Faster, More Cost-Effective Development Using the Qt Cross-Platform Application & UI Framework

Abstract

As an application developer, you need to get the most from your development project, team and budget. If the applications that you develop must be deployed on multiple target platforms, then you are faced with the special needs of maintaining application code bases to serve those multiple platforms. Real world experience has shown that this can best be done when you develop using a true cross-platform framework, thereby reducing the complexity of development and maintenance to that of a single code base. It also helps if the underlying class libraries of that framework take care of most of the target device-specific work for you, particularly in the areas of user interfce design. The Qt cross-platform application and UI framework was created for exactly this purpose, and it is targeted towards developers who need to create cross-platform applications.

NOKIA

Faster, More Cost-Effective Development Using the Qt Cross-Platform Application & UI Framework

As an application developer, you need to get the most from your development project, team and budget. If the applications that you develop must be deployed on multiple target platforms, then you are faced with the special needs of maintaining application code bases to serve those multiple platforms. Real world experience has shown that this can best be done when you develop using a true cross-platform framework, thereby reducing the complexity of development and maintenance to that of a single code base. It also helps if the underlying class libraries of that framework take care of most of the target device-specific work for you, particularly in the areas of user interfce design. The Qt cross-platform application and UI framework was created for exactly this purpose, and it is targeted towards developers who need to create cross-platform applications.

1. What the Qt Cross-Platform Application & UI Framework Can Do for You

Developing a complex application with an appealing rich user interface can be a daunting, resource-intensive, and time consuming challenge. While business logic is often reusable across different operating systems, the UI and system-level interactions may vary greatly across platform. Code can be difficult and expensive to maintain or extend as new features are added and new platforms are supported. The Qt application framework, consisting of both development tools and class libraries, provides a cross-platform foundation that enables developers to create rich, visually appealing user interfaces that take advantage of the latest platform and internet technologies. These include the latest Webkit improvements. Web-enabling your native applications makes it easy for developers to implement Web content directly into native applications through the integration of the Webkit rendering engine. Qt allows developers to focus on features, functions and usability, while dramatically increasing productivity, portability, and maintainability.

The first version of Qt was released in 1994 as a C++ set of developer tools and C++ class libraries and has evolved to include new functionality and new platforms (with its own IDE and SDK). Available alone or integrated into a cross-platform Qt IDE, the Qt developer tools include a layout designer, documentation assistant, and translation assistant. But the power of Qt can be found in its rich collection of class libraries. Qt includes a whole host of cross-platform class libraries that support a large set of features including GUI design, database interfaces, network communications, multimedia, 2D and 3D graphics, web programming, ECMAScript, and sophisticated XML operations.

2. Getting ROI from the Qt Cross-Platform Application & UI Framework

Return on Investment on software projects is one of the most difficult items to quantify, yet feedback and testimonials from Qt application framework users have indicated savings in a number of categories, including development time, project cost, overall code size, language internationalization, code maintenance, programming personnel needed to complete the project, and the learning curve required to use the framework. A great cross-platform application and UI framework can return value to your investment through high reusability of code, a single code base to simplify maintenance, and quicker time-to-market. Here are but a few examples that demonstrate how some companies achieved a high ROI from their usage of Qt.

2.1 Development Time Savings:

2.1.1 CS Virtual Reality (50% time savings)

CS, a leading provider of virtual reality solutions in military/defense, space imaging and aerospace, needed a cross-platform application framework for developing the user interface for a virtual prototyping system. The Qt framework helped to simplify the process by abstracting low-level infrastructure tasks. “We developed GUIs 1.5 times faster with Qt and Qt Designer compared to basic development techniques,” says Dr. Olivier Balet, Technical and Project Director, CS Virtual Reality Department.

2.1.2 Barco (30% time savings)

Barco, a global technology company, designs and develops visualization products for a variety of selected professional markets: medical imaging, media & entertainment, infrastructure & utilities, traffic & transportation, defense & security, education & training and corporate AV. The Barco digital projector's Communicator touch panel was built using Qt allowing for a strong GUI and one-source code multi-platform deployment. “Without Qt, it would not have been possible to support native applications on all four targeted platforms,” said Barco’s Philippe Martin. “Qt shortened our product development time by 30%.”

2.1.3 Quadstone (20% time savings)

Quadstone Paramics is a leading provider of traffic microsimulation software. The software is comprised of a suite of microscopic traffic simulation tools that are being used in over 40 countries worldwide by 1000’s of customers including commercial consultants, cutting edge transportation researchers and state-funded government agencies. Quadstone realized that its existing development platform, Motif, lacked the helper classes that they needed, and it turned to the Qt application framework. “Qt has allowed us to write more stable, flexible, user-oriented and modular code. This means we can create better applications with more user-driven features in a shorter timeframe than would ever have been possible with Motif," says Ewan Speirs, Head of Paramics Development at Quadstone. “It saved us as much as 20% in developer time.”

2.1.4 The Foundry (15% time savings)

The Foundry is a leading developer of visual effects and image processing technologies that boost productivity and workflow in film and video post production. The Foundry used the Qt framework in its development of Nuke 5.0, a powerful compositing application. “Qt allows us to deliver a consistent and attractive cross-platform interface without devoting a huge amount of engineering resource. As a whole, Qt, eliminates most cross-platform worries, provides a consistent cross-platform user experience and facilitates more creative solutions to user interface and work flow issues,” says Andrew Whitmore, engineering manager at The Foundry. “The Qt framework saved the Nuke project approximately 15% of development time.”

2.2 Project Cost Savings

2.2.1 DFS (1 million Euros cost saved)

DFS Deutsche Flugsicherung GmbH is responsible for air traffic control in Germany. DFS moved from a navigation software system purchased by industry suppliers to its own in-house developed

PC/Linux- based system. The new system, called Phoenix, is used at airport towers for flight scheduling and tracking, and was developed using the Qt application framework. “Developing the system in-house on Qt saved DFS an estimated 1 million Euros, and as many of 104 man months of development time,” says Ralf Heidger, Head of Department SH/T at DFS.

2.3 Code Size Reductions

2.3.1 National Instruments (54% less code)

National Instruments transforms the way engineers and scientists around the world design, prototype, and deploy systems for test, control, and embedded design applications. For its next generation developer’s tool, MATRIXx, National Instruments wanted to continue to offer its customers a choice of platforms: UNIX, Linux, Windows XP or Mac, but had to find a way to make the new capabilities available on each of those operating systems. “When we rewrote the GUI code using Qt, it went from 650,000 lines of source code to fewer than 300,000 lines, making it much easier to manage,” says Bob Pizzi, Principal Architect at National Instruments.

2.4 Ease of Internationalization

2.4.1 Volvo

For its ground breaking ITS4mobility system, a device used by public transportation companies to increase transit efficiency, Volvo Mobility Systems needed an interface that would be easy for drivers to use and would support a Linux-based system. Because the system would be sold in a number of different countries, it needed to support multiple languages as well. Called the Human Machine Interface (HMI), it provided drivers with an efficient, informative view with minimal interactions so they could keep their eyes on the road. “Qt was the only product that fulfilled our needs, including the internationalization,” says Goran Vorosy, Marketing Director, Volvo Mobility Systems. “The Volvo team was able to design HMI in only six months, and we were able to get all of ITS4mobility to market in just 18 months.”

2.5 Better Code Maintainability

2.5.1 European Space Agency (days to hours)

To build digital maps of the plant Mars, the Modeling and Simulation group at the European Space Agency (ESA) was using software that was built on one UNIX version but used on other UNIX versions and on PCs running various other operating systems. At one point, the team found itself maintaining four separate versions of the software for four different operating environments: IRIX, Solaris, Linux and Windows. Every time there was an upgrade or revision, the development team had to rewrite portions of the code. The team rewrote its software completely using Qt, and now sees a very different time frame for engineering change orders (ECOs). “With Qt, the process of issuing a new release has gone from taking a few days to taking a few hours,” says Peter van der Plas, Simulation Engineer, Modeling and Simulation Group, European Space Agency.

2.6 Reduced Personnel Requirements

2.6.1 Digital Film Tools

With three Emmy awards and hundreds of feature films on its resume, Digital Film Tools is widely known for its visual effects. To develop its Dfx Digital Filter Suite, a software application that includes a comprehensive array of effects and gels to simulate the effects of optical filters, Digital Film Tools needed a solution that would work on Linux, Windows and Mac. The cross platform capability was essential to keep the company from having to write the same UI code three times – a process which would have been time consuming and costly. “By developing with Qt, we have saved a huge amount of time,” says Paul Miller, Lead Developer at Digital Film Tools. “We are able to write a single set of source code for all platforms. There’s no way we could have written native code for all operating systems without hiring additional people.”

2.7 Faster Learning Curve

2.7.1 Siemens

The development team at Siemens AG was tasked with designing a new GUI for the software that controls part of its manufacturing assembly line. The team attended a one week course as an introduction to Qt. After only one week, they were proficient enough to implement some of the over 250 dialogs that would be needed. Soon, all of the criteria were being met with developers leveraging Qt’s Plug-In concept and its Qt Styles to get the desired functionality as well as the updated user interface look and feel. The resulting design was so user friendly that manufacturing line operators can use the software without having to be trained.

“It became very clear to us that using Qt would enable us to create a clear and easy design,” says Jose Carlos Arcas, Lead Developer, Siemens AG Automation & Drives EA. “In fact, one person who joined the team was a student with no experience at all. Within 2-1/2 months, the student was implementing MFC controls as Qt widgets. They could easily be integrated within Qt Designer and we could use the Qt style function right out of the box.”

3. CODE LESS. CREATE MORE. DEPLOY EVERYWHERE.

Qt facilitates the development of C++ code that runs across a wide variety of hardware platforms, and that can interact with code written in other languages such as Java™ and a variety of popular scripting languages such as Python and JavaScript™, and other ECMA-based scripting languages. The Qt framework supports platforms such as Microsoft Windows, Mac OS, Linux, Windows CE, and Embedded Linux. It also supports other IDEs, such as Microsoft Visual Studio and Eclipse.

The advantages of such a development environment include the following:

- **Single cross-platform native code base:** The idea of write-once, run everywhere coding has been around for years and it brings with it a compelling economic argument. With applications developed with the Qt framework, you get native code performance rather than scripting of virtual machine performance, while still maintaining a single code base that targets all Windows variants, including Windows CE, variants of Unix, and Mac OS 64-bit support. The Qt application and UI framework allows you to deploy source code from one device target, such as the desktop PC environment, to another, such as embedded

operating systems, without rewriting it. It gives the development manager the flexibility of reassigning development resources regardless of the platform, and it future-proofs the code against future changes in the platform itself. This can be very useful as the operating systems themselves seem to be constantly undergoing revisions. The Qt framework allows developers to focus on building core value into their applications instead of having to spend their time maintaining APIs.

- **Quicker time-to-market:** When developing an application for deployment in multiple hardware environments, the single code base mentioned above clearly accelerates the completion of the project across all of the deployment targets. The time saved can be a multiple corresponding to the number of different deployment environments. Often the UI portion can be 50% or more of the overall design effort. If the application framework does the heavy lifting on the UI design and platform dependencies, this can save developers time.
- **Cheaper maintenance costs:** Again, there is nothing like a single code base to reduce the costs of software maintenance. This is true for both post-deployment software changes that may be categorized as maintenance, and for pre-deployment software changes that usually come in the form of engineering change orders (ECOs). In the real deployment world, the underlying operating environments of all deployment targets can vary from one OS release or security patch to another. How well an application framework can insulate the developer from those underlying changes also contributes to the overall efficiency of code maintenance, and of the quickness of servicing ECOs.
- **Accomplishing more with less:** When faced with budget cutbacks, personnel cutbacks, and time-to-market pressure, the natural reaction of any application development manager is to ask for a reduction in new feature sets required. The overwhelmingly probable response that such a manager receives from the marketing department is “No. This is what we want. Make it happen, and it has to happen in this time frame.” To the extent that the development team’s application development framework can result in increased efficiency, application development managers have more flexibility in managing the expectations of their marketing counterparts.

The Qt cross-platform application and UI framework was created by Trolltech, a company that was acquired in 2008 by Nokia. While this acquisition considerably expands the potential future use of the Qt application framework for billions of future mobile devices, it in no way affects Qt’s commitment to platform independence, a fundamental precept of an open source software platform. The framework still implements the universal ideal of “Write once, run everywhere.” This enables developers to code less, create more, and deploy everywhere.

About Qt:

Qt is a cross-platform application framework. Using Qt, you can develop applications and user interfaces once, and deploy them across many desktop and embedded operating systems without rewriting the source code. Qt Development Frameworks, formerly Trolltech, was acquired by Nokia in June 2008. For more details about Qt please visit http://qt.nokia.com.

About Nokia

Nokia is the world leader in mobility, driving the transformation and growth of the converging Internet and communications industries. We make a wide range of mobile devices with services and software that enable people to experience music, navigation, video, television, imaging, games, business mobility and more. Developing and growing our offering of consumer Internet services, as well as our enterprise solutions and software, is a key area of focus. We also provide equipment, solutions and services for communications networks through Nokia Siemens Networks.

Code less.
Create more.
Deploy everywhere.

NOKIA

Oslo, Norway +47 21 60 48 00 ■ Redwood City, CA, USA +1 650-551-1676 ■ Beijing, China +86 10 6588 8836 ■ qt-info@nokia.com www.qt.nokia.com