

Formation Qt 6

Durée

4 jours (28 heures)

Présentation

Qt est LE Framework industriel destiné aux applications multiplateformes pour les ordinateurs de bureau, les systèmes embarqués et les téléphones portables. Les plateformes supportées comprennent Linux, OS X, Windows, VxWorks, QNX, Android, iOS, BlackBerry, Sailfish OS et bien d'autres encore !

Qt n'est pas qu'un langage de programmation. Il s'agit d'un cadre de travail écrit en C++. Un préprocesseur, le MOC (Meta-Object Compiler), est utilisé pour étendre le langage C++ avec des fonctionnalités telles que les signaux et les slots. Avant la compilation, le MOC analyse les fichiers sources écrits en C++ étendu Qt et génère à partir de ceux-ci des sources C++ conformes aux normes. Ainsi vos Apps peuvent être compilés par n'importe quel compilateur C++ conforme aux normes comme Clang, GCC, ICC, MinGW et MSVC.

La réalisation d'interface Homme-Machine est un enjeu majeur dans une application. Il existe de nombreuses solutions, mais beaucoup sont laborieuses à mettre en place, peu ergonomiques et difficiles à maintenir.

Lorsqu'on veut créer une application en C++, la solution la plus courante et la plus efficace est d'utiliser Qt6. La formation vous proposera d'apprendre à maîtriser cet outil surpuissant et de créer des interfaces utilisateurs. Comme toutes nos formations, nous vous présenterons la dernière version en date.

Qt6 propose de créer des applications compatibles sur Windows, Mac OS et Linux en s'appuyant sur les éléments graphiques de tous ces systèmes. Vous allez apprendre à créer vos pages avec des boutons connectés à votre code. Vous découvrirez comment dessiner votre application avec l'éditeur graphique QtCreator. À noter que Qt peut s'utiliser avec Java et Python.

Comme dans toutes nos formations, celle-ci vous présentera la toute dernière version de Qt, à savoir [Qt 6](#).

Objectifs

- Découvrir la puissance du Framework Qt5
- Savoir utiliser le concepteur graphique QtCreator
- Savoir développer une nouvelle application
- Connaître les bases du langage QML
- Connaître les mécanismes avancés du Framework

Public visé

- Développeur d'application

Pré-requis

- Connaissance basique en C++ (notion d'objet et pointeur)

Programme de la formation Qt 6

Jour 1 - Renforcement des bases C++ / Qt / POO

C++ et Qt aujourd'hui

- Évolution de C++ de 2011 à nos jours
- Évolution de Qt de la version 5.0 à nos jours
- Qt Widgets vs. QML
- Un mot sur les licences

Programmation Orientée Objet

- Révision des fondamentaux
- Principaux Design Patterns... et anti-patterns !
- Introduction aux principes SOLID

Les fondements du C++ moderne

- Sémantique de valeur vs sémantique de référence
- Gestion robuste et automatisée des ressources
- Programmation-objet sans héritage

Les concepts essentiels de Qt

- MVC (Modèle-Vue-Contrôleur)
- Les extensions apportées au C++ grâce au MOC
- Le modèle objet spécifique de Qt
- Fonctionnalités avancées des signaux/slots
- Divergences et pièges entre C++ traditionnel et Qt

Jour 2 - Découverte de QML / QtQuick

Présentation de QML / QtQuick

- Syntaxe du langage et principaux concepts
- Différences clés avec les widgets

- Présentation des principaux composants textuels et graphiques
- Ajout du support de TableView pour cacher les lignes et les colonnes

Découverte de QML avec QtCreator

- Créer des composants réutilisables
- Positionner ses éléments
- Fonctionnement du property binding
- TP : Création d'une application en QML avec QtCreator

Interagir avec l'utilisateur

- Gestion de la souris et des écrans multipoints
- Gestion du clavier
- Animation et transitions d'états

Présenter des données complexes

- Repeater et Delegate
- Utilisation d'un modèle
- Utilisation d'un proxy de modèle

Jour 3 - Approfondir Qt / QML

Notions avancées

- Fonctionnement interne du moteur QML
- Utilisation d'un loader
- Création dynamique de composants

Interactions C++ / QML

- Lier son code C++ aux couches de présentation QML
- Écrire une extension QML en C++
- Gérer la durée de vie des objets C++ exposés à QML

Machine à états et Qt

- Principe et utilité
- Fonctionnement de QStateMachine
- Module Qt SCXML

QML dans une vraie application

- Traduire ses interfaces (Internationalisation)
- Appliquer des thèmes / styles personnalisés
- Optimiser et déboguer son code
- Bonnes pratiques et écueils à éviter

Jour 4 - Architecturer et développer un projet complexe

Être efficace avec QMake

- Organiser son projet en modules
- Supporter plusieurs plateformes et modes de compilation
- Intégration d'une bibliothèque externe (Qwt)
- Tests unitaires et intégration continue

Multitâche et programmation asynchrone

- Parallélisme vs concurrence
- Programmation synchrone vs asynchrone avec Qt
- Approche moderne du parallélisme avec Qt

Développement de plug-ins avec Qt

- Le système de plug-ins de Qt
- Contraintes techniques : ABI et compatibilité binaire
- Plug-ins out-of-process : les mécanismes IPC de Qt
- Plug-ins vs. scripting (extensibilité)

Outillage et conclusion

- Outils et autres ressources utiles qui n'ont pas pu être présentés
- Trucs et astuces en vrac
- Questions / réponses sur l'ensemble des sujets abordés
- Conclusion

Nouvelles fonctionnalités

- Qt GUI
 - Ajouter QImage::convertTo new API
 - QPainterPath : supporte les méthodes clear, reserve et capacity avec la même sémantique que QVector::clear() (les allocations sont préservées)
- Qt Network
 - Fenêtres : Prise en charge de la Voie de communication protégée pour la prise SSL
 - Support d'agafage OCSP
- Qt QML
 - Amélioration du support des énumérations déclarées en C++.
 - JavaScript "null" comme valeur de liaison est maintenant optimisé lors de la compilation.
 - QML génère maintenant des tables de fonctions sur des fenêtres 64 bits, ce qui permet de dérouler la pile via des fonctions JITed (QTBUG-50061).
- Qt Quick
 - Ajout du support de TableView pour cacher les lignes et les colonnes
- Contrôles Qt Quick 2
 - Ajouter SplitView
 - Ajouter une propriété de cache à l'icône
- Qt Bluetooth

- Qt 3D
 - Ajout de la prise en charge de l'importation et de l'exportation des poignées de texture OpenGL
 - Ajouté des nœuds de framegraph pour les objets de clôture
 - Ajout d'un prélèvement basé sur la priorité
 - Ajout du support initial de l'importation de scène glTF 2.0
- Compositeur Qt Wayland
- Qt WebEngine
 - Application-local client certificate store
 - Prise en charge des certificats clients par QML
 - Affichage PDF via l'extension Chromium interne
 - API des notifications Web
 - Intercepteurs de requêtes d'url spécifiques aux threads et aux pages.
- Qt WebSockets
- Qt Emplacement
 - Ajout de la prise en charge des GeoPolygons avec trous, reflétés dans MapPolygons et MapPolygonObjects.
 - Introduction de l'interopérabilité avec GeoJson avec la fonctionnalité d'import/export
- Test Qt
- Qt Multimédia
 - Lecture sans faille dans QML VideoOutput en utilisant la propriété flushMode
 - Ajout du support de GStreamer pour Windows/macOS
 - Ajout des en-têtes HTTP et des rôles audio pour Android
 - Ajout de QT_MULTIMEDIA_PREFERRED_PLUGINS pour spécifier les plugins préférés
 - Permis de construire le WMF avec DirectShow ou désactivé par l'option de configuration -no-wmf ou -no-directshow
 - Introduction de QT_PA_CHANNEL_MAP pour QAudioOutput et PulseAudio
 - Sondes vidéo/audio dans DirectShow
 - QMedia La classe QMediaResource est maintenant obsolète
- Wayland
 - Nouvelle intégration Shell pour fullscreen-shell-unstable-v1.
- Qt Lottie (TP)
- Qt pour WebAssembly
- Qt pour l'automatisation
- Qt KNX
 - API client sécurisée ajoutée
- Qt OPC UA
 - API C++ en TP
 - Ajout de l'API QML (TP)
 - Ajout de l'API sécurisée client C++ (TP)
 - UaCpp et Open62542 sont identiques en termes de fonctionnalités
- Qt CoAP (TP)
- Modules obsolètes
 - Script Qt
 - Contrôles rapides Qt 1
 - Qt XmlModel XmlListModel

Modules annexes (optionnels) - 2 jours supplémentaires

- Base de données (SQLite)
- Sérialisation avec QStream
- Mise en réseau avec QNetwork
- Responsive design avec QML

Sociétés concernées

Cette formation s'adresse aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiple permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.