BogDan Vatra <bogdan@kdab.com>
KDAB, Qt on Android

How many of you have an Android device ?

Well, you are in luck !

We are going to learn how to use Qt and Qt Creator to target your Android device !

BogDan Vatra <bogdan@kdab.com>
KDAB, Qt on Android

Step by step Qt on Android tutorial

Who am I ?

- I am Bogdan Daniel Vatra (AKA BogDan).
- C/C++ developer for over 14 years.
- Qt developer for over 11 years.
- Initial author of Qt port on Android.
- The author and the current leader of Necessitas project (Qt4 port on Android).
- Active KDE (necessitas) and qt-project contributor.
- Last but not least a KDABian !

Overview

- **Qt status.**
- Development setup for Android.
- Using Qt Creator for Android.
- Deployment options.

Qt Essentials status

Qt Core

- **5.1 & 5.2**
  - missing system semaphores and shared memory.
- **5.3**
  - Shared memory is on my TODO list

Qt Essentials status

Qt Multimedia

- **5.1**
  - video and audio works
  - missing camera support
- **5.2**
  - brings camera support
- **5.3**
  - ATM no other plans

Qt Essentials status

Qt Network

- **5.1**
  - missing SSL support
- **5.2**
  - brings SSL support
- **5.3**
  - ATM no other plans

Qt Essentials status

Qt Quick Controls

- **5.1**
  - missing android native style
- **5.2**
  - brings android native style
- **5.3**
  - ATM no other plans

Qt Essentials status

Qt SQL

- **5.1**
- **5.2**
- **5.3**
  - only sqlite is provided by Qt-Project SDK

Qt Essentials status

Qt WebKit & Qt WebKitWidgets

- **5.1**
- **5.2**
  - missing
- **5.3**
  - we'll see, any volunteer(s) ?

Qt Essentials status

Qt Widgets

- **5.1**
  - missing android native style
- **5.2**
  - brings android native style
- **5.3**
  - ATM no other plans

Qt Essentials status

- **Qt GUI**
- **Qt QML**
- **Qt Quick**
- **Qt Quick Layouts**
- **Qt Test**
  - **just work on all Qt versions**

Qt Add-Ons status

Qt Android Extras

- **5.1**
  - missing
- **5.2**
  - additional functionality for development on Android
    - QJNIEnvironment, access to the JNI Environment
    - QJNIObject, C++ wrapper around a Java class
- **5.3**
  - android services/binder support is on my TODO list

Qt Add-Ons status

Qt Bluetooth

- **5.1**
  - missing
- **5.2**
  - missing
- **5.3**
  - on my TODO list

Qt Add-Ons status

Qt NFC

- **5.1**
  - missing
- **5.2**
  - missing
- **5.3**
  - on my TODO list

Qt Add-Ons status

Qt Positioning

- **5.1**
  - missing
- **5.2**
  - missing
- **5.3**
  - on my TODO list

Qt Add-Ons status

Qt D-Bus

- **5.1**
- **5.2**
- **5.3**
  - missing, android uses the binder IPC.

Qt Add-Ons status

Qt Sensors

- **5.1**
  - commonly used sensors
- **5.2**
  - more sensors added
- **5.3**
  - ATM no other plans

Qt Add-Ons status

Qt PrintSupport

- **5.1**
- **5.2**
- **5.3**
  - missing, no native print support on Android

Qt Add-Ons status

Qt OpenGL

- **5.1**
- **5.2**
  - limited to one top level widget
  - can't mix QGLWidget with other QWidget
- **5.3**
  - there is hope to use one more top level widget
  - can mix QGLWidget with other QWidgets

Qt Add-Ons status

Qt SerialPort

- **5.1**
- **5.2**
  - missing
- **5.3**
  - any volunteer(s) ?

Qt Add-Ons status

- **Qt Concurrent**
- **Qt Declarative**
- **Qt GraphicalEffects**
- **Qt ImageFormats**
- **Qt Script**
- **Qt ScriptTools**
- **Qt SVG**
- **Qt XML**
- **Qt XMLPatterns**
  - **just work on all Qt versions**

Overview

- Qt status.
- **Development setup for Android.**
- Using Qt Creator for Android.
- Deployment options.

Setting up the development environment for Android

Supported platforms:

- GNU/Linux
- Windows
- Mac

For a painless experience I do recommend GNU/Linux. For the rest of the presentation I'll refer only to GNU/Linux.

Setting up the development environment for Android

Install **ant** and **(open) JDK 6** (JDK 7 has a known issue when signing the package which is fixed in Qt Creator 3.0).

On debian based systems you can use the following command:
apt-get install ant openjdk-6-jdk

Setting up the development environment for Android

Download QtProject's SDK from

http://qt-project.org/download

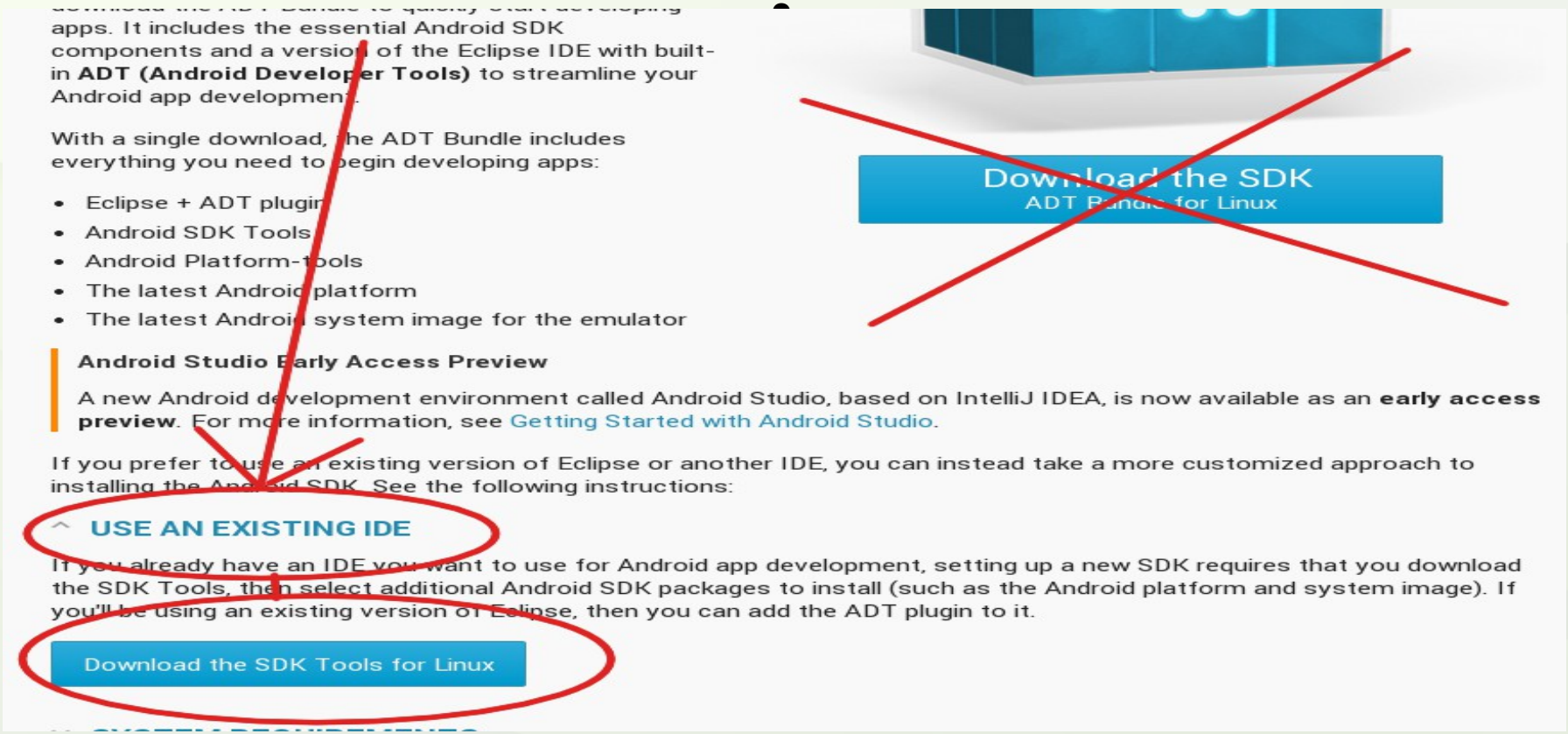
Setting up the development environment for Android

Qt Developer Days 2013

Download Android SDK (ver. 22+) from

http://developer.android.com/sdk/index.html

You need to download ONLY the SDK not ADT Bundle or Android Studio !



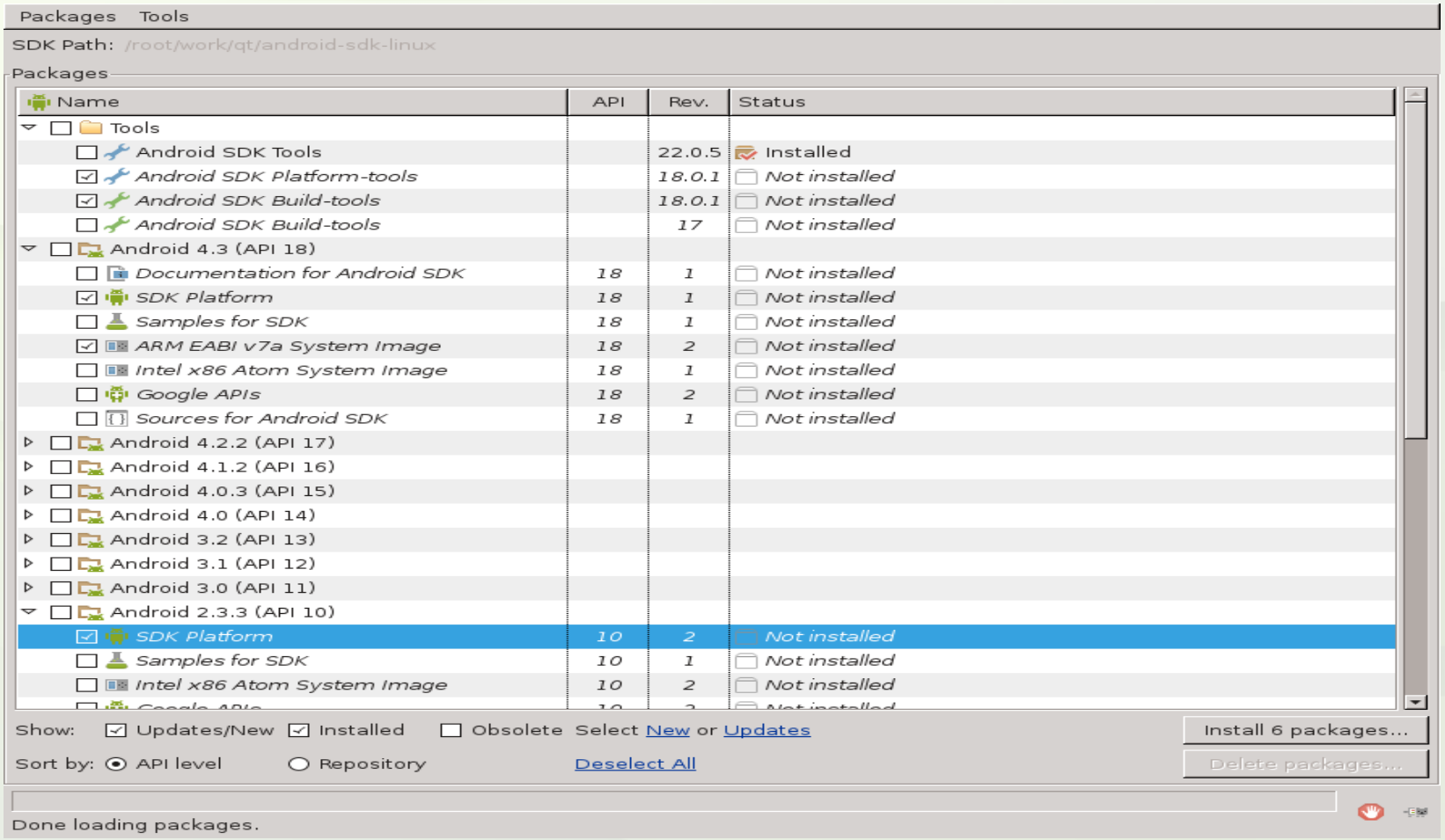
Setting up the development environment for Android

Download Android NDK (ver. r9+) from
http://developer.android.com/tools/sdk/ndk/index.html

Setting up the development environment for Android

Qt Developer Days 2013

Extract the NDK&SDK and run **android-sdk/tools/android** tool and install **SDK Tools, SDK Build-tools** and **Android API-10 SDK Platform**. If you are planning to build Qt yourself you'll need to install also **API-11**.



Setting up the development environment for Android

Enable **USB Debugging** on your device.
On GNU/Linux you have to set the **USB permissions** for your device.

Android provides a detailed page on this matter.
Please check:
http://developer.android.com/tools/device.html

run
android-sdk/platform-tools/adb devices
to see if you enabled the USB debuging and set the permissions correctly.

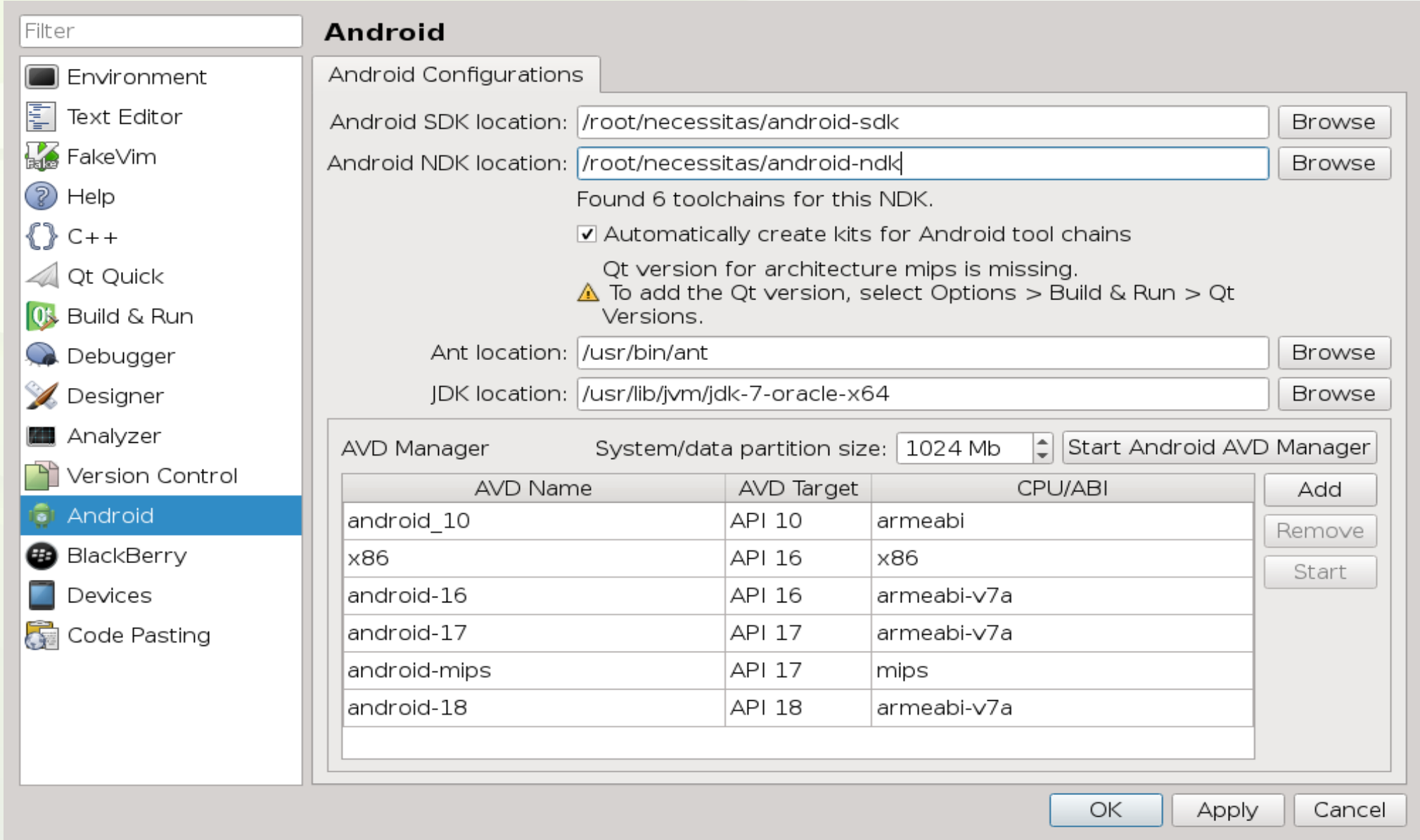
Setting up the development environment for Android

- Setting up Qt Creator for Android
  - Go to **Tools->Option->Android** settings page
    - Set **Android SDK location**
    - Set **Android NDK location**
    - Make sure **Automatically creates kits for Android tool chains** is **checked**.
    - Set **Ant locaion**
    - Set **JDK location**
    - Click **Apply** button **!**

- If you don’t see the Android page, it means that the plugin is disabled and you must first enable it (**Help->About plugins**)

Setting up the development environment for Android

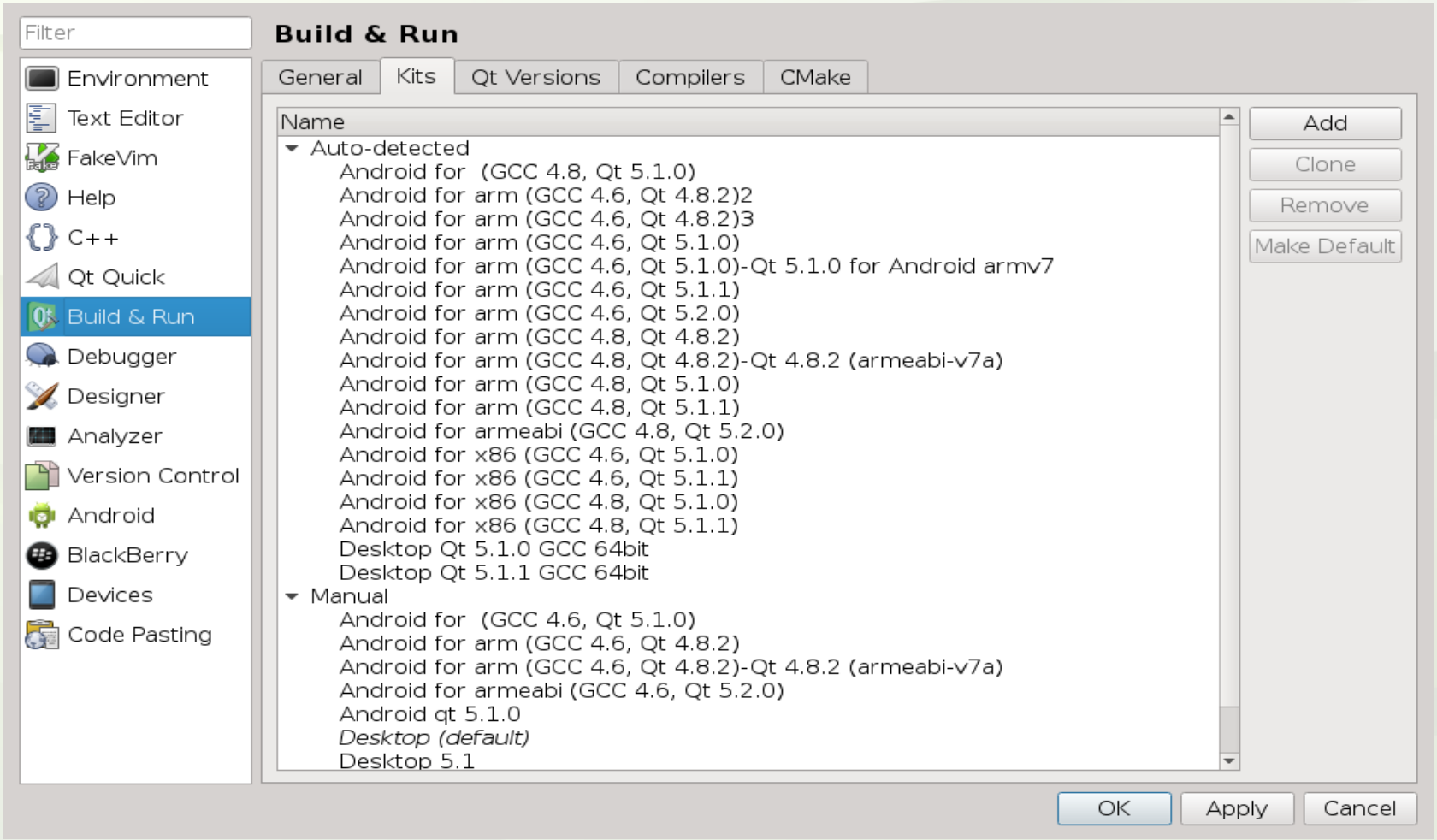
Qt Developer Days 2013



Setting up the development environment for Android

Qt Developer Days 2013

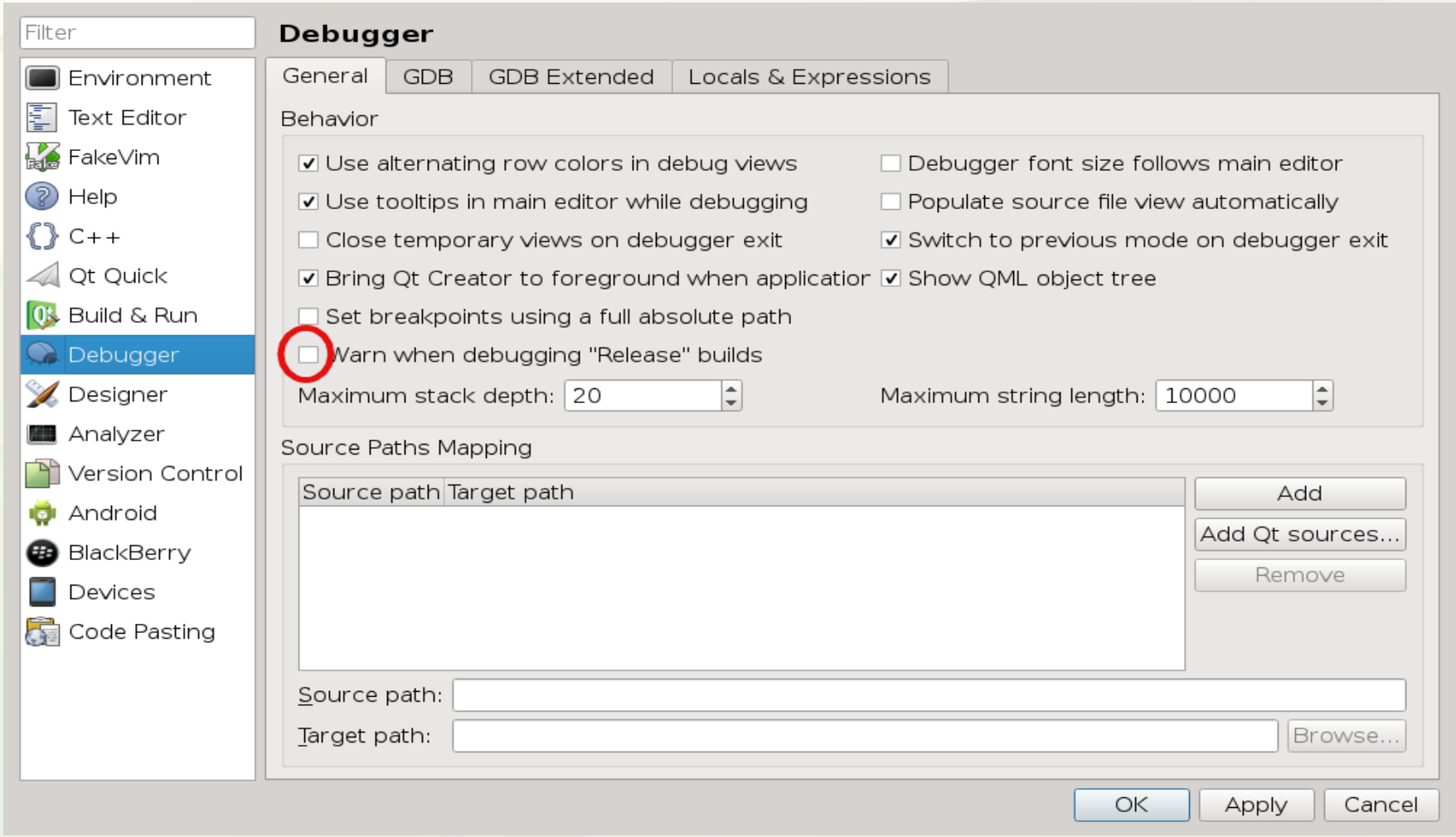
- Setting up Qt Creator for Android
  - check if Qt Creator created the Android kits



Setting up the development environment for Android

Qt Developer Days 2013

- Setting up Qt Creator for Android
  - Uncheck **Warn when debugging "Release" builds**



Overview

- ✔ Qt status.
- ✔ Development setup for Android.
- ➔ **Using Qt Creator for Android.**
- • Deployment options.

Using Qt Creator for Android

Qt Developer Days 2013

This part will cover only the Qt Creator Android specific part not the whole Qt Creator.

- Open/Create a project
- Choose an Android KIT

Using Qt Creator for Android

- Select an Android KIT

Using Qt Creator for Android

After the previous step, Qt Creator will create and add a few files to **android** folder. Most of these files are specific to your project and you should add them to your project SCM.

- The following files are needed to build an android application
  - AndroidManifest.xml
  - version.xml
  - res/*
  - src/*
- Autogenerated file, should not be added to your project SCM.
  - build.xml
  - local.properties
  - proguard-project.txt
  - project.properties
  - assets/*
  - bin/*
  - gen/*
  - libs/*

Using Qt Creator for Android

Warning: **assets** and **libs** folders are not cleaned automatically, so if you are are removing libs/resources, make sure you are removing these folders before you build the final android package.

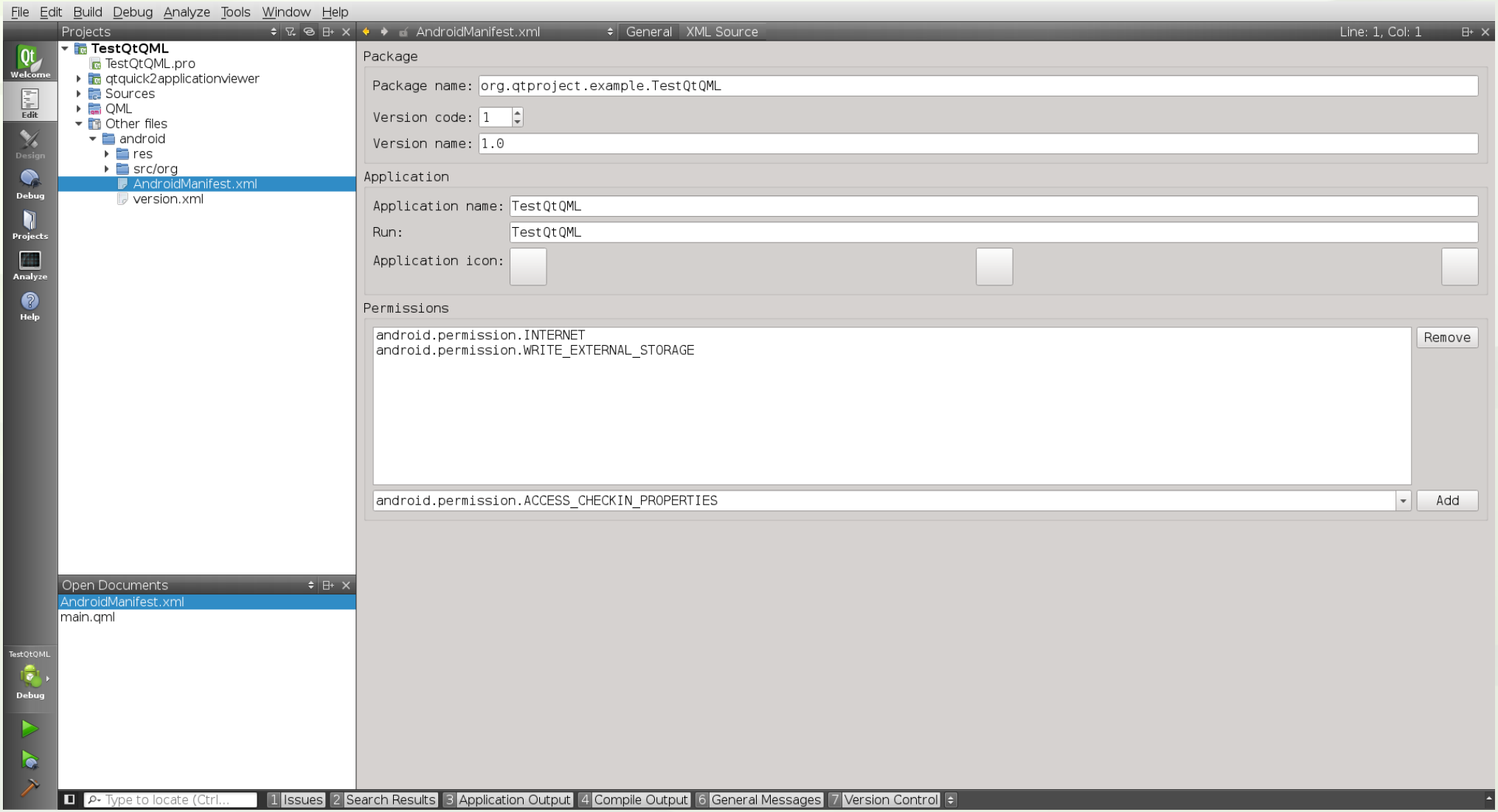
Using Qt Creator for Android

- Setting up the AndroidManifest.xml
  - Package name, reversed URL (e.g. com.kdab.application)
  - Version code, this field is used by Google Play to upgrade your existing applications
  - Version name, is the version displayed in Android settings
  - Application name, is the name displayed in Android launcher
  - Run, this is the application that java part will run
  - Permissions. Here you must add all permissions that your application needs to access (e.g. internet, sd card, sensors, etc.).

Using Qt Creator for Android

Qt Developer Days 2013

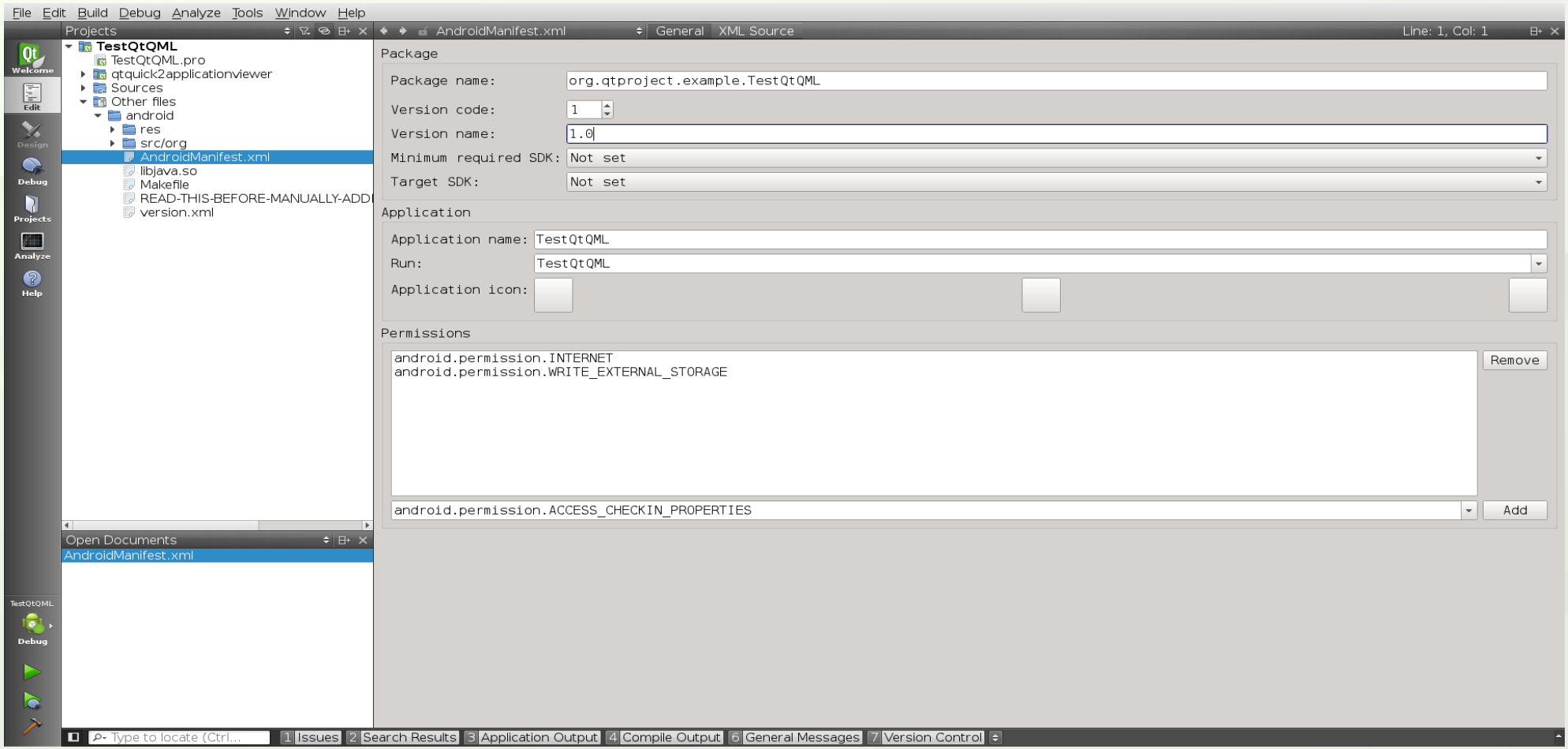
- Setting up the AndroidManifest.xml



Using Qt Creator for Android

Qt Developer Days 2013

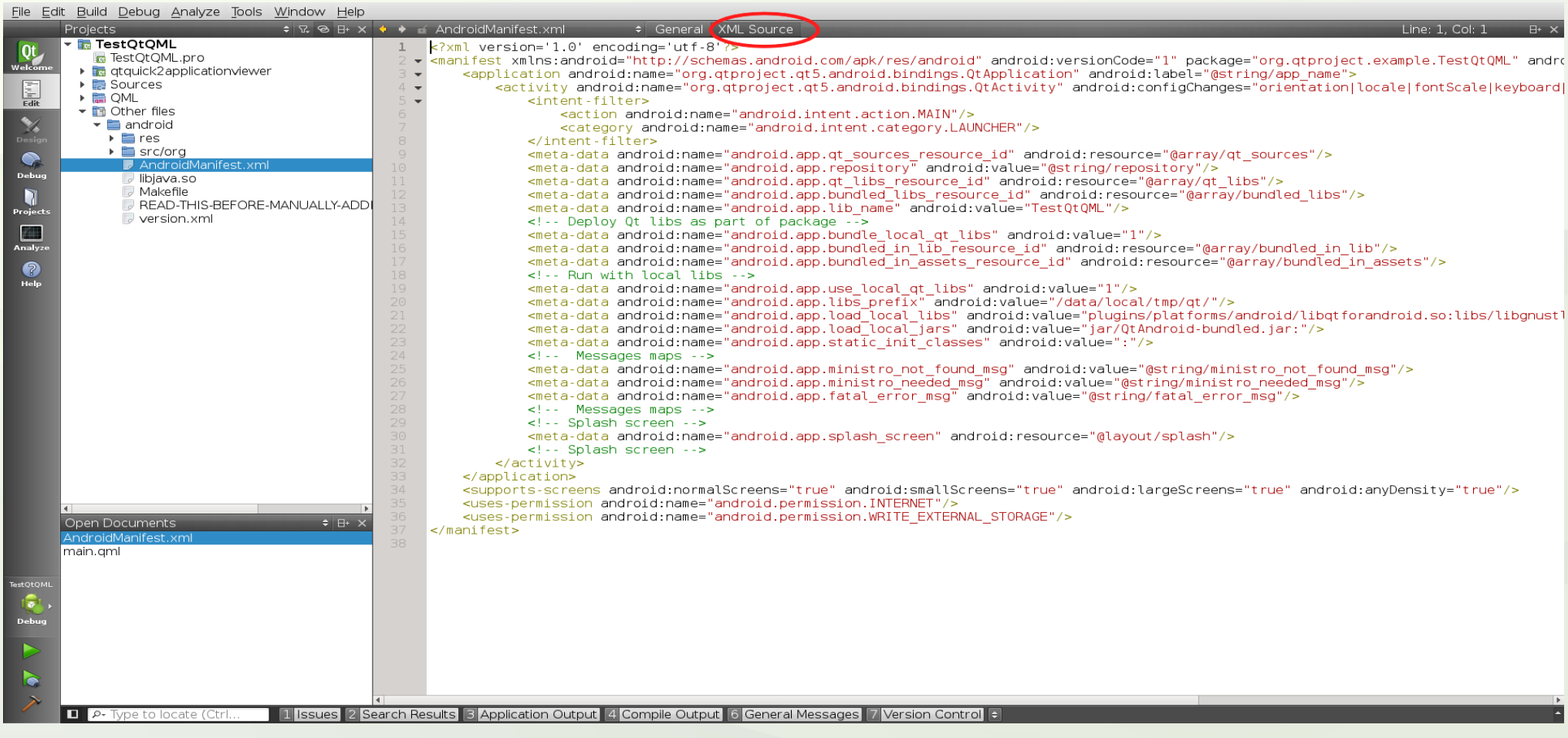
- Setting up the AndroidManifest.xml
  - Qt Creator 3.0 allows you to choose Minimum and Target SDK



Using Qt Creator for Android

Qt Developer Days 2013

- Setting up the AndroidManifest.xml
  - Android manifest is quite complicated sometime you need to edit manually



Using Qt Creator for Android

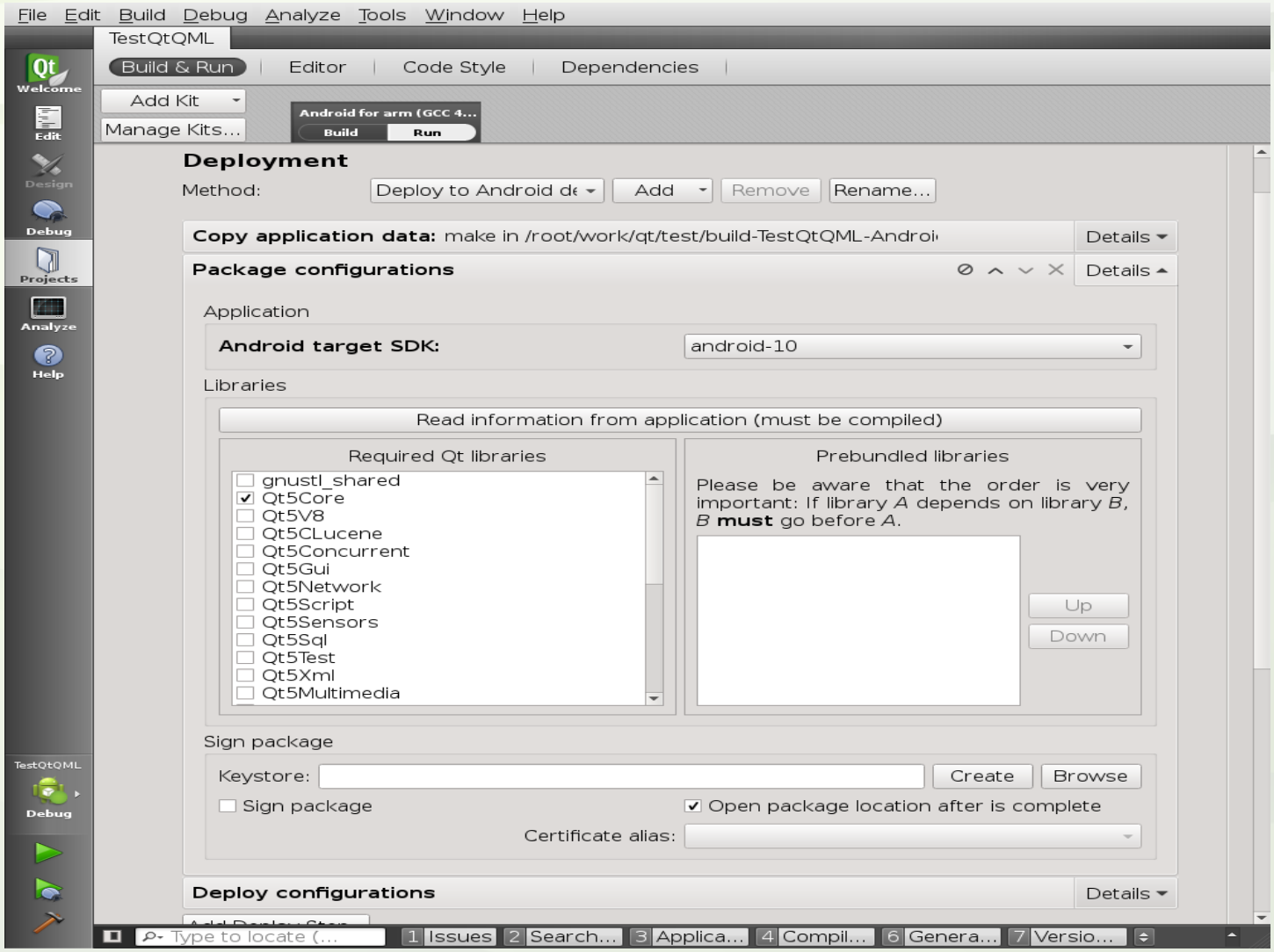
- Setting up the AndroidManifest.xml
  - For more informations about Android Manifest, please check

    http://developer.android.com/guide/topics/manifest/manifest-intro.html

Using Qt Creator for Android

Qt Developer Days 2013

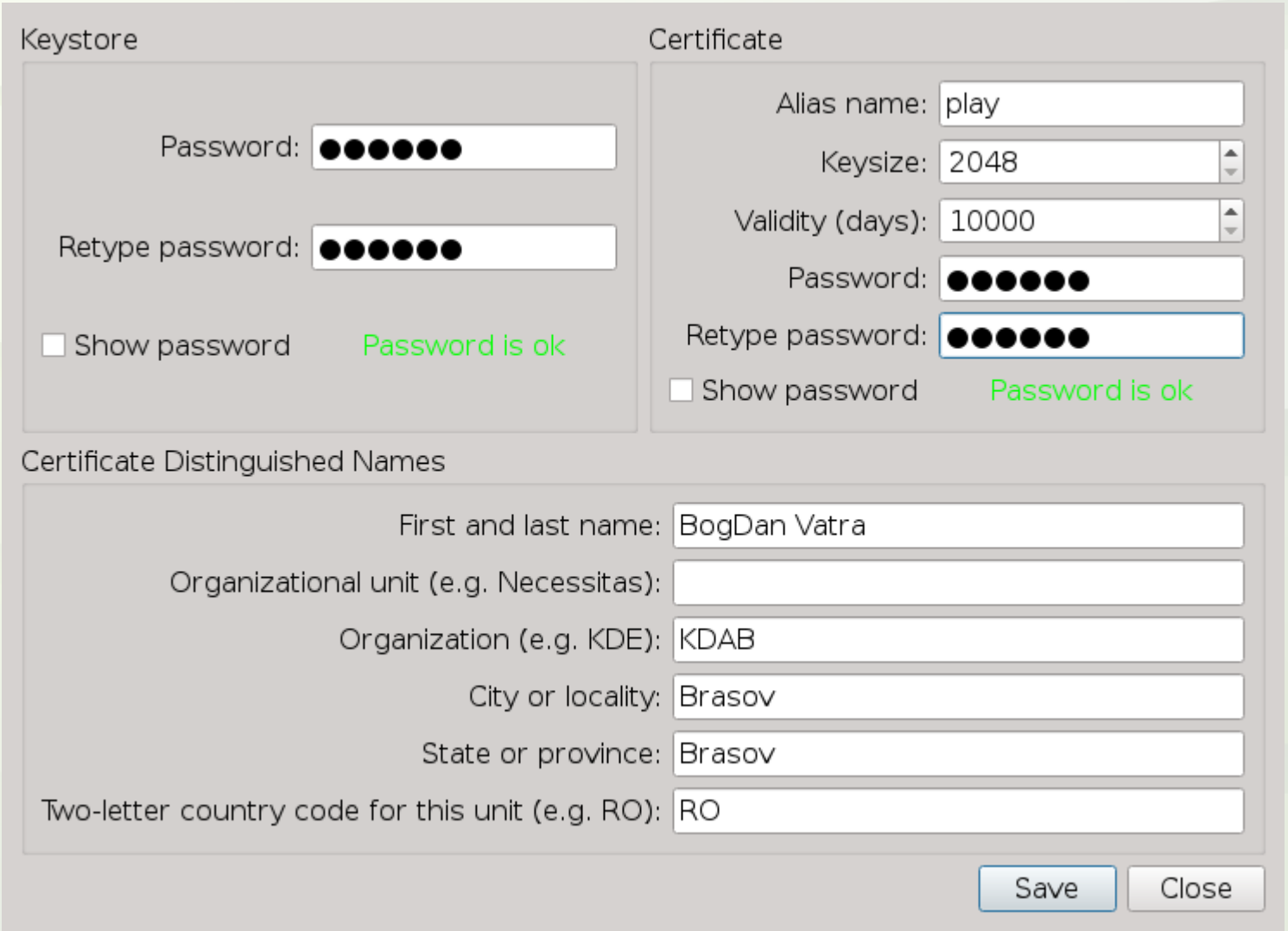
- Package configuration



Using Qt Creator for Android

Qt Developer Days 2013

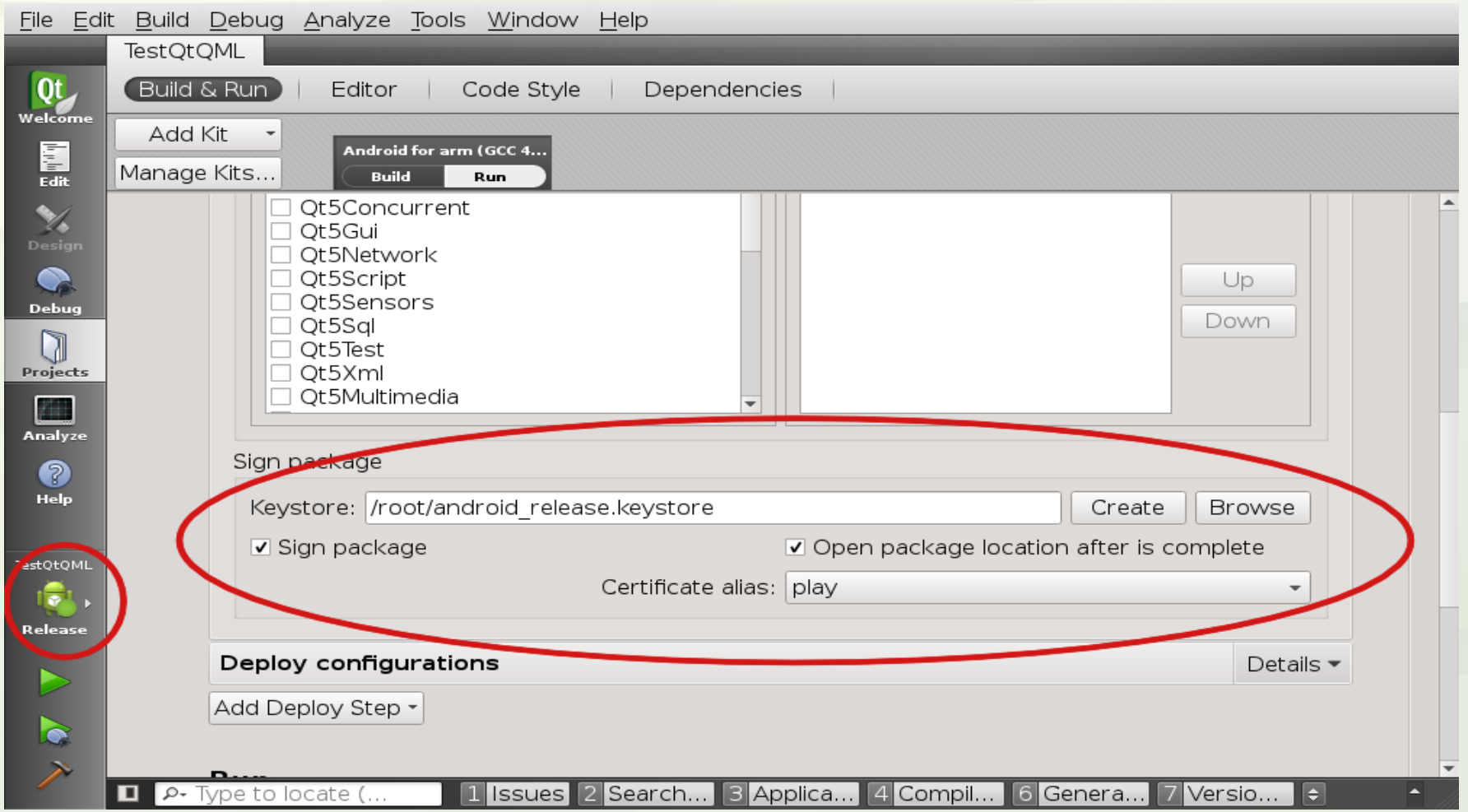
- Signing the application.
  - create a certificate



Using Qt Creator for Android

Qt Developer Days 2013

- Signing the application.
  - switch to release mode
  - open keystore and check “sign package”



Using Qt Creator for Android

- Signing the application
  - If you want to target more than one platform with the same package, then you must build and run in release mode the application for every platform and then sign it !
  - When Qt Creator opens the location of the signed package, there you will have a few .apk files. Only the one with "**-signed**" in the name is the one which is signed and ready for publishing.

Overview

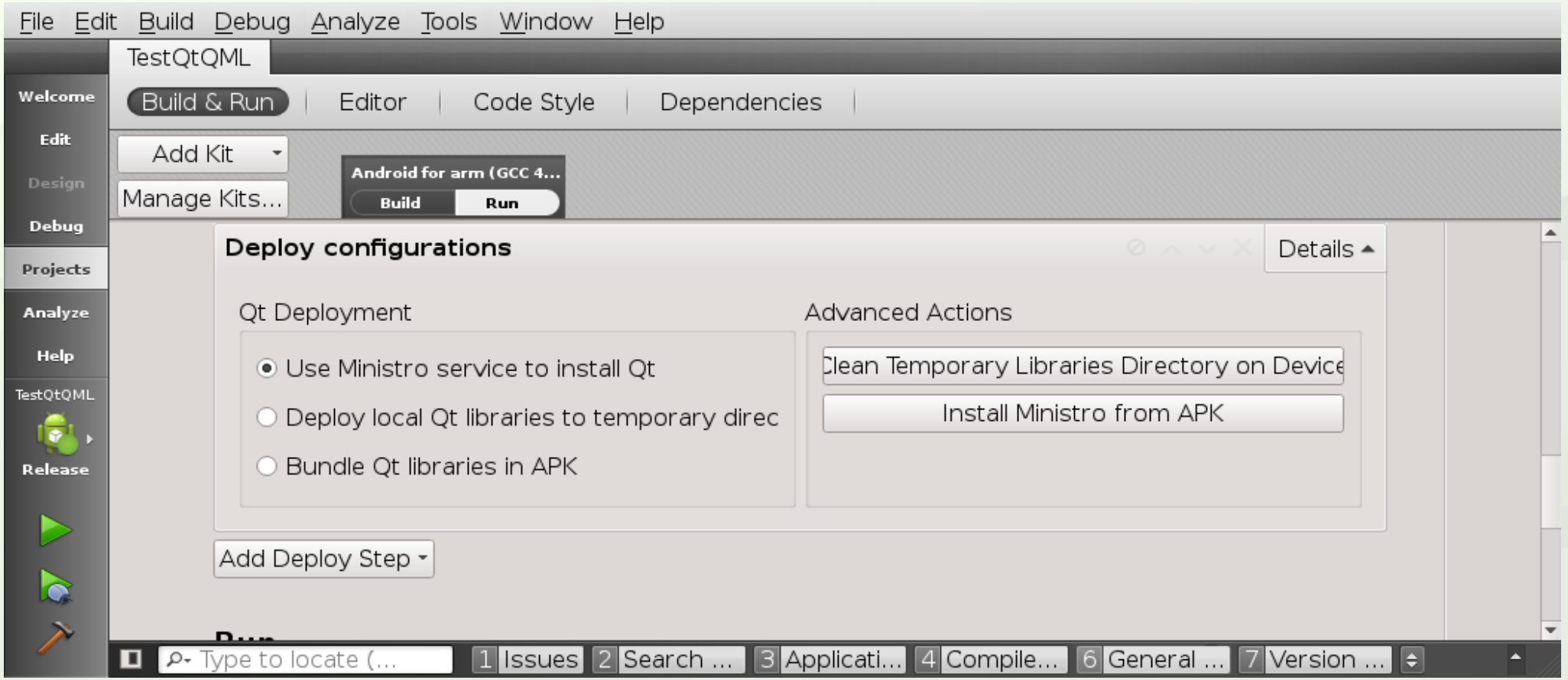
- Qt status.
- Development setup for Android.
- Using Qt Creator for Android.
- **Deployment options.**

Choosing the right deploying system

Qt Developer Days 2013

Qt Creator supports three deploying systems.

- Use Ministro service to install Qt.
- Deploy local Qt libraries to temporary directory.
- Bundle Qt libraries into the APK.



Choosing the right deploying system

- **Deploy local Qt libraries to temporary directory.**
  - This deploy system is used mostly by Qt hackers when hacking on Qt itself.

Choosing the right deploying system

- **Bundle Qt libraries into the APK.**
  - This feature was added recently to Qt Creator. Beside your application and your resources Qt Creator adds all Qt libraries your application needs to run.
  - Pro
    - The APK contains everything it needs to run.
  - Con
    - The APK is HUGE due to Qt libs which are pretty big (+40Mb/platform).
    - All Qt libs must be unpacked! So your application will need a lot of free space to run (+50Mb)
    - Most of the low-end devices users can't afford to spend that much free space.
    - Due to big size you can't target more than one platform/apk. You must create an apk for every platform (armv5, armv7, x86).
    - No VFP on armv5 devices or NEON on armv7 devices.
    - Qt not shared with other Qt apps.
    - No separate libs update.

Choosing the right deploying system

- **Use Ministro service to install Qt.**
  - Why Ministro was invented?
    - In 2009/2010 most devices have limited free space (<100 Mb).
    - Google Market had a lower package size limit than today's 50Mb limit/apk.

Choosing the right deploying system

- **Use Ministro service to install Qt.**
  - How it works
    - your package will contain **ONLY** your application's .so file(s), its resources.
    - application starts and connect to Ministro service
      - opens Android play for the user to install Ministro
    - sends to Ministro the dependencies list
      - downloads missing files
    - Ministro sends back another list with everything the application needs to load.
    - The Application loads everything and starts the Qt application.

Choosing the right deploying system

- **Use Ministro service to install Qt.**
  - Pro
    - Using Ministro, the user needs to download *ONLY* once the Qt libs.
    - Ministro can detect VFP on amv5 and NEON on armv7 download respective libs.
    - Ministro can update Qt libs, without requiring app update.
    - You can easily target all Android platforms with a single APK.
    - You can use your own Ministro sources with your own libraries.
  - Con
    - Not very user friendly?
    - Ministro upgrades Qt libraries and it might break things?

Choosing the right deploying system

- **Use Ministro service to install Qt.**
  - Ministro uses a Debian like release scheme with three different repositories
    - **unstable**
    - **testing**
    - **stable**

Choosing the right deploying system

- **Use Ministro service to install Qt.**
  - Every major Qt release will use a different location for Ministro.
    - http://download.qt-project.org/ministro/android/qt5/qt-5.1/
    - http://download.qt-project.org/ministro/android/qt5/qt-5.2/

That's all folks!

Thank you for your time !

Any questions?