



# The Qt Automotive Suite

Automotive software needs to be small and efficient, and maximize cost-sensitive hardware; yet consumers want rich user experiences and rapid availability of new apps. Automakers often select technologies that meet these opposing demands but by doing so inadvertently lock themselves into limited hardware options, a narrow pool of developers, and immature frameworks. Those who choose broadly adopted open HMI technologies (like HTML5) end up with systems that are slow to boot, make excessive demands on subsystems, and require endless optimizations to be responsive.

When you add to this a globally distributed development environment that places burdens on reliability, synchronization, and release schedules, the ability to develop reliable and desirable automotive infotainment and instrument clusters becomes a serious challenge.

The Qt Automotive Suite is a development toolkit for rapidly creating high-quality infotainment and instrument cluster software. It leverages the huge embedded developer community of Qt, the raw performance and intricate control of C++, and the powerful design tools of QML.

The Qt Automotive Suite includes many features specifically designed for automotive deployment.

### Features at a glance

- A full-featured infotainment reference for a rapid head start on development
- An application manager for downloading both OEM-certified and safely contained third-party apps
- A vehicle hardware emulator for the software-only simplicity needed in broad development distributions
- Physical-based rendering for realistic-looking controls, dials, and gauges
- A tuned startup procedure for meeting early audio/video requirements
- A live-execution model and debugger for rapid-issue resolution, fast prototyping and short deployment cycles
- A tool that creates custom SDKs for distributed development and supply chain consistency
- Automotive-friendly licensing for lower costs and non-intrusive IP management

## Powerful development environment without proprietary lock-in

The Qt Automotive Suite solves the problem of needing a powerful development environment without proprietary lock-in. At its heart is Qt, an active and open software development framework that is the leading cross-platform framework for C++, supported by hundreds of suppliers and hundreds of thousands of developers across the globe. Qt supports every CPU, chip-set, OS, and mobile environment found in automotive development: from ARM to x86, TI to Nvidia, Linux to QNX, and iOS to Android.

Because it uses processor-efficient C++, the Qt Automotive Suite gives bare-to-the-metal performance on commodity hardware. It runs native code for creating precise timing control and responsive user interactions, processing continuous vehicle bus messages, driving rigidly clocked peripherals, and meeting narrow boot-up windows. High performance graphics are no problem either: QML's declarative environment with hardware-accelerated OpenGL ES supports rapid design iteration and 60 frames-per-second rendering. The fact that the Qt Automotive Suite can address all these issues efficiently on lower-cost hardware keeps the bill-of-material costs from skyrocketing.

The Qt Automotive Suite also addresses the huge challenge in managing a global supply chain. With the ability to create your own custom SDKs, your development team can pre-package all of your system's libraries, tools, and frameworks—Qt, third-party, and in-house—into a single simple installer,



making a consistent development environment for everyone in your world-wide team. It can also be easily updated for new developments, or provided to external parties for app development, leading to smoother integration and fewer build challenges. With the device emulator included in the Qt Automotive Suite, the entire running system can be simulated on a desktop machine, removing the massive cost and management headaches of vehicle simulation harnesses and mock-up bucks for hundreds of developers. Creation of a true open third-party infotainment app environment becomes easy with the Qt Automotive Suite's customized SDK, hardware simulator, and app manager with strict process separation.



The Qt Automotive Suite is a full-featured C++ framework tailored to create fast, small, and beautiful HMIs that can readily evolve as needs change.

## Specifically designed for infotainment systems and instrument clusters

The Qt Automotive Suite is designed for complex automotive graphical applications: both infotainment systems and instrument clusters.

For infotainment systems, it provides a reference IVI design called Neptune that allows developers to leverage existing components, facelift a custom UX, or learn how to build their own architecture. With all the necessary software scaffolding in place, Neptune allows your engineering team to quickly create unique, differentiated experiences.

For instrument clusters, it provides the Qt 3D toolkit and a physically based rendering (PBR) system that allows developers to show gauges and dials using photo-realistic models. It lets designers create

surfaces that simulate the absorption, reflection, and scattering of light with enhanced realism for custom dashboard skins with style and originality.

For both types of development, the Qt Automotive Suite provides a programmer-friendly open development model, giving your team complete transparency on the entire content of the suite as well as the ability to influence its development through feedback and contributions.

# Qt Automotive Suite contents

## Qt

- Cross-platform, cross-CPU, and cross-OS development
- OpenGL scene graphs, GPU-accelerated rendering
- Particle systems, sprites, and shader effects
- Qt Quick for declarative UIs
- Integrated 2D & 3D content
- Audio, video, camera, radio, and image support
- Internationalization, dynamic languages, and RTL layouts
- Chromium-based HTML5 engine, JSON support
- Map viewer, Bluetooth, NFC, GPS, sensors, multimedia, virtual keyboard, etc
- D-Bus support, SQL interface to SQLite
- Core libraries for strings, objects, events, plug-ins, compression, secure hash ...

- IPv4, IPv6, and SSL
- PBR shaders for photo-realistic physics based rendering

## GammaRay

- Qt Quick inspector – live HMI tweaking and performance assessment
- Graphical state machine debugger
- Inspector for Qt 3D scene and frame graphs
- Diagnostics and runtime overrides for QtIVI

## Qt for Device Creation

- Device emulation, simulators
- Qt Safe Renderer for safety critical UI elements

## QML Live

- Instant preview and modification of QML-based on-target HMI

## Qt Creator

- Cross-platform IDE
- Embedded target debugging and profiling
- Graphical design tools supporting C++ and JS
- Static code analysis

## Automotive specific add-ons

- QtIVI - flexible middleware API generation infrastructure
- QtSCXML - compiled and dynamic SCXML state machines
- Qt Application Manager – app container, app lifecycle management, process isolation, example app store
- Qt Wayland Compositor – composition and input handling
- Neptune – reference IVI implementation
- Custom SDK to package developer toolkits for internal or external use



[www.kdab.com](http://www.kdab.com)

## About the KDAB Group

The KDAB Group is the world's leading software consultancy for architecture, development and design of Qt, C++ and OpenGL applications across desktop, embedded and mobile platforms. KDAB is the biggest independent contributor to Qt. Our experts build run-times, mix native and web technologies, and solve hardware stack performance issues and porting problems for hundreds of customers, many among the Fortune 500. KDAB's tools and extensive experience in creating, debugging, profiling and porting complex applications help developers worldwide to deliver successful projects. KDAB's trainers, all full-time developers, provide market leading, hands-on, training for Qt, OpenGL and modern C++ in multiple languages. Founded in 1999, KDAB has offices throughout North America and Europe.