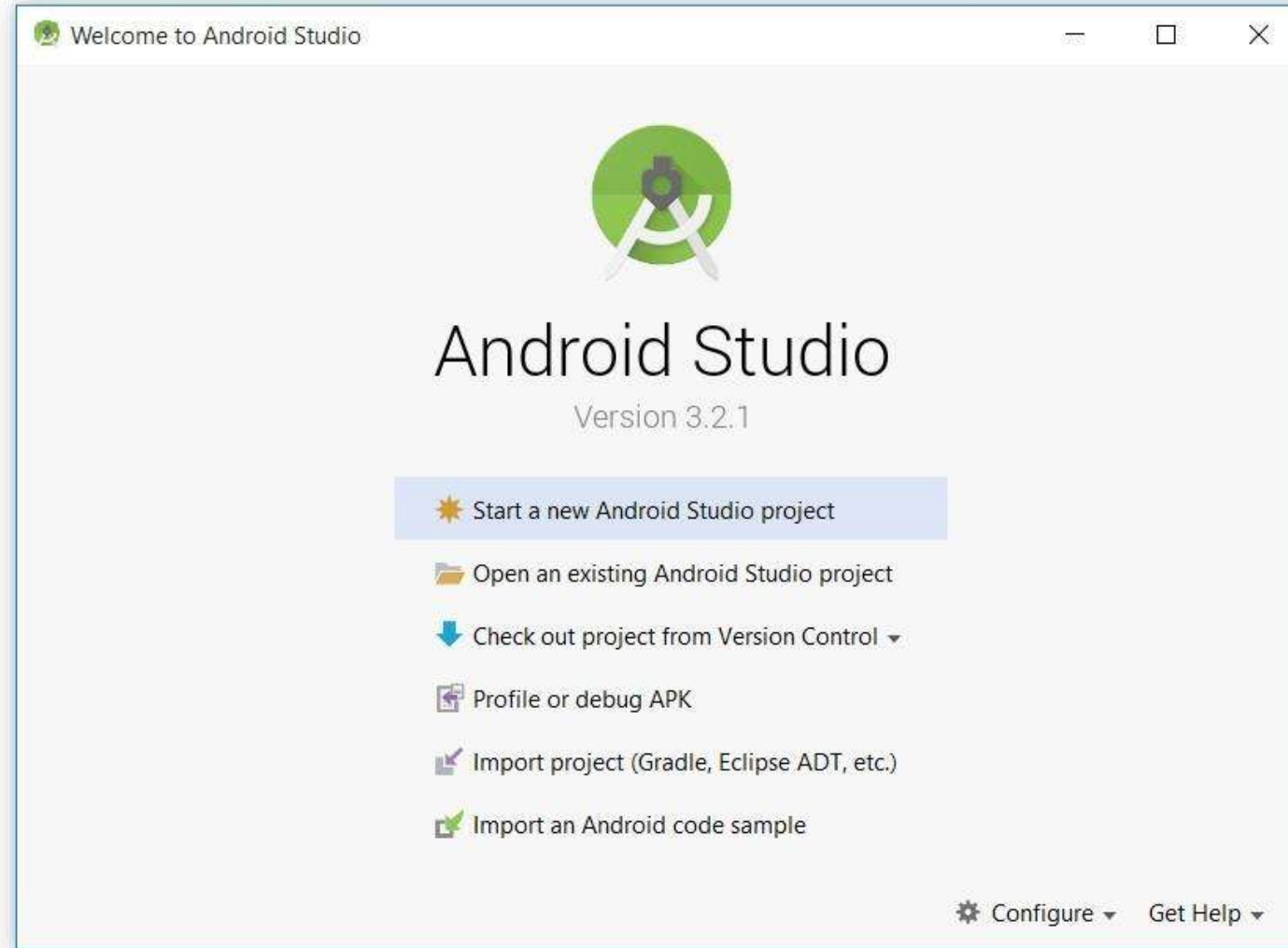
# Running Android Studio

It can take several minutes for this part of the setup to finish.



# Running Android Studio

Finally, click **Finish** to complete the wizard. The **Welcome to Android Studio** dialog box appear.



This dialog box is used to start up a new Android Studio project, work with an existing project, and more. It can be accessed by selecting **Android Studio** from the Windows **Start** menu, or the equivalent on another platform.

