to TOC

KDAB | Training

Training Guide

www.kdab.com

Contact us: **training@kdab.com**

to TOC

About KDAB:

The original Qt services company and the leading Qt contributor.

Successfully delivering hundreds of C++, OpenGL and Qt projects. Thousands of developers trained since 1999.

Germany office
+49 (0)30 5213 254 70
info-de@kdab.com

French office
+33 (0)4 90 84 08 53
info-fr@kdab.com

UK office
+44 (0)1625 809908
info-uk@kdab.com

Swedish office
+46 (0)563 540090
info@kdab.com

US office
+1.866.777.5322
info-us@kdab.com

Dear Reader,

A word about training with KDAB

I remember the very first Qt training I did 15 years ago, back then the training material was an amazing 300 pages long. I had to prepare for it in less than a week. Since then our training offering has grown quite a bit. In 2017 we added seven brand new trainings including Qt 3D, Usability, Qt Automotive, C++ and Advanced Modern OpenGL, so our slides have gone way beyond the 3500 mark. We still offer a diverse range of others from Advanced C++ to CMake and Debugging as well, of course, as good old Qt, (for which we have more than 1800 slides alone). I'm glad I don't have to prepare all that material in just one week.

Back in the day my only credentials to becoming a trainer were that I had presented at a few conferences and local Linux user groups before. Today you don't become a trainer that easily in KDAB. Now we have a program where new trainers need to join a series of trainings and participate in presentation workshops to learn how to "handle the crowd" as we call it.

Some things have, however, not changed, most noticeable is that, as back then, we do not employ people only doing trainings – our trainers do trainings at most once every 6 weeks, and in between work on real world customer projects. To the students this means that they will get up to date experience and best practices.

Another thing that has not changed is the enthusiasm about both doing trainings and the topics being taught. It's a safe bet that in every single training you will find the trainer eagerly discussing corner cases with students at the end of the training day, whether on-site or at one of our customised facilities in the UK or Germany.

Most of our scheduled trainings these days are 3 days long. Sometimes we offer longer ones on-site for specific project needs and occasionally we offer one day trainings at special events like Qt World Summit. One thing is different at these events, though: class size. In our regular scheduled training we never sell more than at most 12 seats, and in onsite trainings we urge the customer not to put more people into the class than that. The reason is simple: The more people there are in class the less the training will be for your exact needs. Also the more people in class, the less time the trainer will have to help and guide you during labs. In a regular training we split the time approximately 50/50 between presented material and labs. This gives participants maximum opportunity to play, practice and learn at their own rate, so they can immediately put to use what they have learnt in a KDAB training when they get back to work.

We hold scheduled training classes throughout the year in Europe, USA and Asia and look forward to seeing you at one of them!

Jesper K. Pedersen
Training Director

to TOC

About KDAB Trainings

Improve your productivity with Qt, OpenGL and C++

KDAB is the world's largest independent source of Qt knowledge and the market leader of training for Qt, OpenGL, Qt 3D and C++.

With over 10 years' experience and a wealth of well-structured, constantly up-dated training material, we offer hands-on, practical programming training for beginners as well as experienced developers, in Europe, Asia and the USA.

Our experienced instructors are all expert developers working on real-life customer projects – you will get the best-in-class training from Qt, OpenGL, Qt 3D and C++ experts.

We run scheduled 2 day and 3 day courses for a maximum of 12 developers at a time, occasionally offering shorter courses at special events such as Qt World Summit. Customized longer courses can be arranged in-house.

Shell, IBM, Boeing, Schlumberger, Accenture, EADS, Motorola, Ericsson and Siemens are examples of the many companies whose engineers have become more productive as a result of training with KDAB.

Choose a scheduled training if you:

- can wait for and travel to a training course in one of our training facilities
- need training for yourself or only a few engineers in your company
- like to discuss your experience and get inspiration from other engineers that participate.

Choose an on-site training if you:

- want your training at a specific date and location you choose
- are a group of developers with similar needs; e.g. working with the same project or the same company
- want a more tailored training – with a focus on a set of particular topics.

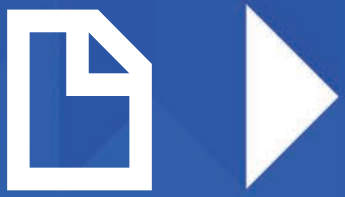
The majority of our scheduled trainings are held in Germany, the United Kingdom, France and the United States in either English, French or German.

Many other language options are available in-house on request.

See our Training Schedule:

Contact us: **training@kdab.com**

Our Training Courses



KDAB | Training

INTRODUCTION

Qt Widgets for the Desktop

Improve your team's Qt programming skills

This introductory Qt course:

- is designed to take development teams in your organization, new to Qt, from the basics to a deep functional understanding of the best Qt practices.
- The Qt training equips developers with the Qt skills and know-how to boost their productivity at work
- offers hands-on Qt training with different kinds of focus, depending on your needs
- is delivered by authorized KDAB trainers with real-life practical experience

Course Contents

- Designing forms and dialogs with Qt Designer
- Layout management
- Event handling
- Model/View programming
- Multithreading with Qt

Find out more:

Why learn Qt?

Qt 5 is a modern, mature and highly featured framework to develop application software that can run across desktop and/or embedded and/or mobile platforms. Qt is often used in a combination with other technologies. KDAB's Qt experts are the maintainers (steer the development) of Qt for Android, Qt for QNX and Qt for Windows CE as well as the areas of Qt 3D (OpenGL), Qt Widgets, Qt WebChannel and parts of Qt Core.

Qt is used by tens of thousands of companies across 70+ industries and in all regions of the world.It is available for development with open source license or with a commercial license. KDAB can assist you in making the choice of licensing as well as evaluating the fit of Qt to your project.

With KDAB as the original Qt Training provider and still the market leader in Qt Training, you will always enjoy a tailored training with up to-date materials, hands-on programming exercises and trainers who are active project engineers.

Target audience:	Developers targeting desktop.
Prerequisite:	Students should have a functional knowledge of C++ but no previous experience with Qt is required. Developers with Qt/Widgets experience will easily benefit from this course which is regularly updated.
Duration:	3 days

INTRODUCTION

Qt/QML

Learn Qt/QML with courses tailored to your team‘s needs

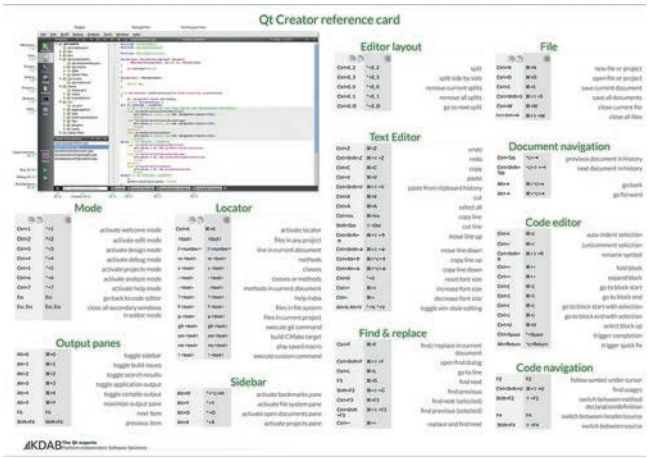
This introductory Qt/QML course:

- is designed to take development teams in your organization, new to Qt or QML, from the basics to a deep functional understanding of best practices.
- The Qt/QML training equips developers with the skills and know-how to boost their productivity at work
- offers hands-on Qt training with different kinds of focus, depending on your needs
- is delivered by authorized KDAB trainers with real-life practical experience

Course Contents

- Connecting a QML UX with C++ business logic
- Complex list views including data provided from C++ models
- Custom objects implemented using Qt Quick scene graph
- Includes time for core topics to establish a working knowledge of QML
- Profiling and best practices

Find out more:



Target audience:	Developers targeting desktop, embedded or mobile platforms.
Prerequisite:	Students should have a functional knowledge of C++ but no previous experience with Qt is required. Developers with Qt/QML experience will easily benefit from this course which is regularly updated.
Duration:	3 days

Contact us: **training@kdab.com**

INTRODUCTION

Qt/QML for Embedded

Learn Qt/QML for Embedded tailored to your team‘s needs

The introductory Qt course “Programming with QML for Embedded”:

- is designed to take development teams in your organization, new to Qt or QML for embedded, from the basics to a deep functional understanding of best practices.
- The Qt/QML for Embedded training equips developers with the skills and know-how to boost their productivity at work
- offers hands-on Qt/QML training with different kinds of focus, depending on your needs
- is delivered by authorized KDAB trainers with real-life practical experience

Course Contents

- Connecting a QML UX with C++ business logic
- Complex list views including data provided from C++ models
- Custom objects implemented using Qt Quick scene graph
- Introduction to the OpenGL pipeline
- Developing for Embedded with Qt Creator
- Performance tuning

Find out more:

Why learn Qt?

Qt 5 is a modern, mature and highly featured framework to develop application software that can run across desktop and/or embedded and/or mobile platforms. Qt is often used in a combination with other technologies. KDAB’s Qt experts are the maintainers (steer the development) of Qt for Android, Qt for QNX and Qt for Windows CE as well as the areas of Qt 3D (OpenGL), Qt Widgets, Qt WebChannel and parts of Qt Core.

Qt is used by tens of thousands of companies across 70+ industries and in all regions of the world.It is available for development with open source license or with a commercial license. KDAB can assist you in making the choice of licensing as well as evaluating the fit of Qt to your project.

With KDAB as the original Qt Training provider and still the market leader in Qt Training, you will always enjoy a tailored training with up to-date materials, hands-on programming exercises and trainers who are active project engineers.

Target audience:	Developers targeting embedded platforms.
Prerequisite:	Students should have a functional knowledge of C++ but no previous experience with Qt is required. Developers with Qt/Widgets or Qt/QML experience will easily benefit from this course which is regularly updated.
Duration:	3 days

Contact us: **training@kdab.com**

ADVANCED

QML

Take your team's understanding of QML to the cutting edge

The advanced QML course:

- is designed to take programmers who already know QML to a deeper level functional understanding
- equips you with the cutting edge QML skills and know-how to boost your productivity at work
- offers hands-on training with different kinds of focus, depending on your needs
- is delivered by authorized KDAB trainers with real-life practical experience

Course Contents

- Advanced integration of QML, JavaScript and C++
- Using OpenGL in custom QML elements
- Analysing and profiling the Qt Quick scene graph
- Understanding and removing bottlenecks that affect a QML UI
- Best practices for achieving optimal performances, even on constrained hardware resources.

Find out more:

Target audience:	Experienced QML developers targeting embedded and/or mobile platforms.
Prerequisite:	Developers enrolled in this Advanced QML course should already have a functional knowledge of QML. Developers with more than one year of Qt/QML experience will also benefit from this course which is regularly updated.
Duration:	3 days

Contact us: **training@kdab.com**

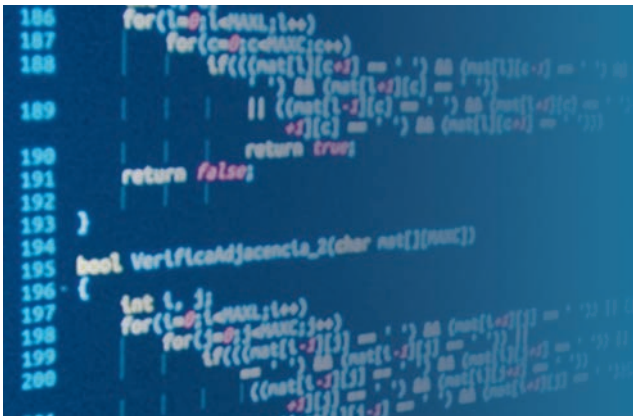
INTRODUCTION

C++

Including best practices for Qt

This training introduces developers to the C++ language, with a focus on how it is used with Qt.The course is targeted at software engineers with prior knowledge in system or application development, for example experience with software written in C.

Rather than going into all the corners of C++, this training focuses on the parts which are needed when developing with Qt. As an example we will not delve a whole lot on exceptions in C++, as they are seldom used with Qt.



Course Contents

- Introduction to the base language features
- Custom data types such as classes
- Life time and memory management of objects
- Code organisation features such as namespaces
- Runtime polymorphism
- Introduction to generic programming with templates
- Usage of templates in the C++ standard library
- Aspects of functional programming such as lambdas
- Best practices for C++ with Qt

Find out more:

Why learn C++?

C++ is widely used and has become a de facto standard, with extensive libraries. This course provides a base to build on for our Advanced C++: What's new in C++11 / C++14 or any of our Introductory Programming with Qt courses.

Target audience:	Developers who want to get started with C++.
Prerequisite:	The course is suitable for developers with no prior C++ knowledge.
Duration:	3 days

ADVANCED

What's new in C++11/C++14?

Learn the relevant library and language changes

In this hands-on C++11/C++14 training for professional C++ developers, you will learn the language changes and the standard library changes introduced in C++11 as well as the changes from C++14.

In class, C++11/C++14 will be demonstrated with the aid of many examples, and you will get the opportunity to use C++11/C++14 right away in our lab projects.

Since not every developer has a C++14-capable compiler yet, features only available in C++14 are clearly marked as such in the material.

The C++11/C++14 training also teaches some advanced C++11/14 features like variadic templates and perfect forwarding, but they are introduced gently, and some of them are optional.

Course Contents

Important language changes, including:

- Lambdas, range based for loops, strongly typed enums
- Uniform initialization, move semantics
- Functional programming, including lambda, bind and function objects
- Template meta programming, including variadic templates
- Multithreading

Find out more:

Why learn about C++11/C++14?

C++11 is a new major version of the C++ standard, released in 2011, and brings many new features to C++ that make the language safer, faster as well as easier and more fun to use. Professional C++ development teams will sooner or later come into contact with C++11/C++14 and there are obvious benefits from introducing its advantages early.

Because of all the numerous changes, the C++ creator, Bjarne Stroustrup, said "C++11 feels like a new language."

Target audience:	Professional C++ developers.
Prerequisite:	Students are required to have basic C++ knowledge, for example knowing about inheritance and virtual functions.
Duration:	3 days

INTRODUCTION

Qt 3D

Integrate 3D content into your applications

In this course you will learn Qt 3D and its use of modern programmable shader-based pipelines which can be applied to both OpenGL and OpenGL ES.

In the Qt 3D training you will learn how to:

- write Qt 3D applications from the ground up
- develop your own materials using Qt 3D and GLSL
- implement your own geometries
- take control of Qt 3D's rendering algorithm using the Frame Graph
- integrate Qt 3D inside a QtQuick application.

In class, our expert trainers will walk you through all necessary topics using a mixture of presentations, examples, and hands-on exercises. By the end of the course you will be armed with all the knowledge you need to become immediately productive with Qt 3D and GLSL, and, in addition, take away over 60 working examples for future reference.

Course Contents

- Overview: Features, Entity Component System
- Drawing: Geometries, Materials and Lights
- User Input: Picking, Keyboard Handling, Logical Devices
- Integration and Helpers: Dynamic Scenes, QtQuick Integration
- Scene Graph: Graphics Pipeline, GLSL, Coordinate Systems, Texturing
- Frame Graph: Viewports, Layers, Selecting Shaders at Runtime, Post-processing Effects
- Advanced Topics: Procedural Texturing, Instanced Rendering

Find out more:

Why Learn Qt 3D?

Integrating 3D content in applications is becoming a clear trend in many fields and is likely to become even more important in the future with the growth of Virtual Reality and Augmented Reality. Qt has always allowed you to integrate with OpenGL fairly easily, but managing the rendering code itself was still a very challenging task, often limited to a few specialized team members.

Qt 3D provides Qt with advanced 3D features with very flexible APIs available both in C++ and QML. It allows you to take full advantage of the enormous parallel processing power of today's GPUs and maximizes performance by scheduling work across all CPU cores.

Target audience:	Developers who want to conveniently and efficiently integrate Qt 3D.
Prerequisite:	This course requires use of Qt 5.7.1 or higher and assumes prior knowledge of Qt Quick. No prior OpenGL knowledge is required.
Duration:	3 days

INTRODUCTION

Modern OpenGL

Skill up on the latest hardware accelerated graphics

This training provides a comprehensive introduction to modern OpenGL development. Beginning with the basic concepts, all the fundamental topics to develop flexible, high performance OpenGL code that run on the desktop and embedded / mobile devices will be covered. Key techniques including lighting, texturing, framebuffer objects and transformations are introduced, in a format suitable for any developer working in C or C++.

Course Contents

- Basic OpenGL window and context management
- The modern OpenGL pipeline
- Rendering geometry with GLSL shaders and vertex buffer objects
- Transformations and coordinate systems, projection and camera handling
- Fundamentals of lighting including the standard Phong model
- Working with frame-buffer-objects (FBOs)

Find out more:

Why learn modern OpenGL?

OpenGL is the industry standard for high performance graphics and computation on desktop, mobile and embedded platforms. Whether the requirement is visualizing large data sets, creating engaging user interfaces or breathtaking real-time visuals, getting the topics covered in this course is essential.

Target audience:	Developers who want to start working with hardware-accelerated graphics, or update their OpenGL knowledge based on current best practices.
Prerequisite:	This course is suitable for developers with no prior graphics experience, but familiarity with C and/or C++ is necessary.
Duration:	3 days

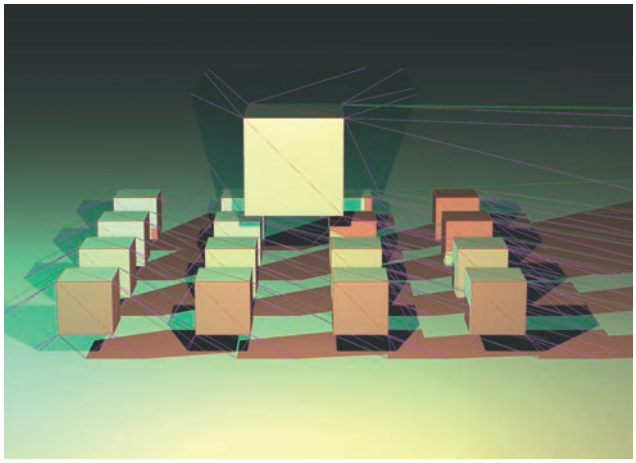
ADVANCED

Modern OpenGL: Rendering and Effects

Learn to bring advanced visuals into your projekt

This training explores the implementation of many different rendering techniques to achieve cutting-edge visuals in OpenGL applications.

Techniques are explored in depth with many examples, analysis of shader code and implementation details.



Course Contents

- Normal mapping and parallax mapping
- Environment mapping
- Procedural texturing
- High dynamic range (HDR) rendering
- Physically based rendering (PBR)
- Deferred rendering
- Shadowing via shadow maps and shadow volumes
- Screen-space ambient occlusion
- Image-processing filters and effects: blur, noise
- Anti-aliasing techniques
- Dealing with transparency
- Picking

Find out more:

Why learn advanced rendering?

Hundreds of papers, textbooks and examples of different rendering techniques exist. In this course we explore some of the most widely used and generally applicable, to give students the confidence to create their own versions and gain a deep understanding of GLSL.

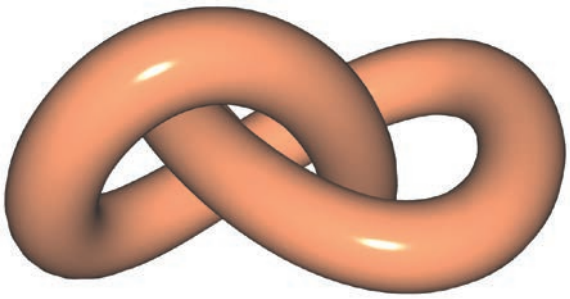
Target audience:	Developers wanting to understand and incorporate realistic advanced visuals into their projects, for realism, visualization or artistic effect.
Prerequisite:	Developers already working with OpenGL, comfortable with the basics of specifying geometry, writing basic shaders and working with image data.
Duration:	3 days

ADVANCED

Modern OpenGL: Pipeline and Performance

Get the maximum out of your hardware

This training explores strategies to increase the performance of new and existing OpenGL code, with multi-pass rendering and use of uniform buffers, shader storage buffers and indirect drawing to reduce driver overhead.



Course Contents

- Cost of state changes, batching and sorting the modern OpenGL pipeline
- Culling, occlusion queries, spatial data-structures
- Debugging and profiling OpenGL via extensions, timers and tools
- Synchronization and timer queries
- Buffering and streaming strategies for large data sets
- Instanced rendering and multi-draw indirect
- Uniform and shader storage buffer objects
- Shader subroutines
- The OpenGL memory model and image load/store
- Geometry and tessellation shaders
- Transform Feedback and Compute shaders

Find out more:

Why learn about advanced pipeline and performance?

Getting the best from available hardware resources, especially on constrained systems, means deeply understanding the costs of different graphics operations, and how to optimise the rendering architecture to meet visual requirements. This course teaches how to increase performance effectively.

Target audience:	Developers wanting to create or improve existing rendering code, using every technique at their disposal to understand and maximise performance, and extract the full potential from their hardware.
Prerequisite:	Developers already working with OpenGL, comfortable with the basics of specifying geometry, writing basic shaders and working with image data.
Duration:	3 days

INTRODUCTION

Qt Automotive Suite

Learn the benefits of the comprehensive package

This training introduces you to the Qt Automotive Suite, Qt's specialized variant for automotive in-vehicle infotainment (IVI) systems.

The course is targeted at Qt developers who want to learn about the additional runtime components and development tools provided by the Qt Automotive Suite, as well as how to best employ these components in their IVI project.

Course Contents

- Device setup and target development with Qt Creator
- Device image creation and customization
- Multi-process application management with Qt ApplicationManager and Wayland
- Qt Virtual Keyboard
- Creating and integrating SCXML-based state machines
- Interfacing with CAN buses using Qt SerialBus
- Developing vehicle data and middleware access APIs using Qt IVI
- Integrating 2D and 3D UI content
- GammaRay runtime introspection

Find out more:

Why learn Qt Automotive Suite?

Qt Automotive Suite extends the regular Qt frameworks and tools into a comprehensive package tailored to the needs of automotive HMI projects. Additional components support you with interfacing with middleware services and with application management. Extensions to QtCreator ease on-target development and powerful runtime inspection tooling gives you new insights at runtime.

In this course you will learn about the additional APIs and components provided by the Qt Automotive Suite, as well as how to effectively employ the development and diagnostic tools included.

Target audience:	Developers needing to get up to speed with the Qt Automotive Suite.
Prerequisite:	Basic Qt or QML knowledge.
Duration:	3 days

INTRODUCTION

Qt for Automotive Development

Design your IVI architecture with Qt

This training covers techniques for designing and implementing large-scale Qt Quick based user interfaces, as they are commonly found in automotive in-vehicle infotainment systems.

This course is targeted at Qt developers faced with having to architect a complex HMI system that will hold up to changing feature requirements and challenging performance goals.

A particular focus is put on:

- scalability
- start-up performance
- maintainability

Course Contents

- Multi-page application architectures
- Single- or multi-process system architectures
- Input, event and focus handling
- Seamless integration of 2D and 3D UI content
- Application infrastructure such as styling/theming, popup management, localization and LTR/RTL vs LHD/RHD layouting
- Startup and runtime performance considerations, profiling and optimization techniques

Find out more:

Why learn about Qt for Automotive Development?

In this course you will learn how to apply Qt effectively when designing the architecture of an IVI system, and what pitfalls to avoid.

In-vehicle infotainment projects tend to be particularly complex. Not only do they have a huge feature scope, but at the same time high requirements for startup and runtime performance. They also need to support a wide range of international markets and a highly variable product line, all while keeping the codebase maintainable for years to come. Qt provides you with the comprehensive toolbox needed to achieve all this.

Target audience:	Developers working on IVI systems using Qt.
Prerequisite:	Basic Qt or QML knowledge.
Duration:	3 days

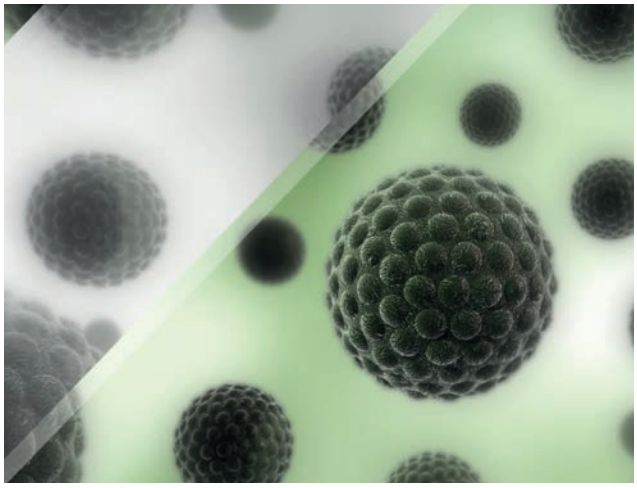
Contact us: **training@kdab.com**

INTRODUCTION

Debugging & Profiling for Qt development

Get up-to-speed with the latest Qt debugging and performance tools

This training gives an introduction to various tools which help developers and testers in finding bugs and performance issues.



The tools presented cover a wide range of problems, from general purpose debugging and CPU profiling to Qt specific high-level analyzers. Often, it is relatively simple to run a tool, but interpreting the results, or even just using some of the more advanced tools, requires deep technical knowledge.

We offer two options for this training: One with a focus on tooling available to developers working with Microsoft Windows on the desktop. The other focuses on what a developer should know for being efficient on both desktop and embedded Linux.

Course Contents

Debugging on Windows:

- general purpose debuggers: Visual Studio, CDB
- memory error detectors: Application Verifier, AddressSanitizers
- thread error detectors: ThreadSanitizer
- tracing: Dependency Walker, Process Monitor, API Monitor
- various Qt-builtin features
- GammaRay to investigate internals of Qt applications
- OpenGL: apitrace

Profiling on Windows:

- CPU: Intel VTune Amplifier XE, Visual Studio, Windows Performance Analyzer
- heap memory: Visual Studio, Windows Performance Analyzer
- QML: QML profiler
- OpenGL: apitrace, NVidia nSight

Debugging on Linux:

- general purpose debuggers: GDB
- memory error detectors: valgrind's memcheck, AddressSanitizers
- thread error detectors: ThreadSanitizer
- tracing: ldd, strace
- various Qt-builtin features
- GammaRay to investigate internals of Qt applications
- OpenGL: apitrace

>> read more

Profiling on Linux:

- CPU: valgrind's callgrind, Linux perf, Intel VTune Amplifier XE
- heap memory: valgrind's massif, heaptrack
- QML: QML profiler
- OpenGL: apitrace

Testing on Windows and Linux:

- Qt TestLib: unit tests and benchmarks
- static code analysis: clang analyzer, Coverity
- code coverage: gcov

Find out more:

Why learn about Debugging and Profiling?

The time spent writing code is often dwarfed by the time required to find bugs and improve performance. This training makes your development workflow more efficient: You will learn what tool to use in which situation, how to set it up and run it on an application. And, finally, you will learn how to analyze and interpret the results obtained from the various tools.

Target audience:	Developers who want to find and fix problems.
Prerequisite:	Knowing the basics of C++, Qt and QML.
Duration:	3 days

INTRODUCTION

CMake

The best thing a build system can do is not get in the way

CMake is the de facto standard build system for C and C++ outside of frameworks that require their own. It has earned this place by supporting the situations and special cases that arise in real projects.

CMake even has strong support for building Qt applications, and it is a good alternative if you hit limitations in qmake.

This course will teach the basics of creating and building projects with CMake. In recent years, CMake has introduced some cleaner and more precise constructs. The course will focus on the new constructs where possible.

Course Contents

- Build system overview; targets and dependencies
- Building executables and libraries
- CMake language and debugging
- Platform-independence
- Using and writing package finders
- Code generators
- Cross compilation

Find out more:

Why learn CMake?

CMake has broad functionality that covers many real world problems. Learning CMake enables you to solve advanced build requirements. This includes cross-platform builds, feature detection based on platform or available libraries, built-time configurable feature switches and custom build steps.

Target audience:	C and C++ Developers.
Prerequisite:	Experience with build systems.
Duration:	2 days

INTRODUCTION

Testing Qt with Squish

Learn how to test your Qt application with Squish

What is the most important thing about a tool for automatic testing? Often the answer is not that it will reliably test your application, but that it will reliably test your application tomorrow even when you have adapted the application slightly!

The aim of this training is that you will learn to use Squish for testing your application, over and over again – much more effortlessly than before. Techniques you will learn include recording a script with Squish and then adapting this into a piece of reusable code that is much less likely to break with the next version of the application it tests.

Testing Qt with Squish – Course Contents

- Recording your first script, verifying the result
- Getting stable scripts by programming
- Optional, Python introduction and/or Java script introduction
- Refactoring your scripts
- Object identification, data driven testing, Qt event handling, file access
- Customized object identification, automatic test runs, special purpose Squish

Find out more:

Why learn about Squish GUI Tester?

Manual testing of user interfaces in applications is often a very complex and error-prone activity. Squish is a proven GUI test automation tool for functional GUI regression tests. Companies in all types of industries, including KDAB, use Squish to reduce the time spent on GUI testing software releases while increasing the quality of their applications.

Target audience:	Testers or programmers to support testers.
Prerequisite:	Limited programming experience. (You know the basics of programming, but you don't necessarily have much experience).
Duration:	3 days

INTRODUCTION

User Centered Development and Usability

It needs more than excellent code

In this course you will learn how to determine user requirements so you can provide solutions for your product's interaction challenges that are both mentally satisfying and visually appealing.

You will also learn how to uncover and estimate the effects of usability deficits and establish benchmarks so as to steer the GUI development for an optimum result.

With this introductory training you will not only understand the theory, but also take away tools and strategies you can directly introduce to your product development team.

In order to tighten your newly gained knowledge, extensive exercises accompany the theory: We will develop an application from the first idea to user testing the final design.

Some of the topics covered

- ISO 9241-210 or "what is a User Centered Development Process?"
- Assess users' needs and make them actionable (e.g. Personas)
- How to communicate user requirements in your team
- Designing for the mind: Translating user needs into interfaces
- Tools and strategies for rapid UI prototyping
- Testing the Usability: Guerilla or The Lab?
- Understand the why: Psychological background
- How to talk with users - getting the feedback right.
- Interweaving UX processes with general development

Find out more:

Why learn about User Centered Development and Usability?

Users ultimately decide about the success of your product - at the latest when they are supposed to use it.
With this training we provide you with strategies for turning your users into allies early in the development process. These strategies will not only lead to better products and higher customer satisfaction, but will also help you to improve the development process itself.

Target audience:	Developers, Development Leads, Product Managers, Managers, Decision Makers wanting to learn how to involve users to boost a product's success.
Prerequisite:	Being part of a product development team.
Duration:	3 days

Our Trainers at KDAB

KDAB | Training

Our Trainers at KDAB

When not conducting training sessions, all KDAB trainers are actively working on client projects, whether developing, consulting or mentoring.

Jim Albamont

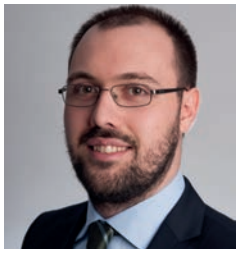
Senior Software Engineer at KDAB, Jim has actively developed with Qt since 2001. He has a background in computer graphics and data visualization, including 6 years as the lead developer of a 3D visualization tool for financial data built with Qt and OpenGL. He has held Qt training classes throughout the US where he is based. Jim holds an MSc in Computer Science. *Language: English*

Nicolas Arnaud-Cormos

Senior Software Engineer and team lead at KDAB, Nicolas has actively developed with Qt since 2001 and is a founding member of Qtfr, the French Qt community site. He has worked on multiple Qt widgets or QML projects, with a particular emphasis on API design and software architecture. He has held Qt trainings for companies such as Michelin, Ford and ST-Ericsson. Nicolas holds an MSc in Computer Science. *Languages: French, English*

Björn Balazs

Senior Experience Engineer at KDAB, Björn has specialized in Usability and User Experience since 1999 and holds a Diploma in Psychology. A longtime member of the KDE visual design group, Björn has worked for many free software projects. For the past ten years, he has regularly given trainings on user-centric development. *Languages: German, English*



Giuseppe D'Angelo

Senior Software Engineer at KDAB, Giuseppe is a long time contributor to Qt, having used Qt and C++ since 2000, and is an Approver in the Qt Project. His contributions in Qt range from containers and regular expressions to GUI, Widgets and OpenGL. A free software passionate and UNIX specialist, before joining KDAB, he organized conferences on opensource around Italy. He holds a BSc in Computer Science. *Languages: Italian, English*

David Faure

Senior Software Engineer who also heads up KDAB's French office, David is a Qt user since its beginning. He has made numerous contributions to Qt, including new classes for QtCore in Qt 5. David is well known in the KDE project for his work on the web browser and especially on KDE Frameworks. He has taught Qt development at numerous conferences and to companies such as Michelin, Schlumberger and Orange. He has become a specialist in multithreading with Qt, as well as performance optimizations. David holds an MSc in Computer Science. *Languages: French, English*

Contact us: **training@kdab.com**

Steffen Hansen

Senior Software Engineer and team lead at KDAB, Steffen has actively developed with Qt since 1997 and was responsible for several of the components of the KDE Desktop such as the display manager kdm and the service manager kded. He has taught numerous Qt classes for companies such as ChevronTexaco, BBC and J.D. Edwards. Steffen holds an MSc in Computer Science. *Languages: Danish, English*

Sean Harmer

Dr. Sean Harmer is a Senior Software Engineer at KDAB where he heads up our UK office and also leads the 3D R&D team. He has been developing with C++ and Qt since 1998 and is Qt 3D Maintainer and lead developer in the Qt Project. Sean has broad experience and a keen interest in scientific visualization and animation in OpenGL and Qt. He holds a PhD in Astrophysics along with a Masters in Mathematics and Astrophysics. *Language: English*

Andreas Hartmetz

Software engineer at KDAB, Andreas Hartmetz has been contributing to KDE base libraries since 2007, when KDE switched to CMake for KDE 4.0. An expert on CMake, Andreas recently took the lead on a project successfully porting to CMake from a sophisticated GNU Make build system. He is the main author of KDAB's CMake training material. Andreas studied Physics in Heidelberg and Berlin. *Languages: German, English*

Tobias Koenig

Senior Software Engineer at KDAB, Tobias has actively developed with Qt since 2001 and has been an active KDE contributor during this time. His contributions have been mainly to the KDE PIM project and the KDE libraries, but also to other open source projects. He holds an MSc in Computer Science. *Languages: German, English*

Kevin Krammer

Senior Software Engineer and team lead at KDAB, Kevin has actively developed with Qt and contributed consistently to KDE since 2000. He is a founding member of the QtCentre website and has mentored at Google's Summer of Code program for 10 years. One of KDAB's most experienced trainers, Kevin keeps our training material up-to-date and has trained engineers from Blackberry, Lockheed Martin, Siemens and Safegate and many others. Kevin holds a BSc in Software and Communications Engineering. *Languages: German, English*

Volker Krause

Senior Software Engineer at KDAB, Volker coordinates KDAB's activities in the automotive space. He is a long-term contributor to KDE and Qt and has been developing with Qt since 2002. Volker's experience spans desktop and embedded platforms, with a special focus on development tooling. In 2010 he started the GammaRay project to address the need for QML debugging tools, and has led the project since then. Volker holds an MSc in Computer Science. *Languages: German, English*

Contact us: **training@kdab.com**

Mike Krus

Dr. Mike Krus is a Senior Software Engineer at KDAB. He has been developing with C++ since 1996 and Qt since 2004. He has a broad range of experience in scientific applications, mainly in civil engineering and oil & gas industries. His range of expertise includes C++, QML and interactive 3D visualization software design on desktop and mobile as well as MacOS development. Mike is the Qt maintainer for the tvOS platform and is very interested in building mobile applications with Qt, mainly on iOS. He has a PhD in Computer Science. *Languages: English, French*

Paul Lemire

Senior Software Engineer at KDAB, Paul is a Qt approver and active contributor to the Qt 3D module where he is one of the main developers. He has been developing Qt based C++ and QML applications since 2010 and has particular interest and expertise in OpenGL and GPU assisted computing. Paul holds a Master's degree in Computer Science. *Languages: French, English*

András Mantia

Senior Software Engineer at KDAB, András has actively developed with Qt since 2002. He is a core developer of KDE's web development environment Quanta Plus and contributor to other parts of KDE. András speaks regularly at free software events about Qt-based products and has held trainings for companies such as Accenture, TietoEnator and Nokia.
Languages: Hungarian, Romanian, English

Thomas McGuire

Thomas is a Senior Software Engineer and team lead at KDAB. He has actively developed with Qt since 2006, when he started contributing to the open source KDE project. Since joining KDAB in 2008, Thomas has focused on QtQuick and QML, especially on embedded Linux. In addition, his interests are performance analysis and optimization, as well as modern C++. He is a contributor and approver for the Qt project. Thomas holds an MSc in Computer Science.
Languages: German, English

Marc Mutz

Marc is a Senior Software Engineer with KDAB and author of the "Effective Qt" series of articles. He originated KDAB's "In-depth Multithreading With Qt" and C++11 courses, and runs "-Wmarc", a blog about Qt, C++ and Boost. A regular contributor to Qt and maintainer of the QtWidgets module, he has actively used the framework for more than a decade, first as a KDE contributor, and then on the job. Marc is a sought-after speaker at conferences on Qt and C++ topics and has an MSc in Theoretical Physics. *Languages: German, English, Norwegian*

Tobias Nätterlund

Senior Software Engineer and team lead at KDAB, Tobias has actively developed with Qt since 2006. He is the main author of the Squish training material and has been involved in numerous projects and workshops helping customers getting started with Squish, as well as integrating Squish with existing test frameworks or user applications. Tobias has held Squish training courses in the United States, Finland, Germany and Sweden. He holds an MSc in Computer Science.
Languages: Swedish, English

Kévin Ottens

Dr. Kévin Ottens is a Senior Software Engineer and team lead at KDAB. He is a main Qt 3D developer in the Qt project as well as a long term contributor to the KDE community. His work has a strong emphasis on API design, frameworks architecture, Test Driven Development and Agile approaches (XP, Scrum, Kanban...). For many years, Kévin has given talks on and taught Qt, KDE, C++11/14 and development best practices at conferences, universities and to companies such as Ubisoft, Siemens or Nokia. He holds a PhD in Computer Science and Artificial Intelligence. *Languages: French, English*

Jesper K. Pedersen

Senior Software Engineer at KDAB, Jesper has actively developed with Qt since 1998 and initiated numerous KDE programs and components. He is the main author of the course material we use for the Qt courses and has taught more than 70 Qt classes for companies such as Boeing, IBM and Veritas. He holds an MSc in Computer Science. *Languages: Danish, English*

Nuno Pinheiro

Senior UX/UI Designer at KDAB, Nuno did the first QML training for designers and actively uses the QML language for fast UX/UI prototyping and UI solutions deployment. His works include general illustrations, UI design, corporate design, interactive mock-ups, animation examples and much more. Known for his contribution to the award winning Oxygen Project where he is the current coordinator, his computer art is used on KDE computer platforms worldwide. Nuno has an MSc in Civil Engineering. *Languages: Portuguese, English.*

James Turner

Senior Software Engineer and team lead at KDAB, James has been developing with Qt since 2002. He contributes to the current maintenance of Mac platform support as well as the development of OpenGL and 3D support in Qt. James has a background in user-interface, graphics and simulation development as well as a long history of development on OS-X and prior versions of Mac OS. He is a lead developer on FlightGear, the open-source flight simulator, and holds a BSc in Computer Science. *Language: English*

Milian Wolff

Senior Software Engineer at KDAB, Milian leads the R&D in tooling and profiling in which he has a special interest. Milian created Massif-Visualizer and heaptrack, both of which are now used regularly to improve the performance of C++ and Qt applications. When not applying his knowledge to improving code base performance for KDAB's customers, Milian maintains QtWebChannel for the Qt Project and is co-maintainer of the KDevelop IDE. In 2015, Milian won KDE's Akademy Award for his work on Clang integration. He has a Masters Degree in Physics. *Languages: German, English*

Testimonials for KDAB Training

Read what our customers say

One of the best run and presented programming training courses I have participated in. The coverage of modern OpenGL techniques and API usage was comprehensive, and the examples and labs demonstrated how theory could be applied in practice.

I would highly recommend this course to anyone working in the scientific computing and data visualisation fields.

Ben Fletcher, Senior Software Engineer, Defence Science & Technology Organisation (DSTO), Department of Defence, Adelaide, Australia

Excellent presentation and instructor whose knowledge was exceptional. I have rarely had an instructor who had such a mastery of his course.

Didier Donner, Paradigm, Nancy, FR

Once again all expectations were more than fulfilled. The training was more than just reading a presentation, it was a good combination of the necessary theory, practical use cases and a nice sense of humor which prevents you from falling asleep.

Geert Depaemelaere, R&D Project engineer, Torex Retail, UK

Most trainees were new to the Qt toolkit. Particularly helpful was the well balanced mixture of teaching theory and doing exercises. The instructor was extremely knowledgeable and addressed all issues raised. I would recommend the course to everyone interested in Qt and/or multi-platform programming.

Paul Norbert, Siemens AG, Erlangen, Germany

The trainer was highly rated for his extensive knowledge, professionalism, teaching skills and patience. His clear explanations and ability to answer questions as well as his hands-on approach and readiness to give a hand to anybody facing a problem was much appreciated.

Claudiu Stefanescu, R&D UK, Orange Labs

Well organized, well-paced, the exercises are very interesting. It was both enjoyable and felt thorough on the topics at hand. Very much recommended!

Cédric Dourneau, SISW, France

The exercises were well designed for all levels of Qt programming experience providing additional challenges for more advanced users. The instructor had a good balance of real programming experience and training expertise. His energy and enthusiasm kept the course interesting and enjoyable.

Jennifer Patton, Software Engineer, Pelco, Fresno, Ca, USA

I found the course to be excellent. It provided a good grounding in the basics of writing programs with Qt, with plenty of useful and realistic examples. The trainer provided many anecdotes and examples of real-world experience of using Qt. I would sincerely recommend the training course to everyone who has an interest in the Qt tool-kit and/or mixed system programming.

Conor O'Neill, Radio Planning, Logica, Bristol, UK

The knowledge and presentation skills of the instructors is exceptional. The whole training was excellent.

Shure Inc., Chicago, IL, USA

to TOC