
Introduction to Qt/QML for Desktop

Based on Qt 5.15, created on March 16, 2023

The logo for KDAB, featuring a stylized white lightning bolt icon to the left of the text 'KDAB' in a white, sans-serif font, all contained within a blue rectangular shape with a diagonal cut at the bottom-left corner.

KDAB

The Qt, OpenGL and C++ Experts

- The Story of Qt
- "Hello World" with QtQuick
- Introducing Qt Creator
- Practical Tips for Developers

No part of this publication may be made available to others than the named licensee which is shown on every page by any means, electronic, mechanical, photocopying, recording or otherwise, or translated into any language, without the prior written permission of the publisher.

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
- QtQuick Input Handlers
- WebEngine
- Animations
- States and Transitions
- Declarative State Machines
- Painting and Effects
- JavaScript in QML
- Drag and Drop
- Component Design

Mixed C++/QML Topics

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

Supplemental Topics

C++ Topics

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

Model/View

- Concepts
- Models
- ObjectModel