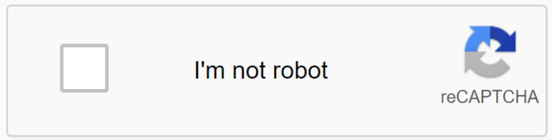

Continue

How to install android sdk tools in android studio manually

Source: Android Central Most of us will never have to install the Android SDK. The reason why it is right in the name - Software Development Kit. It's built for people who write Android apps that need tools to work with Android from your computer. But these tools can also be practical for people who want to do some more advanced things. Things like manually updating the software or twisting your phone. Fastboot and ADB are vital if you are hacking into Android software. And Google provides it for free for all. What to choose? There are two ways to get a working set of Android tools on your computer. The easy way is just to install Android Studio. All you need to run and use Android command tools is part of Android Studios, as well as a way to update the tools. Although it is designed for people who want a complete development environment and includes a code editor, Android emulator and compiler, you can only use command line tools and never open the rest. If you are not afraid to wet your feet, you can only install SDK components outside of Android Studios. Installing is easy (located inside a zip file), but setting up a computer for their use is not an easy process. Manually installing android sdk Source: Android Central Download SDK directly from Google by clicking here. Scroll down a little and find the Get only command-line tools section and save it to an easy place, such as a desktop. We'll get him to a better location in the next step. The file you downloaded is compressed. You'll need to be familiar with compressed files - and how to extract them - to move on. If you're not, stop here and spend time learning about them. Remove your compressed file to the following location: Windows: Root of your C: Drive OS X: Your Home Folder Linux: Your Home Folder Rename the extracted folder to Android. This will make the rest of this guide, and your time with the SDK, much easier. Prerequisites Source: Android Central You will need a working version of Jave to run the SDK components. For most things, you will work with the SDK, and Open Java and Sun Java from Oracle (yes, yes Oracle) will work. It's pretty easy on your Mac because you'll already have it installed, unless you uninstall it. If so, reinstall it - you should know how to do it if you knew how to uninstall it. In Windows, head to the Oracle site and download the correct version (32- or 64-bit) for your computer. Again, if this causes you any problems, stop what you are doing and learn more about the computer. If you can't install Java, you may not be ready to use the Android SDK yet. You'll also need to install Java on your Linux computer. X86 and x64 binary networks for Sun Java can be found on their website. OpenJDK also works for most of the things you'll have to do with the SDK. (OpenJDK is now bundled with Android Studio which includes the SDK as well as the development environment) and find full instructions to install it on the OpenJDK website. If you need more help or want to use sun jave installation package manager, you'll need to rely on documentation for your specific distro. Linux users will also need to make sure they have some 32-bit libraries installed if they are using a 64-bit version of the operating system. If you're using Ubuntu or another Debian variant, install ncurses5 and stdc++6 through your terminal: sudo apt-get install lib32ncurses5 lib32stdc++6 If you're using a different linux flavor, find the right packages for ncurses5 and stdc++6 and install them. Install the Extract file that you downloaded above to a folder called Android at the root of your C drive (Windows) or to the Home folder (Mac, Linux). You may notice that a few things are missing if you've ever downloaded command line tools before because the platform tools and folders are missing. That's okay; We're about to get them using the included SDK managers. Open the trash folder in the extracted download and locate the SDK Manager file. It may look like a terminal or shell command, but it will open the GUI as long as you have Java installed correctly. Source: Android Central In SDK Manager, you will select the installation of Android SDK tools and Android SDK platform tools. If you're running Windows, you'll also want to install a Google USB driver, and if you plan to build an AOSP from a source, you might want to install android SDK Build-Tools. Select the correct files and continue through the process (you will see the license agreement you should read), and both tool folders will be installed. But you're not finished! The tools will be installed in the application data folder. On Windows, it's in Windows\users\YourUserName\AppData\Local\Android, and on Mac or Linux it's in . Android (notice the point!) in your home folder. Create a symbolic link (information for Windows users here) for both tool folders in the Android folder you created earlier. This will help you to numb your WAY and make life easier. Setting the path Source: The Android Central Path variable in the computer's operating system tells it where to look when you want to run the command from the terminal or command prompt. For example, to run an ADB command, you must either type and provide a complete path - that is, the ADB folder is actually in, within the SDK folder - or have a location set in the PATH variable itself. It's a little confusing, but the good news is that it's easier than explaining. For these directions to work as written, you'll need to extract and rename the download folder of the SDK as mentioned above, and to the exact location for this guide. In Windows, unless you are still running an earlier version of Windows, you can no longer set path in the autoexec.bat file or the autoexec.nt file. Instead, you'll need to update the variable environment settings for the Environment. Here's how it works on Windows 10 Machine: Start button on your keyboard. Start typing the words Environment Variables. As you type, you'll see a choice to edit system environment variables. Select him. In the Environment Variables window, select the PATH LINES item in the user variables for (your user name), and then click the Edit button. Add the full path to Android SDK tools and Android SDK folders tools for platform in the editing frame, separated by a semi-thick hose. It should look like this: C:\Android\tools; C:\Android\platform tools For earlier versions of Windows, see the documentation that came with your computer to help set up PATH. And, again: If you've installed an SDK somewhere other than \Android, you'll need to adjust accordingly. On Mac Source: Android Central You can set your PATH variable on the machine that runs OS X in your bash profile. It's easy, and it's all done in one file. Your home folder contains a file named .bash_profile. Open it with any text editor. Never touch the .bashrc or .bash_profile files that you can find in /etc. Directory! You can see an empty file or it may be full of other information. All we need to do is add a few lines to the top of the file: export PATH=$HOME/Android/tools:$PATH export PATH=$HOME/Android/platform-tools:$PATH (Did we mention that if your SDK is in a different location, you'll need to adjust things accordingly? All right, i'm going to have to Save the file and rescue the computer so that the new PATH is properly procured. On Linux Setting the PATH on a Linux computer is almost the same as on a Mac, just edit the second file. Using your favorite text editor, open the ~/.bashrc file. There will probably be more entries. If you get an error that the file doesn't exist, simply create a new file and save it as ~/.bashrc when you're done. At the end of the .bashrc file, you want to add the following two lines: export PATH=$HOME/Android/tools:$PATH PATH EXPORT=$HOME/Android/platform-tools:$PATH Save file and close the terminal window. Open a new instance of the terminal and type this command: the source of ~/.bashrc Your session will refer to the changes you made and the SDK will be in your PATH. Wrapping Source: Android Central Now you should have a working set of Android command tools and be able to do things like flash the latest factory images or manually update your phone with a zip file. And because you did it yourself, you have what you need to fix it when things go wrong. Good luck and have fun! Each week, the Android Central Podcast brings you the latest tech news, analysis and hot takes, with well-known co-hosts and special guests. Subscribe to Pocket Casts: Audio Subscribe to Spotify: Audio Subscribe to iTunes: Audio We can earn a commission to purchase using our links. learn more. The Android Software Development Kit (SDK) includes various components, including SDK tools, creation tools, and platform tools. SDK Tools include inventories of inventories emulator, hierarchical browser, SDK manager, and ProGuard. Construction tools primarily include aapt (Android packaging tool for creating . APK), dx (An Android tool that converts .java files into .dex files). Platform Tools include Android debug shell, sqlite3 and Systrace. The Android SDK can be installed automatically using the latest version of Gradle or manually downloading the Android SDK in several different ways. Below is an overview of all the different approaches. Installing android sdk (Automated Path) Gradle 2.2.0 now supports automatic download dependency. Be sure to upgrade to the latest gradle version. The gradle addiction management supplement has now been depreciated. Install for Ubuntu Linux If you are using Ubuntu 15.04 or 15.10, be sure to install the following packages. Otherwise, you may notice no such file or directory when running trying to execute an aapt program that is part of the Android SDK tool: sudo apt-get to install libc6-dev-i386 lib32z1 openjdk-8-jdk Installing Android SDK (via Homebrew) Assuming you have macOS / OS X running, you can use Homebrew to install android sdk (via Homebrew) Assuming you have macOS / OS X running, you can use Homebrew to install Android SDK (via Homebrew) Assuming you have macOS / OS X running, you can use Homebrew to install android sdk (via Homebrew) Assuming you have macOS / OS X running, you can use Homebrew to install android sdk (via Homebrew) Assuming you have macOS / OS X running, you can use Homebrew to install Android SDK (via Homebrew) Assuming you have macOS / OS X running, you can use Homebrew to install Android SDK (via Homebrew) Assuming you have macOS / OS X running , you can use Homebrew to install Android SDK (via Homebrew) Under Install Homebrew - Package Manager for macOS / OS X Run following commands: brew caskroom / cask brew cask install android-sdk It will install Android SDK tools in / usr / local / Basement / android-sdk /<version number=> Installing Android SDK (Manual Mode) You will need to download the Android SDK without Android Studios in the package. Go to the Android SDK and navigate to the SDK Tools only section. Copy the download URL that is appropriate for your build machine OS. Use wget with the correct SDK URL: $ wget Unzip and set the content within the home directory. Directory name names can be anything you want, but save files to someone easy to find (i.e. ~/android-sdk). Run sdkmanager tools: $ tools / bin / sdkmanager -- update $ tools / bin / sdkmanager platform; android-25 build-tools;25.0.2 extras;google;m2repository extras;android;m2repository $ tools/bin/sdkmanager --licenses Now is the time to set path variables and other variables of your environment that will be used to locate Android. Edit .bash_profile If you are not using bash, edit the right configuration file for your environment. # Android export ANDROID_SDK_ROOT=/Users/ciandroid/android-sdk-macosx export PATH=$PATH:$ANDROID_SDK_ROOT/tools Save and cancel. Reload .bash_profile: Installing using the GUI On request, type Android and press Enter to run Android SDK Manager in the window. If this doesn't work, your PATH variable is not set with the Android SDK location. You'll want to install the same Android SDK packages on your build machine as you did to make Gradle run locally. Before you start, see the build.gradle file in your project. Installation packages Here are the sdk package names you will definitely want to choose: Tools > Tools for Android SDK Tools > Android SDK </version> </version> Tools > Android SDK build-tools One version of android platform. For example, Android 5.1.1 (API 22). It should be the one you called in android: composessdkVersion section of your build.gradle file. You'll also want to download add-ons: Android Support Repository Android Support Library Note: Choose Android SDK Build-tools for the version of Android that you specified in the build.gradle file as Android: buildToolsVersion target. If your build.gradle says android { buildToolsVersion '21' ... } then be sure to download that API version in Android SDK Manager. Installing using the command line you can also download SDK packages using the command line with a -no-ui parameter. android update sdk --no-ui --all If you want to be selective regarding installation, you can use android list to view all packages and apply options -filter for selective installation: sudo android update sdk --no-ui --filter platform-tools,tools If you decide to be selective about which packages will be installed, be sure to turn on additional Android Maven repository. Otherwise, you may not be able to use the latest support design library. android update sdk --no-ui --all -- filter extra-android-m2repository There is currently no filter for direct installation of the build tool. See this ticket for more information. Information.