Why you should be excited about Qt 5

Thiago Macieira, Qt Core Maintainer
Software Architect, Intel OTC
Berlin, Nov 13-14, 2012
Santa Clara, Dec 6-7, 2012
Who am I?

• Open Source developer for 15 years
• Software Architect at Intel’s Open Source Technology Center (OTC) since last year
  – Living in Portland, Oregon
• Maintainer of two modules in the Qt Project
  – QtCore and QtDBus
• MBA and double degree in Engineering
• Previously, led the “Qt Open Governance” project
  – Ended with the creation of the Qt Project
We’ve come a long way

• 1999-06-25: Qt 2.0
• 2001-10-15: Qt 3.0
• 2005-06-27: Qt 4.0
Habemus Betam 2

• Released today
• Get it while it’s fresh!
  – In your hotel, so you don’t kill the connection for everyone
  – Or the USB sticks

http://qt-project.org/downloads
Goals

• Works for everyone (desktop, embedded, mobile)
• New features
• State-of-the-art UIs
• Increased modularity
• Reduced footprint
• Compatible with Qt 4.x
What is new in Qt 5.0?

• Qt Quick 2

• New graphics stack

• New modular structure

• All platforms based on QPA
New features

• Performance improvements
• OpenGL w/ ANGLE support
• Wayland support
• JSON support
• Mimetype support
• QStandardPaths
• XCB instead of Xlib
• QRegularExpression based on PCRE

• QDnsLookup
• New Signal/Slot connection mechanism
• C++11 support
• WebKit2 architecture in QtWebKit
• V8 as JS engine
• …
New features: C++11 support

• Certain new functionality only in C++11
  – Inline UTF-16 support for QString
  – New signal / slot syntax
  – Performance

• Qt continues to support C++98 (for now)
New features: new signal/slot connection syntax

• Compile-time checking of:
  – Existence of the signal and the slot
  – Argument compatibility

• Works best with C++11

• Advantages:
  – Proper namespace and typedef support
  – Automatic type promotion & demotion
    long → int
    const char * → QByteArrayList or QString

```cpp
QObject::connect(s, &SenderObject::signal1, r1, &ReceiverObject::slot1);
QObject::connect(s, &SenderObject::signal1, [=]() { s->dumpObjectInfo(); });
```
UIs: new style for desktops

• New modern style, called Fusion
• Replaces previous styles
  – Motif, Plastique, Cleanlooks
  – Still available in a separate module
UIs: QML in the spotlight

• QML is our bet in “the next step in GUI programming”
• Easier interaction with designers
• Easier to modify, update, maintain
• Implementations:
  – Qt Quick 1
  – Qt Quick 2
  – Cascades
  – Desktop Components
UIs: Qt Quick 2 and OpenGL (ES)

• Qt 5 supports two rendering paradigms
  – Software rasteriser/QPainter: used by QWidgets
  – OpenGL (ES): used by Qt Quick 2

• QPainter
  – Optimised for many years

• Qt Quick 2 is built and optimized for OpenGL
  – Even SW rasterisation can give great performance
Modularity: Structure of Qt 5

SDK

Tools

Qt Add-Ons

Qt Essentials

Platform:

Linux
Wayland
X11
Windows
Mac OS X
Your platform
...
Modularity: Qt 5 Platforms

Reference platforms
• Linux* (X11 and Wayland)
• Mac OS X*
• Windows*

Other supported platforms
• QNX, DirectFB
• Android, iOS being worked on

* Other names and brands may be claimed as property of others
Modularity: Tiers

Tier 1
• Tested all the time, especially at release time
• Team behind it is committed to supporting it

Tier 2
• Tested at release, but not all fixes in
• May raise in the future

Tier 3
• Code is present, but support level is unknown
Modularity: Qt Essentials

Always available:

• Qt Core
• Qt Network
• Qt Gui
• Qt Widgets (for desktops only)
• Qt WebKit
• Qt Multimedia
• Qt Sql
• Qt Quick 1 and 2
## Modularity: Add-on modules

### For specific contexts

<table>
<thead>
<tr>
<th>Qt Widgets (desktop)</th>
<th>Qt WebKit (WebKit1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Qt Quick components</td>
<td>Qt Multimedia Widgets</td>
</tr>
<tr>
<td>Qt Svg</td>
<td>Qt Feedback</td>
</tr>
<tr>
<td>Qt Xml</td>
<td>Qt 3D</td>
</tr>
<tr>
<td>Qt XmlPatterns</td>
<td>Qt Compositor</td>
</tr>
<tr>
<td>Qt Script, Qt Script Tools</td>
<td>Qt Wayland</td>
</tr>
<tr>
<td>Qt Concurrent</td>
<td>Qt Graphical Effects</td>
</tr>
<tr>
<td>Qt Print Support</td>
<td>Qt Json Db</td>
</tr>
<tr>
<td>Qt Help</td>
<td>Qt Mime Type</td>
</tr>
<tr>
<td>Qt UiTools</td>
<td>Qt Organizer</td>
</tr>
<tr>
<td>Qt Designer</td>
<td>Qt Contacts</td>
</tr>
<tr>
<td>Qt ActiveQt (Windows)</td>
<td>Qt Versit</td>
</tr>
</tbody>
</table>

And more being added...
Modularity: why addons?

• Simpler to maintain for us
  – Different release schedules
  – Quicker to QA and test

• Simpler for new projects to be added
Footprint: Lean and mean QtGui

- Widget classes moved to a separate module (QtWidgets)
- QtGui concentrates on basic tasks
  - Window management (QWindow) & windowing system integration
  - Raster painting (QPainter, QImage)
  - OpenGL / OpenGL ES support
Footprint: Widgets in Qt 5

• In “Done” state of development
  – Will fix important bugs
  – Community is not adding new features or improving performance

• Currently recommended for desktop GUI

• Does not require OpenGL or JS engine
Compatibility: Migrating from Qt 4 to Qt 5

• Compatible with Qt 4 with very few exceptions
• Qt Widgets are supported in Qt 5
• Doesn’t require migration to Qt Quick
• Doesn’t require OpenGL
• Doesn’t require JavaScript
Compatibility: Embedded support

**EGLFS**
- Uses EGL
- Full screen applications
- Single process only

**DirectFB**
- Blitting acceleration
- Input handling
- OpenGL support possible

**Wayland**
- Designed for HW acceleration
- Wayland 1.0
- Qt Compositor API
- Integrates with other Wayland clients and servers
Done under the Qt Project

The Qt Project celebrated 1 year on October 22

http://qt-project.org
Current state

We’re in Beta!

• Feature freeze: March 2012
• Alpha: April 4 2012
• Beta 1: August 30 2012
• Beta 2: today
• Final: by the end of the year
After 5.0: Releases

Hybrid time / quality based release schedule

• 2 feature releases per year
After 5.0: Embedded systems

Recover time lost

• Cooperate with vendors and embedded Linux distributions
  – Ready for your use
  – Yocto Project, Open Embedded, etc.

• Investigate Android on embedded

• Tooling support
  – Cross-compiling, remote debugging, deployment, etc.
  – Integrated into Qt Creator
After 5.0: More platforms

• Definite support for Blackberry and QNX
  – RIM contributing directly to the Qt Project

• Investigating support for
  – Android
  – IOS
  – Windows 8 / RT
After 5.0: Others

- Full Desktop support in Qt Quick
- Integrated Software OpenGL renderer
- Continue to evolve the WebKit2 based Qt WebKit
- More processor architectures for JavaScript
- Improvement on the V4 engine
- Your ideas?
Questions?

Thiago Macieira

thiago.macieira@intel.com

Links:
Website: http://qt-project.org
Mailing lists: http://lists.qt-project.org
IRC: #qt and #qt-labs on Freenode