

Digia's Lars Knoll Discusses The Qt GUI Framework

[William Wong](#) [Lars Knoll](#)

Mon, 2013-05-20 20:12

Qt, pronounced “cute” or “cue-tee,” is a cross-platform graphical user interface (GUI) framework. It has a long history, starting with Nokia and eventually winding up with Digia. The [Qt Project](#) is the open-source version supported by [Digia](#). Qt is licensed under GPL v3 and LGPL v2. It is also available as a commercial license from Digia.

Applications using Qt that use the open-source licenses need to follow the same license, so their source would need to be made available. The commercial license allows closed-source projects. Digia also provides support options. Qt runs on a wide variety of platforms. It also is the base for the open-source [KDE](#) Linux desktop environment.

Related Articles

- [Embedded: Embedded Graphics Package Supports X11 And Qt](#)
- [Consumer Linux Gets Graphics](#)

Qt is written in C++, but it has its own scripting language as well. Its graphical presentation is its most notable part, but it also provides non-graphical features such as network support, thread management, file handling, and XML parsing and SQL support.

Qt's cross-platform application programming interface (API) allows an application to run on a variety of platforms. Qt also runs on all major operating systems as well as a number of real-time operating systems and embedded operating systems. Digia's CTO and Qt Project chief maintainer Lars Knoll recently explained the framework.

Wong: Qt is used on PC projects like KDE. Can it be used in embedded applications?

Knoll: Absolutely. Qt has supported development for embedded Linux for almost 15 years now. Nowadays, Qt is supported on many embedded operating systems. Apart from Linux, there are versions supporting Windows Embedded, QNX, Integrity, and vxWorks. We are also working on a version to support device creation using Qt on an Android base layer.

Wong: What are the requirements for an embedded Qt environment?

Knoll: That depends a bit on the use case. Qt can run on devices with an ARM 9 processor and 64 Mbytes of RAM. To get a fully smooth user experience, a GPU supporting OpenGL ES 2.0 is a requirement. Qt does, for example, allow you to do smooth user interfaces in 1080p resolution on a Raspberry Pi with an ARM 11 processor and 256 Mbytes of RAM.

Wong: What programming languages and environments does Qt support?

Knoll: Qt applications are usually developed in C++. Our technology for creating smooth user interfaces called Qt Quick is using our own language called QML, which makes it extremely easy to describe the UI in a declarative fashion. In addition to this, we have a full JavaScript integration available that can be used for scripting, prototyping, and smaller tasks. The WebKit integration makes it possible to integrate HTML5 into your application. Bindings to other languages such as Python do also exist.

Wong: How do you design or describe Qt user interfaces?

Knoll: There are two ways to do user interfaces with Qt. The first and traditional one is by using the Qt Widgets module. User interfaces are implemented through C++ in this model, and they are mainly static. Creation of the user interfaces is supported by a WYSIWYG tool called Qt Designer. The other way to create user interfaces with Qt is through the Qt Quick technology. Qt Quick allows the creation of smoothly animated, touch-based user interfaces. It uses a declarative language called QML to make implementation of the user interface very easy and intuitive. All backend/application logic can still be implemented in C++. It allows UX designers to directly participate in the development of the application or device and test their designs on the real hardware.

Wong: What is QtScript?

Knoll: QtScript is a module in Qt that offers a full JavaScript engine that can be integrated into applications. The JavaScript engine can be used for many purposes, such as application scripting.

Wong: What is the relationship between Qt and HTML5?

Knoll: Qt has a module called Qt WebKit. Qt WebKit is using the WebKit project to offer an up to date implementation of the HTML5 standard. The module offers a C++ and Qt Quick API to integrate Web content into any application written with Qt. The API is flexible enough to implement a fully functional Web browser on top of it, but can also be used to very easily embed some HTML code into Qt. Qt WebKit together with many of the other parts of Qt (such as our network stack, the XML support or the JSON support) makes it very easy to integrate Web services into your application.

Source URL: <http://electronicdesign.com/dev-tools/digia-s-lars-knoll-discusses-qt-gui-framework>