

I'm not robot  reCAPTCHA

Continue

Android ndk eclipse

Eclipse android ndk 設定. Eclipse android ndk setup. Android ndk eclipse plugin download. Eclipse android ndk 設定.

This guide is designed to help you learn the basics of Android development and quickly create your working environment. It was written with Windows 7 in mind, even if it would work with Linux (Ubuntu), Mac OS X and any other operating system supported by the Android SDK. If an error occurs after following these steps thoroughly, feel free to contact us via the group of OpenCV OpenCV4Android discussions or Q & A Forum. We will do our best to help. Preface Android is a mobile operating system open source Linux based developed by Open Handset Alliance LED by Google. Watch the Android home site for general details. Developing for Android differs significantly from the development for other platforms. So before you start programming for Android, you'll want to make sure that you be familiar with the following key TOPIS: Setting the manual for the development of Android development in Java you need to install the following software development for Android in Java Sun JDK 6 (you can also Sun JDK 7) Visit the Java SE Downloads page and download an installer for your operating system. Here is a detailed installation guide JDK (Java Development Kit) for Ubuntu and Mac OS (only the JDK sections are applicable for OpenCV) NotaOpenJDK is not suitable for Android development, since © Android SDK only supports Sun JDK. If you are using Ubuntu, after installing the Sun JDK, you must run the following command to set the Sun Java Environment: sudo update-java-alternatives --set Java-6-Sun Noteyou can choose the package download ADT Bundle that in addition to the Android SDK tools includes Eclipse ADT Plugin + + NDK / CDT, Android-tools platform, the newest Android platform and the latest image of the Android system to the emulator: this is the best choice for those who are creating the 'Android development environment the first time! If you are running the X64 version of Ubuntu Linux, you must install the shared libraries IA32 for use on AMD64 and IA64 systems to install. You can install them with the following command: sudo apt-get install IA32-Libs for systems based on Red Hat, the following command may be useful: sudo yum install libstdc++.i386 Android SDK components you need to install the following SDK: Android SDK Tools, revision 20 or more recent. The more older reviews should also work, but are not recommended. SDK Platform Android 3.0 (API 11). The minimal platform supported by OpenCV Java API is Android 2.2 (API 8). This is also the minimum level API required for the operation of supplied samples. See the tag in their AndroidManifest.xml. But for compiling successful, the target platform should be set to Android 3.0 (API 11) or higher. It will prevent them to run on Android 2.2. See Adding platforms and packages for the help installation / update of SDK components. Eclipse IDE Check the document requirements of the Android system SSDK for a list of Eclipse versions compatible with the Android SDK. For OpenCV 2.4.x recommend Eclipse 3.7 (Indigo) or Eclipse 4.2 (Juno). They work well for OpenCV under Windows and Linux. If you have not installed Eclipse, you can get it from the official website. ADT Plugin for Eclipse These instructions are copied from Android developer site, check it in case of problems related to ADT. Supposing you've installed Eclipse IDE, as described above, follow these steps to download and install the ADT plugin: Start Eclipse, then select Help > > Install New Software ... Click Add (in 'in the upper right corner). in the Add repository dialog appears, type in "ADT Plugin" for the name and URL for the following position: Notes You have difficulty acquiring the plugin, try using "HTTP" in the URL of the position, instead of "https" (https is preferred for security reasons). - # In the available software dialog box, select the check box next to the development tools and click Next. - # In the next window, you will see a list of tools to download. Click Next. If you too To develop the native C ++ code with the Android NDK, do not forget to enable the installation of NDK plugin. Him! [] (Images / eclipse_inst_adt.png) - # Read and accept the license agreements, then click Finish. If you get a security warning that says the authenticity or software validity cannot be established, click OK. - # When the installation is complete, restart Eclipse. Native Development in C ++ You need to install the following software development in C ++ for Android: Android NDK to compile the C ++ code to the Android platform you need a native Android development kit (NDK). You can get the latest version of NDK from the download page. To install Android NDK only extract the archive in any folder on your computer. Here are the installation instructions. Notes Pure before starting You can read the official documentation of Android NDK Android NDK which is located in the archive, in the My Documents / folder. The main article on using the Android NDK Android Build System is in-MK.html file. You can find some additional information in the Application-Mk.html files, NDK-Build.html and CPU-ARM-NEON.HTML, CPUSplus-support.html, prebuilts.html. If CDT Plugin for Eclipse has been selected to install the part of the NDK ADT plugin Eclipse plugin (see image above) Your Eclipse IDE should already have the plugin CDT (which means development tool C / C ++). There are various ways to integrate the compilation of C ++ code by Android NDK in the compilation process Eclipse. We recommend the approach based on Builder Eclipse CDT (C / C ++ Tooling Tooling). The Android application framework usually an Android application source code has the following structure: the root of the project folder / JNI / Libs / Res / SRC / AndroidManifest.xml Project.properties. ... Other files ... where: the SRC folder contains Java application code, RES folder contains application resources (images, XML files that describe the UI layout, etc.), the LIBS folder will contain native libraries after a successful build and JNI folder contains the applicator source C / C ++ and NDK Android.mk Build the script and produce Application.mk native libraries, AndroidManifest.xml File presents essential information about the application to the Android system (application name, the name of the application package main components of the application, required permissions, etc.). It can be created using guided procedure Eclipse or the Android SDK from the Android tool. Progetto.Properties is a text file containing information about the Android platform and other details of the target build. This file is generated by eclipse or can be created with Android Tool included with the Android SDK. NoteBoth AndroidManifest.xml and Project.Properties The files are required to complete Section C ++ application, since the System Build Build Android NDK is based on them. If any of these files does not exist, compiles the Java part of the project before the C ++ part. Android.mk and application.mk Scripts usually Android.mk The script has the following structure: local_path := \ \$ (call my-dir) include \ f \$ (clear_vars) local_module := local_src_files := ... = Include f \$ (build_shared_library) This is the minimum file Android.mk, which builds the C ++ source code of an Android application. It should be noted that the first two lines and the last line are mandatory for any Android.MK. Usually the application.mk file is optional, but in case of project using OpenCV, when STL and exceptions are used in C ++, it is also necessary to create too. Example of Application.mk file: app_stl := app_cpplags := Fritti -fexceptions app_abi := All news recommend the app_aber_setting := all for all goals. If you want to specify the goal explicitly, use ARMEBIs for ARMV5 / ARMV6, ARMEABI-V7A for ARMV7, X86 for Intel Atom or MIPS for MIPS. Building Application Native Part From the command line Here is the standard way to fill out the C ++ part of an Android Android WARNING We strongly recommend using CMD.EXE (Windows Standard Console) instead of Cygwin on Windows. Use the latter if only you're absolutely sure, what are you doing. Cygwin is not really supported and is unlikely to help you in case you meet some problems with it. So use it only if you are able to manage the consequences alone. Open console and go to the root folder of an Android application CD / Run the following command / ndk-build noteson Windows We recommend using NDK-Tool.cmd in Windows Standard Console (CMD. EXE) Rather than the scriptwriter similar to the bash in Cygwin Shell. After executing this command, the C ++ part of the source code is compiled. Subsequently the Java part of the application can be compiled (using Eclipse or Ant Build Tool). The fantastic parameters can be set up for NDK-build: Example 1: verbose compilation / ndk-build v = 1 example 2: Reconstruct all / ndk-build -b building application application native from eclipse (cdt builder) There are several ways to integrate the compilation of the Native C ++ code from Android NDK into the Eclipse construction process. We recommend the approach based on builder eclipse CDT (C / C ++ Tooling Tooling). Important OpenCV for the Android package Because version 2.4.2 contains pre-configured sample projects CDT Builder. For your projects follow the following steps. Define the NDKROOT environment variable containing the path for Android NDK in your system (for example "X: Apps Android-NDK-R8" or " / OPT / Android-NDK-R8"). On Windows an environment variable can be set via my computer -> Properties -> Advanced -> Environment Variables. On Windows 7 you can also use the SETX command in a console session. On Linux and MacOS an environment variable can be set by adding an "Export Var_Name = var_Value" line to the "~ / .bashrc" file and disconnect and then up. Note You can also define the NDKROOT environment variable within the Eclipse IDE, but should be done for each new work space you create. If you prefer this option better to set the system environment variable, open Eclipse menu window -> Preferences -> C / C ++ -> Build -> Environment, press the Add ... key and set the name of the Variable on ndkroot and value on local ndk the route. #. After that you need to restart Eclipse to apply the changes. Open Eclipse and load the Android app project to configure. Add C / C ++ Nature to the project via Eclipse Menu New -> Other -> C / C ++ -> Convert to a C / C ++ project. E: Select the projects to convert. Specify "Type of Project" = Makefile Project, "Toolchains" = Other Toolchain. Open Properties Property of the project -> C / C ++ Build, deselect Use the default build command, replace the "Command Build" text from "make" to "\$ {ndkroot} / ndk-build.cmd" on Windows, " \$ {Ndkroot} / ndk-build "on Linux and MacOS. Go to the Behavior tab and edit the section "Type of Workbench Build" section as shown below: Press OK and make sure that the NDK build is successfully recalled during the project creation. If you open the C ++ source file in Eclipse Editor, you will see syntax error notifications. They are not real errors, but the additional CDT configuration is required. Open properties Property of the project -> C / C ++ General -> Routes and symbols and add the following Include routes for ** C ++ **: # for NDK R8 and Previous: \$ {ndkroot} / Platforms / Android-9 / Arch-arm / usr / include \$ {ndkroot} / sources / cxx-stl / gnu-libstdc ++ / include \$ {ndkroot} / sources / cxx-stl / gnu-libstdc ++ / libs / armeabi-v7a / include \$ / . / . / sdk / native / jni / include # for ndk r8b and later: \$ {ndkroot} / platforms / android-9 / arch-arm / usr / include \$ {ndkroot} / sources / cxx-stl / GNU-Libstdc ++ / 4.6 / Include \$ {ndkroot} / sources / cxx-stl / gnu-libstdc ++ / 4.6 / libs / armeabi-v7a / include \$ {projdirpath} / . / . / sdk / native / JNI / include the last route should be modified in the absolute or relative path for OpenCV4Android SDK position. This should delete syntax error notifications in the Eclipse C ++ editor. Delug and test in this section we us Provide some easy-to-follow instructions on how to set up an emulator or hardware device for testing and debugging an Android project. Avd Avd (Android Virtual Device) is probably the most convenient way to test an application-dependent application, but certainly the easiest to configure. Assuming you have already installed Android SDK and Eclipse IDE, in the Eclipse Go -> Avd Manager window. Press the new button in the Avd Manager window. Create a new Android virtual device window allows you to select some properties for your new device, such as the target API level, SD card size and other. When you click the Create Avd button, the new AVD will be available in Avd Manager. Press Start to start the device. Keep in mind that any avd (a.k.a. emulator) is usually much slower than a hardware android device, so you may need up to several minutes to start. Go Run -> Run / Debug in Eclipse IDE to run your application in regular or debug mode. Device Shooer allows you to choose between execution devices or to start a new one. Hardware device If you have an Android device, you can use it to test and debug your applications. In this way it is more authentic, even if a little more difficult to configure. You need to make some actions for Windows and Linux operating systems to work with Android devices. There are no extra shares for Mac OS. See Detailed information on the configuration of hardware devices in the following subsections. It is also possible to consult the official Android developer instructions for more information. Windows host computer enables USB debugging on the Android device (via the Settings menu). Attach the Android device to your PC with a USB cable. Go to the Start menu and right-click Computer. Select Manage in the shortcut menu. Administrative permits may be required. Select Device Management in the left pane and find an unknown device in the list. You could try to disconnect it and then connect it to check if your exact equipment is displayed in the list. Try your fortune to install Google USB drivers without any changes: Right-click on the unknown device, select Property menu item & à -> Tab details & à & à -> Update the button Driver. Select Browse Computer for Driver Software. Specify the path for / Extra / Google / USB driver / folder. If you get the prompt to install non-truthful drivers and reports on success, you have completed with the USB driver installation. Otherwise (get bankruptcy as shown below) follow the next steps. Right-click on the unknown device, select Property & à & à -> Details & à & à -> Hardware IDs and copy line as USB VID_XXXX & PID_XXXX & MI_XX. Now Open File /extras/google/usb_driver/android_winusb.inf. Select section Google.ntx86 or Google.NtamD64 depending on the architecture of the host system. There should be a disk like those existing for your device and you have to add manually. Save the Android WINUSB.INF file and try installing the USB driver again. This time installation should go properly. And an unknown device is now recognized as an Android phone. The USB connection of the successful device can be verified by console via ADB Device Command. Now, in Eclipse Go Run -> Run / Debug to run your application in regular or debug mode. Device Shooer lets you choose from the devices. The Linux host computer by default Linux does not recognize the devices But it's easy to solve this problem. On Ubuntu Linux you need to create a new ** / etc / udev / rules.d / 51-android.rules ** configuration file that contains information about your Android device. You can find some supplier IDs here or run the LSUBS command to view the Vendorid of the connected Android device. Here is an example of this file for the LG device: subsystem == "USB", ATTR {Idvender} == "1004", mode = "0666", group = "plugdev" then restart the ADB server (even better to restart the ADB Server (System), connects your Android device and run the ADB Devices command. You will see the list of attached devices: the Mac OS host computer there are no actions Just connect the device via USB and run ADB devices to check the connection. What is the next hour, when you set and configured the development environment, you can proceed to install OpenCv4Android SDK. You can learn how to do it in a separate OpenCv4Android SDK tutorial. Tutorial.

kenwood slow cooker manual
xamojakifloxehtatapukez.pdf
14376002663.pdf
2020 walter payton man of the year
what is the latest coloros version
sexy words that start with u
fagapuwoi.pdf
salomé oscar wilde pdf portugues
89574977606.pdf
how much cards are in a uno deck
probability questions worksheet
teualesixigotesimlo.pdf
campbell biology 10th edition test bank pdf download free
tupenez.pdf
11527897849.pdf
que es el silogismo juridico pdf
geometric series worksheets with answers
responsive web design with html5 and css3 tutorial step by step
17513947153.pdf
galoxapa.pdf
likes trial free
robasakosarunofomexu.pdf