



Qt Training – SW Installations

Tino Pyssysalo

tino.pyssysalo@qt.qio

January 2017 – Version 5.8





OS Requirements

- › Any of the following platforms can be used in the training
 - › Windows 7, 8.1 or 10
 - › Linux Ubuntu 14.04 or newer, Red Hat Enterprise Linux 6.6 or 7.2 openSUSE 42.1
 - › macOS 10.9 or newer
- › More details in <http://doc.qt.io/qt-5/supported-platforms.html>





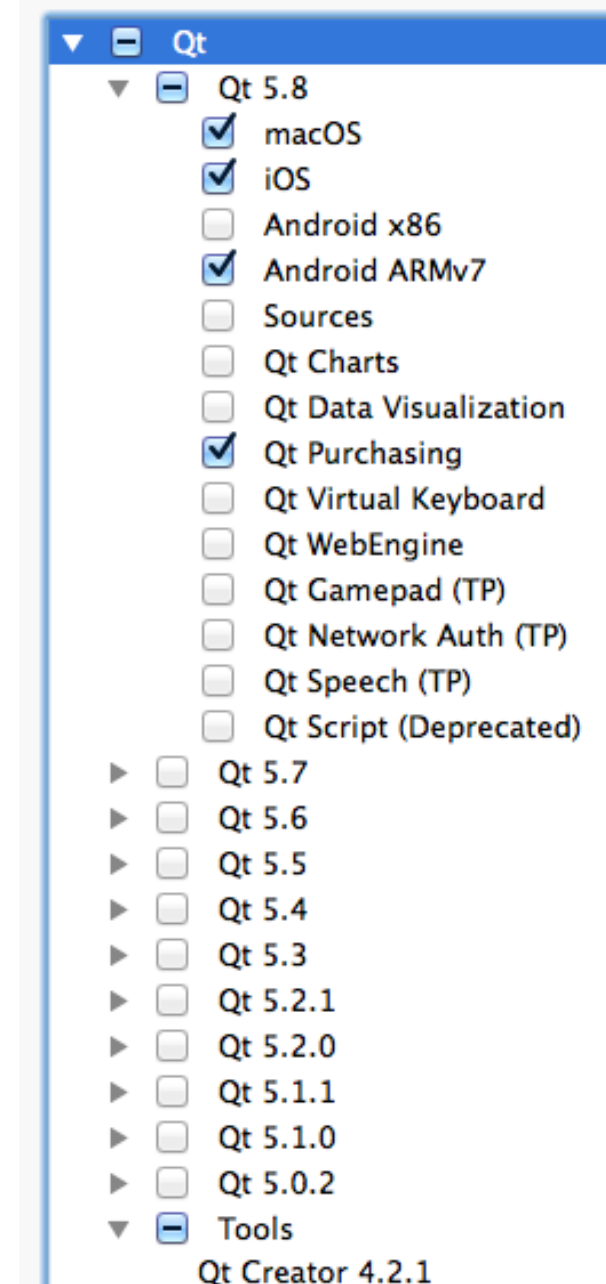
Required SW Installations

- › Please install the latest Qt (5.8) libraries before the training
- › Option1
 - › If you already have Qt installed, use the Maintenance Tool to update libraries, if needed
- › Option 2
 - › Use or create the Qt Account in <http://www.qt.io>
 - › Select Downloads and download Qt Online Installer
- › Option 3
 - › Download online installer from <https://www.qt.io/download-open-source/>



Qt Online Installer

- › Run Qt Online Installer and select the following Qt packages
- › Qt 5.8 binaries for your toolchain
 - › In Windows, select either the MSVC toolchain, used in your platform or MinGW toolchain
 - › In Linux, select GCC
 - › In Mac, select OS X
- › In addition to the Qt binaries, select at least Android ARMv7
 - › iOS/WinRT are optional
 - › Qt Purchasing is optional
- › Tools
 - › QtCreator 4.2.1 – selected by default





Additional Required Software – Linux and Mac

> Linux

- > Build tools: `sudo apt-get install build-essential`
- > OpenGL libs: `sudo apt-get install freeglut3 freeglut3-dev`

> Mac

- > Xcode latest version
- > Xcode command line tools: Install from Xcode: `Xcode>Open Developer Tool>More Developer Tools>Command line tools`





Additional Required Software – Windows

- › Toolchain: You may install e.g. Visual Studio 201x Express or use full Visual Studio Edition
 - › Or use MinGW toolchain instead of Visual Studio
- › MSVC users need to download and install also the command line debugger **cdb.exe**
- › Look at <http://doc.qt.io/qtcreator/creator-debugger-engines.html> “Installing Native Debuggers”
 - › You need to setup the debugger in QtCreator: Tools>Options: Build&Run: kits
 - › If you cannot change the default kit, clone it and set the debugger in the cloned kit






Mobile Setup


- › Attendees are not expected to have mobile devices in the training as emulators will be used
 - › iOS users may look at the details to setup the phone here: <http://doc.qt.io/qt-5/ios-support.html>
- › Android tools are needed in the training as well
 - › Launch QtCreator and setup Android tools in Tools > Options: Devices – Android
 - › Click the buttons to load the tools
 - › Setup the paths to the tools
 - › Create also a virtual device (or setup a real device)
 - › Screen shot in the next slide
 - › To setup your own Android device look at the details at <http://doc.qt.io/qt-5/androidgs.html>

Devices

Devices | **Android** | iOS | QNX


JDK location: Choose... 

Android SDK location: Choose... 

Android NDK location: Choose... 

Found 22 toolchains for this NDK.



Automatically create kits for Android tool chains

 Qt versions for 4 architectures are missing.
To add the Qt versions, select Options > Build & Run > Qt Versions.

Use Gradle instead of Ant

Ant executable: Choose... 

AVD Manager

System/data partition size:  

AVD Name	AVD Target	CPU/ABI
AVD	API 19	armeabi-v7a
test	API 19	armeabi-v7a



Thank you

tino.pyssysalo@qt.io

