Android Developers

Create and Manage Virtual Devices

In this document

Dependencies and prerequisites

Android Studio 2.0 or higher

SDK Tools 25.0.10 or higher

adb integration enabled through **Tools** > **Android** > **Enable ADB Integration**

Active network connection for certain operations, such as downloading system images

An Android Virtual Device (AVD) definition lets you define the characteristics of an Android phone, tablet, Android Wear, or Android TV device that you want to simulate in the Android Emulator (https://developer.android.com/studio/run/emulator.html). The AVD Manager helps you easily create and manage AVDs.

About AVDs

This site uses cookies to store your preferences for site-specific language and display options.

OK

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 Nexus phone devices, and you can define and import hardware profiles as needed. You can override some of the settings in your AVD, if needed.

To effectively test your app, you should create an AVD that models each device type that your app is designed to support.

System image

The AVD Manager helps you choose a system image for your AVD by providing recommendations. It also lets you download system images, some with add-on libraries, like Google APIs, which your app might require. x86 system images run the fastest in the emulator. Android Wear and Android TV devices tend to run best (and have the largest installed base) on recent releases, while users of Android phones and tablets tend to use slightly older releases, as shown in the API level dashboards (https://developer.android.com/about/dashboards/index.html).

We recommend that you create an AVD for each API level that your app could potentially support based on the `<uses-sdk>` (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html) setting in your manifest. For example, you might want to test with all API levels that are equal to and higher than the `minSdkVersion` (https://developer.android.com/studio/publish/versioning.html) setting. By testing with API levels higher than required by your app, you ensure app forward-compatibility when users download system updates.

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

Just as with a real device, for apps to use certains features defined in an AVD, such as the camera, it must have the

This site uses cookies to store your preferences for site-specific language and display options.

OK

manifest. See Hardware Profile Properties (#hpproperties) and AVD Properties (#avdproperties) for lists of features you can define in your AVDs.
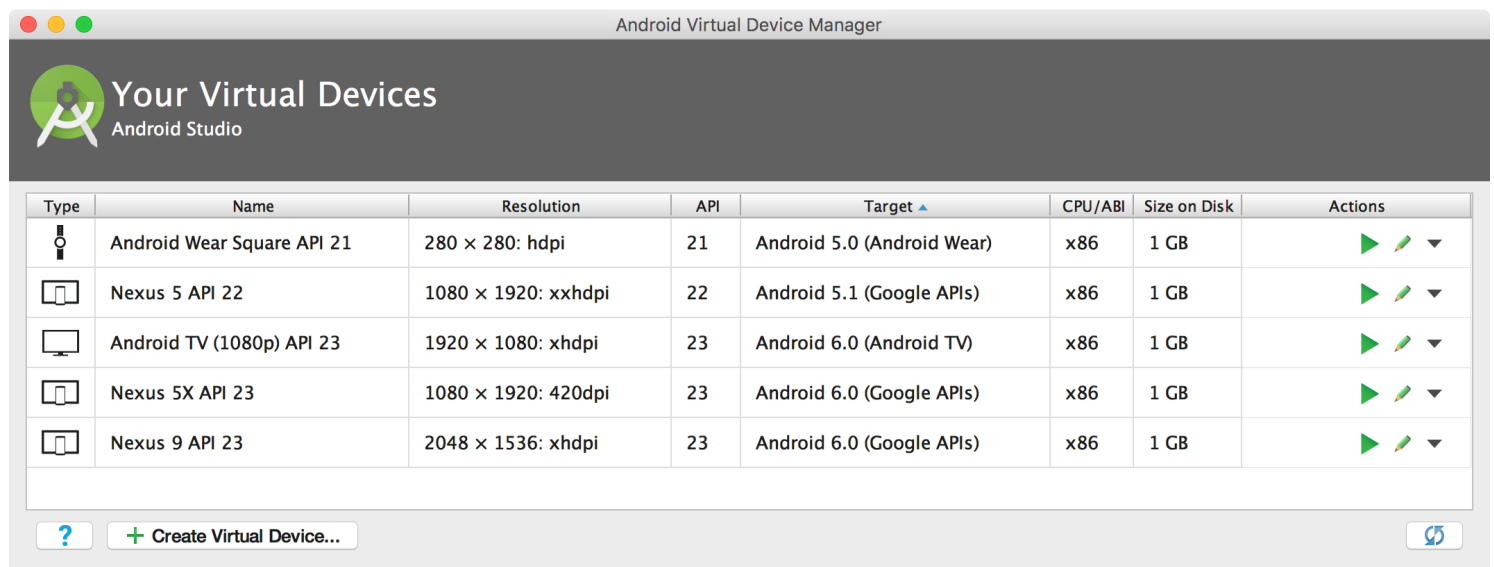
Viewing and Managing Your AVDs

The AVD Manager lets you manage your AVDs all in one place.

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

- In Android Studio, select **Tools** > **Android** > **AVD Manager**.

- Click **AVD Manager** 🔳 in the toolbar.

The AVD Manager appears.

Type	Name	Resolution	API	Target ▲	CPU/ABI	Size on Disk	Actions
📱	Android Wear Square API 21	280 × 280: hdpi	21	Android 5.0 (Android Wear)	x86	1 GB	▶ ✏ ▼
📱	Nexus 5 API 22	1080 × 1920: xxhdpi	22	Android 5.1 (Google APIs)	x86	1 GB	▶ ✏ ▼
🖥	Android TV (1080p) API 23	1920 × 1080: xhdpi	23	Android 6.0 (Android TV)	x86	1 GB	▶ ✏ ▼
📱	Nexus 5X API 23	1080 × 1920: 420dpi	23	Android 6.0 (Google APIs)	x86	1 GB	▶ ✏ ▼
📱	Nexus 9 API 23	2048 × 1536: xhdpi	23	Android 6.0 (Google APIs)	x86	1 GB	▶ ✏ ▼

It displays any AVDs you've already defined. When you first install Android Studio, it creates one AVD. If you defined AVDs for Android Emulator 24.0.*x* or lower, you need to recreate them.

From this page you can:

- Define a new AVD (#createavd) or hardware profile (#createhp).

- Edit an existing AVD (#workingavd) or hardware profile (#workinghp).

- Delete an AVD (#workingavd) or hardware profile (#workinghp).

- Import or export (#importexporthp) hardware profile definitions.

- Run (#emulator) an AVD to start the emulator.

- Clear (#emulator) data and start fresh, from the same state as when you first ran the emulator.

- Show (#workingavd) the associated AVD `.ini` and `.img` files on disk.

- View (#workingavd) AVD configuration details that you can include in any bug reports to the Android Studio team.

Creating an AVD

You can create a new AVD from the beginning, or duplicate an AVD (#copyavd) and change some properties.

To create a new AVD:

1. From the *Your Virtual Devices* (#viewing) page of the AVD Manager, click **Create Virtual Device**.

   Alternatively, run your app (https://developer.android.com/studio/run/index.html#RunningApp) from within Android Studio. In the *Select Deployment Target* dialog, click **Create New Emulator**.

   The *Select Hardware* page appears.
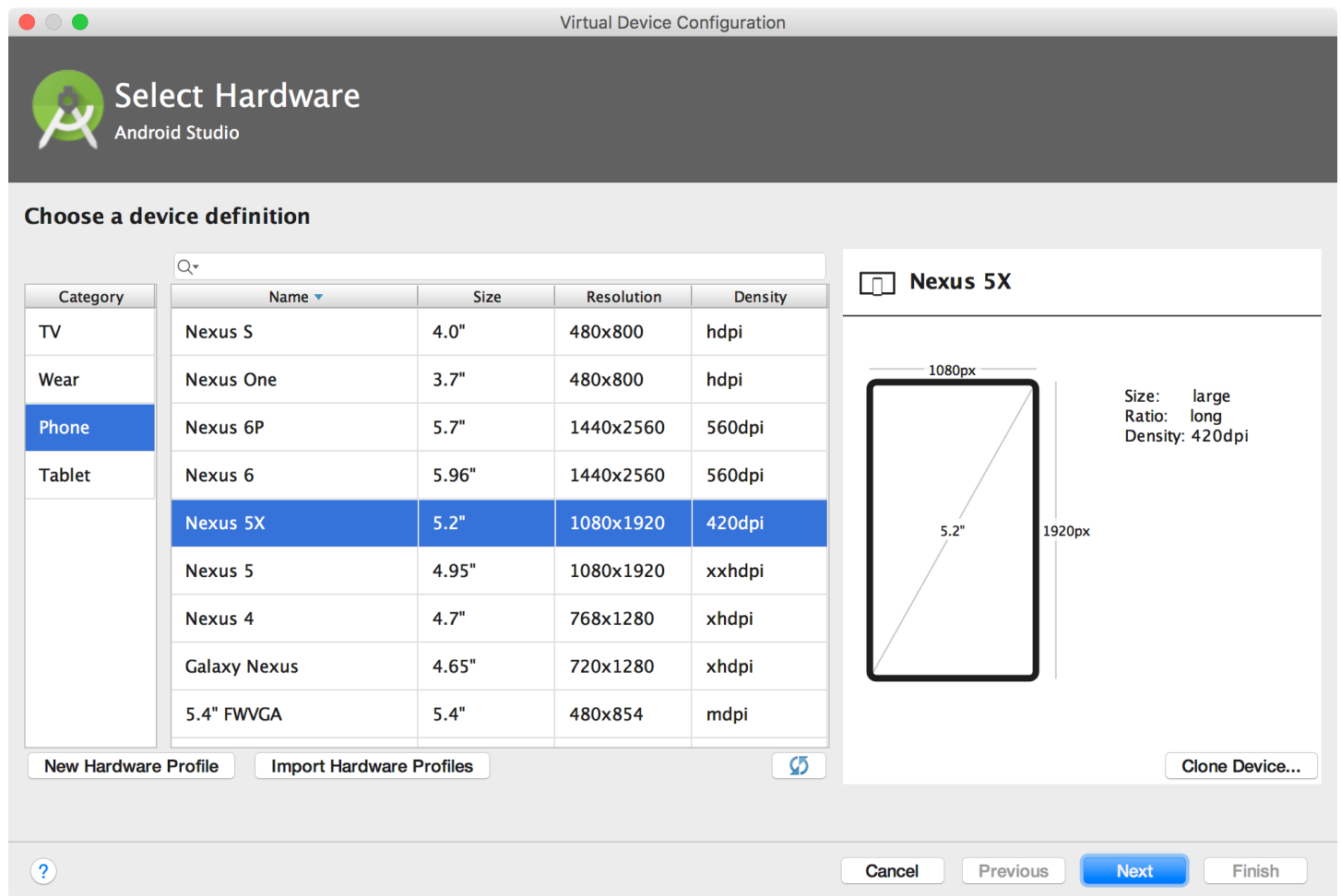
The *System Image* page appears.
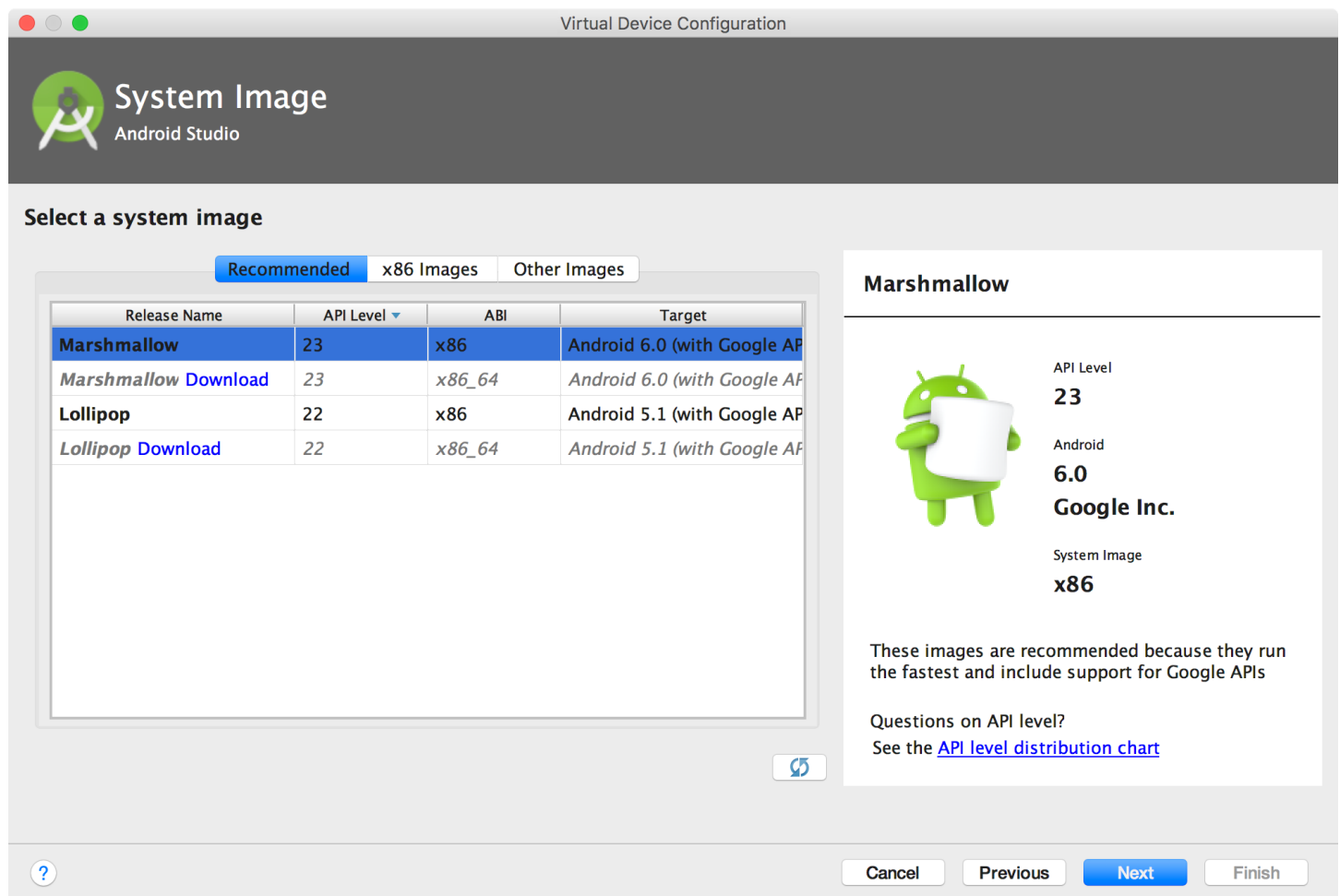
3. Select the system image for a particular API level, and then click **Next**.

   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.html) 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.html).

   If your app declares a `<uses-library>` (https://developer.android.com/guide/topics/manifest/uses-library-element.html) 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
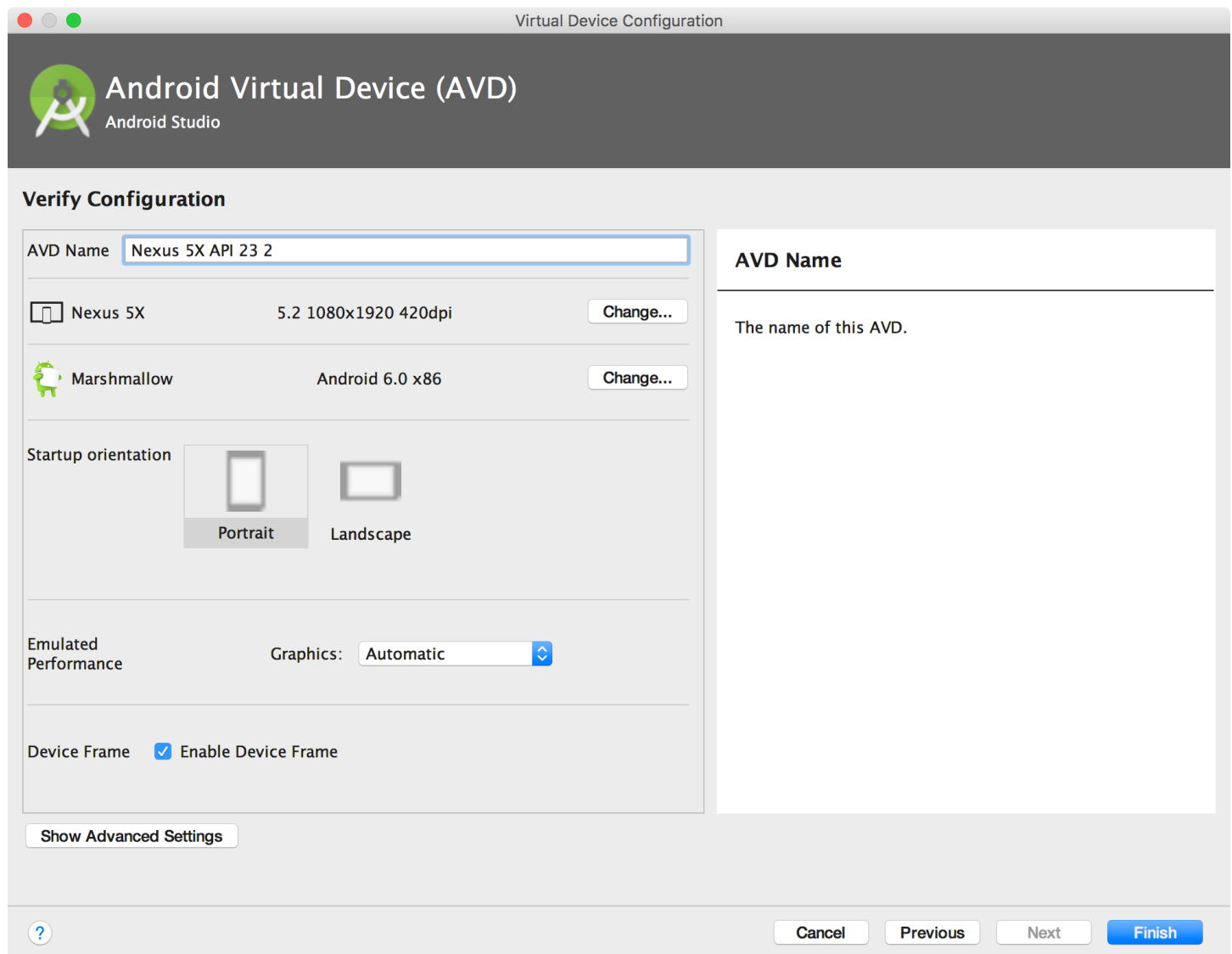
4. Change AVD properties (#avdproperties) as needed, and then click **Finish**.

   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* (#viewing) page of the AVD Manager, right-click an AVD and select **Duplicate**.

   Or click Menu ▼ and select **Duplicate**.

   The *Verify Configuration* (#verifyconfigpage) page appears.

2. Click **Change** or **Previous** if you need to make changes on the *System Image* (#systemimagepage) and *Select Hardware*
   (#selecthardwarepage) pages.

3. Make your changes, and then click **Finish**.

This site uses cookies to store your preferences for site-specific language and display options.

   OK

Creating 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 (#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* (#selecthardwarepage) page, click **New Hardware Profile**.

2. In the *Configure Hardware Profile* page, change the hardware profile properties (#hpproperties) as needed.

3. Click **Finish**.

   Your new hardware profile appears in the *Select Hardware* page. You can optionally create an AVD (#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* (#selecthardwarepage) page, select a hardware profile and click **Clone Device**.

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

2. In the *Configure Hardware Profile* page, change the hardware profile properties (#hpproperties) as needed.

3. Click **Finish**.

   Your new hardware profile appears in the *Select Hardware* page. You can optionally create an AVD (#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.

Working with Existing AVDs

From the *Your Virtual Devices* (#viewing) page, you can perform the following operations on an existing AVD:

- 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**.

Working with Existing Hardware Profiles

From the *Select Hardware* (#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 (#copyhp).

- To delete a hardware profile, right-click it and select **Delete**.

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

Running and Stopping an Emulator, and Clearing Data

From the *Your Virtual Devices* (#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**.

Importing and Exporting Hardware Profiles

From the *Select Hardware* (#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* (#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 • Android Wear • Android TV
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 Android Wear 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	Select this option if your device has a hardware keyboard. Deselect it if it doesn't. If you

This site uses cookies to store your preferences for site-specific language and display options.

OK

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 Creating Emulator Skins

This site uses cookies to store your preferences for site-specific language and display options.

OK

AVD Properties

You can specify the following properties for AVD configurations in the *Verify Configuration* (#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* (#selecthardwarepage) page.
System Image	Click **Change** to select a different system image in the *System Image* (#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.

This site uses cookies to store your preferences for site-specific language and display options.

OK

	This option is available only if it's selected in the hardware profile; it's not available for Android Wear 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 • HSPDA - High-Speed Downlink Packet Access • 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.
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.html#vm).
Memory and	The amount of nonremovable memory space available on the device. This value is set by the

This site uses cookies to store your preferences for site-specific language and display options.

OK

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.html) and AVD data directory (https://developer.android.com/studio/run/emulator-commandline.html#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 Creating Emulator Skins (https://developer.android.com/tools/devices/managing-avds.html#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 Android Wear and Android TV.

Creating Emulator Skins

An Android emulator skin is a collection of files that define the visual and control elements of an emulator display. If the skin definitions available in the AVD settings don't meet your requirements, you can create your own custom skin definition, and then apply it to your AVD.

Each emulator skin contains:

- A `hardware.ini` file

- Layout files for supported orientations (landscape, portrait) and physical configuration

- Image files for display elements, such as background, keys and buttons

To create and use a custom skin:

---

2. Define the visual appearance of the skin in a text file named `layout.` This file defines many characteristics of the skin, such as the size and image assets for specific buttons. For example:

```
parts {
    device {
        display {
            width   320
            height  480
            x        0
            y        0
        }
    }

    portrait {
        background {
            image background_port.png
        }

        buttons {
            power {
                image   button_vertical.png
                x 1229
                y 616
            }
        }
    }
    ...
}
```

3. Add the bitmap files of the device images in the same directory.

4. Specify additional hardware-specific device configurations in a `hardware.ini` file for the device settings, such as `hw.keyboard` and `hw.lcd.density`.

5. Archive the files in the skin folder and select the archive file as a custom skin.

For more detailed information about creating emulator skins, see the Android Emulator Skin File Specification (https://android.googlesource.com/platform/external/qemu/+/emu-master-dev/android/docs/ANDROID-SKIN-FILES.TXT) in the tools source code.

This site uses cookies to store your preferences for site-specific language and display options.

OK