**Lab Manual for Application for Mobile Devices**

**Lab No. 2**

# Introduction to AVD(Android Virtual Device )and ADB(Android Debug Bridge)

Objectives

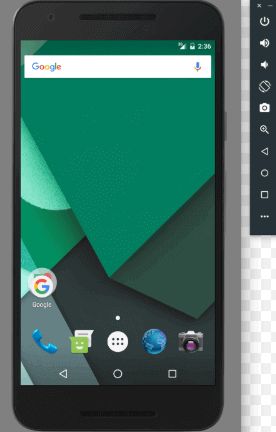
The purpose of this lab is to familiarize with basics of Android Virtual Device and Android Data Bridge

**LAB # 02**

**Introduction to AVD and ADB**

## **Android Emulator**

The **Android emulator** is an **Android Virtual Device (AVD),** which represents a specific Android device. We can use the Android emulator as a target device to execute and test our Android application on our PC. The Android emulator provides almost all the functionality of a real device. We can get the incoming phone calls and text messages. It also gives the location of the device and simulates different network speeds. Android emulator simulates rotation and other hardware sensors. It accesses the Google Play store, and much more



Testing Android applications on emulator are sometimes faster and easier than doing on a real device. For example, we can transfer data faster to the emulator than to a real device connected through USB.

The Android emulator comes with predefined configurations for several Android phones, Wear OS, tablet, Android TV devices.

## **Requirement and recommendations**

The Android emulator takes additional requirements beyond the basic system requirement for Android Studio. These requirements are given below

* SDK Tools 26.1.1 or higher
* 64-bit processor
* Windows: CPU with UG (unrestricted guest) support
* HAXM 6.2.1 or later (recommended HAXM 7.2.0 or later)

## **Install the emulator**

The Android emulator is installed while installing the Android Studio. However some components of emulator may or may not be installed while installing Android Studio. To install the emulator component, select the **Android Emulator** component in the **SDK Tools** tab of the **SDK Manager.**

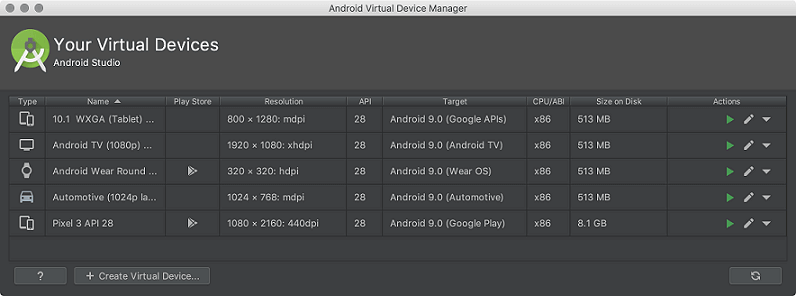
## **Run an Android app on the Emulator**

We can run an Android app form the Android Studio project, or we can run an app which is installed on the Android Emulator as we run any app on a device.

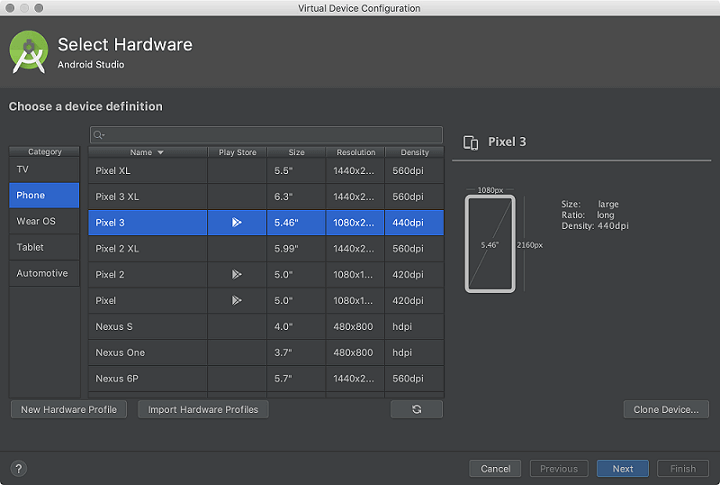
To start the Android Emulator and run an application in our project:

**1.** In **Android Studio,** we need to create an Android Virtual Device (AVD) that the emulator can use to install and run your app. To create a new AVD:-

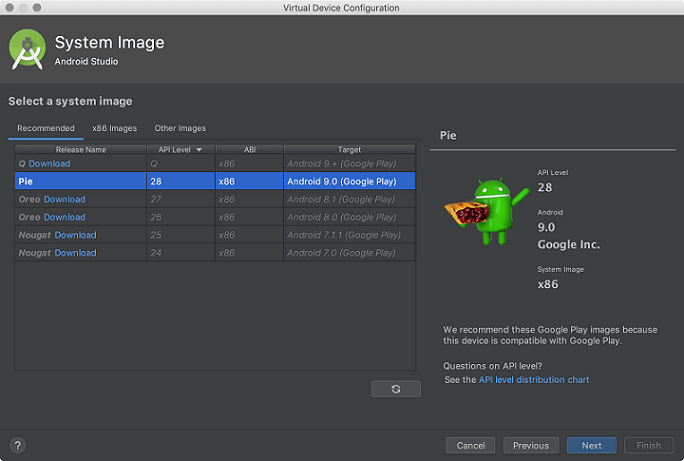
**1.1** Open the AVD Manager by clicking **Tools > AVD Manager.**



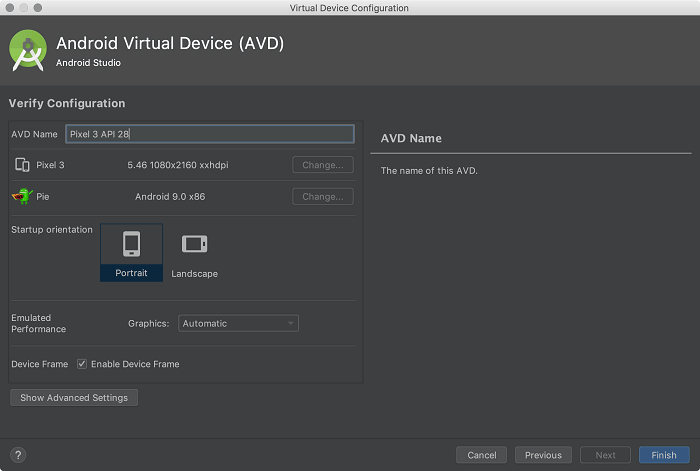
**1.2** Click on Create **Virtual** Device, at the bottom of the AVD Manager dialog. Then **Select Hardware** page appears



**1.3** Select a hardware profile and then click **Next.** If we don?t see the hardware profile we want, then we can create or import a hardware profile. The **System Image** page appears



**1.4** Select the system image for the particular API level and click **Next.** This leads to open a **Verify Configuration** page



**1.5** Change AVD properties if needed, and then click **Finish.**

**2.** In the toolbar, choose the AVD, which we want to run our app from the target device from the drop-down menu.

**3.** Click **Run.**

## **Enabling USB Debugging**

To communicate with an Android device over the Android Debug Bridge (adb), enable USB debugging on the device.

1. On the Android device, open the settings.
2. Tap **Developer Settings**. The developer settings are hidden by default. If the developer settings are not included in the settings menu of the device:
   1. Depending on whether the device is a phone or a pad, scroll down and tap **About phone** or **About Pad**.
   2. Scroll down again and tap **Build Number** seven times.
3. In the **Developer settings** window, check **USB-Debugging**.
4. Set the USB mode of the device to **Media device (MTP)**, which is the default setting. For additional information, refer to the documentation of the device.

# Introduction to ADB (Android Debug Bridge)

ADB (Android Debug Bridge) is a powerful command-line tool that allows developers and advanced users to interact with Android devices over a USB or Wi-Fi connection. Here's a tutorial on some of the most useful ADB commands:

1. adb devices: This command lists all the devices that are currently connected to your computer via USB or Wi-Fi.
2. adb shell: This command opens a shell on the target device, allowing you to execute commands directly on the device.
3. adb logcat: This command displays the device log in real-time. It's useful for debugging and troubleshooting.
4. adb push: This command pushes a file from your computer to the device.
5. adb pull: This command pulls a file from the device to your computer.
6. adb install: This command installs an APK (Android Package) file onto the device.
7. adb uninstall: This command uninstalls an app from the device.
8. adb reboot: This command reboots the device.
9. adb backup: This command creates a backup of the device's data and apps.
10. adb restore: This command restores a backup created using the adb backup command.
11. adb shell dumpsys: This command dumps system information about the device, such as running processes, battery usage, and more.
12. adb shell pm list packages: This command lists all the packages installed on the device.
13. adb shell am start -n com.package.name/.activity.name: This command starts an activity in an app. Replace "com.package.name" with the package name of the app and ".activity.name" with the name of the activity you want to start.
14. adb shell wm size: This command sets the screen resolution of the device.
15. adb shell input text "text": This command simulates typing on the device. Replace "text" with the text you want to type.

These are just a few of the many ADB commands available. Learning how to use ADB can greatly enhance your Android development and troubleshooting skills.

## **Time Boxing**

|  |  |  |
| --- | --- | --- |
| Activity Name | Activity Time | Total Time |
| Login Systems + Setting up android studio Environment | 3 mints + 5 mints | 8 mints |
| Walk through Theory & Tasks | 60 mints | 60 mints |
| Implement Tasks | 80 mints | 80 mints |
| Evaluation Time | 30 mints | 30 mints |
|  | Total Duration | 178 mints |

## **Objectives/Outcomes**

The purpose of this lab is to familiarize with basic of Android studio IDE

• Understanding Android Virtual Device

• Understanding USB Debugging and Android Data Bridge

## **Lab Tasks/Practical Work**

* Developing and running application.
  + Configuring Android Device to run application.
    - Android Virtual Device.
      * Creating, Starting, Stopping, Docking etc.
    - Using ADB
      * Enabling developer options.
      * Enabling USB debugging within developer options.
      * Listing attached devices from command prompt.
      * USB debugging authorizations on device.
      * Attaching device within network.
  + Running a template-based application on device.