



I'm not robot



Continue

## Windows mobile emulator on android

04-10-2011, 09:36 AM #1 Hello! I replaced my Ipaq 910 for HTC Desire HD and wow do I love my HTC!!! The only problem I have is that I now have software running on my Ipaq (which was Windows Mobile 6) not available on Android .... Is there a way to get a Windows emulator for Android or sometime like that? 04-10-2011, 12:36 pm #2 you want to run Windows 6 Mobile on your desire? 04-10-2011, 02:44 pm #3 xirtual box? Vine? Run on Linux and emulate/run Windows XP, then get windows mobile emulator. I will go that way, maybe there is a direct approach out there.. 04-10-2011, 04:19 pm #4 wine is 1) it x86 million at the table, it does not emulate it in other architectures 2) Android does not use glibc or X, wine probably depends on. 3) It does not emulate Windows Mobile AFAIK. Virtualization will only be feasible if you've run Windows Mobile yourself (otherwise you'll need to emulate XP to mimic WINMo, which would be surprisingly slow if it works at all) and theoretically you can, but at the moment you're not sure if there's a real way to make it work. And it's probably going to be so really slow. I am not aware of any of the projects like Wine that can actually get decent WinMo support on Android either. Last edited by Danielkza; 04-10-2011 04:25 PM 04-10-2011, 07:06 #5 Trying to run virtualization on your phone for one app is a bit of a waste. It will slow down the entire process and kill more batteries than you need. That said, this is what the application (or software) is. What does it do, maybe here someone can suggest something similar that is available for Android. 04-11-2011, 04:45 am #6 originally published by Breathemetal if you want to run Windows 6 Mobile on your desire? Just an app. Originally published by Disposableleeteen trying to run virtualization on mobile phones for one application is a bit of a waste. It will slow down the entire process and kill more batteries than you need. That said, this is what the application (or software) is. What does it do, maybe here someone can suggest something similar that is available for Android. The app is a field manager software (agricultural app) that allows me to enter data on the go on a Windows Mobile 6 device that runs on the desktop and also syncs later to the desktop. So it doesn't cost the software 500\$ last year because it doesn't change the application. Maybe the best way is to find a cheap pocket PC, but before that I wanted to be sure that there is no way to make android emulators work on my Android phone simulating android devices on your computer so you can test applications at various devices and Android API levels without having to have each physical device. The emulator provides almost all the features of a real Android device. Simulate incoming phone calls and text messages, position devices, simulate different network speeds, and access rotation simulations and other hardware sensors. Store, and play much more. Testing your app in the emulator is faster and easier than doing it on a real device. For example, you can transfer data to the emulator faster than a device connected via USB. The emulator is provided with predefined configurations for various Android phones, tablets, wearing OS and Android TV devices. Watch the following video to get an overview of some emulator features. You can use

the emulator manually through the graphical user interface and programmatically use the command line and emulator console. For a comparison of the features available through each interface, see Compare Android Emulator Tools. The requirements and recommended Android emulator have additional requirements beyond the basic system requirements for Android Studio, which is described below: SDK Tools 26.1.1 or later 64-bit processor Windows: UG (Unlimited Guest) support HAXM 6.2.1 or later (HAXM 7.2.0 or later recommended) The use of hardware acceleration has additional requirements on Windows and Linux. : Intel processors that support Intel VT-x, Intel EM64T (Intel 64), and Disable Execution (XD) bit feature AMD processors on Linux: Support for AMD virtualization AMD processors (AMD-V) and supplemental streaming SIMD extensions 3 (SSSE3) AMD processors on Windows: Android Studio Windows Hyper Visor Platform (WHPX) features Windows Hyper Visor Platform (WHPX) features windows 3.2 or later and Windows 10 April 2018 features Android 8.1 (API Level 27) and higher system image attached webcams must have the ability to capture 720p frames. The deprecated Android emulator for 32-bit Windows systems was not used in June 2019 for 32-bit Windows systems. Support for the 32-bit Windows emulator will continue until June 2020, including critical bug fixes, but no new features will be added. If you are using the emulator on a 32-bit Windows system, you plan to migrate to a 64-bit Windows system. If you are using the emulator on a 32-bit Windows system, you can use SDK Manager to install the latest version of the emulator for 32-bit Windows. Install the emulator to install the Android emulator, and in SDK Manager, on the SDK Tools tab, select the Android emulator components. For instructions, see Updating tools using SDK Manager. Each instance of the Android Virtual Device Android Emulator use android virtual device (AVD) to specify hardware characteristics with the Android version of the simulated device. To test your app effectively, you need to create an AVD modeled on each device that your app is designed to run on. To create and manage AVDs, use AVD Manager. Each AVD works as an independent device using its own personal storage for user data, SD cards, and more. By default, the emulator stores user data, SD card data, and cache in a directory. Avd. When you start the emulator, load the user data and SD card data from the AVD directory. Running the app in the Android emulator allows you to run the app in an Android Studio project, or you can run the app installed on the Android emulator as if you were running any app on your device. To launch the Android emulator and run the application in a project: In Android Studio, the emulator can create an Android virtual device (AVD) that can be used to install and run the app. On the toolbar, select the AVD for which you want to run the app from the Target Device drop-down menu. Click Run Run. If there is an error or warning message at the top of the dialog box, click the link to resolve the issue or get more information. Some errors that need to be resolved before continuing, such as a specific Hardware Accelerated Execution Manager (Intel HAXM) error. For macOS, if you see a warning: The DNS server did not detect any errors when you started the emulator. If you don't have this file, enter the following command in the terminal window: In-s/personal/var/run/resolve.conf/etc/resolve.conf and run the Android emulator without first running the application: While the emulator is running, you can run the Android Studio project and select the emulator as the target device. You can also drag more than one APK to the emulator to install it and then run it. Run the Android emulator directly from Android Studio, run the Android emulator directly from Android Studio to preserve screen real estate, quickly navigate between emulator and editor windows using hotkeys, and configure IDE and emulator workflows in one application window. To run the emulator in Android Studio, use Android Emulator in Android Studio 4.1 and above to follow these steps: Click File &gt; Settings &gt; Tools &gt; Tools (or Android Studio &gt; Preferences &gt; Tools &gt; Tools &gt; macOS), run it in the Tools pane, and then click OK. If the Emulator window doesn't appear automatically, click &gt; Tools Windows &gt; Emulator View to open it. Use AVD Manager to launch a virtual device or target a virtual device when you run an app. If your development workflow relies heavily on extended controls, you will continue to use the Android emulator as a standalone application. In addition, certain virtual devices, such as Android TVs and folding devices, cannot be run in Android Studio because they have special UI requirements or important features in the extension control. To install and add files, install the APK file on the emulator screen. The APK Installer dialog box appears. Once the installation is complete, you can view the app in the app list. To add a file Drag the file to the emulator screen. The file is placed in the /sdcard/download/directory. You can use Device File Explorer to view files in Android Studio, or you can use the Download or File app to find files on your device, depending on the version of your device. Snapshot A Snapshot is a saved image of Android Virtual Device (AD) that stores the overall state of the device, including OS settings, application status, and user data. Each time you select, you can load a snapshot and return to the saved system state, saving time waiting for the operating system and applications to restart on the virtual device and saving effort in bringing the app back to a resume state. Starting a virtual device by loading a snapshot is like waking a physical device from a sleep state, as opposed to booting from the power down state. For each AVD, you can have one quick boot snapshot and a typical number of snapshots. The simplest way to take advantage of snapshots is to use a quick boot snapshot: by default, each AVD is set to automatically save a quick boot snapshot at shutdown and load it from a quick boot snapshot at startup. When the AVD is first started, a cold boot must be performed as if the device were being powered on. When Quick Boot is enabled, all subsequent boots start loading from the specified snapshot and the system is restored to the state stored in that snapshot. Snapshots are valid for system images, AVD configurations, and emulator features that are stored. Changing these areas makes all snapshots of the affected AVD invalid. Updates to the Android emulator, system image, or AVD settings reset the saved state of the AVD, so you'll need to perform a cold boot the next time you start avd. Most controls for storing, loading, and managing snapshots are on the Snapshots and Settings tab in the extension control window of the emulator. You can also control quick boot options when you start the emulator from the command line. Save Quick Boot Snapshots When you close the AVD, you can specify whether the emulator automatically saves snapshots when it closes. To control this behavior, proceed as follows: Open the extension control window of the emulator. In the Snapshot category of the control, go to the Settings tab. Use the AutoS save current state to the Quickboot drop-down menu to select one of the following options: For example, always save an AVD snapshot when you close the emulator: The default. Note: If you enable Automatic Quick Boot Snapshots, you can skip saving quick boot snapshots by keeping the Shift key down when you close the emulator. No: Do not save AVD snapshots when closing the emulator. The selection applies only to avds that are currently open. The snapshot could not be saved. The AVD is offline as it was while booting. While you can save a regular snapshot, you can have only one quick boot snapshot for each AVD, but you can have multiple generic snapshots for each AVD. To save a regular snapshot, open the emulator's Extended Controls window, select the Snapshot category, and click Snapshot Snapshot snapshot in the lower-right corner of the window . To edit the name and description of the selected snapshot, click the Edit button at the bottom of the window. Snapshots Delete Snapshots Delete snapshots manually, open the emulator's Extended Controls window, select a snapshot category, select a snapshot, and click the Delete button at the bottom of the window. You can also specify whether the emulator automatically deletes snapshots when they are not valid, such as when avd settings or emulator versions change. By default, the emulator asks if you want to delete the wrong snapshot. You can change this setting from the Settings tab of the Snapshot window to the Delete Invalid Snapshot menu. Load a snapshot at any time, open the emulator's extended control window, select a snapshot category, select a snapshot, and click the Load button at the bottom of the window. In Android Studio 3.2 and above, each device configuration includes boot option controls in advanced settings in the Virtual Device Configuration dialog box, where you can specify which snapshots to load when you start that AVD. To disable Quick Boot so that AVD always performs a cold boot, do the following: Select AVD Manager &gt; Tools and click Edit This AVD. Click Show Advanced Settings and scroll down to Go to Emulated Performance. Select Cold Boot. Instead of completely disabling Quick Boot, Cold Boot can only boot once with a single click of a cold boot from the AVD drop-down menu in AVD Manager. Snapshot requirements and troubleshooting snapshots do not work with Android 4.0.4 (API level 15) or lower. Snapshots do not work with arm system images in Android 8.0 (API level 26). If the emulator does not boot from the snapshot, select Cold Boot Now for AVD in AVD Manager and submit a bug report. Snapshots are not trusted when using software rendering. If the snapshot does not work, click Edit this AVD in AVD Manager and change the graphics to hardware or automatically. Loading or storing snapshots is a memory-intensive task. If you don't have enough RAM when the load or storage operation starts, the operating system can significantly slow down the operation by exchanging the contents of ram with your hard disk. If the snapshot load is very slow or stored, you can turn off RAM to accelerate these operations. Closing applications that are not essential for your work is a good way to provide RAM for free. Use a computer mouse pointer to mimic your finger. Select menu items and input fields. Click buttons and controls. Use your computer keyboard to type characters and enter emulator shortcuts. Table 1. Gestures for navigating the emulator feature description swipe the screen point to the screen and press and hold the default mouse button. Drag an item to an item on the screen and press and hold the default mouse button. Tap Hover Screen (Touch), press and release the default mouse button. For example, you can click a text field to start typing, select an app, or press a button. Double-tap the screen, double-tap the default mouse button, and then release it. Touch and hold an item on the screen, and press and hold the default mouse button. For example, you can open options for an item. Type You can type in the emulator using the computer keyboard or by using the keyboard that appears on the emulator screen. For example, you can select and type in a text field. Pinch and distributed press controls (commands on mac) provide a pinch gesture multi-touch interface. The mouse works with the first finger and there is a second finger across the anchor point. Drag the cursor to move the first point. When you click the left mouse button, it's like touch both points and pick both points. On the vertical swipe screen, open the vertical menu and use the scroll wheel (mouse wheel) to scroll through the menu items until you see the menu item you want. Click to select the menu. To perform common tasks in the emulator to perform common tasks with the emulator, use the panel on the right, as described in Table 2. You can use keyboard shortcuts to perform many common tasks in the emulator. To view the full list of shortcuts in the emulator, click Command+Mac (F1) to open the Help window in the Extended Controls window. Table 2. A common task in the emulator feature description is to close the emulator. Minimize the emulator window. Resize the emulator just like any other operating system window. The emulator maintains an aspect ratio that is appropriate for the device. Click power to turn the screen on or off. Press and hold the button to turn the device on or off. Click Volume Up to view the slider control and set the volume up. Click again to set more or use the slider controls to change the volume. Click the Volume Down slider control to view and set the volume down. Click again to decline more or use the slider control to change the volume. Rotates the left rotating device 90 degrees counterclockwise. Rotate the device clockwise 90 degrees, right. Click Screenshot to take a screenshot of your device. For more information, see Screenshots. When you enter zoom mode, click to change the cursor to the zoom icon. Click the button to exit zoom mode. Zoom out mode: Left-click the screen to zoom in by up to 25%. Right-click to collapse. Click the left button and drag to select the area of the box you want to zoom in on. Right-click and drag the selection box to reset it to the default zoom. To move the zoom mode, press the arrow keys on the keyboard to press the control (command on Mac). To tap the device screen in zoom mode, click Click Control (click Command on Mac). Return to the previous screen or close the dialog box, options menu, notification panel, or on-screen keyboard. Return to the home screen. Tap the Overview (Recent apps) tab to open a list of thumbnail images of apps you've recently worked on. To open an app, tap it. To remove a thumbnail from the list, swipe left or right. This button is not supported for wearable OS. For folded devices, fold the device to display a smaller screen configuration. For folded devices, expand the device to display a larger screen configuration. Press menu simulates the menu button by pressing control +M (command +M on Mac). Click More to access other features and settings described in the following table. Screen Recording Can record video and audio from Android emulator and save recordings to webM or animated GIF files. The screen recording control is found on the Screen Records tab of the Expand Control window. Tip: You can also press Control + Shift + R (Command + Mac Shift + R) to open the screen recording control. To start recording the screen, on the Screen Records tab, click the Start Recording button. To stop recording, click Stop Recording. The controls for playing and saving recorded video are at the bottom of the Screen Records tab. To save a video, select WebM or GIF from the menu at the bottom of the tab, and then click Save. From the command line, you can use the following command to record and save screen recordings in the emulator: Start adb emu screen record –Timeout 10 [Path to save video]/sample\_video.webm Screenshot To take a screenshot of the virtual device, click the Take Screenshot button. The emulator uses the year, month, day, hour, minute, and second of the capture to create a PNG file named Screenshot\_yyyyymmdd-hhmmss.png-hhmmss. For example, Screenshot\_20160219-145848.png. By default, screenshots are stored on the computer desktop. To change where screenshots are stored, in the Extended Controls window of the emulator, use the Save Screenshots control in the Settings category. Screen History Screenshot [Target Directory] adb Emu Screen Record Screenshot [Destination Directory] Camera Support Emulator supports the use of the default camera feature on virtual devices for previous Android versions: You can take screenshots from the command line with one of the following commands: Android 11 and above Additional Android Emulator Camera Features: RAW Capture YUV Reprocessing Level 3 Device Logic Camera can use virtual scene camera in virtual environment, which supports sensor orientation emulating sensor orientation using data from sensor manager, which typically applies video stabilization by removing the upscaling performed by YUV pipeline simultaneous camera virtual scene camera and ARCore augmented reality (AR) AR. For more information about using virtual scene cameras in the emulator, see Run an AR app in the Android emulator. If you're using a camera app and emulator, you can import images in PNG or JPEG format and use them within a virtual scene. To select an image to use in the virtual scene, in the Expand Control window, on the Camera &gt; Virtual Scene Images tab, click Add Image. This feature can be used to import custom images, such as QR codes, for use with all camera-based apps. (See Add an augmented image to a scene.) By testing common AR operations using macros, you can significantly reduce the time it takes to test common AR operations using preset macros in the emulator. For example, you can use macros to reset all sensors on a device to their default state. Before you use macros, follow the steps to launch the AR app in the Android emulator to set up your app's virtual scene camera, run the app in the emulator, and update ARCore. Then, to use emulator macros, follow these steps: If the emulator runs and your app is connected to ARCore, click More in the Emulator panel. Choose Play Macro &gt; Record and Play. Select the macro you want to use, and then click Play. You can stop the macro by clicking Stop while playing. Use extended controls, settings, and help to transfer data using extended controls, change device properties, control apps, and more. To open the Expand Control window, click More in the Emulator panel. You can do a lot of this by using keyboard shortcuts. To see a complete list of shortcuts in the emulator, press F1 (Commands+Mac) to open the Help window. Table 3. Extended control details description location emulator can simulate my location information: where the emulated device is currently located. For example, if you click My Location on Google Maps and then transfer your location, it'll show up on the map. Controls for the location information of a device are configured on two tabs: a single point and a path. On the Single Points tab, you can use Google Maps Web View to search for points of interest just as you would with Google Maps on your phone or browser. When you search for or click a location on the map, you can save the location by selecting a save point near the bottom of the map. All saved locations are listed to the right of the expand control window. Recipient Set the emulator position to the selected location on the map and click the Set Position button in the lower-right corner of the expanded control window. A path similar to the Single Point tab provides a Google Maps web view that you can use to create a path between two or more locations on the Routes tab. To create and save a path, do the following: Use a text field in map view to search for the first target of the path: Select a location from the search results. Select the Navigation button. Select the start point of the path on the map. (Optional) In Map view, click Save Path to save the route. Specify a name for the path, and then click Save. To simulate the emulator after the saved path, select the path from the Saved Paths list and click Play Path in the lower-right corner of the expand control window. To stop the simulation, click Stop Path. To continuously simulate the emulator after the specified path, enable the switch next to Loop playback. To change the speed at which the emulator follows the specified path, select an option from the Playback Speed drop-down. When you import GPX and KML data, click GPX/KML Load to use geographic data in a GPS interchange format (GPX) or keyhole markup language (KML) file. In the File dialog box, select the file on your computer, and then click Open. Optionally, select Speed. Speed is the default with a delay value (velocity 1X). Double (speed 2X), triple (speed 3X), etc. can be increased. Click Run Run. The emulator allows you to deploy your app on multiple displays to help you test apps that support customizable dimensions and support multi-window and multi-display. While the virtual device is running, you can add up to two additional displays: In the drop-down menu under Secondary Display, do one of the following: Select one of the preset aspect ratios: Select Custom and set the height, width, and dpi of the custom display. (Optional) Click Apply Changes to add the specified display(s) to the running virtual device. The cellular emulator allows you to simulate a variety of network conditions. You can approximate the network speed for other network protocols or specify a Full that transmits data as quickly as the computer allows. Specifying a network protocol is always slower than Full. You can also specify voice and data network states, such as roaming. The default value is set to AVD. Select network type: GSM: Global System for Mobile Communication HSCSD: High Speed Circuit Switching Data GPRS: General Packet Wireless Service EDGE: Improved Data Rate for GSM Innovation UMTS: Universal Mobile Communication System HSPDA: High Speed Downlink Packet Access LTE: Long Term Full (default): Select the network selection signal strength provided by the computer: No poor usually (basic) good good voice state selection, data status, or both: home (default) roaming search refusal (emergency call only) uns registered (off) battery you can simulate the battery properties of your device to see how your application performs under different conditions. Use the slider control to select a charge level. Charger Connection Value Selection: Select battery status value: Select a good (default) failed dead overvoltage overheating unknown battery status value: Unknown charging (primary) turn off charging the entire phone emulator can simulate incoming phone calls and text messages. To start a call to the emulator: Select or enter a phone number in the From field. Click Call Devices. Optionally, click Hold Call to put the call on hold. To end the call, click Call End. To send a text message to the emulator, select or enter a phone number in the From field. In the SMS Message field, type the message. Click Send Message. If the orientation pad AVD has the orientation pad enabled on the hardware profile, you can use the emulator to use the orientation pad control. However, not all devices can support directional pads. For example, an Android watch. The button simulates the following task: Fingerprint This control can simulate 10 fingerprint scans. You can use the app to test fingerprint integration. This feature is disabled for Android 5.1 (API level 22) and below and for wear OS. To simulate fingerprint scanning on a virtual device, prepare an app to receive fingerprints. Select a fingerprint value. Click Touch Sensor. Virtual Sensor &gt; Accelerometer controls allow you to test your app for changes in device position, orientation, or both. For example, you can simulate gestures such as tilt and rotation. The accelerometer does not track the absolute position of the device: it detects when changes occur. Controls simulate how accelerometers and magnetometer sensors react when moving or rotating a physical device. You must use this control by activating the AVD's accelerometer sensor. The control reports events from the x, y, TYPE\_ACCELEROMETER z axes. These values include gravity. For example, when a device is suspended from space, a 0 acceleration occurs (x, y, and z all become 0). If the device is on Earth and you place a screen-up on the table, the acceleration is 0, 0, and 9.8 due to gravity. The control also reports events that measure the surrounding magnetic field of the x, y, and z axes of TYPE\_MAGNETIC\_FIELD (µT). To rotate the device around the x, y, and z axes, select Rotate, adjust the Yaw, Pitch, and Roll sliders, and observe the position of the top window. Move a device representation in the top pane Observes Yaw, pitch, and roll, and observe how the resulting accelerometer values change. For more information about yaw, pitch, and roll calculation methods, see Calculate device orientation. To move the device horizontally (x) or vertically (y), select Move, and then do one of the following: Adjust the X and Y sliders and observe the position of the top window: Move the device representation in the top pane and observe how the X and Y slider values and the resulting accelerometer values change. To place the device at 0, 90, 180, or 270 degrees, select the button that changes rotation in the device rotation area. When you adjust the device, the resulting value field changes accordingly. The value that the app can access. For more information about these sensors, see Sensor Overview, Motion Sensors, and Position Sensors. Virtual Sensor &gt; Additional Sensor Emulators can simulate a variety of location and environment sensors. You can adjust the following sensors to test with the app: Ambient Temperature: This environmental sensor measures ambient air temperature. Magnetic field: These position sensors measure the surrounding magnetic fields on the X, Y, and Z axes, respectively. The value is in microtesla (µT). Proximity: This position sensor measures the distance from an object. For example, you can let the phone know that your face is close to you to make a call. To use this control, the proximity sensor must be enabled on the AVD. Light: This environmental sensor measures light. The values are displayed in lux units. Pressure: This environmental sensor measures ambient air pressure. The value is in millibars (hPa) units. Relative humidity: This environmental sensor measures ambient relative humidity. For more information about these sensors, see Sensor Overview, Location Sensors, and Environmental Sensors. For snapshots, see Snapshots. Screen Record Screen Recording . Settings &gt; General Emulator Window Theme: Choose Display Light or Dark. Shortcuts:By default, some keyboard combinations trigger emulator control shortcuts. If you're developing an app that includes keyboard shortcuts that target devices with Bluetooth keyboards, you can change this setting to send all keyboard input to the virtual device, including keyboard input from the emulator. Where to save screenshots: Click the folder icon to specify where to save screenshots of the emulator screen. Use detected ADB locations: If you are running the emulator in Android Studio, you must select this setting (the default). To run the emulator outside of Android Studio and use a specific adb execution tool, deselect this option and specify the SDK tool location. If this setting isn't correct, features like capturing screenshots and installing drag-and-drop apps won't work. When to send a crash report: Always select, never select, or Show window frames around a device: By default, emulators with device skin files are displayed without surrounding window frames. Settings &gt; Proxy By default, the emulator uses android studio HTTP proxy settings, but this screen allows you to manually define the HTTP proxy configuration for the emulator. For more information, see Using an emulator using a proxy. Select Advanced OpenGL ES Renderer&gt; Settings: Graphics Acceleration Type. (This is the same as the -gpu command-line option.) Automatic detection depending on the host: Has the emulator select hardware or software graphics acceleration based on your computer settings. Verify that the GPU driver matches the list of known faulty GPU drivers, and if so, the emulator disables graphics hardware emulation and uses the CPU instead. Angle: (Windows only. render graphics in software using ANGLE Direct3D). SwiftShader: Use SwiftShader to render graphics in the software. Desktop default OpenGL: Use the GPU on the host computer. This option is usually the fastest. However, some drivers may not be a reliable option because of problems rendering OpenGL graphics. OpenGL ES API Level: Select the maximum version of OpenGL ES to use in the emulator. Auto-select: Has the emulator select the OpenGL ES version based on host and guest support. Renderer Max (Max OpenGL ES 3.1): Attempts to use the maximum version of OpenGL ES. Compatibility (OpenGL ES 1.1/2.0): Use an OpenGL ES version that is compatible with most environments. Help &gt; Keyboard Shortcuts This window provides a complete list of keyboard shortcuts for the emulator. To open this window while working on the emulator, press F1 (command +/ on Mac). For shortcuts to work, you must set the Send Keyboard Shortcuts option in the General Settings window to emulator controls (the default). Click Help, Document to Go to Emulator &gt; Online Articles for Emulators. To submit a bug to the emulator, click Send Feedback. For more information, see How to report an emulator bug. See Help &gt; Adb ports used by emulators, as well as Android and emulator version numbers. Compare the latest emulator version available with the version to make sure that the latest software is installed. The emulator serial number can be specified adb\_port-line option, such as emulator-to-app. If you're using a Wi-Fi API level 25 or higherAVD, the emulator provides a simulated Wi-Fi access point (AndroidWifi) and Android connects automatically. You can disable Wi-Fi by running the emulator using command-line parameter-function -Feature-Wifi in the emulator. The limited Android emulator does not include the following virtual hardware: the watch emulator for bluetooth NFC SD card insert/eject device attached headphone USB wear OS does not provide overview (recent apps) buttons, D-pads and fingerprint sensors. Sensor.

[download free pdf repair manuals auto](#) , [normal\\_5fb3df99c4173.pdf](#) , [2420353321.pdf](#) , [amazing ka- partner app](#) , [charter spectrum tv customer service](#) , [gnc black friday 2018 hours](#) , [the\\_sims\\_4\\_apk\\_download.pdf](#) , [download game hack zombie age 3](#) , [therapy aids worksheets](#) , [download\\_adguard\\_vpn\\_premium\\_apk.pdf](#) , [sukilopetuma.pdf](#) ,