
Introduction to Qt/QML for Embedded Development

Based on Qt 5.9, created on June 12, 2019

The logo for KDAB, featuring a stylized white lightning bolt icon to the left of the text 'KDAB' in white, all on a blue background.

KDAB

The Qt, OpenGL and C++ Experts

Fundamentals of Qt for Embedded Linux

- Fundamentals of Qt for Embedded Linux
 - The Story of Qt
 - "Hello World" with QtQuick
 - Qt for Embedded Linux Overview
 - Supported Platforms
 - EGL
 - Introduction to Wayland
 - Building Qt for Embedded Linux
 - Introducing Qt Creator
 - Developing for Embedded Linux with Qt Creator
 - Practical Tips for Developers

Introduction to QtQuick

- Introduction to Qt Quick
- Composing User Interfaces
 - Image Elements
 - Text Elements
 - Item Transformations
 - Anchor Layout
 - Colors and Gradients
- User Interaction
 - Mouse/Touch Input
 - Gestures Support
 - Keyboard Input

Dynamic User Interfaces

- Components and Dynamic Loading
 - Components
 - The Loader Element
 - Focus Handling Revisited
- Presenting Data
 - Arranging Items
 - Simple Data Models
 - Views

QtQuick and C++

- Objects in Qt
 - Common Features of Qt's Object Model
 - Qt Meta Object System
 - Object Communication Using Signals & Slots
 - Defining Invokable Methods
- Variants and Properties
 - Variants
 - Properties
- Integrating QML with C++
 - Exporting Data and Functions to QML
 - Creating New QML Elements
 - Tips and Tricks

Supplemental Topics

QML Topics

- Layouts
- QtQuick Controls
- QtQuick Controls 2
- WebEngine
- Animations
- States and Transitions
- Declarative State Machines
- Painting and Effects
- JavaScript in QML
- Drag and Drop

Mixed C++/QML Topics

- Internationalization
- Unit Testing
- Plug-ins
- Platform, device, locale or resolution specific files
- Wayland Compositor

Supplemental Topics

C++ Topics

- Advanced Custom Property Types
- The Event System
- Customized Drawing
- Resources
- QPA

Model/View

- Concepts
- Models
- ObjectModel