

# Migrating from Qt 4 to Qt 5



**Nils Christian Roscher-Nielsen**  
Product Manager, The Qt Company

**David Faure**  
Managing Director and migration expert, KDAB France

# Qt History

**1995**  
 Troll Tech 1st public release on 20 May,  
 **Qt 0.90** for X11/Linux  
 » Commercial & open source (FreeQt license)

**1998**  
 KDE Free Qt Foundation - guarantees Qt availability for free software development

**2000**  
 » New Qt windowing system - Qt/Embedded - (a.k.a. QWS & Qtopia Core)  
 » Both Qt/X11 & Qt/Embedded under GPL + commercial licenses  
 » GPL v2 with Qt 2.2

**2005**  
 **Qt 4.0** - Total makeover with full re-architecture  
 under commercial & GPL 2.0 (or later) for all  
 platforms, even Windows (Famous Qt 4 dance video)

**2008**  
 Nokia acquires Trolltech - "Qt Software at Nokia"

**2010**  
 Qt Quick launched, WebKit integration, & Symbian support in Qt 4.7

**2012**  
 » Digia acquires all rights to Qt - "Digia, Qt"  
 » **Qt 5.0** - Major overhaul - new modularized codebase,  
 consolidation of QPA (Qt Platform Abstraction), Qt Quick 2,  
 and more support for mobile (WinRT).

**2014**  
 "Digia, Qt" becomes its own entity "The Qt Company", subsidiary of Digia



**1990**  
 Qt conceived by Haavard Nord and Eirik Chambe-Eng on a  
 park bench in Trondheim, Norway.

**1996**  
 » Customer #1 - European Space Agency  
 » **Qt 1.0** - full X11 support free for free software development plus Windows  
 » KDE project established with Qt as its underlying library

**1999**  
 **Qt 2.0** - Qt/X11 open source with QPL  
 (Q Public License)

**2001**  
 **Qt 3.0** - "multiple database environments, multiple  
 languages, multiple monitors" with Mac OS X  
 support and a new Qt Designer GUI builder

**2006**  
 » Trolltech IPO on Oslo Stock Exchange  
 » Greenphone developer smartphone  
 » Qtopia in millions of devices worldwide from Sharp to Motorola

**2009**  
 Qt Creator launched and Qt 4.5 brought LGPL v2.1

**2011**  
 » Digia acquires Qt commercial licensing business  
 » Qt Project launched  
 » Nokia N9 - Nokia MeeGo "Harmattan" smartphone

**2013**  
 Boot to Qt pre-built software stack & Qt WebEngine

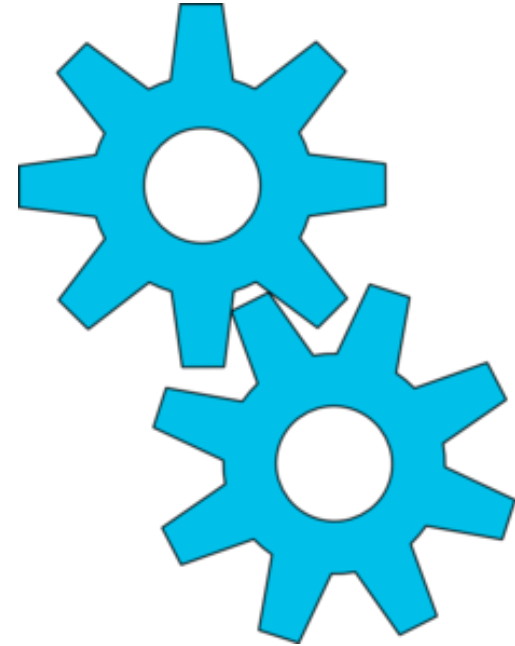
**2015**  
 20th anniversary of Qt's 1st public release  
 » One Qt Site Unification Completed  
 » 800K+ Qt developers worldwide



# Moving to Qt 5

## Motivation

- New user interface requirements
- Embedded devices
- New technologies available
  
- 7 years of Qt 4
  - Time to fix many smaller and larger issues with a new major release

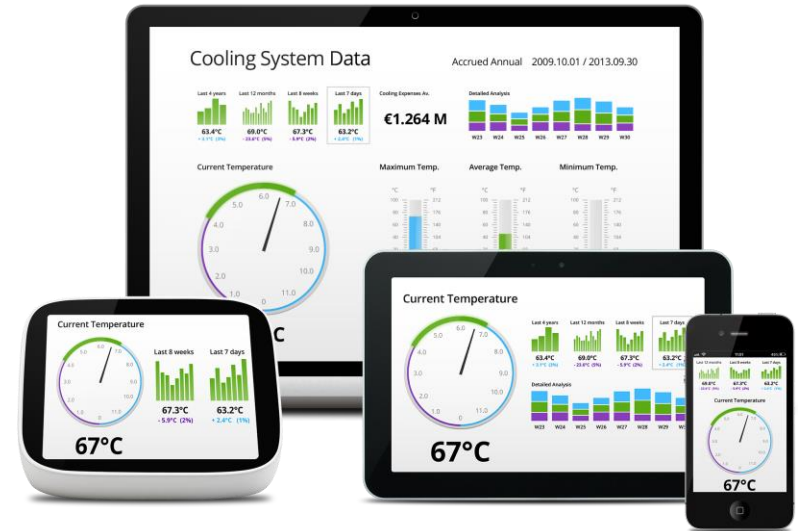


# QML / Qt Quick

## Age of the new User Interfaces



- New industry standards
- More devices than ever
- 60 frames per seconds
- Multi modal interaction
- Enter the SceneGraph
  - Powerful QML User Interfaces
  - Full utilization of OpenGL hardware
  - Full control of your User Interface on all devices



# Embedded Devices

## Qt powers the world

- Qt Platform Abstraction
  - Enables easy porting to any platform or operating system
- Modular architecture
  - Easier to tailor for embedded HW
- Boot to Qt
  - Premade embedded Linux based stack for device creation
- Device deployment
  - Greatly improved tooling
  - On device debugging and profiling



# Wide Platform support

- Seamless experiences across all major platforms
  - Windows, Mac, Linux
  - Windows Phone, iOS and Android
  - Jolla, Tizen, Ubuntu Touch, BB10, and more
  - VxWorks and QNX
- 
- High DPI Support
  - Dynamic GL switching
  - Simplified deployment process
  - Charts and 3D Visualization
  - Location and positioning



# Increased speed of development

For your own applications and for Qt itself

- Qt Creator 3
  - Stable Plugin API
  - Qt Quick Designer
  - QML Profiler
- Modularization
  - More stable and reliable Qt code base
  - Faster module development
  - Easier to create and maintain new modules
- Qt Open Governance model



# Qt UI Offering – Choose the Best of All Worlds



## Qt Quick

C++ on the back, declarative UI design (QML) in the front for beautiful, modern touch-based User Experiences.



## Qt Widgets

Customizable C++ UI controls for traditional desktop look-and-feel. Great for more static embedded UIs for limited devices / Operating Systems



## Web / Hybrid

Use HTML5 for dynamic web documents, Qt Quick for native interaction.





# 20 years of improvement

## 3 years of Qt 5

- Continuous improvement.
  - Speed
  - Stability
  - Performance
  - Features and functionality
- Qt 5.6 introduces Long Term Support
- Still moving into new areas and are reaching more developers and customers than ever



# Migration from Qt 4 to Qt 5

# Migration steps



- Phase 0: Preparing the Qt 4 code base
- Phase 1: Compiling with Qt 5
- Phase 2: Adjusting runtime behavior for Qt 5
- Phase 3: Porting away from deprecated API in Qt 5

# Phase 0

## Preparing the Qt 4 code base

- Porting away from Qt3Support
- Preparing for char \* to QString conversions being UTF-8
- Making QByteArray to char \* conversions explicit
- Fixing include statements
- Enable detection of virtual reimplementations errors

- Comes from having ported from Qt 3 to Qt 4
- Major undertaking
  - 172 deprecated classes, all named Q3\*
  - 346 deprecated methods in QtCore
  - 779 deprecated methods in QtGui
  - out of scope for this presentation, contact me if this applies to you

# Preparing for UTF-8

- `QString("ß")` used latin1 by default in Qt 4
- This was configurable globally with `setCodecForCStrings()`
- Qt 5 uses UTF-8 and only UTF-8
- Prepare by setting `setCodecForCStrings("utf-8")`
- Convert literals by saving source files as utf8
- `char*` variables and `QByteArrays` more problematic
- `QT_NO_CAST_FROM_ASCII` can help, but creates much work

# Making QByteArray to char\* explicit



- Qt 4:

```
const char *name = metaMethod.signature(); otherApi(name);
```

- naive port to Qt 5:

```
const char *name = metaMethod.methodSignature(); otherApi(name);
```

Dangling pointer!

- Enable `-DQT_NO_CAST_FROM_BYTEARRAY` today already.

- `#include <QtGui/QWidget>` won't work anymore
- QtGui was split into QtGui and QtWidgets
- Solution: `#include <QWidget>`, works in Qt 4 and Qt 5
  - assuming a reasonable build system (e.g. qmake or cmake)
- Qt 5 provides a perl script for this



# Enable detection of overrides

- One example:
  - Qt 4: `virtual void QAbstractItemView::dataChanged(const QModelIndex & topLeft, const QModelIndex & bottomRight);`
  - Qt 5: `virtual void QAbstractItemView::dataChanged(const QModelIndex & topLeft, const QModelIndex & bottomRight, const QVector<int> & roles = QVector<int>());`
- Your reimplementation just won't be called anymore. Silent error.
- Let the compiler tell you, add `override` or `Q_DECL_OVERRIDE`.
- Can be automated with `clang-modernize`.
- More examples: `QTranslator::translate`, `QTcpServer::incomingConnection...`

# Phase 1

## Compiling with Qt 5

- Adjusting the build system
- System-specific GUI code: from X11/Win32/QWS to QPA
- Native event filtering API
- Plugin loading
- Widget styles
- ...

- Link to QtWidgets and QtPrintSupport where necessary
- Enable deprecated methods in Qt 5 (until phase 3)
- Windows: Qt 5 is now compiled with builtin wchar\_t, third-party libraries must be recompiled the same way.
- CMake: find Qt 5 using the files provided by Qt
- Phonon no longer part of Qt, add as 3<sup>rd</sup> party dependency

- `Q_WS_*` does not exist anymore
- Qt uses plugins for platform abstraction (QPA)
- XLib no longer used, replaced with XCB
- Native handles no longer accessible for pixmaps, fonts, cursor, etc.
- `QInputContext` no longer exists. For virtual keyboards, see Qt Virtual Keyboard.

- `x11Event`, `winEvent`, `winEventFilter`, `setEventFilter` no longer exist
- New generic mechanism, `installNativeEventFilter`

- `Q_EXPORT_PLUGIN` removed
- Plugins now incorporate JSON data
- `Q_PLUGIN_METADATA` in the header makes moc create the entry point

- No API change, but concrete classes no longer part of the API
- Cannot inherit from QPlastiqueStyle, QWindowsStyle etc. anymore
- Solution : port to QProxyStyle

## Phase 2

### Fixing runtime behavior with Qt 5

- QHash and QSet iteration order
- Only windows have a native window ID / handle now
- QDateTime parsing behaviour changes
- QTime: null time behaves differently in some APIs
- Behavior change in QByteArray-to-QString conversions in Qt 5.6, under discussion



# Phase 3

## Avoiding deprecated methods

- `QUrl::addQueryItem`
- `qInstallMsgHandler`
- `QAbstractItemModel::setRoleNames`
- `QAbstractItemModel::reset`
- `toAscii`, `fromAscii`
- `QWebKit` is deprecated, in favour of `QWebEngine`
- ...

# This is too much!



- KDAB to the rescue
- Free estimation
- Huge experience with migrations
- Faster and cheaper than doing it internally
  - Automated scripts
  - Know-how for manual parts
  - Deep knowledge of Qt and the Qt 3 / Qt 4 / Qt 5 differences

# Thank You!

[www.kdab.com](http://www.kdab.com)

[www.qt.io](http://www.qt.io)

David Faure

<david.faure@kdab.com>

Nils Christian Roscher-Nielsen

<nils.roscher-Nielsen@theqtcompany.com>