Programming with Qt Quick for Embedded Linux

Produced by Nokia, Qt Development Frameworks and Klarälvdalens Datakonsult AB

Material based on Qt 4.7, created on April 23, 2012
Embedded Programming with Qt

- Fundamentals of Qt for Embedded Linux
  - The Story of Qt
  - Qt for Embedded Linux Overview
  - Building Qt for Embedded Linux
  - Introducing Qt Creator
  - Developing for Embedded Linux with Qt Creator
  - Practical Tips for Developers

QtQuick Basics

- Introduction to Qt Quick
  - Meet Qt Quick
  - Concepts
- Composing User Interfaces
  - Nested Elements
  - Graphical Elements
  - Text Elements
  - Anchor Layout
- User Interaction
  - Mouse Input
  - Keyboard Input
- Components

Animations and Data Presentation

- Animations
- Presenting Data
  - Arranging Items
  - Simple Data Models
  - Views
  - Customizing the Views
  - The Path View

Under the hood - The Qt C++ API

- Objects in Qt
  - Common Features of Qt's Object Model
  - Object Communication using Signals & Slots
- Core Classes
  - String Handling
  - Container Classes
  - Variants and Properties
- Customized Drawing
  - Painting on Widgets
  - Color Handling
  - Painting Operations
Integrating QtQuick with Qt

- C++ Integration
  - Declarative Environment
  - Exporting C++ objects to QML
  - Exporting Classes to QML
    - Exporting Non-GUI Classes
    - Exporting GUI Classes
  - Using Custom Types

QtQuick Application Logic

- State Machines in C++
- State Machines in QtQuick
  - States
  - Transitions
- The Loader Element

Model/View programming

- Exporting Data from Qt to QtQuick
  - Model/View Concepts in C++
  - Custom Models in C++
  - Proxy Models in C++
  - Value based models in C++
  - Integrating C++ models with QtQuick
- Specialized QtQuick Models
  - VisualItemModel
  - Accessing models from Java Script
  - XML Models

Additional Embedded Programming Topics

- Qt for Embedded Linux Adaptation
  - QVFb: the Qt Virtual Framebuffer
  - Font handling
  - Customizing Qt Features
  - Performance Tuning
Supplemental Topics

Graphics and Styling
  • Graphics View (p. 546)
  • OpenGL in Qt (p. 627)
  • 2D transformations (p. 635)

Interprocess Communication
  • Networking (p. 645)
  • QtDBus (p. 681)
  • QtDBus Part II (p. 695)
  • QProcess (p. 707)
  • Shared Memory (p. 718)

Interaction
  • Gestures (p. 730)
  • Touch (p. 746)

Multithreading
  • Foundation (p. 758)
  • QtConcurrent (p. 793)
  • Atomic Operations (p. 815)

XML and SQL
  • Using XML from Qt (p. 823)
  • QtXmlPatterns (p. 858)
  • XML Schema (p. 889)
  • SQL Database API (p. 898)

Supplemental Topics

Application Infrastructure
  • Event System (p. 914)
  • Qt Script (p. 932)
  • Internationalization (p. 982)
  • QtHelp (p. 1008)
  • Plug-ins (p. 1029)
  • Resources (p. 1051)
  • File I/O (p. 1065)
  • Qt Smart Pointers (p. 1071)

Qt Components
  • WebKit (p. 1085)
  • mmap (p. 1120)
  • Phonon (p. 1126)
  • QImage (p. 1144)
  • Custom Image Formats (p. 1162)

Supplemental Topics

Qt Quick Topics
  • Modules (p. 1193)
  • Plug-ins (p. 1203)
  • Internationalization with QtQuick (p. 1213)

Development and Testing
  • Qt Licensing (p. 1219)
  • Deployment (p. 1226)
  • Linux Dev. tools (p. 1232)
  • QMake (p. 1260)
  • Portability (p. 1284)
  • Qt Debugging Aids (p. 1292)
  • QTestLib (p. 1300)
  • Testing with Squish (p. 1315)
  • Introduction to Git (p. 1391)
  • Introduction to CMake (p. 1454)

Qt for Embedded Linux
  • QWS - Qt Window System (p. 1475)

Qt Mobility
  • Introduction (p. 1505)
  • Multimedia, Sensors, Feedback (p. 1554)
  • Contacts, Organizer, Document Gallery, Messaging, Versit (p. 1586)

Symbian
  • Qt for Symbian (p. 1602)