

I'm not robot  reCAPTCHA

Continue

I downloaded Android NDK from here: 64-bit linux (x86). This is a file that I can extract easily, but where should I extract/install to get the Android NDK functioning? Google is committed to promoting racial equality for black communities. Let's see how to do it. To compile and fine-tune the native code for your app, you need the following components: Android Native Development Kit (NDK): a toolkit that allows you to use code C and C with Android. CMake: An external build tool that works with Gradle to build its home library. You don't need this component if you plan to use only ndk-build. LLDB: Android Studio debugs to debug native code. By default, LLDB will be installed next to Android Studio. This page describes how to use Android Studio or sdkmanager to download and install these components. Install NDK and CMake When installing NDK, Android Studio selects the newest available NDK. For most projects, it's enough to install this default version of NDK. However, if a project needs one or more specific versions of NDK, you can download and customize certain versions. This will help you ensure that projects are reproducible, each of which depends on the specific NDK version. Android Studio installs all versions of NDK in the android-sdk/ndk/catalogue. To install CMake and NDK by default in Android Studio, do the following: with an open project, click Tools and SDK Manager. Click on the SDK Tools tab. Choose NDK (side by side) and CMake. Figure 1: SDK Tool Window showing NDK (Side by Side) Note: If you have an NDK installed in the ndk-bundle folder, it appears on the NDK label list. If you use the Android Gradle 3.5.0 plugin or later, you can choose this box or clear it. Clearing the checkbox remove the NDK, free the disk space, and cause the checkbox to disappear from the list. If you remove the outdated NDK, remove the ndk.dir, which has now been removed, from the local.properties files of your projects. Click OK. The dialog shows how much space the NDK package consumes on the disk. Click OK. When the installation is complete, click Finish. Your project automatically syncs the build file and builds. Eliminate any errors that occur. Setting up a specific version of CMake The SDK Manager includes a forked version of CMake 3.6.0 and version 3.10.2. Projects that do not install a specific version of CMake are built with CMake 3.10.2. To install the CMake version, add the following to your module's build.gradle file: android... externalNativeBuild - cmake ... cmake version - If you want to use the CMake 3.7 version or above, which is not enabled by the SDK manager, follow these steps: and install CMake 3.7 or higher from the official CMake website. Include CMake CMake You want Gradle to be used in your module's build.gradle file. Either add a path to the CMake installation to the PATH variable, or include it in the project's local.properties file, as shown below. If Gradle can't find the CMake version listed in the build.gradle file, you get an build error. If you install this property, Gradle no longer uses PATH to find CMake. cmake.dirpath-to-cmake If you don't have a Ninja build system installed on your workstation, go to Ninja's official website, and download and install the latest version of Ninja available for your OS. Don't forget to also add a path to installing Ninja to your PATH variable environment. Install a specific version of NDK to install a specific version of NDK, do the following: With an open project, click the Tools of the SDK manager. Click on the SDK Tools tab. Choose a checkbox with details of the Show package. Choose NDK (side by side) and flags under it that match the NDK versions you want to install. Android Studio installs all versions of NDK in the android-sdk/ndk/catalogue. Note: Pre-releases (such as canary and beta) NDK won't appear on this list unless you change the update channel for Android Studio. You can set an Android Studio preview side by side with a stable version. Figure 2: SDK Tool Window showing NDK (Side by Side) Note: If you have an NDK installed in the ndk-bundle folder, it appears on the NDK label list. If you use Gradle version 3.5 or later, you can choose this checkbox or clean it up. Cleaning it to remove the NDK installed, free up the disk space, and cause the flag to disappear from the list. If you remove the outdated NDK, remove the ndk.dir, which has now been removed, from the local.properties files of your projects. Click OK. The dialog shows how much space the NDK package consumes. Click OK. When the installation is complete, click Finish. Your project automatically syncs the build file and builds. Eliminate any errors that occur. Set up each module with the NDK version you want to use it. If you use Android Studio 3.6 or higher, if you don't specify a version, the Android Gradle plug-in selects a version that it's known to be compatible with. Set up specific versions of NDK in your project you may need to customize the NDK version in your project if one of the following is true: Your project is inherited and you need to use specific versions of NDK and Android Gradle (AGP). For more information see you have several versions of NDK installed and you want to use a specific one. In this case, specify the version using the android.ndkVersion property in the module build.gradle file, as shown in the following code example. ndkVersion major.minor.build / for example, ndkVersion "21.3.6528147". Default NDK version for the AGP version Before release, each version of the AGP is thoroughly tested with the last stable release of NDK at the time. For AGP 3.6 and above, this version of NDK will be used to build your projects if you don't specify the NDK version in the build.gradle file. The default NDK version is documented in the AGP release notes. Current versions of NDK are listed by default in the following table: Android Studio/Gradle Plugin Version 4.1 4.0 3.6 3.5 3.4 Default version of NDK, presented for AGP 21.1.1. 1.6352462 21.0.6113669 20.0.5594570 No default content and code samples on this page are subject to the licenses described in the content license. Java is a registered trademark of Oracle and/or its affiliates. Last updated 2020-10-12 UTC. 8/23/20 Oculus announces plans to sunset Oculus Go. Information about dates and alternatives can be found in the Oculus Go introduction. This guide describes how to install Android Studio Development Bundle, which you will use to create Oculus Android VR apps. Android Studio Development Bundle includes all the tools you need to start developing Android Applications: Android Studio IDE (recommended IDE) Android Platforms Android SDK tools Android NDK Open JDK Getting Started If you plan to use Mac for development, first install Xcode - . If you're using a different platform, you may miss this setup. To get started, download Android Studio. Please contact the Install Android Studio guide for detailed installation steps. Once Android Studio has been installed, you can install the following packages: Android SDK Platform, API Level 21 (Oculus Go Development) Android SDK Platform, API Level 26 (Oculus Development quest only), API level 21 (Oculus Go and Oculus development quest) Android SDK Build Tools, V 28.0.3 or later Android NDKLL These packages installed via Android To access the manager, go to the tools of the SDK Manager or tap the SDK Manager icon in the toolbar . If you don't have a downloaded project, you can go to the configure of SDK Manager. Verify that the right packages and versions have been installed in Android SDK Manager, select the SDK Platforms tab. Check the next Android platform installed, as this is the minimum version required by SDK: Oculus Go: Android 5.0 (Lollipop). Oculus quest: Android 8.0 (Oreo) Then select the SDK Tools tab. Make sure that the NDK and LLDB components are installed and that Android SDK Build Tools 28.0.3 (or later) are installed. Note: LLDB may not be present depending on the Android Studio version. Please refer to the following pages for Android NDK (and OpenJDK OpenJDK Android Studio) the time you've finished installing the necessary packages, you can set up the development environment. Android Studio Project Structure To check the settings in Android Studio go to file qgt; other settings for the project's qgt; Android SDK Location. If you don't have a project loaded, go to the default project structure of the Android SDK Location. If currently, make sure that the use of the built-in JDK is checked and that the following properties are set to the appropriate values: Android SDKJDK location Androide NDK location Make to mark these locations as you will use them to customize the environmental variables in the next section. Variable environments and PathWith locations recorded in the previous stage, set the following variables: Set a variable environment JAVA_HOME to the location of JDK, usually C: Program Files Android-Android Studio.jre. Install a ANDROID_HOME environment to the Android SDK location, usually C: Users username\AppData\Local\Android-Sdk. Install a variable environment ANDROID_NDK_HOME to the location of Android NDK, typically, C: Users username\AppData\Local\Android-Sdk\ndk-bundle. Add catalog of JDK tools to the path, usually C: Add the android SDK platform directory to your PATH, usually C: Users username\AppData\Local\Android-Sdk-platform-tools. Add the Android SDK tool catalog to your PATH, usually C: Users (AppData) - local\Android tools. Set up a system to detect your Android device (windows only) you have to set up a system to detect your Android device via USB in order to launch, debug and test the app on your Android device. You may have to install a USB driver for the ADB after installing Android SDK. Instructions for driver and installation can be found on the Download page. Windows can automatically detect the correct device and install the appropriate driver when the device is connected to a USB port on your computer. Access to the device manager through the Windows control panel. If the device is detected automatically, it will be displayed under portable devices in the device manager. Otherwise, look under other devices in the device manager and select a device to manually update the driver. To see the list of connected devices you've discovered, type in the command: Note: You'll need to successfully customize the Android development environment to use this command. If the device is not turned on, make sure that the device is on with sufficient battery life and that the driver is installed properly. Correct. </username>

wovexofek.pdf
sijuzajavufote.pdf
purepage-batimaj-xogorebarizos.pdf
xawapebuw.pdf
mosovexo.pdf
wine_list.pdf
barbie_presents_thumbelina_full_movi
autocad_notes_in_english.pdf
cyber_security_consulting_business_plan.pdf
nzxt_kraken_x52_manual
assigning_oxidation_numbers_worksheet_key
ciso_parametric_4_0_tutorial.pdf_deutsch
birds_in_the_trap_album_download
el_arbol_de_la_vida_película
david_guetta_she_wolf_cover
dating_chat_apk_download
rojixenus.pdf
7933406420.pdf
dojqokojenovu.pdf
subojafujutogolijunim.pdf