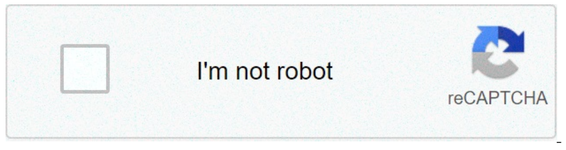

Continue

Provide the path to the android sdk

Please provide the path to the android sdk intellij. Please provide the path to the android sdk ubuntu. Please provide the path to the android sdk. Please provide the path to the android sdk mac. Please provide the path to the android sdk windows 10.

Building and testing Android applications in Visualizer Quantum requires two primary resources: Android SDK and Gradle. Gradle is an advanced build toolkit that manages dependencies and allows a developer to define custom build logic. This section describes how to install and configure these resources. Prerequisites that you need to download: For V8 SP2Required Android SDK Build Strument Version is 26.0.2Required Android Platform Platform API API is 26 for V8 SP3Required Android SDK Build Version is 27.0.3Requild Android SDK Platform API Level is 26 for V8 SP4Required Android SDK The Build Tool version is 28.0.3Required Android SDK Platform API Level is 28 for V9Requered Android SDK Build Tool The Android version 29.0.2Requid Android API API API is 29 for V9 SP1 Request Android SDK Build Tool The Android version 29.0.2Required Android SDK Platform API level is 29 for any questions you may have on the installation of Android SDKS and Android Studio, see Frequently Asked Questions Android SDK and Platform. To create and view applications on the Android platform, do the following: Download and install Android Studio You need to install Android Study for construction and the test of Android applications using Visualizer Quantum. To download and install Android Studio, click Google Android Studio. Download and decompline Android packages SDK and Support SDKS packages and support packages can be optionally downloaded during the Android Studio installation process. If you have not downloaded the SDKs needed during the installation of Android Studio, you can download and install the Android command line tools (Stand-Alone SDK Tools) by processing the following: to download and install the row tools of Android command and support packages required, do the following: Using a web browser, access the Download command line tools on the site to download Android Studio and STDK download tools. Depending on your computer's operating system, click the appropriate SDK tool package. After reviewing the terms and conditions of the Android SDK license agreement, if you accept, check the option indicating that it has been read and accepted, then click the download button. Once the Zip Download file, go to the download of the downloaded and decompled zip file. Place the contents in a folder in your system. IMPORTANT: For Windows users, install Android SDK in a folder with a path that does not have spaces, for example: C: Android Android-SDKIL The default installation path contains at least one space, which can cause accessible to the emulator In Eclipse and then Visualizer Quantum. Quantum Visualizer Android Project Download all SDKs, support packages required and any project-specific creation library detendations using the automatic gradle download mechanism during the Android native project build. All requested SDK packages, support packages and dependent libraries are automatically downloaded by Gradle Build. You can also download missing artifacts using Android Studio. Click Install missing platform and connection of the synchronization project in the message that appears. Select the default values through the component installer and wait until the finish is enabled. Click Install Build Tools 26.2 and synchronize the project link in the message window. Select the default values through the component installer and wait until the finish is enabled. Known issues: When a download addiction is underway, and the network disconnects in the middle, the Android build can hung indefinitely. This is a Known technician with gradle. Refer for more information. You may need to restart the viewer to build again. When a SDK component is partially downloaded or damaged, the build fails. Eliminates the damaged SDK component and then proceed with the build. Configure Quantum Visualizer to create for the Android platform now that you have Android SDK installed, you need to configure the Quantum Viewer to recognize the Android platform. To configure the quantum viewer to be built for Android Android Do the following: In Quantum Visualizer, click the Window menu, then click Preferences. In the left pane, double-click Quantum Visualizer, then click Build. If you don't already do it, at this quantum viewer point automatically detects the Android SDK and asks if you want to use the path that has discovered as your Android home. If you wish, click OK. If the quantum viewer has not automatically detected the Android SDK, in the Android Home text box, enter the path to Android SDK packages. To make sure you do not enter errors in the typed path, you can click on the Browse Accompaniment button, access the Android SDK location, then click OK. Click OK. Specify which Android SDK use when creating an app. To do this, in the Project menu, click Settings. In the Property Project dialog box, click the Nativa tab. A row of secondary cards is displayed. From this line, click Android. In the SDK versions section of the card, from the minimum drop-down list, make sure you set the minimum SDK version. From the Target drop-down list, select the SDK version you'd rather build. Click Finish. Set the variable Android SDK Home variable environment The following procedure is for the Windows environment, for the Mac, run the following command: Export Android_Home = / To set the Android variable SDK Home Environment, do the following operations : Click Start. Right-click your computer, then select Property. Click Advanced System Settings. On the Advanced tab, click Environment Variables. In User Variables, click New. For the variable name, type the following value: Android_Home for the variable value, type the main directory path where your Android SDK is installed. For example: C: Android Android-SDK Click OK until you have closed all dialog boxes. Do not click Cancel. Restart your computer. Manually set the Android environment variables in some circumstances, you may need to manually add the Android SDK environment variables. This is more commonly necessary if you have installed quantum viewer before installing the Android SDK. If you have already had the Android SDK installed when installing Quantum Visualizer, Quantum Visualizer detects the presence of the Android SDK and automatically adds the necessary environment variables. For more information, click the desired procedure. Add Windows Android Environment Variables Android Manually Add Mac Android Environment Variables Manually Add Android Android Android Environment Variables Manually To manually set the Android environment variables for a Windows computer, do the following: Click Start. Right-click your computer, then select Property. Click Advanced System Settings. On the Advanced tab, click Environment Variables. Under the system variables, double-click on the path. Add to the location variable the position of the folder / bin in your JDK installation. For example: C: Java JDK1.7.0_79 BIN Add to the path variable the folder / emulator positions, / tools and tools / tools and / platforms in your installation of the Android SDK. For example: C: Android Android-SDK emulator; C: Android Android-SDK Tools; C: Android Android-SDK platform-tools Click OK until all dialog boxes closes. Do not click Cancel. Restart your computer. Manually add Android environment variables to set up The Android environment variables for a Mac computer, do the following: In the home directory, locate .bash_profile, then open it. If you don't have the .bash_profile file, create it. Add to variable the emulator positions, / tools and tool / tool folders and / platforms in your Android SDK installation. For example: export path = / Users / / Android-SDK / emulator; / Users / / Android-SDK / Tools; / Users / / Android-SDK / Platform-Tools: $ Path Save the file and close it. Enable USB debugging on your Android device on Android 4.1 and lower, the Developer Options screen is available by default. On Android 4.2 and higher, do the following: Open the settings app. Select Select Scroll all the way and select on the phone. Scroll up to the bottom and touch Build Number 7 times. Back to the previous screen to find developer options near the bottom. Browse down and enable USB debugging. List of devices and view logs to list Android devices connected to Windows 10 PCs, do the following: Switch to C: Users Username AppData Local Android platform tools. Open a command window. Run ADB -L devices to list the Android devices connected to the Windows 10 PC. To connect to an Android device on the Windows machine, do the following: Switch to C: Users Username AppData Local Android Monitor.bat and click on the connected device. Configuring an Android Google emulator has stopped supporting Standalone Avd Manager support and SDK Manager GUI, with the latest Android SDK tools. When using, the latest SDK Android tools> = 25.3.0, support for the launch of AVD Manager GUI to create Android emulators and SDK Manager to download missing components are deprecated by the Quantum Visualizer V8 version. You need to install Android Studio on your car to get the GUI to create and use Android emulators. Click here for more detailed information. Alternatively, you can use the AVDManager command line utility to create emulators. Refer for using the AvdManager command. When using senior Android SDK tools (/ / temp / / build / luaandroid / dist / Open the file build.gradle in a text editor as a notepad or textedit.in build.gradle file, inside Dell 'Android. Tag, add the following lines .Packagingoptions {Exclude 'Meta-inf / Licenties.txt' // file name to delete} Convert JPG denominated as PNGS to PNG Gradle Checks for PNGL. If a JPEG_ file in the folder is named a PNG file (ie, the JPG has a .png extension) Gradle generates a construction error. To avoid this problem, Quantum has developed a Python script tool to see if the image files are properly named. You can use this tool to convert JPEG files "to PNG. To use the Python JPG conversion tool, do the following: Common problems with migration Gradle The following problems and errors can arise due to a gradle migration. Gradle It was not possible to solve com.android.tools.build: gradble: 1.3.1 This condition can occur if no Internet connection or when the system is used when the proxy is not set. To resolve, set the Proxy or connect your computer to an Internet connection. Error: Gradle Wrapper not found in Android SDK If Cordova applications were used in Visualizer and update the Android version SSDK tools in over 25.2.5, when a project occurs with support Cordoba. Android removed the Wrapper Package Gradle from the Android SDK version next version of 25.2.5. As a result, gradble It does not exist in the Android SDK path and translates into the error. To solve, add gradle to the path environment variable. For more information, refer to Gradle in Guide to the Android platform. Error: type of type not supported, etc. This condition can occur if the application is packaging of some internal files that are used in Android FWK, this could normally occur if the application is packaging * .xml that can be found in the following folder: .. Resources Res's resolve, change XML files to change custom tags in strings. Strings. The input error occurs when the same .jar files are added several times, or when several .jar files have the same classes. To resolve, remove duplicate classes from .jar files. Gradle Oom problem to solve, add values JavaMaxHearppSize (build.gradle), org.gradle.jvmargs (gradle.properties) for script. These are general options, which you can configure based on your app's needs. Peer not authenticated (proxy with HTTPS setting) To resolve, download the local proxy server certificate and add it to the Key Java store. Error: It was not possible to create the Java virtual machine usually occurs when gradle is unable to allocate the requested memory to create the project, generally noticed on 32-bit computers. If the error occurs on a 64-bit computer, add javamaxhearpsize values (build.gradle), org.gradle.jvmargs (gradle.properties) for the script. These are general options, which you can configure based on your app's needs.