Unity 2017. 3 android sdk

I don't know why but last week I found myself several times in the situation that I had to build an Android in Unity3D on a new computer or reinstall it. Every time I have to run the whole Android SDK setup process and mess up some issues because the installation is not as easy as it should be. This article is kind of a cheatsheet for me but maybe others appreciate it too. The first problem is that you can install the Android SDK either as part of an Android Studio bundle or using the SDK Tools and CLI. Since I don't like downloading and installing large software for just one scenario, I'm happy to know that there is a very fast CLI option to install SDKs without much overhead. The second issue related to Unity 5.6.2p1 (and earlier) failed to build Android SDK Tools 26.0, so for now only version 25 is supported by Unity. 1. JDK Installation Download and install Java Development Kit because you can't do much without it. The current version is 1.8.0. 2. Installation of Android SDK 25 Tools Under normal circumstances we will download the SDK Tools from the Android Studio page. Since we need an older version 25, we can download it directly from the google repository. After downloading the extract content to your SDK path. In our case, we'll use c:\Users\markey\AppData\Local\Android\SDK25\tools. Stop here for a while and note that we created a tool directory inside the entire SDK directory. That's because only one tool from the entire SDK and the other packages will be installed at the same level. 3. Install the package using the CLI and SDKManager As the next step we will start the CLI and open our newly created tool directory: cd c:\Users\markey\AppData\Local\Android\SDK25\tools\bin\ Now try to list all your installed packages using the sdkmanager command: sdkmanager --list This should generate something like that: Installed packages: Path | Version | Description | LocationU ------- | ------- | ------- | ------- tools | 25.2.5 | Android SDK Tools 25.2.5 | tools\ Available Packages: Path | Version | Description ------- | ------- | ------- ... add-ons;addon-g..._apis-google-24 | 1st | Google API ... build-tools;25.0.0 | 25.0.0 | Android SDK Build Tools 25 build-tools;25.0.1 | 25.0.1 | Android SDK Build Tools 25.0.1 build-tools;25.0.2 | 25.0.2 | Android SDK Build Tools 25.0.2 build-tools;25.0.3 | 25.0.3 | Android SDK Build Tools 25.0.3 build-tools;26.0.0 | 26.0.0 | Android SDK Build-Tools 26 ... platform tools | 26.0.0 | Android SDK Platform-Tools ... platform;android-25 | 3 | Platform Android SDK Platform 25;android-26 | 1st | Platform Android SDK 26 ... tools | 26.0.2 | Update Available Android SDK Tools: ID | Installed | Available ------- | ------- | ------- tools | 25.2.5 | 26.0.2 which we can see the tool is already installed, so we have to install three other packages of platform-tools, platforms and build-tools. Cli commands for is the sdkmanager package. If you want to explore sdkmanager more deep then I recommend you to check the documentation. For platforms and build-tools we have to determine the right version, so don't forget to choose the one with version 25: sdkmanager platform-tools sdkmanager platform;android-25 sdkmanager build-tools;25.0.3 4. Set the Android path in Unity Go to Unity and switch projects to the Android platform. Then go to External Tools (see image below) and set the SDK path to c:\Users\markey\AppData\Local\Android\SDK25\tools and also the JDK path to where you installed it in the first step. In our case the path is C:\Program Files\Java\jdk1.8.0_131\. Now, try to build your project and voilà! Other VersionsUda access other versions offline! Whether you're building an Android app in Unity or programming it from scratch, you'll need to set up the Android Software Development Kit (SDK) before you can create and run any code on your Android device. 1. Download Android SDK Download Android SDK from Android Studio and SDK Tools download pages. You can use Android Studio and SDK bundles, or just download the SDK command line tool. 2. Install the Android SDK Install or uninstall the Android SDK. After installing, open the Android SDK Manager and add at least one Android SDK Platform, Platform Tools, Build Tools, and USB drivers if you're using Windows. 3. Enable USB debugging on your device To enable USB debugging, you need to enable the Developer option. To do this, find the build number in your device's Settings menu. The location of the build number varies between devices. Android stock settings can be found by navigating to Settings &gt; About phone &gt; Build. For different android devices and versions, see your hardware manufacturer. Build number as shown in Android 5.0 (Lollipop) on Samsung Galaxy Note 3 Note: On operating systems older than Android 4.2 (Jelly Bean), the Developer option is not hidden. Go to Settings &gt; Developer, then turn on USB debugging. After navigating to the build number using the instructions above, tap the build number seven times. A pop-up notification that says you are now X steps away from being a developer appears, with X being a number that is counted down with every extra tap. On the seventh tap, the Developer option is unlocked. Go to Settings &gt; Developer options, and check the USB debugging checkbox to enable debugging mode when the device is connected to the computer via USB. Developer options as shown in Android 5.0 (Lollipop) - Samsung Galaxy Note 3 4. Connect your Android device to the SDK Connect your Android device to your computer using a USB cable. If you are developing on a Windows computer, you need to install a driver suitable for your device. For more information about connecting your Android device to the SDK, SDK, to the Running Your App section of the Android Developer documentation. 5. Configure the Android SDK path in Unity The first time you create Project for Android (or if Unity then fails to find the SDK), you'll be prompted to find the folder where you installed the Android SDK. Select the root folder for your SDK installation. If you want to change the location of the Android SDK, in the menu bar go to Unity &gt; Preferences &gt; External Tools. 6. Download and set up Android NDK If you are using il2CPP back end scripting for Android, you need an Android Native Development Kit (NDK). It contains the toolchains (such as compilers and linkers) needed to build the necessary libraries, and eventually generates an output package (APK). If you are not targeting the IL2CPP back end, you can skip this step. Download the latest version of Android NDK from the NDK Downloads webpage, then extract it to the directory. The first time you create a project for Android using IL2CPP, you'll be prompted to find the folder where you installed Android NDK. Select the root folder for your NDK installation. If you want to change the location of Android NDK, in the Unity Editor, navigate to the menu: Unity &gt; Preferences... to display the Unity Preferences dialog box. Here, click External Tools. Do you find this page useful? Please rate: Thanks for the ranking of this page! Report a problem on this page Thank you for telling us! This page has been flagged for review based on your feedback. If you have time, you can provide more information to help us fix the problem faster. Provide more information Whether you built an Android app in Unity or programmed it from scratch, you'll need to set up the Android Software Development Kit (SDK) before you can create and run any code on your Android device. 1. Install the Java Development Kit Download and install the Java Development Kit (JDK). Unity requires a 64-bit (1.8) version of JDK 8. 2. Download the Android SDK You can install the Android SDK using the command line tool or through Android Studio. Android Studio provides GUI-based tools that are easy to use, but install additional software on your computer. Using the command line tool is a smaller download and does not install additional software, but it can be more challenging to use. 2a. Install the Android SDK using the Install or uninstall the Android SDK command line tool. After installing, open the Android SDK Manager and add: at least one Android SDK Platform, Platform Tools, Build Tools, and USB drivers if you're using Windows. To install the Android platform SDK and related tools: Download the Android Software command line tool. Unzip the tool folder to a location on your hard drive. Open the command prompt window. Navigate to a folder in the location where you unzipped the tool folder: install the folder &gt; tools &gt; bin Use command line tool to retrieve a list of packages that you can install. Installable packages include platform SDKs, Build Tools, Platform tools, and other tools. sdkmanager –list Select the version of the SDK Platform to install. The platform SDK takes the following form in the list: platform;android-xx. Xx indicates the SDK level. The bigger the number, the newer the package. Usually, you can install the latest available version. But, there may be cases where Google has released a new version of the SDK that caused an error when you built your Unity Project. In this case you will need to uninstall the SDK and install an earlier version. The general format of the command for package installation is sdkmanager &lt;package name=&gt;. You can install the appropriate Platform Tools and Build Tools at the same time. Example: sdkmanager platform-tools platforms;android–27 build-tools;27.0.3 If you are running in Windows, install the USB device driver. sdkmanager extra;google;usb_driver Installs the SDK in a directory named platform in the directory where you unzipped the tool folder. Example: c:\&lt;install folder=&gt;\platforms 2b. Install the SDK using Android Studio Install android studio from the Android developer portal. The Android developer portal provides detailed installation instructions. Note: Android Studio provides some ease of use benefits, but isn't fully tested for compatibility with Unity installations. If you run into an error, Unity recommends using the command line method. When installing the Android platform SDK and other tools, you can usually install the latest available version. There may be cases where Google has released a new version of the SDK that caused an error when you built your Unity Project. In this case you will need to uninstall the SDK and install an earlier version. Install the associated Platform and Build tools at the same time. If you are running Windows, install the USB device driver. 3. Enable USB debugging on your device To enable USB debugging, you must enable the Developer option on your device. To do this, find the build number in your device's Settings menu. The location of the build number varies between devices. Android stock settings can be found by navigating to Settings &gt; About phone &gt; Build. For specific information about your Android device and version, see your hardware manufacturer. Build number as shown in Android 5.0 (Lollipop) on Samsung Galaxy Note 3 Note: On Android versions prior to 4.2 (Jelly Bean), the Developer option is not hidden. Go to Settings &gt; Developer, then turn on USB debugging. After navigating to the build number using the instructions above, tap the build number seven times. Pop-up notifications that say you are now X step away from being appears, with X being the calculated number&lt;/install&gt; &lt;/package&gt; &lt;/package&gt; with each additional tap. On the seventh tap, the Developer option is unlocked. Connect the device to your computer using a USB cable. If you develop it on a Windows computer, you may need to install a device-specific USB driver. See the creation website for your device for additional information. The setup process is different for Windows and macOS, and is described in detail on the Android developer's website. For more information about connecting your Android device to the SDK, see the Running Your App section of the Android Developer documentation. Go to Settings &gt; Developer options, and check the USB debugging checkbox to enable debugging mode when the device is connected to the computer via USB. Developer options as shown in Android 5.0 (Lollipop) - Samsung Galaxy Note 3 4. Configure the Android SDK path in Unity When you first create Project for Android (or if Unity then fails to find the SDK), Unity prompts you to find the folder where you installed the Android SDK. If you installed the SDK using sdkmanager, you can find the folder in &lt;android tools= install= location=&gt;\platforms\&lt;android sdk= folder=&gt;. Example: c:\&lt;android tools= install= location=&gt;\platforms\android–27 If you installed the SDK when installing Android Studio, you can find a location in the Android Studio SDK Manager. To open the SDK Manager from Android Studio, click Tools &gt; Android &gt; SDK Manager or click SDK Manager in the toolbar. SDK Manager toolbar button To change the location of the Android SDK, in the menu bar go to Unity &gt; Preferences &gt; External Tools. 5. Download and set up Android NDK If you are using il2CPP scripting backend for Android, you need Android Native Development Kit (NDK). It contains the toolchains (such as compilers and linkers) needed to build the necessary libraries, and eventually generates an output package (APK). If you are not targeting the IL2CPP back end, you can skip this step. Download Android NDK r13b (64-bit) version from the NDK Downloads webpage. Extract the android-ndk-r13b folder to a directory on your computer and note its location. The first time you create Project for Android using IL2CPP, you're prompted to find the folder where you installed Android NDK. Select the root folder for your NDK installation. To change the location of android NDK, in the Unity Editor, navigate to the menu: Unity &gt; To display the Unity Preferences dialog box. Here, click External Tools. Tool. &lt;/android&gt;&lt;/android&gt;&lt;/android&gt;