

**(Flutter)**

Flutter Lab 1 - Session 1

**Total Time:**

**3 hours**

**Pre-Lab Activities:**

* No Pre-Lab Activity

**Learning Outcomes:**

* Perform the execution, debugging, testing, and profiling of mobile apps in modern IDEs.

**Lab Tasks:**

* + To install and configure Android Studio.
  + To install and configure the Emulators.
  + To explore the Android Studio functionalities.

**Student Activities:**

* + To explore Android Studio installation.
  + To explore the Emulators.
  + Android Studio functionalities.

**Lab Solutions:**

# Lab Tasks

# 1 – Install Android Studio.

**System Requirements**

* Microsoft Windows 7/8/10 (32-bit or 64-bit)
* 4 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

**Installation:** Perform the following steps to install the Android Studio.

**Step 1:**Head over to [**this link**](https://developer.android.com/studio/#downloads) to get the Android Studio executable or zip file.

**Step 2:**Click on the **Download Android Studio** Button.

Graphical user interface, text, application

Description automatically generated

Click on the “I have read and agree with the above terms and conditions” checkbox followed by the download button.

Graphical user interface, text, application, email

Description automatically generated

Click on the Save file button in the appeared prompt box and the file will start downloading.

**Step 3:**After the downloading has finished, open the file from downloads and run it. It will prompt the following dialog box.

Graphical user interface, text, application, chat or text message

Description automatically generated

Click on next. In the next prompt, it’ll ask for a path for installation. Choose a path and hit next.

**Step 4:**It will start the installation, and once it is completed, it will be like the image shown below.

Graphical user interface, text, application

Description automatically generated

Click on next.

Graphical user interface, text, application, chat or text message

Description automatically generated

**Step 5:**Once “**Finish**” is clicked, it will ask whether the previous settings need to be imported [if the android studio had been installed earlier], or not. It is better to choose the ‘Don’t import Settings option’.

Graphical user interface, text, application

Description automatically generated

Click the **OK**button.

**Step 6:**This will start the Android Studio.

Graphical user interface, application

Description automatically generated

Meanwhile, it will be finding the available SDK components.

Graphical user interface, text, application

Description automatically generated

**Step 7:**After it has found the SDK components, it will redirect to the Welcome dialog box.

Graphical user interface, application

Description automatically generated

Click on **Next**.

Graphical user interface, text, application

Description automatically generated

Choose Standard and click on Next. Now choose the theme, whether the **Light**theme or the **Dark**one. The light one is called the **IntelliJ**theme whereas the dark theme is called **Darcula**. Choose as required.

Graphical user interface, text, application

Description automatically generated

Click on the **Next**button.

**Step 8:**Now it is time to download the SDK components.

Table

Description automatically generated

Click on Finish. Components begin to download let it complete.

Text

Description automatically generated

The Android Studio has been successfully configured. Now it’s time to launch and build apps. Click on the Finish button to launch it.

**Step 9:**Click on **Start a new Android Studio project** to build a new app.

Graphical user interface, application

Description automatically generated

# 2 – Create and manage virtual devices

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](https://developer.android.com/studio/run/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** AVD Manager icon in the toolbar.

A picture containing text

Description automatically generated

## About AVDs

An AVD contains a hardware profile, system image, storage area, skin, and other properties.

We recommend that you create an AVD for each system image that your app could potentially support based on the [<uses-sdk>](https://developer.android.com/guide/topics/manifest/uses-sdk-element) setting in your manifest.

### Hardware profile

The hardware profile defines the characteristics of a device as shipped from the factory. The AVD Manager comes preloaded with certain hardware profiles, such as Pixel devices, and you can define or customize the hardware profiles as needed.

Notice that only some hardware profiles are indicated to include **Play Store**. This indicates that these profiles are fully [CTS](https://source.android.com/compatibility/cts/) compliant and may use system images that include the Play Store app.

### System images

A system image labeled with **Google APIs** includes access to [Google Play services](https://developers.google.com/android/guides/overview). A system image labeled with the Google Play logo in the **Play Store** column includes the Google Play Store app and access to Google Play services, including a **Google Play** tab in the **Extended controls** dialog that provides a convenient button for updating Google Play services on the device.

To ensure app security and a consistent experience with physical devices, system images with the Google Play Store included are signed with a release key, which means that you cannot get elevated privileges (root) with these images. If you require elevated privileges (root) to aid with your app troubleshooting, you can use the Android Open Source Project (AOSP) system images that do not include Google apps or services.

### Storage area

The AVD has a dedicated storage area on your development machine. It stores the device user data, such as installed apps and settings, as well as an emulated SD card. If needed, you can use the AVD Manager to wipe user data, so the device has the same data as if it were new.

### Skin

An emulator skin specifies the appearance of a device. The AVD Manager provides some predefined skins. You can also define your own, or use skins provided by third parties.

### AVD and app features

Be sure your AVD definition includes the device features your app depends on. See [Hardware Profile Properties](https://developer.android.com/studio/run/managing-avds#hpproperties) and [AVD Properties](https://developer.android.com/studio/run/managing-avds#avdproperties) for lists of features you can define in your AVDs.

## Create an AVD

**Tip:** If you want to launch your app into an emulator, instead [run your app from Android Studio](https://developer.android.com/studio/run#RunningApp) and then in the **Select Deployment Target** dialog that appears, click **Create New Virtual Device**.

**To create a new AVD:**

1. Open the AVD Manager by clicking **Tools > AVD Manager**. A picture containing text

   Description automatically generated
2. Click **Create Virtual Device**, at the bottom of the AVD Manager dialog.

The **Select Hardware** page appears.

Graphical user interface

Description automatically generated

Notice that only some hardware profiles are indicated to include **Play Store**. This indicates that these profiles are fully [CTS](https://source.android.com/compatibility/cts/) compliant and may use system images that include the Play Store app.

1. Select a hardware profile, and then click **Next**.

If you don't see the hardware profile you want, you can [create](https://developer.android.com/studio/run/managing-avds#createhp) or [import](https://developer.android.com/studio/run/managing-avds#importexporthp) a hardware profile.

The **System Image** page appears.

Graphical user interface, application

Description automatically generated

1. Select the system image for a particular API level, and then click **Next**.

The **Recommended** tab lists recommended system images. The other tabs include a more complete list. The right pane describes the selected system image. x86 images run the fastest in the emulator.

If you see **Download** next to the system image, you need to click it to download the system image. You must be connected to the internet to download it.

The API level of the target device is important, because your app won't be able to run on a system image with an API level that's less than that required by your app, as specified in the [minSdkVersion](https://developer.android.com/guide/topics/manifest/uses-sdk-element) attribute of the app manifest file. For more information about the relationship between system API level and minSdkVersion, see [Versioning Your Apps](https://developer.android.com/studio/publish/versioning).

If your app declares a [<uses-library>](https://developer.android.com/guide/topics/manifest/uses-library-element) element in the manifest file, the app requires a system image in which that external library is present. If you want to run your app on an emulator, create an AVD that includes the required library. To do so, you might need to use an add-on component for the AVD platform; for example, the Google APIs add-on contains the Google Maps library.

The **Verify Configuration** page appears.

Graphical user interface

Description automatically generated

1. Change [AVD properties](https://developer.android.com/studio/run/managing-avds#avdproperties) as needed, and then click **Finish**.

Click **Show Advanced Settings** to show more settings, such as the skin.

The new AVD appears in the **Your Virtual Devices** page or the **Select Deployment Target** dialog.

To create an AVD starting with a copy:

1. From the [**Your Virtual Devices**](https://developer.android.com/studio/run/managing-avds#viewing) page of the AVD Manager, right-click an AVD and select **Duplicate**.

Or click Menu  and select **Duplicate**.

The [**Verify Configuration**](https://developer.android.com/studio/run/managing-avds#verifyconfigpage) page appears.

1. Click **Change** or **Previous** if you need to make changes on the [**System Image**](https://developer.android.com/studio/run/managing-avds#systemimagepage) and [**Select Hardware**](https://developer.android.com/studio/run/managing-avds#selecthardwarepage) pages.
2. Make your changes, and then click **Finish**.

The AVD appears in the **Your Virtual Devices** page.

## Create a hardware profile

The AVD Manager provides predefined hardware profiles for common devices so you can easily add them to your AVD definitions. If you need to define a different device, you can create a new hardware profile. You can define a new hardware profile from the beginning, or [copy a hardware profile](https://developer.android.com/studio/run/managing-avds#copyavd) as a start. The preloaded hardware profiles aren't editable.

To create a new hardware profile from the beginning:

1. In the [**Select Hardware**](https://developer.android.com/studio/run/managing-avds#selecthardwarepage) page, click **New Hardware Profile**.
2. In the **Configure Hardware Profile** page, change the [hardware profile properties](https://developer.android.com/studio/run/managing-avds#hpproperties) as needed.
3. Click **Finish**.

Your new hardware profile appears in the **Select Hardware** page. You can optionally [create an AVD](https://developer.android.com/studio/run/managing-avds#createavd) that uses the hardware profile by clicking **Next**. Or, click **Cancel** to return to the **Your Virtual Devices** page or **Select Deployment Target** dialog.

To create a hardware profile starting with a copy:

1. In the [**Select Hardware**](https://developer.android.com/studio/run/managing-avds#selecthardwarepage) page, select a hardware profile and click **Clone Device**.

Or right-click a hardware profile and select **Clone**.

1. In the **Configure Hardware Profile** page, change the [hardware profile properties](https://developer.android.com/studio/run/managing-avds#hpproperties) as needed.
2. Click **Finish**.

Your new hardware profile appears in the **Select Hardware** page. You can optionally [create an AVD](https://developer.android.com/studio/run/managing-avds#createavd) that uses the hardware profile by clicking **Next**. Or, click **Cancel** to return to the **Your Virtual Devices** page or **Select Deployment Target** dialog.

## Edit existing AVDs

From the [**Your Virtual Devices**](https://developer.android.com/studio/run/managing-avds#viewing) page, you can perform the following operations on an existing AVD:

* To edit an AVD, click **Edit this AVD**  and [make your changes](https://developer.android.com/studio/run/managing-avds#copyavd).
* To delete an AVD, right-click an AVD and select **Delete**. Or click Menu  and select **Delete**.
* To show the associated AVD .ini and .img files on disk, right-click an AVD and select **Show on Disk**. Or click Menu  and select **Show on Disk**.
* To view AVD configuration details that you can include in any bug reports to the Android Studio team, right-click an AVD and select **View Details**. Or click Menu  and select **View Details**.

## Edit existing hardware profiles

From the [**Select Hardware**](https://developer.android.com/studio/run/managing-avds#selecthardwarepage) page, you can perform the following operations on an existing hardware profile:

* To edit a hardware profile, select it and click **Edit Device**. Or right-click a hardware profile and select **Edit**. Next, [make your changes](https://developer.android.com/studio/run/managing-avds#copyhp).
* To delete a hardware profile, right-click it and select **Delete**.

You can't edit or delete the predefined hardware profiles.

## Run and stop an emulator, and clear data

From the [**Your Virtual Devices**](https://developer.android.com/studio/run/managing-avds#viewing) page, you can perform the following operations on an emulator:

* To run an emulator that uses an AVD, double-click the AVD. Or click **Launch** .
* To stop a running emulator, right-click an AVD and select **Stop**. Or click Menu  and select **Stop**.
* To clear the data for an emulator, and return it to the same state as when it was first defined, right-click an AVD and select **Wipe Data**. Or click Menu  and select **Wipe Data**.

## Import and export hardware profiles

From the [**Select Hardware**](https://developer.android.com/studio/run/managing-avds#selecthardwarepage) page, you can import and export hardware profiles:

* To import a hardware profile, click **Import Hardware Profiles** and select the XML file containing the definition on your computer.
* To export a hardware profile, right-click it and select **Export**. Specify the location where you want to store the XML file containing the definition.

## Hardware profile properties

You can specify the following properties of hardware profiles in the [**Configure Hardware Profile**](https://developer.android.com/studio/run/managing-avds#createhp) page. AVD configuration properties override hardware profile properties, and emulator properties that you set while the emulator is running override them both.

The predefined hardware profiles included with the AVD Manager aren't editable. However, you can copy them and edit the copies.

|  |  |
| --- | --- |
| **Hardware Profile Property** | **Description** |
| Device Name | Name of the hardware profile. The name can contain uppercase or lowercase letters, numbers from 0 to 9, periods (.), underscores (\_), parentheses ( () ), and spaces. The name of the file storing the hardware profile is derived from the hardware profile name. |
| Device Type | Select one of the following:   * Phone/Tablet * Wear OS * Android TV * Chrome OS Device * Android Automotive |
| Screen Size | The physical size of the screen, in inches, measured at the diagonal. If the size is larger than your computer screen, it’s reduced in size at launch. |
| Screen Resolution | Type a width and height in pixels to specify the total number of pixels on the simulated screen. |
| Round | Select this option if the device has a round screen, such as some Wear OS devices. |
| Memory: RAM | Type a RAM size for the device and select the units, one of B (byte), KB (kilobyte), MB (megabyte), GB (gigabyte), or TB (terabyte). |
| Input: Has Hardware Buttons (Back/Home/Menu) | Select this option if your device has hardware navigation buttons. Deselect it if these buttons are implemented in software only. If you select this option, the buttons won’t appear on the screen. You can use the emulator side panel to "press" the buttons, in either case. |
| Input: Has Hardware Keyboard | Select this option if your device has a hardware keyboard. Deselect it if it doesn’t. If you select this option, a keyboard won’t appear on the screen. You can use your computer keyboard to send keystrokes to the emulator, in either case. |
| Navigation Style | Select one of the following:   * None - No hardware controls. Navigation is through the software. * D-pad - Directional Pad support. * Trackball * Wheel   These options are for actual hardware controls on the device itself. However, the events sent to the device by an external controller are the same. |
| Supported Device States | Select one or both options:   * Portrait - Oriented taller than wide. * Landscape - Oriented wider than tall.   If you select both, you can switch between orientations in the emulator. You must select at least one option to continue. |
| Cameras | To enable the camera, select one or both options:   * Back-Facing Camera - The lens faces away from the user. * Front-Facing Camera - The lens faces toward the user.   Later, you can use a webcam or a photo provided by the emulator to simulate taking a photo with the camera. |
| Sensors: Accelerometer | Select if the device has hardware that helps the device determine its orientation. |
| Sensors: Gyroscope | Select if the device has hardware that detects rotation or twist. In combination with an accelerometer, it can provide smoother orientation detection and support a six-axis orientation system. |
| Sensors: GPS | Select if the device has hardware that supports the Global Positioning System (GPS) satellite-based navigation system. |
| Sensors: Proximity Sensor | Select if the device has hardware that detects if the device is close to your face during a phone call to disable input from the screen. |
| Default Skin | Select a skin that controls what the device looks like when displayed in the emulator. Remember that specifying a screen size that's too big for the resolution can mean that the screen is cut off, so you can't see the whole screen. See [Create an emulator skin](https://developer.android.com/tools/devices/managing-avds#skins) for more information. |

## AVD properties

You can specify the following properties for AVD configurations in the [**Verify Configuration**](https://developer.android.com/studio/run/managing-avds#verifyconfigpage) page. The AVD configuration specifies the interaction between the development computer and the emulator, as well as properties you want to override in the hardware profile.

AVD configuration properties override hardware profile properties. Emulator properties that you set while the emulator is running override them both.

|  |  |
| --- | --- |
| **AVD Property** | **Description** |
| AVD Name | Name of the AVD. The name can contain uppercase or lowercase letters, numbers from 0 to 9, periods (.), underscores (\_), parentheses ( () ), dashes (-), and spaces. The name of the file storing the AVD configuration is derived from the AVD name. |
| AVD ID (Advanced) | The AVD filename is derived from the ID, and you can use the ID to refer to the AVD from the command line. |
| Hardware Profile | Click **Change** to select a different hardware profile in the [Select Hardware](https://developer.android.com/studio/run/managing-avds#selecthardwarepage) page. |
| System Image | Click **Change** to select a different system image in the [System Image](https://developer.android.com/studio/run/managing-avds#systemimagepage) page. An active internet connection is required to download a new image. |
| Startup Orientation | Select one option for the initial emulator orientation:   * Portrait - Oriented taller than wide. * Landscape - Oriented wider than tall.   An option is enabled only if it’s selected in the hardware profile. When running the AVD in the emulator, you can change the orientation if both portrait and landscape are supported in the hardware profile. |
| Camera (Advanced) | To enable a camera, select one or both options:   * Front - The lens faces away from the user. * Back - The lens faces toward the user.   The **Emulated** setting produces a software-generated image, while the **Webcam** setting uses your development computer webcam to take a picture.  This option is available only if it's selected in the hardware profile; it's not available for Wear OS and Android TV. |
| Network: Speed (Advanced) | Select a network protocol to determine the speed of data transfer:   * GSM - Global System for Mobile Communications * HSCSD - High-Speed Circuit-Switched Data * GPRS - Generic Packet Radio Service * EDGE - Enhanced Data rates for GSM Evolution * UMTS - Universal Mobile Telecommunications System * HSDPA - High-Speed Downlink Packet Access * LTE - Long-Term Evolution * Full (default) - Transfer data as quickly as your computer allows. |
| Network: Latency (Advanced) | Select a network protocol to set how much time (delay) it takes for the protocol to transfer a data packet from one point to another point. |
| Emulated Performance: Graphics | Select how graphics are rendered in the emulator:   * Hardware - Use your computer graphics card for faster rendering. * Software - Emulate the graphics in software, which is useful if you're having a problem with rendering in your graphics card. * Automatic - Let the emulator decide the best option based on your graphics card. |
| Emulated Performance: Boot option (Advanced) | * Cold boot - Start the device each time by powering up from the device-off state. * Quick boot - Start the device by loading the device state from a saved snapshot. For details, see [Run the emulator with Quick Boot](https://developer.android.com/studio/run/emulator#quickboot). |
| Emulated Performance: Multi-Core CPU (Advanced) | Select the number of processor cores on your computer that you’d like to use for the emulator. Using more processor cores speeds up the emulator. |
| Memory and Storage: RAM | The amount of RAM on the device. This value is set by the hardware manufacturer, but you can override it, if needed, such as for faster emulator operation. Increasing the size uses more resources on your computer. Type a RAM size and select the units, one of B (byte), KB (kilobyte), MB (megabyte), GB (gigabyte), or TB (terabyte). |
| Memory and Storage: VM Heap | The VM heap size. This value is set by the hardware manufacturer, but you can override it, if needed. Type a heap size and select the units, one of B (byte), KB (kilobyte), MB (megabyte), GB (gigabyte), or TB (terabyte). For more information on Android VMs, see [Memory Management for Different Virtual Machines](https://developer.android.com/tools/help/am-memory#vm). |
| Memory and Storage: Internal Storage | The amount of nonremovable memory space available on the device. This value is set by the hardware manufacturer, but you can override it, if needed. Type a size and select the units, one of B (byte), KB (kilobyte), MB (megabyte), GB (gigabyte), or TB (terabyte). |
| Memory and Storage: SD Card | The amount of removable memory space available to store data on the device. To use a virtual SD card managed by Android Studio, select **Studio-managed**, type a size, and select the units, one of B (byte), KB (kilobyte), MB (megabyte), GB (gigabyte), or TB (terabyte). A minimum of 100 MB is recommended to use the camera. To manage the space in a file, select **External file** and click **...** to specify the file and location. For more information, see [mksdcard](https://developer.android.com/tools/help/mksdcard) and [AVD data directory](https://developer.android.com/studio/run/emulator-commandline#data-filedir). |
| Device Frame: Enable Device Frame | Select to enable a frame around the emulator window that mimics the look of a real device. |
| Custom Skin Definition (Advanced) | Select a skin that controls what the device looks like when displayed in the emulator. Remember that specifying a screen size that's too big for the skin can mean that the screen is cut off, so you can't see the whole screen. See [Create an emulator skin](https://developer.android.com/tools/devices/managing-avds#skins) for more information. |
| Keyboard: Enable Keyboard Input (Advanced) | Select this option if you want to use your hardware keyboard to interact with the emulator. It's disabled for Wear OS and Android TV. |

# 3 - Run apps on a hardware device

When building an Android app, it's important that you always test your app on a real device before releasing it to users. This page describes how to set up your development environment and Android device for testing and debugging over an Android Debug Bridge (ADB) connection.

## Set up a device for development

Before you can start debugging on your device, decide if you want to connect the device to using a USB cable or Wi-Fi. Then do the following:

1. On the device, open the **Settings** app, select **Developer options**, and then enable **USB debugging** (if applicable).

**Note:** If you do not see **Developer options**, follow the instructions to [enable developer options](https://developer.android.com/studio/debug/dev-options).

## Connect to your device using USB

## When you are set up and plugged in over USB, you can click **Run**  in Android Studio to [build and run your app on the device](https://developer.android.com/studio/run).

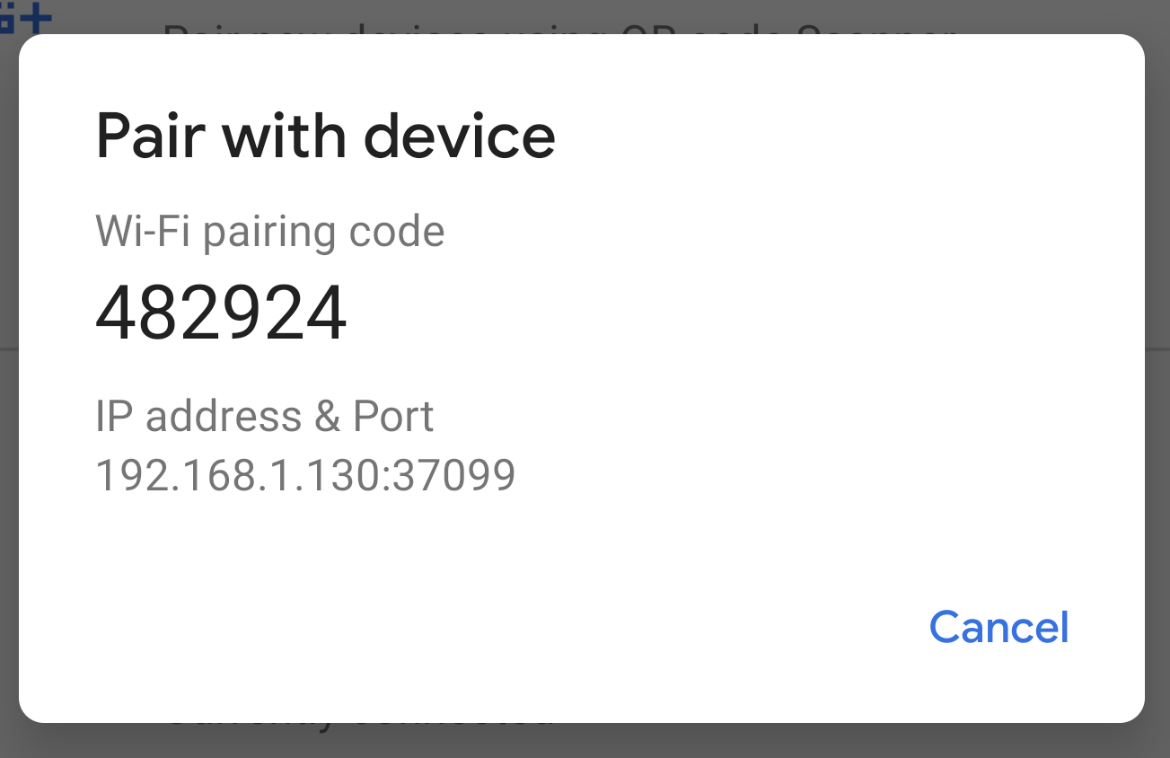
You can also use [adb](https://developer.android.com/studio/command-line/adb) to issue commands, as follows:

* Verify that your device is connected by running the adb devices command from your **android\_sdk**/platform-tools/ directory. If connected, you'll see the device listed.
* Issue any [adb command](https://developer.android.com/studio/command-line/adb" \l "issuingcommands) with the -d flag to target your device.

### Connect to your device using Wi-Fi

Android 11 (and later) supports deploying and debugging your app wirelessly from your workstation via Android Debug Bridge (adb). For example, you can deploy your debuggable app to multiple remote devices without physically connecting your device via USB and contending with common USB connection issues, such as driver installation.

To use wireless debugging, you need to pair your device to your workstation using a pairing code. Your workstation and device must be connected to the same wireless network. To connect to your device, follow these steps:



1. On your workstation, update to the latest version of the [SDK Platform-Tools](https://developer.android.com/studio/releases/platform-tools).
2. Enable the **Wireless debugging** option under **Developer options**.
3. On the dialog that asks **Allow wireless debugging on this network?**, click **Allow**.
4. Select **Pair device with pairing code**. Take note of the pairing code, IP address, and port number displayed on the device (see image).
5. On your workstation, open a terminal and navigate to **android\_sdk**/platform-tools.
6. Run adb pair **ipaddr**:**port**. Use the IP address and port number from step 3.
7. When prompted, enter the pairing code that you received in step 4. A message should indicate that your device has been successfully paired.
8. Enter pairing code: 482924
9. Successfully paired to 192.168.1.130:37099 [guid=adb-235XY]
10. Run adb connect **ipaddr**:**port**. Use the IP address and port under **Wireless debugging** (see image below).

