+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Updated 05/02/2025

Sign up

Advanced QML training : modern applications

3 days (21 hours)

Presentation

Our advanced QML training course will teach you how to fluid, modern applications while keeping your code concise and easy to maintain.

During this course, you'll learn how to expose C++ types to QML, and how to manipulate properties to help you design an application that meets your users' expectations.

You'll learn about advanced QML concepts such as list types (primitive, QObject, variants...), signal management, call grouping with a C++ proxy and events.

We'll also look at a complex aspect of this technology: multithreading with Qt Concurrent. This course will enable you optimize the performance of your applications using C++ profiling and Qt Quick Test unit tests.

This advanced QML course is based on the latest version of Qt, Qt 6.8.

Objectives

- Exposing C++ types to QML
- Manage Qt connections and events.
- Using advanced QML concepts
- Optimize your application's performance

Target audience

Application developer.

Prerequisites

Significant experience with QT and QML.

Technical requirements

- QT creator installed

QML Advanced training program

Introduction C++/QML

interactions

- Exposing C++ types to QML
  - A reminder of the meta types system
  - Properties
  - Type conversions
  - Methods
- Memory management
  - Default behavior
  - Personalized behavior
- List data
  - Primitive type list: QList<QString>
  - QObject type list: QQmlListProperty<MyType>
  - Variant list : QVariantList
  - List model: QAbstractListModel
  - Special case of the QML ListModel
- Creating and managing signals in QML
  - Declaring a signal in QML
  - The different ways of processing a signal
- Accessing the QML tree from C++
  - Simple case: find an element by objectName or type
  - Advanced case: finding an element by its properties
  - Private headers for QML components

How Qt connections and events work

- Qt events
  - Creation and dispatch
  - Propagation

- Debugging connections
  - List connections (public API)
  - List connections (private API)
  - For unit tests: QSignalSpy
- Example - KeyEvents in QML

Advanced use of templates

- Reminders
- TP - Grouped call history
  - Simple case: grouping in QML
  - Advanced case: C++ proxy model

High-level multithreading (Qt Concurrent)

- Run tasks in parallel
- Control the number of parallel threads

Performance optimization

- C++ profiling: how to use the callgrind tool from the valgrind suite
- QML profiling: connect to a remote program, interpret results
- Load complex components on demand (lazy-loading)
- Avoid over-drawing the entire screen
- Avoid complex bindings and manual positioning
- Limit type conversions
- Reuse components as much as possible
- Avoid writing a property in a loop
- RAM/CPU trade-off: create a pool of items in advance (pooling)

Discover QML unit tests (Qt Quick Test) Q&A session

Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as enrolment is finalized, the learner receives a self-assessment questionnaire enabling us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and objectives.

This questionnaire also enables us to anticipate any connection or internal security problems (intra-company or virtual classroom) that could be problematic for the follow-up and smooth running of the training session. This questionnaire also enables us to anticipate any connection or internal security difficulties within the company (intra-company or virtual classroom) that could be problematic for the follow-up and smooth running of the training session.

Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

Organization

The course alternates theoretical input from the trainer, supported by examples, brainstorming sessions and group work.

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Sanction

A certificate will be issued to each trainee who completes the course.