qt installation and setup

QT for Automotive

download qt:

from japan server:
aria2c -x 16 -s 16 \
https://ftp.jaist.ac.jp/pub/qtproject/official_releases/online_installers/qt-online-installer-linux-x64-online.run

Germany
aria2c -x 16 -s 16 \
https://ftp.fau.de/qtproject/official_releases/online_installers/qt-online-installer-linux-x64-online.run

Singapore
aria2c -x 16 -s 16 \
https://mirror.xtom.com.hk/qtproject/official_releases/online_installers/qt-online-installer-linux-x64-online.run

Download qt creator alone:
(Select the version appropriate for your use case):
aria2c -x 16 -s 16 \https://download.qt.io/official_releases/qtcreator/18.0/18.0.1/qt-creator-opensource-linux-x86_64-18.0.1.run

verfiy:
file qt-online-installer-linux-x64-online.run

expectged output:
ELF 64-bit LSB executable, x86-64

check size:
ls -lh qt-online-installer-linux-x64-online.run

make exe and run:

chmod +x qt-online-installer-linux-x64-online.run
./qt-online-installer-linux-x64-online.run

Install qt with mirror:
QT_MIRROR=https://ftp.jaist.ac.jp/pub/qtproject ./qt-online-installer-linux-x64-online.run

Load maintenance too from mirror:
QT_MIRROR=https://ftp.jaist.ac.jp/pub/qtproject ~/Qt/MaintenanceTool

for andorid:
select qt 5.15.2 (check android)
Check qt creator 18.xx
in build tools:
cmake
Ninja
(
In this process we selected qt version 5.12.2
And inside 5.12.2 check android, and inside build tools check cmake and ninja
)

—> After installation configure android:
Download the adnorid zip fle from:
https://drive.google.com/drive/folders/1jH-qAA0iSIlyGTAequeCct-YRAOdYF6M

After extracting, add the file to the Android SDK Location
Example: /home/votarytech/Android/Sdk

If any android sdk is red, then :

Step 1 Set JAVA_HOME explicitly (must do this)

Even if Java works, Qt Creator needs this **explicitly set**.

Run in terminal:

```
readlink -f $(which java)
```

You should see something like:

```
/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
(if not working or something is showing red, try change java version to
java17(java-17-openjdk-amd64))
```

Now set JAVA_HOME:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export PATH=$JAVA_HOME/bin:$PATH
```

Make it permanent:

```
echo 'export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64' >> ~/.bashrc echo 'export
PATH=$JAVA_HOME/bin:$PATH' >> ~/.bashrc source ~/.bashrc

Install Java 17 (if not installed) (for cmd line tools)
sudo apt update
sudo apt install openjdk-17-jdk

Check:

/usr/lib/jvm/

You should see:

java-8-openjdk-amd64 java-17-openjdk-amd64
```

Step B — Use Java 17 ONLY to fix Android SDK

Run **ONLY in terminal** (not permanent):

```
export JAVA_HOME=/usr/lib/jvm/java-17-openjdk-amd64
export PATH=$JAVA_HOME/bin:$PATH
```

Verify:

```
java -version
```

Must show:

```
openjdk version "17"
```

Now test:

```
~/Android/Sdk/cmdline-tools/latest/bin/sdkmanager --version
```

Setting java version globally:

```
sudo update-alternatives --config java
sudo update-alternatives --config javac
```

Step C — Install required Android components (Android 10)

Android 10 = **API level 29**

```
sdkmanager \ "platform-tools" \ "platforms;android-29" \ "build-tools;29.0.3" \
"ndk;21.4.7075529"
```

Accept licenses:

```
sdkmanager --licenses
```

Step D — Switch BACK to Java 8 (CRITICAL)

Qt 5.15 **WILL FAIL with Java 17**

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export PATH=$JAVA_HOME/bin:$PATH
```

Verify:

```
java -version
```

```
Must show:

openjdk version "1.8.0_xxx"
```

alias use_java17='export JAVA_HOME=/usr/lib/jvm/java-17-openjdk-amd64; export PATH=$JAVA_HOME/bin:$PATH'
use_java17 (now this command will switch to java17)
alias use_java8='export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64; export PATH=$JAVA_HOME/bin:$PATH'
use_java8(now this command will switch to java8)

Run **exactly this** (use Java 17 terminal):

```
sdkmanager --list
```

Install **one platform (minimum)**:

```
sdkmanager "platforms;android-30"
```

Also install:

```
sdkmanager "build-tools;30.0.3"
sdkmanager "platform-tools"
```

Then:

```
sdkmanager --licenses
Accept all if prompted (y)
```

Disabling ipv6:
sudo sysctl -w net.ipv6.conf.all.disable_ipv6=1

Hard reset:
rm -rf ~/Qt
rm -rf ~/.cache/Qt
rm -rf ~/.config/Qt*
rm -rf ~/.local/share/Qt*
rm -rf ~/.local/share/QtProject

Disabling ipv6:
sudo sysctl -w net.ipv6.conf.all.disable_ipv6=1
sudo sysctl -w net.ipv6.conf.default.disable_ipv6=1

OPENING AND SAVING BASH:

nano ~/.bashrc
Edit the necessary and Ctrl o, enter, ctrl x
source ~/.bashrc

android emulator

Only do after sdk manager is installed and works with java 17(didnt work with java 8)

Step 1: Enable x86_64 ABI

In **Qt Creator**:

```
Projects → Android → Build Settings
Android ABIs:
☑ x86_64
☐ arm64-v8a
☐ armeabi-v7a
```

📌 Emulator works **best with x86_64**.

Step 2: Create Emulator (AVD)

Run:

```
~/Android/Sdk/emulator/emulator -list-avds
```

If none exist, create one:

```
~/Android/Sdk/cmdline-tools/latest/bin/sdkmanager \
"system-images;android-30;google_apis;x86_64"
```

Then:

```
~/Android/Sdk/cmdline-tools/latest/bin/avdmanager create avd \ -n
qt_test_emulator \ -k "system-images;android-30;google_apis;x86_64"

Or:

~/Android/Sdk/cmdline-tools/latest/bin/avdmanager create avd \
-n QtEmulatorX86 \
-k "system-images;android-30;google_apis;x86_64" \
-d pixel \
--force
```

When asked:

Do you wish to create a custom hardware profile? [no]

Step 3: Start Emulator

```
~/Android/Sdk/emulator/emulator -avd qt_test_emulator
```

Start emulator in SAFE MODE

First run (no snapshots, software GPU):

```
~/Android/Sdk/emulator/emulator \
-avd QtEmulatorX86 \
-no-snapshot \
-gpu swiftshader_indirect
```

Step 4: Deploy from Qt Creator

1. **Select Android kit**
2. **Run (▸)**
3. Qt will automatically detect the emulator
4. APK installs & launches