Android Developers

Run Apps on the Android Emulator

In this document

See also

The Android Emulator simulates a device and displays it on your development computer. It lets you prototype, develop, and test Android apps without using a hardware device. The emulator supports Android phone, tablet, Android Wear, and Android TV devices. It comes with predefined device types so you can get started quickly, and you can create your own device definitions and emulator skins.

The Android Emulator is fast, powerful, and feature-rich. It can transfer information faster than using a connected hardware device, speeding up the development process. The multi-core feature lets the emulator take advantage of multiple core processors on your development computer to improve emulator performance even more.

You can launch an app on the emulator when you run your project, or you can drag an APK file onto the emulator to install it. As with a hardware device, after you install an app on a virtual device, it remains until you uninstall or replace it. If needed, you can test how multiple apps, such as your own or system apps, work with each other.

Features for trying out your apps

You interact with the emulator just as you would with a hardware device, but using your mouse and keyboard, and emulator buttons and controls. The emulator supports virtual hardware buttons and touchscreens, including two-finger operations, as well as directional pads (D-pads), trackballs, wheels, and various sensors. You can dynamically resize the emulator window as needed, zoom in and out, change the orientation, and even take a screenshot.

When your app is running on the emulator, it can use the services of the Android platform to invoke other apps, access the network, play audio and video, accept audio input, store and retrieve data, notify the user, and render graphical transitions and themes. The emulator has controls that let you easily send incoming phone calls and text messages, specify the location of the device, simulate fingerprint scans, specify network speed and status, and simulate battery properties. The emulator can simulate an SD card and internal data storage; you can drag a file, such as a graphics or data file, onto the emulator to store it.

Watch the following video for a fast-paced visual overview of some emulator features.

Android Virtual Device configurations

The emulator uses an Android Virtual Device (AVD) configuration to determine the look, functionality, and system image of the simulated device. AVDs let you define certain hardware aspects of your emulated devices and allow you to create many configurations to test different Android platforms and hardware permutations.

Each AVD functions as an independent device, with its own private storage for user data, SD card, and so on. When you launch the emulator with an AVD configuration, it automatically loads the user data and SD card data from the AVD directory. By default, the emulator stores the user data, SD card data, and cache in the AVD directory.

To create and manage AVDs, use the AVD Manager (https://developer.android.com/tools/devices/managing-avds.html).

System images

The Android Emulator runs a full Android system stack, down to the kernel level, that includes a set of preinstalled apps (such as the dialer) that you can access from your apps. You can choose which version of the Android system you want to run in the emulator when creating AVDs.

The Android system images available through the AVD Manager contain code for the Android Linux kernel, the native libraries, the VM, and the various Android packages (such as the Android framework and preinstalled apps).

Dependencies and prerequisites

The version of Android Emulator described in this page requires the following:

- Android Studio 2.0 or higher
- Android Emulator (a standalone package for versions 25.3.0 and higher, previous versions indluded in the SDK Tools package)
- SDK Tools 25.0.10 or higher
- System requirements (https://developer.android.com/sdk/index.html#Requirements) for the accelerated emulator
- Newly created AVDs to replace any AVDs for emulator 24.0.*x* or lower
- Active network connection for certain operations, such as testing app features that require it
- adb integration enabled through **Tools** > **Android** > **Enable ADB Integration**

What's not supported

The Android Emulator supports most features of a device, but doesn't include virtual hardware for:

- WiFi

- SD card insert/eject
- Device-attached headphones
- USB

The watch emulator for Android Wear doesn't support the Overview (Recent Apps) button, D-pad, and fingerprint sensor.

While most end users of phones and tablets tend to use earlier API levels, Android Wear and Android TV users tend to use the latest releases. Using recent releases can give you a better experience using the emulator.

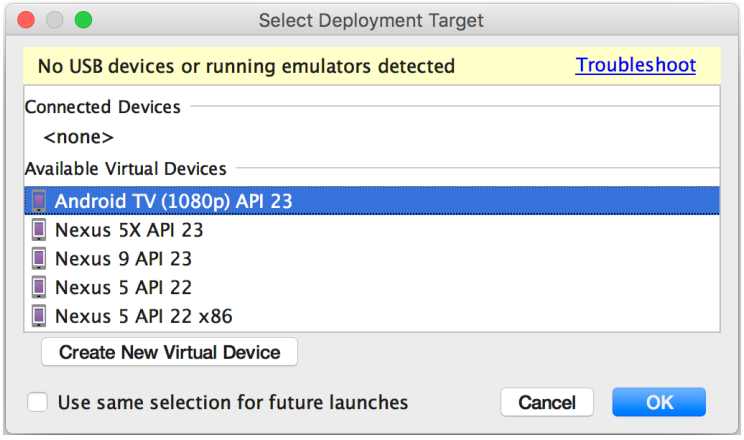
Running an App in the Android Emulator

You can run an app from an Android Studio project. Or, you can run an app that's been installed on the emulator as you would run any app on a device.

To start the emulator and run an app in your project:

1. Open an Android Studio project and click **Run** ▶.

   The *Select Deployment Target* dialog appears.



2. If you receive an error or warning message at the top of the dialog, click the link to correct the problem or get more information.

   The **No USB devices or running emulators detected** warning means that you don't currently have any emulators running, or any detected hardware devices connected to your computer. If you don't have hardware devices connected to your computer, or any emulators running, you can ignore it.

   Some errors you must fix before you can continue, such as certain Hardware Accelerated Execution Manager (Intel® HAXM) errors.

   For Mac, if you see a `Warning: No DNS servers found` error when starting the emulator, check to see whether you have an `/etc/resolv.conf` file. If not, enter the following command in a terminal window:

```
ln -s /private/var/run/resolv.conf /etc/resolv.conf
```

3. In the *Select Deployment Target* dialog, select an existing emulator definition, and then click **OK**.

   If you don't see a definition you want to use, click **Create New Virtual Device** to launch the AVD Manager. After you define a new AVD, in the *Select Deployment Target* dialog, click **OK**.

   If you want to use this emulator definition as the default for your project, select **Use same selection for future launches**.

   The emulator launches and displays your app.

4. Test your app in the emulator.

   You can use the features described in the following sections:

   - Navigating on the Screen (#navigate)
   - Performing Basic Tasks in the Emulator (#tasks)

The emulator device stores the installed app so you can run it again, if needed. You need to uninstall an app to remove it. If you run the project again on the same emulator, it replaces the app with the new version.

Launching the Android Emulator without Running an App

To start the emulator:

1. Open the AVD Manager (https://developer.android.com/tools/devices/managing-avds.html).
2. Double-click an AVD, or click **Run** ▶.

   The Android Emulator appears.

   While the emulator is running, you can run Android Studio projects and choose the emulator as the target device. You can also drag one or more APKs onto the emulator to install them, and then run them.

Navigating on the Screen

Use your computer mouse pointer to mimic your finger on the touchscreen; select menu items and input fields; and click buttons and controls. Use your computer keyboard to type characters and enter emulator shortcuts.

Feature	Description
Swipe the screen	Point to the screen, press and hold the primary mouse button, swipe across the screen, and then release.
Drag an item	Point to an item on the screen, press and hold the primary mouse button, move the item, and then release.
Tap (touch)	Point to the screen, press the primary mouse button, and then release. For example, you could click a text field to start typing in it, select an app, or press a button.
Double tap	Point to the screen, press the primary mouse button quickly twice, and then release.
Touch and hold	Point to an item on the screen, press the primary mouse button, hold, and then release. For example, you could open options for an item.
Type	You can type in the emulator by using your computer keyboard, or using a keyboard that pops up on the emulator screen. For example, you could type in a text field after you selected it.
Pinch and spread	Pressing Control or Command (⌘) brings up a pinch gesture multi-touch interface. The mouse acts as the first finger, and across the anchor point is the second finger. Drag the cursor to move the first point. Clicking the left mouse button acts like touching down both points, and releasing acts like picking both up.
Vertical swipe	Open a vertical menu on the screen and use the scroll wheel (also called the mouse wheel) to scroll through the menu items until you see the one you want. Click the menu item to select it.

Performing Basic Tasks in the Emulator

The panel on the right side of the emulator lets you perform various tasks. You can also drag files onto the emulator to install apps and download files.

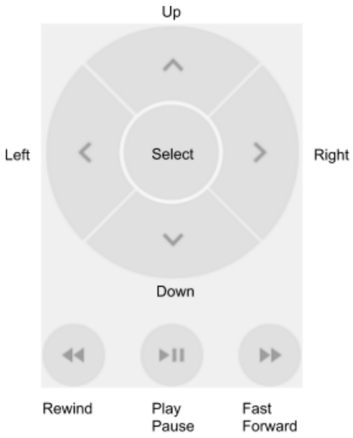
Feature	Description	Keyboard Shortcut
Close ✕	Close the emulator.	
Minimize —	Minimize the emulator window.	
Resize	Resize the emulator as you would any other operating system window. The emulator maintains an aspect ratio appropriate for your device.	Command+Up and Command+Down

Volume Up	Click to view a slider control and turn the volume up. Click again to turn it up more, or use the slider control to change the volume.	Control+= Command+=
Volume Down	Click to view a slider control and turn the volume down. Click again to turn it down more, or use the slider control to change the volume.	Control+- Command+-
Rotate Left	Rotate the phone 90 degrees counterclockwise.	Control+Left Command+Left
Rotate Right	Rotate the phone 90 degrees clockwise.	Control+Right Command+Right
Take Screenshot	Click to take a screenshot of the device. The default save location is your computer desktop. To change the save location, select **...** > **Settings**. The emulator creates a file with the name `Screenshot_yyyymmdd-hhmmss.png` using the year, month, day, hour, minute, and second of the capture, for example, `Screenshot_20160219-145848.png`.	Control+S Command+S
Enter Zoom Mode	Click so the cursor changes to the zoom icon: • Left-click the screen to zoom in by 25%, up to a maximum of about twice the screen resolution of the virtual device. • Right-click to zoom out. • Left-click and drag to select a box-shaped area to zoom in on. • Right-click and drag a selection box to reset to default zoom. • Control-click to touch the screen while in zoom mode. Click Enter Zoom Mode again to return to normal screen size.	Control+Z Command+Z While in zoom mode: Control+Up Control+Down Control+Shift+Up Control+Shift+Down Control+Shift+Left Control+Shift+Right Command+Up and Command+Down Command+Shift+Up and Command+Shift+Down Command+Shift+Left and Command+Shift+Right
Back	Return to the previous screen, or close a dialog box, an options menu, the Notifications panel, or the onscreen keyboard.	Control+Backspace Command+Backspace
Home	Return to the Home screen. Press and hold to open the item specific to your API level.	Control+H Command+H
Overview (Recent Apps)	Tap to open a list of thumbnail images of apps you've worked with recently. To open an app, tap it. To remove a thumbnail from the list, swipe it left or right. This button isn't supported for Android Wear.	Control+O Command+O
Menu	Type the keyboard shortcut to simulate the Menu button, for example, to open the menu for the selected app.	Control+M Command+M
More	Click to access other features and settings, described in the next table.	
Install an APK	Drag an APK file onto the emulator screen. An APK Installer dialog appears. When the installation completes, you can view the app in your apps list.The app didn't install if a dialog appears that says "APK failed to install."	
Add a file	Drag any file onto the emulator screen. It's placed in the `/sdcard/Download` directory. Navigate to the file using the method for the API level. For example, for API 22, this is the navigation path: **Settings** > **Device: Storage & USB** > **Internal Storage** > **Explore** (Virtual SD Card).	
Toggle trackball mode		F6

The extended controls let you send data, change device properties, control apps, and more. To access the controls, select **...** in the emulator panel and then select the option you want in the left panel of the *Extended Controls* dialog.

Feature	Description	Keyboard Shortcuts
Location	The emulator lets you simulate "my location" information: the location where the emulated device is currently located. For example, if you click My Location in Google Maps and then send a location, the map shows it. To send a GPS location: 1. Select **Decimal** or **Sexagesimal**. 2. Specify the location. In decimal mode, enter a **Latitude** value in the range -90.0 to +90.0 degrees and a **Longitude** value in the range -180.0 to +180.0 degrees. In sexigesimal mode, enter a three-part **Latitude** value in the range -90 to +90 degrees, 0 to 59 minutes, and 0.0 to 60.0 seconds. Enter a **Longitude** value in the range -180 to +180 degrees, 0 to 59 minutes, and 0.0 to 60.0 seconds. For the latitude, - indicates south and + indicates north; for the longitude, - indicates west and + indicates east. The + is optional. Optionally specify an **Altitude** value in the range -1,000.0 to +10,000.0 meters. 3. Click **Send**. To use geographic data from a GPS exchange format (GPX) or Keyhole Markup Language (KML) file: 1. Click **Load GPX/KML**. 2. In the file dialog, select a file on your computer and click **Open**. 3. Optionally select a **Speed**. The speed defaults to the **Delay** value (**Speed 1X**). You can increase the speed by double (**Speed 2X**), triple (**Speed 3X**), and so on. 4. Click **Run** ▶.	Control+Shift+L Command+Shift+L
Cellular	The emulator lets you simulate various network conditions. You can approximate the network speed for different network protocols, or you can specify **Full**, which transfers data as quickly as your computer allows. Specifying a network protocol is always slower than **Full**. You can also specify the voice and data network status, such as roaming. The defaults are set in the AVD. Select a **Network type**: • 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 • HSPDA - High-Speed Downlink Packet Access • LTE - Long-Term Evolution • Full (default) Select a **Signal strength**:	Control+Shift+C Command+Shift+C

	• Poor • Moderate (default) • Good • Great Select a **Voice status**, **Data status**, or both: • Home (default) • Roaming • Searching • Denied (emergency calls only) • Unregistered (off)	
Battery	You can simulate the battery properties of a device to see how your app performs under different conditions. To select a **Charge level**, use the slider control. Select a **Charger connection** value: • None • AC charger Select a **Battery health** value: • Good (default) • Failed • Dead • Overvoltage • Overheated • Unknown Select a **Battery status** value: • Unknown • Charging (default) • Discharging • Not charging • Full	Control+Shift+B Command+Shift+B
Phone	The emulator lets you simulate incoming phone calls and text messages. Note that the information flow is one way, from the control to the emulator. For example, the control doesn't change its state if the emulator hangs up; you need to end the call in the control. To initiate a call to the emulator: 1. Select or type a phone number in the **From** field. 2. Click **Call Device**. 3. Optionally click **Hold Call** to put the call on hold. 4. To end the call, click **End Call**. To send a text message to the emulator:	Control+Shift+P Command+Shift+P

	3. Click **Send Message**.	
Directional Pad	If the AVD has the directional pad enabled in the hardware profile, you can use the directional pad controls with the emulator. However, not all devices can support the directional pad; for example, an Android watch. The buttons simulate the following actions:  	Control+Shift+D Command+Shift+D
Fingerprint	This control can simulate 10 different fingerprint scans. You can use it to test fingerprint integration in your app. This feature is disabled for Android 5.1 (API level 22) and lower, and for Android Wear. To simulate a fingerprint scan on the virtual device: 1. Prepare an app to receive a fingerprint. 2. Select a **Fingerprint** value. 3. Click **Touch Sensor**.	Control+Shift+F Command+Shift+F
Virtual sensors > Accelerometer	This control lets you test your app against changes in device position, orientation, or both. For example, you can simulate gestures such as tilt and rotation. The accelerometer doesn't track the absolute position of the device: it just detects when a change is occurring. The control simulates the way accelerometer and magnetometer sensors would respond when you move or rotate a real device. The control reports `TYPE_ACCELEROMETER` (https://developer.android.com/reference/android/hardware/Sensor.html#TYPE_ACCELEROMETER) events on the x, y, and z axis. These values include gravity. For example, if the device is suspended in outer space, it would experience zero acceleration (all of x, y, and z will be 0). When the device is on Earth and laying screen up on top of a table, the acceleration is 0, 0, and 9.8 because of gravity. The control also reports `TYPE_MAGNETIC_FIELD` (https://developer.android.com/reference/android/hardware/Sensor.html#TYPE_MAGNETIC_FIELD) events, which measure the ambient magnetic field on the x, y and z axis in micro-Tesla (μT). To rotate the device around the x, y, and z axes, select **Rotate** and do one of the following: • Adjust the **Yaw**, **Pitch**, and **Roll** sliders and observe the position in the upper pane. • Move the device representation in the upper pane and observe the **Yaw**, **Pitch**, and **Roll** and how the resulting accelerometer values change. See Computing the Device's Orientation (https://developer.android.com/guide/topics/sensors/sensors_position.html#sensors-pos-orient) for more information about how yaw, pitch, and roll are calculated. To move the device horizontally (x) or vertically (y), select **Move** and do one of the following: • Adjust the **X** and **Y** sliders and observe the position in the upper pane. • Move the device representation in the upper pane and observe the **X** and **Y** slider values and how the	Control+Shift+V Command+Shift+V

	• In the **Device rotation** area, select a button to change the rotation (https://developer.android.com/reference/android/view/Display.html#getRotation()). As you adjust the device, the **Resulting Values** fields change accordingly. These are the values that an app can access. For more information about these sensors, see Sensors Overview (https://developer.android.com/guide/topics/sensors/sensors_overview.html), Motion Sensors (https://developer.android.com/guide/topics/sensors/sensors_motion.html), and Position Sensors (https://developer.android.com/guide/topics/sensors/sensors_position.html). You can import the AccelerometerPlay app (https://github.com/googlesamples/android-AccelerometerPlay) to try out the **Accelerometer** control. Select **File** > **New** > **Import Sample** and select the app in the dialog. This app is showcased in the emulator video on this page.	
Virtual sensors > Additional sensors	The emulator can simulate various position and environment sensors. It lets you adjust the following sensors so you can test them with your app: • Ambient temperature - This environmental sensor measures ambient air temperature. • Magnetic field - This position sensor measures the ambient magnetic field at the X, Y and Z axis, respectively. The values are in micro-Tesla (μT). • Proximity - This position sensor measures the distance from an object; for example, it can notify a phone that a face is close to it to make a call. • Light - This environmental sensor measures illuminance. • Pressure - This environmental sensor measures ambient air pressure. • Relative Humidity - This environmental sensor measures ambient relative humidity. For more information about these sensors, see Sensors Overview (https://developer.android.com/guide/topics/sensors/sensors_overview.html), Position Sensors (https://developer.android.com/guide/topics/sensors/sensors_position.html), and Environment Sensors (https://developer.android.com/guide/topics/sensors/sensors_environment.html).	Control+Shift+V Command+Shift+V
Settings	You can specify the following settings: • Emulator window theme - Select Light or Dark. • Send keyboard shortcuts to - By default, some keyboard combinations will trigger emulator control shortcuts. If you're developing an app that includes keyboard shortcuts, such as one targeted at devices with Bluetooth keyboards, you can change this setting to send *all* keyboard input to the virtual device, including input that would be a shortcut in the emulator. • Screenshot save location - Click the folder icon to specify a location to save screenshots of the emulator screen. • Use detected ADB location - If you're running the emulator from Android Studio, you should select this setting (the default). If you run the emulator from outside Android Studio and want it to use a specific adb executable, deselect this option and specify the SDK Tools location. If this setting is incorrect, features such as drag-and-drop app install and file copy, and screenshot capture, won't work. • When to send crash reports - Select Always, Never, or Ask.	Control+Shift+S Command+Shift+S
Help > Keyboard Shortcuts	See the keyboard shortcuts that the emulator accepts. For the shortcuts to work, you need to: • Select **Settings** > **Send keyboard shortcuts to** > **Emulator controls (default)**.	F1 Command/
Help > Emulator Help	To go to the online documentation for the emulator, click **Documentation**. To file a bug against the emulator, click **File a Bug**.	F1 Command/

Help > About	See which adb port the emulator uses, as well as the Android and emulator version numbers. Compare the latest available emulator version with your version to determine if you have the latest software installed. The emulator serial number is **emulator-*adb_port***, which you can specify as an adb command line option, for example.	F1 Command/

Compare Android Emulator Tools

The following table compares the tasks you can perform using the emulator UI (https://developer.android.com/studio/run/emulator.html), AVD Manager, commonly used command-line startup options (https://developer.android.com/studio/run/emulator-commandline.html#startup-options), and the Emulator Console (https://developer.android.com/studio/run/emulator-console.html). In the AVD Manager, you can set hardware profile (https://developer.android.com/studio/run/managing-avds.html#hpproperties) and AVD (https://developer.android.com/studio/run/managing-avds.html#avdproperties) properties.

The table compares features that are similar, but not necessarily with identical functionality, between different emulator tools.

Although not listed in the table, the adb utility also supports commands for interacting with virtual devices, as described in Android Debug Bridge (https://developer.android.com/studio/command-line/adb.html). For example, you can install an app on a virtual device by using Android Studio (https://developer.android.com/studio/run/emulator.html#runningapp), the emulator UI (https://developer.android.com/studio/run/emulator.html#tasks), or adb (https://developer.android.com/studio/command-line/adb.html#move). You can copy a file to a virtual device using the emulator UI or adb (https://developer.android.com/studio/command-line/adb.html#copyfiles), and copy a file from a virtual device using adb.

Remember that to use certain features of the emulator with your apps, you need to enable them through various `<uses-feature>` (https://developer.android.com/guide/topics/manifest/uses-feature-element.html) elements and manifest permission (https://developer.android.com/reference/android/Manifest.permission.html) constants, just as you would if you ran them on a hardware device.

User Interface Control	Hardware Profile Property	AVD Property	Command-Line Startup Option	Console Command
Device Hardware				
	Device Type: Phone/Tablet, Android Wear, Android TV			
Back, Home, Overview, Menu	Input: Has Hardware Buttons (Back/Home/Menu)			
Settings: Send keyboard shortcuts to	Input: Has Hardware Keyboard	Keyboard Input		
Directional Pad Toggle trackball mode (F6)	Navigation Style: None, D-pad, Trackball, Wheel			
Zoom Mode				
Battery				`power` command
Phone				`gsm` and

Rotate	Supported Device States	Startup: Orientation		`rotate` command
	Camera	Camera (can choose webcam)	`-camera-back` `-camera-front` `-webcam-list`	
Take Screenshot Settings: Screenshot save location				
Orientation Location Virtual sensors	Sensors: Accelerometer, Gyroscope, GPS, Proximity Sensor			`geo` command
Fingerprint				`finger` command
Volume			`-noaudio` `-no-audio`	
				`event` command
Disk Images and Memory				
		Memory and Storage: SD Card	`-sdcard`	
		Memory and Storage: Internal Storage		
	RAM	Memory and Storage: RAM	`-memory`	
		Memory and Storage: VM Heap		`vm` command
	In the **Your Virtual Devices** page of the AVD Manager, right-click an AVD and select Wipe Data (https://developer.android.com/studio/run/managing-avds.html#emulator).		`-wipe-data`	
Network				
				`network status` command
			`-dns-server`	
			`-http-proxy`	
	Network: Latency		`-netdelay`	`network delay` command
			`-netfast`	
Cellular: Network Type	Network: Speed		`-netspeed`	`network speed` command
Voice status Data				`gsm` command

strength				
Settings: Use detected ADB location			`-port` `-ports`	
				`redir` command
			`-tcpdump`	`network capture start` and `network capture stop` commands
System				

	In the **System Image** (https://developer.android.com/studio/run/managing-avds.html#systemimagepage) page of the AVD Manager, select a system image capable of VM acceleration (https://developer.android.com/studio/run/emulator-acceleration.html#accel-vm).		`-accel`
			`-accel-check`
			`-no-accel`	

	Multi-Core CPU			
			`-engine`	
	Emulated Performance: Graphics — hardware, software, or auto		`-gpu`	
			`-nojni` `-no-jni`	
			`-selinux {disabled|permissive}`	
			`-timezone`	
			`-version`	
UI				
Settings: Emulator window theme				
Resize	Screen Size and Screen Resolution			`window` command
	Round			
			`-no-boot-anim`	
	Default Skin	Device Frame and Custom Skin Definition		
			`-screen {touch|multi-touch|no-touch}`	
Debug				

			`-debug tags`
			`-debug-tag`
			`-debug-no-tag`	

			`-verbose`	