Android Developers

Configure Hardware Acceleration for the Android Emulator

In this document

See also

- Run Apps in the Android Emulator
- Create and Manage Virtual Devices
- Compare Android Emulator Tools
- Start a Virtual Device from the Command Line
- Send Emulator Console Commands to a Virtual Device
- Set Up Android Emulator Networking

The Android Emulator (https://developer.android.com/studio/run/emulator.html) supports several hardware acceleration features to improve performance, sometimes drastically. Android Studio typically prompts you to install any required software, and suggests the best configuration for your development computer. The following sections describe how you can fine-tune the graphics and virtual machine (VM) acceleration settings, if needed, and manually install VM acceleration software.

Configuring graphics acceleration

Graphics acceleration for the emulator takes advantage of the graphics hardware on your development computer, specifically the graphics processing unit (GPU), to make screen rendering faster. Android devices use OpenGL for Embedded Systems (OpenGL ES (https://developer.android.com/guide/topics/graphics/opengl.html) or GLES) for rendering both 2D and 3D graphics on the screen.

graphics drivers that aren't compatible with the emulator. In this case, hardware graphics rendering can be poor or cause the emulator to crash.

The default is to let the emulator decide if it should use hardware or software graphics acceleration based on your computer setup. If the GPU hardware is compatible, the emulator uses it; otherwise, it uses software GPU emulation.

If you start the emulator from the command line (https://developer.android.com/studio/run/emulator-commandline.html#starting), you can override the graphics acceleration setting in the AVD for that virtual device instance, if needed.

Dependencies and prerequisites

Graphics acceleration has the following requirements:

- SDK Tools - Latest release recommended (Revision 17 minimum)
- SDK Platform - Android 4.0.3, Revision 3, or higher

Configuring graphics acceleration in the AVD Manager

To configure an AVD to use hardware or software graphics acceleration, follow these steps:

1. Open the AVD Manager (https://developer.android.com/studio/run/managing-avds.html#viewing).
2. Create a new AVD (https://developer.android.com/studio/run/managing-avds.html#createavd) or edit an existing AVD (https://developer.android.com/studio/run/managing-avds.html#workingavd).
3. In the **Verify Configuration** (https://developer.android.com/studio/run/managing-avds.html#verifyconfigpage) page, specify the **Emulated Performance: Graphics** setting.
4. To save the AVD, click **Finish**.

Configuring graphics acceleration on the command line

To specify a graphics acceleration type when you run an AVD from the command line, include the `-gpu` option:

```
emulator -avd avd_name -gpu mode [{-option [value]} ... ]
```

Where *`mode`* can be any of the following:

- `auto` - Let the emulator choose hardware or software graphics acceleration based on your computer setup. It checks if your GPU driver matches a preexisting blacklist of known faulty GPU drivers, and if it does, the emulator uses `-gpu off`. Otherwise, the emulator uses `-gpu host`.
- `host` - Use the GPU on your computer. This option is typically the fastest. However, some drivers have issues with rendering OpenGL graphics, so using the computer GPU might not be a reliable option. (Equivalent to `on` and `enabled`, both deprecated.)
- `mesa` - Use the Mesa 3D (http://www.mesa3d.org/) software library to render graphics. This solution has been deprecated in favor of SwiftShader, which is faster and more compliant with the OpenGL ES standards, resulting in fewer rendering errors in certain edge cases.
- `angle` - Use Almost Native Graphics Layer Engine (ANGLE (https://chromium.googlesource.com/angle/angle/+/master/README.md)) and Microsoft DirectX to render graphics in software. It's currently supported on Windows only. When your hardware isn't compatible with the emulator,

- On Windows, Microsoft DirectX drivers normally have fewer issues than OpenGL ES drivers.
- In some cases, ANGLE rendering is slightly poorer when compared to using `-gpu host`.

- `swiftshader` - Use SwiftShader (https://swiftshader.googlesource.com/SwiftShader) to render graphics in software. It's slower than using the GPU, but normally yields good results. It's supported in the Android Emulator 25.3 and higher for all system images.
- `off` - Disable graphics hardware emulation. The emulator doesn't use the GPU for graphics rendering, and instead uses the CPU, which can be much slower and cause improper rendering for some items. It uses an OpenGL ES 1.1 software renderer inside the system image; or, for system images that contain SwiftShader, the emulator uses it. The Android OS run by the emulator includes SwiftShader in API level 19 and API levels 21 and higher. This option is available for legacy reasons only, and is required when using VM snapshots (an experimental feature in Android Studio 2.2). (This option falls back to `-gpu guest` in some situations. It is also equivalent to `-gpu disabled`, which is deprecated.)

Configuring VM acceleration

Many modern CPUs provide extensions for running VMs more efficiently. Taking advantage of these extensions with the emulator requires some additional configuration of your development computer, but can significantly improve the execution speed. This section outlines the VM acceleration requirements (https://developer.android.com/studio/index.html#Requirements), including those for each operating system.

Dependencies and prerequisites

To use VM acceleration with the emulator, you need to meet the following Android development tools requirements:

- Latest SDK Tools recommended (Revision 17 minimum)
- AVD with an x86-based system image, available for Android 2.3.3 (API level 10) and higher

AVDs that use ARM- or MIPS-based system images can't be accelerated using the emulator configurations described in the following sections.

Virtualization extension requirements

Before attempting to use acceleration, you should first determine if your CPU supports one of the following virtualization extensions technologies:

- Intel Virtualization Technology (VT, VT-x, vmx) extensions
- AMD Virtualization (AMD-V, SVM) extensions (Linux only)

Most modern computers do. If you use an older computer and you're not sure, consult the specifications from the manufacturer of your CPU to determine if it supports virtualization extensions. If your CPU doesn't support one of these virtualization technologies, then you can't use VM acceleration.

Virtualization extensions are typically enabled through your computer BIOS and are frequently turned off by default. Check the documentation for your motherboard to find out how to enable virtualization extensions.

VM acceleration restrictions

Note the following restrictions of VM acceleration:

- You can't run software that uses another virtualization technology at the same time that you run the accelerated emulator. For example, VirtualBox, VMWare, and Docker currently use a different virtualization technology, so you can't run them at the same time as the accelerated emulator.

Determining whether HAXM or KVM is installed

Emulator acceleration requires that you install either Intel Hardware Accelerated Execution Manager (Intel HAXM) or Kernel-based Virtual Machine (KVM), which are types of *hypervisors*. If the needed hypervisor isn't installed, Android Studio typically prompts you to install it. The following sections describe how to install them on different operating systems.

Without acceleration, the emulator takes the machine code from the VM and translates it block-by-block to conform to the architecture of the host computer. This process can be quite slow. But, if the VM and the architecture of the host computer match (such as x86 on x86), the emulator can skip translating the code and simply run it directly on the actual CPU using a hypervisor. In this case, the emulator can approach the speed of your actual computer.

Typically, Android Studio prompts you to install the needed software when you do any of the following:

- Install Android Studio and SDK Tools.
- Create and manage AVDs in the AVD Manager.
- Select an AVD to run from the **Select Deployment Target** dialog.

If you ever need to check whether HAXM or KVM is installed, you can use the emulator `-accel-check` command-line option.

For example, here's sample output on a Mac, where *`sdk`* is the Android SDK location:

```
janedoe-macbookpro:tools janedoe$ ./sdk/tools/emulator -accel-check
accel:
0
HAXM version 6.0.3 (3) is installed and usable.
accel
```

For Linux:

```
janedoe:~/Android/Sdk/tools$ ./sdk/tools/emulator -accel-check
accel:
0
KVM (version 12) is installed and usable.
accel
```

Configuring VM acceleration on Windows

VM acceleration for Windows requires the installation of Intel HAXM.

Your computer must have an Intel processor with support for the following:

- Intel VT-x
- Intel EM64T (Intel 64)
- Execute Disable (XD) Bit functionality enabled

- Windows 7
- Windows 8
- Windows 10

On Windows 8 or 10, you must turn off Hyper-V in the Control Panel, or the emulator won't run. Note that installing certain software could turn it back on. In this case, where possible, Android Studio displays a "quick fix" link in certain dialogs that lets you easily turn it off again.

To install the HAXM driver, follow these steps:

1. Open the SDK Manager (https://developer.android.com/studio/intro/update.html#sdk-manager).
2. Click the **SDK Update Sites** tab and then select **Intel HAXM**.
3. Click **OK**.
4. After the download finishes, execute the installer.
   For example, it might be in this location:
   `sdk\extras\intel\Hardware_Accelerated_Execution_Manager\intelhaxm-android.exe`.
5. Use the wizard to complete the installation.
6. After installing HAXM, confirm that the virtualization driver is operating correctly by entering the following command in a Command Prompt window:

   ```
   sc query intelhaxm
   ```

   You should see a status message that includes the following information:

   ```
   SERVICE_NAME: intelhaxm
          ...
          STATE              : 4  RUNNING
          ...
   ```

   For more information, see Installation Instructions for Intel HAXM (https://software.intel.com/en-us/android/articles/installation-instructions-for-intel-hardware-accelerated-execution-manager-windows) .

You can adjust the amount of memory available to the Intel HAXM kernel extension by running the installer again.

You can stop using the virtualization driver by uninstalling it. To remove the software, run the installer or use the Control Panel.

Configuring VM acceleration on Mac

To use VM acceleration on a Mac, you must install the Intel HAXM kernel extension to allow the emulator to make use of CPU virtualization extensions. Android Studio requires Mac OS X 10.8.5 or higher, up to 10.11.4 (El Capitan) Mac OS X; the kernel extension is compatible with Mac OS X 10.6.0 and higher.

To install the Intel HAXM kernel extension, follow these steps:

1. Open the SDK Manager (https://developer.android.com/studio/intro/update.html#sdk-manager).
2. Click the **SDK Update Sites** tab and then select **Intel HAXM**.
3. Click **OK**.

`sdk/extras/intel/Hardware_Accelerated_Execution_Manager/IntelHAXM_version.dmg`.
To begin installation, in the Finder, double-click the **IntelHAXM.dmg** file and then the **IntelHAXM.mpkg** file.

5. Follow the on-screen instructions to complete the installation.

6. After installation finishes, confirm that the new kernel extension is operating correctly by opening a terminal window and running the following command:

   ```
   kextstat | grep intel
   ```

   You should see a status message containing the following extension name, indicating that the kernel extension is loaded:

   ```
   com.intel.kext.intelhaxm
   ```

   For more information, see Installation Instructions for Intel HAXM (https://software.intel.com/en-us/android/articles/installation-instructions-for-intel-hardware-accelerated-execution-manager-mac-os-x) .

You can adjust the amount of memory available to the Intel HAXM kernel extension by running the installer again.

You can stop using the virtualization kernel driver by uninstalling it. Before removing it, shut down any running x86 emulators. To unload the virtualization kernel driver, run the following command in a terminal window:

```
sudo /System/Library/Extensions/intelhaxm.kext/Contents/Resources/uninstall.sh
```

Configuring VM acceleration on Linux

Linux-based systems support VM acceleration through the KVM software package. Follow the instructions for installing KVM on your Linux system, and verify that KVM is enabled. For an example, see Ubuntu KVM Installation (https://help.ubuntu.com/community/KVM/Installation) .

Running KVM requires specific user permissions. Make sure you have sufficient permissions as specified in the KVM installation instructions.

To run an accelerated emulator on a Linux computer, it must also meet these requirements:

- Intel processor with support for Intel VT-x, Intel EM64T (Intel 64), and Execute Disable (XD) Bit functionality enabled, or
- AMD processor with support for AMD Virtualization (AMD-V)

You can use the emulator `-accel-check` (#accel-check) command-line option to check whether you have KVM installed. Alternatively, you can install the `cpu-checker` package containing the `kvm-ok` command. For example:

```
$ sudo apt-get install cpu-checker
$ egrep -c '(vmx|svm)' /proc/cpuinfo
12
$ kvm-ok
INFO: /dev/kvm exists
KVM acceleration can be used
```

If KVM is missing or to ensure you have the latest KVM installed, enter the following command line to install it:

```
$ sudo apt-get install qemu-kvm libvirt-bin ubuntu-vm-builder bridge-utils ia32-libs-multiarch
```