Qt Development Frameworks:

- Founded in 1994
- Norwegian Trolltech acquired by Finnish Nokia in 2008
- More than 250 employees in eight locations worldwide
- Trusted by over 6,000 customers worldwide

Qt: a cross-platform application and UI development framework

- For desktop, embedded and mobile development
- Intuitive C++ class libraries with JavaScript
- Integrated development tools

Qt licenses support all business models

<table>
<thead>
<tr>
<th>License Type</th>
<th>Commercial</th>
<th>LGPL v. 2.1</th>
<th>GPL v. 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>License Cost</td>
<td>License fee charged</td>
<td>No cost</td>
<td>No cost</td>
</tr>
<tr>
<td>Must provide source code for changes to Qt</td>
<td>No, modifications can be closed</td>
<td>Source code must be provided</td>
<td>Source code must be provided</td>
</tr>
<tr>
<td>Can create proprietary application</td>
<td>Yes—no obligation to disclose source code</td>
<td>Yes, if dynamically linked to Qt library</td>
<td>No, application is subject to the GPL</td>
</tr>
<tr>
<td>Support</td>
<td>Yes, with valid maintenance agreement</td>
<td>Not included, available separately</td>
<td>Not included, available separately</td>
</tr>
<tr>
<td>Charge for Runtimes</td>
<td>Yes—in some instances*</td>
<td>No, distribution is royalty free</td>
<td>No, distribution is royalty free</td>
</tr>
</tbody>
</table>

*Runtime charges apply when the Qt-based application is part of a joint hardware and software distribution and the main UI of the device is controlled by Qt.

Why Qt?

- Write code once to target multiple platforms
- Produce compact, high-performance applications
- Focus on innovation, not infrastructure coding
- Choose the license - Commercial, LGPL or GPL - that fits you
- Count on professional services, support and training
- Take part in an active Qt Open Source ecosystem

Medical Business
Medical Devices
Medical Imaging

MeVis

“Qt empowers our productivity and robustness to rethink our development process. We have to handle the complexity of our application with no adverse effect than we would for a simple platform. Qt increases the speed and the quality of our code. It is further, we are very happy with Qt.”

—Florian Link, Director of Software Engineering, MeVis

It’s clear for me that one of the best decisions in my programming career was to move to Qt after almost 10 years of experience with Microsoft APIs. Switching to Qt means that I have a clearer API now while targeting all major computer platforms. That Qt makes this feasible for small companies at reasonable costs makes me indeed very happy.

—Ranier Boebel, General Manager, Brain Innovation

Nokia

15 years of customer success and community growth
Qt Core Classes

- Qt core modules and utility classes
  - Qt kernel: Object model, event handling, application infrastructure
  - Basic input / output: File, I/O, Streams
  - Threading
    - Signals and slots support, atomic reference counting, per-thread event loops, concurrency support
  - Container classes
    - For storing, sorting and retrieving groups of data
    - Robust, easy to use, low memory overhead, highly optimized
    - Alternative to STL (but Qt and STL work well together)
  - Fast XML stream reader and writer
  - IPC classes
    - Shared memory, local domain sockets, memory mapped files
  - Supporting tools: qmake, rcc, configure, moc

Qt GUI Classes

- Provides rich set of GUI components and supporting functionality
  - User interface components
    - Simple to complex widgets and controls, dialogs
  - Model-View-Controller (MVC) itemviews
  - Styles – ensuring native or custom look and feel on target platform(s)
  - Font-aware layout system
  - Accessibility classes
  - Canvas component with Widget support
**Qt OpenGL Classes**

- Allows you to build your user interface in Qt, display and manipulate 3D model in OpenGL®
- Integrates OpenGL canvas with Qt
- Provides frame buffer and pixel buffer abstraction
- Supports accelerating 2D painting with OpenGL
- Mix 2D painting and 3D scenes
- OpenGL graphics system draws everything using OpenGL

**Qt Development Tools: Overview**

- Qt Creator: Cross-platform IDE
- Qt Designer: GUI layout and forms builder
- Qt Linguist: Internationalization toolset
- Qt Assistant: Customizable documentation reader
- qmake: Cross-platform build tool
- Integration with Visual Studio® and Eclipse® IDEs

**Qt Creator: Cross Platform IDE**

- Lightweight cross-platform IDE for Qt
  - An advanced C++ code editor
  - Integrated, context sensitive help system
  - Integrated visual debugger (based on gdb)
  - Integrated source code management
  - Project and build management tools
  - Qt developer tools integrated
- Benefits
  - New developers get up and running faster
  - Boosts developer productivity
  - Tailored to the needs of Qt developers

**Qt Designer**

- Qt Designer is a powerful, drag-and-drop GUI layout and forms builder
- Features
  - Supports forms and dialog creation with instant preview
  - Integrates with Qt layout system
  - Extensive collection of standard widgets
  - Support for custom widgets and dialogs
  - Seamless integration with Microsoft® Visual Studio .NET and Eclipse
  - Allows preview with embedded profiles
- Benefits
  - Greatly speeds the interface design process
  - Enables native look and feel across all supported platforms
  - Developers work within the environment of their choice, leveraging existing skills

**Qt Linguist**

- Qt Linguist is a set of tools that smooth the internationalization workflow
- Features
  - Collects all UI text and presents it to a human translator in a simple window
  - Support for all languages, including CJK
  - Simultaneous support for multiple languages and writing systems from within a single application binary
  - Load and edit multiple translations at once
  - Support for .po and .xliff
- Benefits
  - Greatly speeds the translation/localization process
  - Enables native look and feel across all supported platforms
  - Easily target international markets

**Qt Assistant**

- Qt Assistant is a fully customizable, redistributable help file/documentation browser
- Features
  - Simple, web-browser-like navigation, bookmarking and linking of documentation files
  - Support for rich text and HTML
  - Full text and keyword lookup
  - Can be customized and shipped with Qt applications
- Benefits
  - No longer have to build a help system from scratch
  - Leverage existing HTML skills
  - Deliver documentation in an easily searchable and navigable format to your end users
**Qt qmake**

- A cross-platform application build tool
- Features
  - Reads project source, generates dependency tree, generates platform-specific project and makefiles
  - Integrates with Visual Studio and Xcode
- Benefits
  - Takes the pain out of cross-platform builds
  - Eliminates the need for makefile generation

---

**Qt Evolution and History**

<table>
<thead>
<tr>
<th>Minor version</th>
<th>Feature Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.0 – A new beginning</td>
<td>New API, improve backend implementation to better adapt to platform change</td>
</tr>
<tr>
<td>4.1 – Graphics</td>
<td>QML, Transparency, OpenGL, PDF</td>
</tr>
<tr>
<td>4.2 – Desktop Experience</td>
<td>Gwenview, Kylo, D-Bus, Improved desktop integration, embedded painting performance</td>
</tr>
<tr>
<td>4.3 – Infrastructure</td>
<td>Cross-platform, Desktop integration, Drag &amp; Drop, Networking, OpenGL ES</td>
</tr>
<tr>
<td>4.4 – Advanced UI</td>
<td>WebKit, Multimedia, Phonon, X11, Widgets in Graphicsview, Consumer, GDI, Windows CE</td>
</tr>
<tr>
<td>4.5 – Qt Everywhere</td>
<td>Mac Cocoa, S60, N900, X11, Script Debugging</td>
</tr>
<tr>
<td>4.6 – User Experience</td>
<td>Animation, State Machine, Multi-touch &amp; gestures, Performance</td>
</tr>
<tr>
<td>4.7 – Apps</td>
<td>Qt Quick, Mobility API</td>
</tr>
</tbody>
</table>

---

**Qt 4.5**

- **WebKit improvements**
  - QWebKit can load any NPAPI-compatible plugin
  - HTML5 multimedia tag support through Phonon

- **Qt Everywhere**
  - Qt for Mac OS X ported to Cocoa Framework

**Performance**

- Major improvements in overall performance
- Special focus on new graphics system
- Benchmarking library added to Qt

**Tools**

- Qt Creator 1.0
- Ability improvements in Linguist and Designer
- Qt 15.1

---

**Qt 4.6**

- **User experience**
  - New animation framework
  - Gestures & Multitouch support
  - Improved graphical capabilities

- **Performance**
  - OpenGL graphics engine
  - Optimizations to resource use
  - Architecture-specific improvements to ARM devices

- **Qt Everywhere**
  - Qt ports to Symbian
  - Qt for Windows 7
  - Qt for Mac OS X 10.6

---

**Qt 4.7**

- **Qt Quick**
  - Complete UI creation kit for developers and designers
  - Extremely fast learning curve with support in Qt Creator and the Qt Meta-Object Language (QML)
  - Ideal for iterative development of expressive, animated, modern UIs with high performance

**Performance**

- Continuous focus on performance improvements
- Dedicated performance team
- New benchmark based development culture that guarantees that Qt can only get faster

**New APIs**

- Bearer management with automatic HTTP level roaming
- Numerous small improvements e.g. in the QGL polish library and QWebkit library

**Quality**

- Internal software process improvements that will increase the quality of Qt and make drop-in upgrades of Qt less risky
- Introduced continuous integration system
- Increased focus on bug triage and bug resolution metrics during development

---

**Qt Research Projects**

- **Web and Hybrid Development**
  - Enhance the hybrid development capabilities of Qt and Qt Tools, combining web technologies and Qt technologies
  - WebApp and WebDevice, QML to C++ code generation
  - Symbian state chart compiler

- **Tools Research**
  - New build system with enhanced support for cross-platform application development on mobile/embedded and desktop platforms
  - Tools for enhancing multi-user and/or multi-site project collaboration

- **Mobility Project**
  - New API for web uses that are common in mobile development (may be relevant to embedded and desktop, too)
  - WiMob, Services Framework, Bearer Management, and Contacts API

- **UI and Visualization**
  - 3D user interfaces
  - Next generation widgets
  - Multi-threaded painting
GraphicsView

Scene-based Graphics View
- View stores all items and propagates events
- Scene is a surface for mapping items
- Items & widgets represent a specific object
- Scene uses binary space partitioning for fast item access
- Widget placement uses floating-point numbers (transforming the view)

Widgets on Canvas

WebKit Integration

- An open source HTML rendering component integrated with Qt
- Web standards compliant
  - Support for HTML, XHTML, XML, stylesheets, JavaScript, HTML editing, HTML canvas, AJAX, XML, SVG, some XUL
  - Acid3 Test compliance – Score 100
- Deployable wherever Qt is: cross-platform/cross-version/cross-device
- Interact with Web environment, expose native objects
- NPAPI (Flash) support
- HTML 5 media element
- Various technologies from webkit.org
  - Full page zoom, css effects and animations
  - Squirrelfish JavaScript engine, client side storage

States and Transitions

- A robust state chart implementation
- Harel state-charts and SCXML
- Key benefits:
  - Simplify application semantics
  - Use states to improve maintainability
  - Solidify event-driven programming
  - Write verifiable application logic

Animation API

- Allow users to create dynamic UIs
- Integrates with States and Transitions
- Key features:
  - Animate any QObject property
  - Combine animations in groups
  - Use easing curves
  - Synchronized timer for better CPU utilization

Qt-3D enablers

- Math primitives for matrix multiplication, vectors, quaternions (client-side)
- API for vertex and fragment Shaders, GLSL/ES
- Future research on
  - Stencils
  - Vertex buffers and arrays
  - Texture manipulation
  - Geometry Shaders
- New QGLBuffer class
  - Create and manage vertex, index, pixel pack, and pixel unpack buffer objects
- Improvements in OpenGL Shader API
  - See http://doc.qt.nokia.com/4.7-snapshot/qt-4-7-intro.html for details

Qt Quick

- Qt Declarative
  - A Qt module which can create a canvas-based (QGraphicsView) UI from a declarative description (QML)
- QML
  - The Qt Meta-Object Language allows describing a tree of objects with properties in an extremely intuitive and productive way. It also allows easy integration to native code where any meta-object (QObject) information is made available.
- CSS Influenced with Javascript
  - Creator
    - Full Qt Creator Integration, including New Project Wizards, Syntax Highlighting, Intellisense, Error Lookup, Documentation Integration and QML Inspector
    - Drag-and-drop visual editing of QML files (Preview)
Qt Quick Components

Qt Quick Components

Qt Programming Alternatives

Client Architecture Approaches

Qt Quick Components

Qt Programming Alternatives

Open Source Contribution Model

- Our SCM is now hosted at http://qt.gitorious.org
- All changes and commits public (24hr delay)
- Infrastructure for cloning and merging
- Review infrastructure in the works
- Guidelines
  - Code style and conventions
  - Good documentation
  - Compiles cross platform!
  - Good test coverage
  - No regression in Qt's tests

Open Source Contribution Model

Qt for Nokia platforms

- Qt for millions of Symbian smart phones
  - Symbian, the world's most popular software for smart phones is now supported by Qt.
  - Qt applications are commercially deployable now. Qt will be preinstalled on devices later in 2010.
- Nokia services
  - Nokia's Internet services
  - Qt and Webkit based common web runtime

Qt for Nokia platforms

Recipe for Next Generation Qt

- Basic ingredients
  - C++ and Qt in University Courses
  - Need more Programmers
  - CS courses/Engineering Courses/Medical IT?
  - More Jobs than Programmers
  - Ecosystems

Recipe for Next Generation Qt

Qt in Higher Education

Qt in Higher Education
Qt Books

- Qt: An Introduction to Design Patterns in C++ with Qt 4, Alan and Paul Ezust
- Foundations of Qt Development, Johan Thelin
- C++ GUI Programming with Qt 4, Jasmin Blanchette, Mark Summerfield
- Advanced Qt Programming: Creating Great Software with C++ and Qt 4
- Programming with Qt by Matthias Kalle Dalheimer
- The Book of Qt 4: The Art of Building Qt Applications, Daniel Molkentin