

Qt overview



This article needs to be updated: If you found this article useful, please fix the problems below then delete the `{{ArticleNeedsUpdate}}` template from the article to remove this warning.

Reasons: hamishwilliee (13 Nov 2012)

The article reflects (largely) the state of Qt in Qt 4.4 when Nokia first acquired Qt. Qt is now at Qt 4.8 on Symbian and MeeGo Harmattan devices. Qt is now an open source project, where most of the development team been acquired by Digia - it is further being developed (at time of writing to Qt 5). It is licensed via LGPL. The main UI framework is a declarative language and tools called Qt Quick (QML) rather than C++ widgets as stated here.

This article should be updated to reflect the current state of Qt on Symbian/MeeGo Harmattan devices (as that is relevant to Nokia). It should cover the Qt Mobility project and Qt Quick. It then should link to the Qt project and digia for ongoing Qt development.

What is Qt?

Qt [pronounced 'cute'] is a cross-platform application framework. Using Qt, you can develop applications and user interfaces, and deploy them across many desktop and embedded operating systems without needing to rewrite the source code.

The vision is to have [Qt Everywhere](#) and enable developers to create advanced applications with innovative user experiences while getting the applications to market quickly. Qt allows you to code once and deploy across major device and desktop operating systems.

In October 2008 at the Symbian Smartphone Show, Qt announced the porting of Qt to S60 on Symbian OS and made available a technology preview of the S60 port of Qt.

If you are new to Qt, it is recommended that you become familiar with Qt by [evaluating](#) it before using the Qt for Symbian technology preview. For more information, see [Learning Qt](#).

Qt for Symbian Technology

The [Qt for Symbian Developer's Library](#) is the main source of information for Qt for Symbian, including details about how Qt relates to S60 and Symbian OS.

For more information about installing Qt for Symbian, see the [Qt for Symbian installation guide](#) in the Qt for Symbian Developer's Library. The article [Qt for Symbian - Installation packages](#) describes the current technology preview installation kit and provides a list of the earlier technology previews.

Important Notice

The official product and final release is now available as described above [Official Qt for Symbian Release](#)

Learning Qt

The best way to learn Qt is to try it on one of the fully supported platforms, such as Microsoft Windows. To start with Windows, install the package found on the [Qt for Application Development Platform web page](#).

Note: For Windows Vista, there can be a problem with the file mingwm10.dll missing from the C:\windows\system directory. To solve this problem, do a Google search for "mingwm10.dll download" and get a copy from one of the free websites. When the file has been placed in the correct directory, go to Start > All Programs > Qt by Nokia v4.4.3 (OpenSource) > Designer to open the Qt Designer GUI.

For extensive Qt documentation, see [How to learn Qt](#)

Go through all the examples and demos that are included with the SDK.

Qt Creator IDE

[Qt Creator IDE](#) is a tool for creating Qt applications that eliminates the need for operating system or device emulators. The tool has also been implemented efficiently to minimise download size as well as processor and disk space requirements. The Qt

Creator IDE is easy to install and the IDE enables developers to create Qt applications quickly and easily.

Qt tools

Qt is supplied with several command line and graphical tools to simplify and speed up the development process. Each tool is listed here with a link to its documentation.

- [Qt Designer](#): Create forms visually.
- [Qt Assistant](#): Quickly find the help you need.
- [Qt Linguist](#), lupdate, lrelease: Translate applications to reach international markets.
- [qmake](#): Create makefiles from simple platform-independent project (.pro) files.
- [Meta-Object Compiler \(moc\)](#): Generate meta-object information for QObject subclasses.
- [User Interface Compiler \(uic\)](#): Generate C++ code from user interface files.
- [Configuring Qt \(qtconfig\)](#): X11-based Qt configuration tool with online help.
- [Examples and Demos Launcher](#): A launcher for Qt's Examples and Demonstration programs for Platforms.
- [qt3to4](#) - The Qt 3 to 4 Porting Tool: A tool to assist in porting applications from Qt 3 to Qt 4. (Please note: Code ported from Qt 3 to Qt 4 will not be supported on Qt for Symbian).
- [QtDBus XML compiler \(qdbusxml2cpp\)](#): A tool to convert D-Bus interface descriptions to C++ source code.
- [D-Bus Viewer](#): A tool for examining D-Bus objects and messages.

Qt Books

- [C++ GUI Programming with Qt4. Second Edition](#)
- [Other books about Qt](#)
- [Online Independent Book - The Art of Building Qt Applications](#)

Qt terms

1. **Widgets**: UI components (buttons, message boxes, application windows)
2. **Layout Manager**: Automatic positioning and resizing of child widgets.
3. **Signals and Slots**: Inter-object communication.
4. **Events**: System events (mouse clicks, keyboard, ...)
5. **Actions**: e.g., save action used in tool bar and menu.

Qt Architecture

Qt uses native styles to draw UI

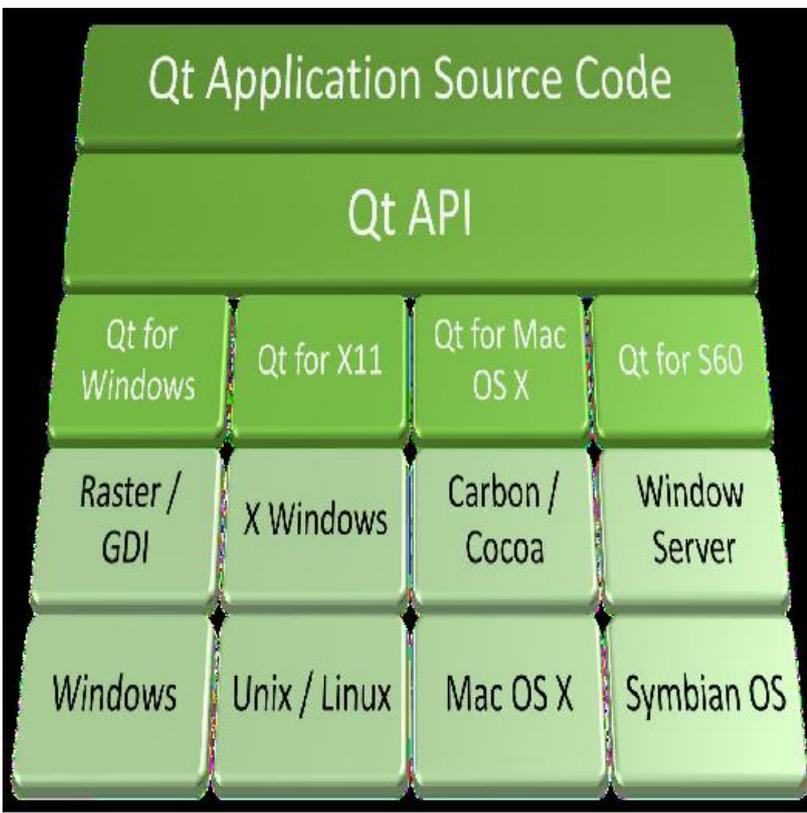
- Widgets emulate exact look & feel
- Can be adapted by the developer
- Built on low level APIs of platform

MFC, Motif, Layered toolkit with thin wrappers. Less performance, less flexibility

- Cross-platform

Single source for multiple platforms Only requires recompilation

Qt Graphical Architecture



Qt Modules

Module	Description
QtCore	Core non-graphical classes used by other modules
QtGui	Graphical user interface (GUI) components
QtNetwork	Classes for network programming
QtOpenGL	OpenGL support classes
QtScript	Classes for evaluating Qt Scripts
QtSql	Classes for database integration using SQL
QtSvg	Classes for displaying contents of SVG files
QtWebKit	Classes for displaying and editing web content
QtXml	Classes for handling XML
QtXmlPatterns	An XQuery& XPathengine for XML and custom data models
Phonon	Multimedia framework classes
Qt3Support	Qt 3 compatibility classes
QtDesigner	Classes for extending Qt Designer
QtUiTools	Create user interfaces from .ui-files at run time
QtHelp	Classes for online help
QtAssistant	Support for online help
QtTest	Tool classes for unit testing

Advantages of Qt

- Target multiple platforms from a single source
- Shorter development time – faster time to market
- Reduced maintenance expense
- Avoid OS-subgroups in development organization
- Enjoy true platform independence

- Target a new platform in weeks, not months
- Rapidly respond to evolving market requirements
- Remain insulated from platform changes
- Qt is actively maintained and developed to support all new mainstream OS variants
- Focus development efforts instead on value-adding innovation
- Qt delivers real, lasting competitive advantage
- Qt increases the productivity of developers by making C++ programming faster, easier and more intuitive
- Qt development tools eliminate common bottlenecks in the development process:
- GUI Design & Layout - Qt Designer
- Translation/Localization - Qt Linguist
- Documentation - Qt Assistant
- Cross-platform build system – qmake
- Qt delivers true platform freedom – targeting a new platform is measured in days or weeks, not months or years
- One source code base means less maintenance time and expense – multiplying results of development efforts
- Full access to complete source code on all platforms enables development teams to adapt and extend Qt to meet their unique needs, expediting the development process.

Nokia to Add LGPL to Qt Licensing Model

[Nokia to Add LGPL to Qt Licensing Model](#)

Qt for Symbian Developer's Library

[Qt for Symbian Developer's Library](#)

DiBo  DISCUSSION BOARD