**Pre Requisites and Version Requirements**

* Ubuntu 22.04.2 LTS
* QT v5.15.5
* OpenSSL v1.1.1 (prebuilt/precompiled libs)
* Java jdk v1.8
* Android SDK Tools (v26.1.1)
* Android commandline\_tools v8.0 (for compatibility with java 1.8)
* Android NDK (Side by side) 21.3.6528147 (Only install using sdkmanager)
* Android Abi *armeabi-v7a*

**Initial Server Setup**

1. $ apt update && apt upgrade
2. $ adduser <user>
3. $ usermod -aG sudo <user>
4. $ su – <user>
5. $ apt install build-essential default-jre openjdk-8-jdk-headless android-sdk android-sdk-platform-23 libc6-i386

**Android Setup**

Note: Android Studio’s SDKManager can also be used for easier setup with proper versions required for QT 5.15.5 as mentioned above.

1. The following bash script will download the SDK for QT 5.15.5. Run the script from the directory you wish to install to, such as /home/username/android\_tools.(do not use sudo)

*#!/bin/bash*

*ndkVersion="21.3.6528147"*

*sdkBuildToolsVersion="30.0.2"*

*sdkApiLevel="android-30"*

*cmdToolsVersion="8.0"*

*repository=https://dl.google.com/android/repository*

*toolsFile=sdk-tools-linux-4333796.zip*

*toolsFolder=android-sdk-tools*

*rm -rf $toolsFolder*

*echo "Downloading SDK tools from $repository"*

*wget -q $repository/$toolsFile*

*unzip -qq $toolsFile -d $toolsFolder*

*rm $toolsFile*

*echo "Configuring environment"*

*export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64*

*export PATH=$JAVA\_HOME/bin:$PATH*

*# Optional workaround for issue with certain JDK/JRE versions*

*#cp $toolsFolder/tools/bin/sdkmanager $toolsFolder/tools/bin/sdkmanager.backup*

*#sed -i 's/^DEFAULT\_JVM\_OPTS.\*/DEFAULT\_JVM\_OPTS='"'\"-Dcom.android.sdklib.toolsdir=\$APP\_HOME\" -XX:+IgnoreUnrecognizedVMOptions --add-modules java.se.ee'"'/' \*

*#        $toolsFolder/tools/bin/sdkmanager*

*echo "Installing SDK packages"*

*cd $toolsFolder/tools/bin*

*echo "y" | ./sdkmanager "platforms;$sdkApiLevel" "platform-tools" "build-tools;$sdkBuildToolsVersion" >> sdkmanager.log*

*echo "y" | ./sdkmanager --install "emulator" >> sdkmanager.log*

*echo "y" | ./sdkmanager --install "system-images;android-21;google\_apis;x86" >> sdkmanager.log*

*echo "no" | ./avdmanager create avd -n x86emulator -k "system-images;android-21;google\_apis;x86" -c 2048M -f >> sdkmanager.log*

*echo "Downgrade cmdline-tools to v8.0 if already exists so it works with java 1.8"*

*echo "y" | ./sdkmanager --uninstall "cmdline-tools;latest" >> sdkmanager.log*

*echo "y" | ./sdkmanager --install "cmdline-tools;$cmdToolsVersion" >> sdkmanager.log*

*echo "Installing NDK"*

*echo "y" | ./sdkmanager --install "ndk;$ndkVersion" >> sdkmanager.log*

*echo "Provisioning complete. Here's the list of packages and avd devices:"*

*./sdkmanager --list*

*./avdmanager list avd*

1. Verify android setup, execute commands
   1. $ java -version (output should be 1.8)
   2. $ ~/android-tools/android-sdk-tools/cmdline-tools/8.0/bin/sdkmanager –version (output 8.0)

**QT Framework Build from Source**

All the commands executed is with the new user created in user’s home ~ directory.

1. $ git clone <https://github.com/KDAB/android_openssl.git>
2. $ mkdir QT && cd QT
3. $ git clone --depth 1 --branch v5.15.5-lts-lgpl <https://github.com/qt/qt5.git>
4. $ cd qt5
5. $ perl init-repository
6. $ export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64
7. $ *export PATH=$JAVA\_HOME/bin:$PATH*
8. $ mkdir build && cd build
9. $ ../configure -c++std c++17 -opensource -confirm-license -nomake tests -nomake examples \

-xplatform android-clang \

-ssl -openssl-runtime -I ~/android\_openssl/ssl\_1.1/include/ -L ~/android\_openssl/ssl\_1.1/include/ \

-prefix "/usr/local/QT-5.15.5" \

-android-ndk-platform android-29 \

-android-sdk ~/android-tools/android-sdk-tools \

-android-ndk ~/android-tools/android-sdk-tools/ndk/21.3.6528147 \

-android-ndk-host linux-x86\_64 \

-skip qttranslations -skip qtserialport -skip qtwebengine -skip qtpurchasing -skip qtvirtualkeyboard \

-no-warnings-are-errors -disable-rpath

*[On success config.summary file will be created with details.]*

1. *[For the next make command you can use tmux to detach as it takes upto an hour or more. Also make sure nproc give proper core count or it gives errors]*

*$ gmake -j$(nproc)*

1. $ sudo gmake install

[*Successful step, creates a QT folder at the prefix location defined in configure step*]

1. To verify installation execute below command and check.

$ /usr/local/QT-5.15.5/bin/qmake -v

[*Should output QT version*]

**QT Creator**

1. $ sudo apt install qtcreator
2. Launch QT creator from GUI.
3. Goto Devices -> Android in Options/Preferences and set following locations:
   1. Java: /usr/lib/jvm/java-8-openjdk-amd64
   2. SDK Location: ~/android-tools/android-sdk-tools
   3. NDK: [*Should get automatically set when SDK is selected*]
   4. Openssl: ~/android\_openssl
   5. If prompting for any other packages install it except cmdline-tools:latest.

Note: Only cmdline-tools:latest will be missing which is not required, we have v8.0

1. Goto Kits -> QT Versions and manually add location to <qt install>/bin/qmake and Apply changes. Once done successfully the error mark against the QT version should be gone. Also Android QT will be detected in the Kits tab.
2. Done !!!

**Verify Setup**

Create a new widget project and run build, it should run successfully and create an apk.