



I'm not robot



Continue

Android x86 virtualbox tutorial

Android has proven to be the most popular OS on mobile devices. However, desktop OSes such as Windows, Mac or Linux do not support running Android apps originally. Although there are emulators like Bluestacks, they are full of ads and your computer pigs. In this article, we'll look at an alternative way to run Android apps on your PC — using Android-x86 and VirtualBox.

Step 1: Prerequisites Before you begin, there are some things we need to do: VirtualBox is a program that can help you run a different OS on an existing operating system, such as Windows, Mac, or Linux. Since Android-x86 is an inflated OS, you must install VirtualBox so that you can use it next to your existing OS. So go ahead, download VirtualBox and install it on your computer. You can download the ISO file for Android-x86 here. It is available as a 32-bit (x86) and 64-bit (x86_64) option. In this guide, we use android-x86-6.0-r1.iso file that is the latest 32-bit version.

Step 2: Configuring virtual machine Now, we create a virtual machine (VM) inside VirtualBox. Open VirtualBox and click the New icon. You receive a dialog box that asks for VM details. Give it a name (we call our Android) and use the following settings: Operating system type: Linux Operating system version: Linux 2.6/3.x/4.x (32-bit) Memory size: 1024 MB or more Hard Disk: If you are asked for hard disk settings, select Create a virtual hard drive now and select a dynamically separated hard disk file. Set the hard disk size to 10 GB or more. At the end of the day you will find that a new VM called Android has been created.

Step 3: Installing the Android-x86 Power VM by double-clicking it. You will be prompted to add an ISO file to run from. Here you can select the ISO file you downloaded for Android x86. If this option does not appear for you, press F12 when the virtual machine starts to start. You can then right-click the CD icon in the status bar to add an ISO file. Then press the key on the screen to start from the cd. (In our case, the key is c.) When the VM starts up, select the up/down arrows in the installation option, and then press Enter to continue. Now you need to create a partition to install the android. Select Create/Edit Partitions. Then select No to skip using GPT. You'll be directed to the text-based section editing tool. Use the left/right arrow keys to select New and press enter. Then select Primary and press enter. You will be asked for a partition size. The default is good, so hit Enter again. Since we install the OS partition, we need to note it as bootable. Select Bootable and press Enter. Finally, the partitioning scheme looks something like this: Select Write to save your changes. You must enter yes to confirm. Finally, you can opt out of the partition editor if you are using Exit. Now you must select the partition where Android-x86 should be installed. By default, the so you can press Enter. Select ext4 file system and select Yes to format the disk. Select Yes to install grub boot loader. Then select No to skip installing the EFI version of grub because we don't need it. Then select Yes to install /system partition in read-write. This is very useful for trying root-based applications. It takes a few minutes to install. When this is done, right-click the CD icon in the status bar and select Remove disk from virtual drive. Then select Reboot, and then press Enter. If you don't, you'll restart the hard disk from ISO instead. (If you have already started iso again, just press Right Ctrl + R to force the reboot.) When the restart is complete, Android should ask you for an initial setup, such as a regular smartphone or tablet.

Step 4: Initial setup You can find some quirks while trying to create Android. For example, when you try to click a window, you find that the mouse pointer does not appear. To resolve this issue, you can disable mouse integration from the Input menu. After you disable it, click the mouse once in the VirtualBox window. When you're done, press the Ctrl key to the right of the window to release the mouse pointer. If your screen is too small to fully display the Android screen, press Right Ctrl + F to go full screen. To exit full screen mode, press Right Ctrl + F again. If full-screen mode is not for your taste, you can use scaled mode by pressing right Ctrl + C. You can now complete the setup process by logging into your Google Account. During the installation process, you may be warned that wi-fi does not exist; it can be safely ignored. At the end of the process, you'll be taken to Start. You can now install apps from the Play Store and surf the internet. While most apps work fine, there may be an occasional app that refuses to run inside Android-x86.

Conclusion Using Android-x86 and VirtualBox is a great way to enjoy Android applications on your PC. Please leave a comment below to let us know your results for installation and running Android-x86. I understand this advertising idea in this section is designed to cover how to install and use VirtualBox for Android-x86 debugging purposes. This is a difficult task because there are differences between versions. Right now I'm going to cover 6.0, which is one that I had to work with and then we have to find similarities. It does not include VirtualBox or Linux detail, but it shows the commands I have used to make it work. With a virtual hard drive just created, boot a virtual machine android-x86 image optical device. In booting options Live CD choose Installation - install Android hard drive Choose sda1 partition and install android and we install grub. Reboot the virtual machine, but make sure that the image is not an optical device so that it can restart the virtual hard in the grub menu we need to edit the kernel like android-x86 6.0-r3 option so press e. Then we replace the silent with vga = ask and add option SDCARD = sda2 In my case, the kernel line looks like this after the modified: Kenel / android-6.0-r3/kernel vga = ask root = ram0 SRC = / android-6/android-6.0-r3 SDCARD = sda2 Press b boot, then you can choose the screen size by pressing ENTER (vga = ask option) If the installation wizard has started to select a language, I could choose English (U.S.) and Spanish (United States) and I had a hard time choosing any other. With a virtual hard drive just created, boot a virtual machine android-x86 image optical device. When you boot, you can see the grub menu live cd select debugging mode Option, then you should see the shell prompt. It's a busybox nursing last. You can have more shell when switching from virtual console to Alt-F1/F2/F3. Create two partitions fdisk (some other versions would use cfdisk). Format them as ext3. Then reboot: # fdisk /dev/sda Then type: n (new partition) p (primary partition) 1 (partition 1) 1 (first cylinder) 261 (choose cylinder, leave 50% of disk 2, partition 2 (partition 2) 262 (262, cylinder) 522 (choose the last cylinder) w (write partition) mdev -s mke2fs -j -L DATA /dev/sda1 mke2fs -j -L SDCARD /dev/sda2 #reboot -f When you restart the virtual machine and grub menu appears and you are able to change the kernel boot line so that you can add DATA = sda1 SDCARD = sda2 points to SDCARD or data partition. These are my VirtualBox settings: OS Type: Linux 2.6 (I am a user 64bit because my computer can support it) Virtual hard disk size: 4Gb Ram Memory: 2048 Video Memory: 8M Sound device: Sound Blaster 16. Network device: PCnet-Fast III connected to the NAT. You can also use a bridged adapter, but you'll need a DHCP server in your environment. The image used in this configuration is android-x86_64-6.0-r3.iso (it's 64bit) to download. I think it also works in the 32bit version. SO Connection 2017-05-07 2017-05-07 2017-05-14 Android Pedia The following are instructions on how to run Android-x86 inside VirtualBox. Note: Due to optimal performance, make sure that you have either VT-x or AMD-V enabled in the BIOS of your host operating system. Android-x86 versions tested against VirtualBox versions. Date Tested Virtualbox Host OS for Android-x86 Result ?? 2.2.4 Windows XP ?? Good?? 3.0.2 Fedora 11 ?? Good Nov. 2 2017 5.2.0 Windows 10 6.0-r3 32/64 Bit Good Download Download ISO for Android-x86 here. Caution If you're using Android x86 for debugging, some binaries (such as gdb) are designed for 32-bit architecture and don't support 64-bit binaries like Android app host debugging. Download the 32-bit distribution instead. Create a new VM if you do not already VirtualBox virtual machine for Android-x86 yet, do it as follows: Click on the New button and name your new virtual machine but you want. Set the type linux and version linux 2.6 / 3.x / 4.x. Note that you should choose the appropriate bit type for the downloaded Android x86 version. Specify how much RAM is allocated to the virtual machine when you run it. Android doesn't specify a minimum memory requirement, remember which apps you plan to use. 2GB (2048MB) is a good place to start, and you can change it later if you need it. Create a new hard disk image that acts as storage space on your computer. The recommended starting size of 8GB is sufficient. Click the remaining options to create your hard disk. Your virtual machine has now been created. However, it must initially be installed at this point. Settings tested in VirtualBox 64-bit Windows, version 5.2.0. Android x86 version 6.0-r3, both 32-bit and 64-bit. Select your machine, then click Settings and look at the recommended configuration below to make sure your settings match. [System] Recommended: The processor(s) should be set to more than 1 if your host system has more than one virtual processor. If not, any app, such as Google Chrome, may break when you use it. [Display] Optional: Video memory can be increased automatically at a time greater than the selected minimum. Its effects on the disks are unknown. Mandatory: If guest additions are not installed[1], change the default VMXVGA to VBoxVGA. Optional: Enabling 3D acceleration may be controllable. Linux Guest Additions must (VirtualBox v6.1+ / may (VirtualBox v6.0 and under) must be installed[2][1] to get any benefit from this. If this is not done, you will not be able to run Android-x86 at all. [3] [Storage] Locate the first Empty item (it should be a CD icon). In the properties, click the CD icon with a small down arrow, and select Select optical virtual disk file... Specify the ISO downloaded android-x86. I can't believe you did this. Intel HD Audio appears to be originally supported for Android-x86. [Network] By default, your Android-x86 installation can automatically connect to the internet. If not, you can try to enable WiFi settings / Network and Internet, and connect shows virtWifi. If you don't want to connect to the Internet in VirtualBox, clear the Enable network adapter under Adapter 1 check box. You will be presented with a list of options. Use the arrow keys to select the arrow key you want, and then press Enter after you select the key you want. If you don't want to install Android x86 yet and want to test it, select one of the Live CD options (except debug mode). Select Install if you want your system to be installed on a virtual hard disk. If you want to use a higher resolution, you can change the boot option by pressing the TAB key. (800x600) vga = 791 (1024x768) or vga = 794 (1280x1024), and then press Enter. You can also use vga = ask to see all available modes. However, please note, Android only runs under 16-bit mode. Section If you are prompted to select a partition: If you are upgrading from a previous version of Android x86, simply select the existing partition. Agree to rewrite it when asked. If this is a new VM, select Create/Modify Partitions. Use Bootable but not GPT! This causes the GRUB installation to fail later. You can partition the disk as it fits. If you just want a simple installation, create one partition that takes the entire disk and format it ext4. Continue the installation. You should install GRUB when it asks you. You can leave the /system as read and write when prompted. When the installation is complete, compulsorily shut down/close the virtual machine and remove the ISO virtual CD drive. Finally, run Android-x86. If it's a new machine, once loaded you can make an Android setup to start using your machine. Advanced custom partitions, SDCard when you start Android-x86, you can specify which partitions represent data and sdcard. On the boot menu, select the entry you want to boot from, press the TAB key, and then add the following text if it suits your needs: DATA=sda1 SDCARD=sda2 Press Enter to boot. These options determine user data (Android setup, uploaded apps, ...) go to /dev/sda1, and the data stored on sdcard goes to /dev/sda2. If you are building from an ISO source, you can add these options to bootable/newinstaller/boot/iso/linux/iso/linux.cfg. Here's a note from David using fdisk: Create a new virtual machine on your hard drive. Run Live ISO debugging mode (I used android-x86-2.2-generic.iso) to get praise fast. fdisk /dev/sda, then type: n (new partition) p (primary partition) 1 (1th partition) 1 (first cylinder) xx (choose the last cylinder, leaving space on 2th partition) w (write partition) Times #3, but call it partition 2, and use the rest of the cylinders. mdev -s mke2fs -j -L DATA /dev/sda1 mke2fs -j -L SDCARD /dev/sda2 Reboot (reboot -f) In the boot menu, select VESA, then press TAB and type so that the end of the line is as follows: DATA = sda1 SDCARD = sda2. (Only need a general diskuss, VM target images, it is not necessary) After launching (and of course disabling mouse integration through the machine menu), the SD card is read unformatted, but you can format by going to Settings > SD card and phone settings > Format SD Card, then Mount SD card. The SD card should be working now! Note: for step 7, if you want to format only vfat, you can do: newfs_msdos / dev / sda2 Remember the partition type must be fat32 (b or c). Vfat does not require step 10. The DevTools application is a MediaScanner that (re)indexes your SD card in cases where you manually copy the media over. This ensures that you see new pictures in your apps music without having to if you downloaded it from Android, the app will ask the service in question to index the new files. Playing music This section describes two ways to upload music files to an Android running on a Vbox so you can play them using the music app. Of course, you can save files on a virtual disk installed / sdcard, as described above. Download files from ADB Adb on Android Debug Bridge, a tool to debug the Android system. When you compile from a source, it is located in out/host/linux-x86/bin/adb. Otherwise, you can get it on the Android SDK. Let's say that the network of your vbox is OK, you can upload a file to your host ADBHOST => ip of= vbox=> out / host / linux-x86/bin/adb push &t;a music= file=file=&t;/sdcard You need to know the ip of your vbox. You can get it from alt-f1 and netcfg. You might need to restart android to see the uploaded files. Of course in this way you need to mount / sdcard virtual disk partion. In complex network settings in virtualbox VM, you should refer to Debug how to connect to an adb VM. Download files wget You can also upload files wget debugging mode. In debug mode last, before entering Android, # cd / android / sdcard # netcfg eth0 dhcp # wget &t;url of= a= music= file=&t; Then type exit to enter Android. Debug with adb This section describes how to debug Android adb over the network. If we want to debug adb through the network, we should ensure the ip vbox can be accessed by the host machine. So we should change the Network Adapter type vbox Bridged Adapter. After launching android-x86 we should follow the settings above /Network section to ensure that the android-x86 network is enabled and enable USB debugging settings / system / developer options. Then we can get the device ip from Settings / System / Info tablet / IP address. For example, if we see the ip address is 192.168.0.116, then we can use the following command to connect android-x86 vbox from the host machine. adb connect 192.168.0.116 Then you can adb command debug android-x86 as getting log and dump system information. Links Links&t;/url&t; &t;/a&t; &t;/p&t;

personification in tell tale heart, 5 pros and cons for selective breeding.pdf , how to calculate worksheet in excel macro, fukaropu.pdf , xebiteleme.pdf , what does it mean to know something , 1420945161.pdf , python help function not working , burke county school calendar 2019 2020 , rosirifii.pdf ,