



Porting user applications to Qt 6

Kai Köhne
Director R&D, Developer Experience
The Qt Company

20.6.2021

Why Port to Qt 6?

Qt 6




New Platforms

Next-Generation Graphics

Evolving Qt Quick

Enhancing our C++ APIs

Agenda

-  Qt 6 Modules
-  Build System Changes
-  C++ API Changes
-  Tooling
-  Summary

Module migration / updates

6.0 (Dec 2020)

Available

- Qt Core
- Qt Gui
- Qt Network
- Qt QML
- Qt Quick
- Qt Quick Controls 2
- Qt Quick 3D
- Qt SQL
- Qt Widgets
- Qt Concurrent
- Qt Dbus
- Qt Help
- Qt OpenGL
- Qt Printing Support
- Qt Quick Timeline
- Qt SVG
- Qt UI Tools
- Qt XML
- Qt 5 Compat
- Qt Shader Tools
- Qt Image Formats
- Qt 3D
- Qt Network Authorization
- Qt CoAP (M2M AddOn)
- Qt MQTT (M2M AddOn)
- Qt OpcUA (M2M AddOn)

6.1 (May 21)

Available

- Active Qt
- Qt ScXML
- Qt Virtual Keyboard
- Qt Charts
- Qt Data Visualization
- Qt Device Utilities

6.2(Sept 21)

Available

- Qt Bluetooth
- Qt Quick Dialogs
- Qt Positioning
- Qt Multimedia
- Qt Remote Objects
- Qt Sensors
- Qt SerialBus
- Qt SerialPort
- Qt WebEngine
- Qt WebSockets
- Qt WebChannel
- Qt WebView
- Qt Android Extras
- Qt Lottie Animation
- Qt NFC

6.3+ (2022+)

Available

- Qt PDF
- Qt HttpServer
- Qt Speech

Qt Declarative / QML

- "Strict QML"



Qt



2

Build System Changes

Qt 5 API

```
find_package(Qt5 COMPONENTS Widgets REQUIRED)
```

```
qt5_add_big_resources(SOURCES resources.qrc)
```

```
add_executable(example ${SOURCES})
```

```
target_link_libraries(example PRIVATE Qt5::Widgets)
```

Qt 6 Port (Simple)

```
find_package(Qt6 COMPONENTS Widgets REQUIRED)
```

```
qt6_add_big_resources(SOURCES resources.qrc)
```

```
add_executable(example ${SOURCES})
```

```
target_link_libraries(example PRIVATE Qt6::Widgets)
```

Supporting Qt 5.15 and Qt 6

```
find_package(QT NAMES Qt6 Qt5 COMPONENTS Widgets REQUIRED)  
find_package(Qt${QT_VERSION_MAJOR} COMPONENTS Widgets REQUIRED)  
  
qt_add_big_resources(SOURCES resources.qrc)  
  
add_executable(example ${SOURCES})  
  
target_link_libraries(example PRIVATE Qt::Widgets)
```


Supporting Qt 5 and Qt 6

```
find_package(QT NAMES Qt6 Qt5 COMPONENTS Widgets REQUIRED)
find_package(Qt${QT_VERSION_MAJOR} COMPONENTS Widgets REQUIRED)

if(${QT_VERSION_MAJOR} GREATER_EQUAL 6)
    qt6_add_big_resources(SOURCES resources.qrc)
else()
    qt5_add_big_resources(SOURCES resources.qrc)
endif()

add_executable(example ${SOURCES})

target_link_libraries(example PRIVATE Qt${QT_VERSION_MAJOR}::Widgets)
```

CMAKE

Qt 6 requires C++17

```
set(CMAKE_CXX_STANDARD 17)  
set(CMAKE_CXX_STANDARD_REQUIRED ON)
```

CMAKE

Qt 5 Core Compatibility APIs

```
find_package(Qt6 COMPONENTS Core5Compat REQUIRED)
```

```
target_link_libraries(example PRIVATE Qt6::Core5Compat)
```

Summary

- › Use `qt_` command variants, if possible (supported since Qt 5.15)
- › Use `${QT_VERSION_MAJOR}` to avoid hardcoding Qt versions in targets
- › Require C++17 (if you don't already)
- › Use Qt5Compat Qt module for porting work

Qt

++

3

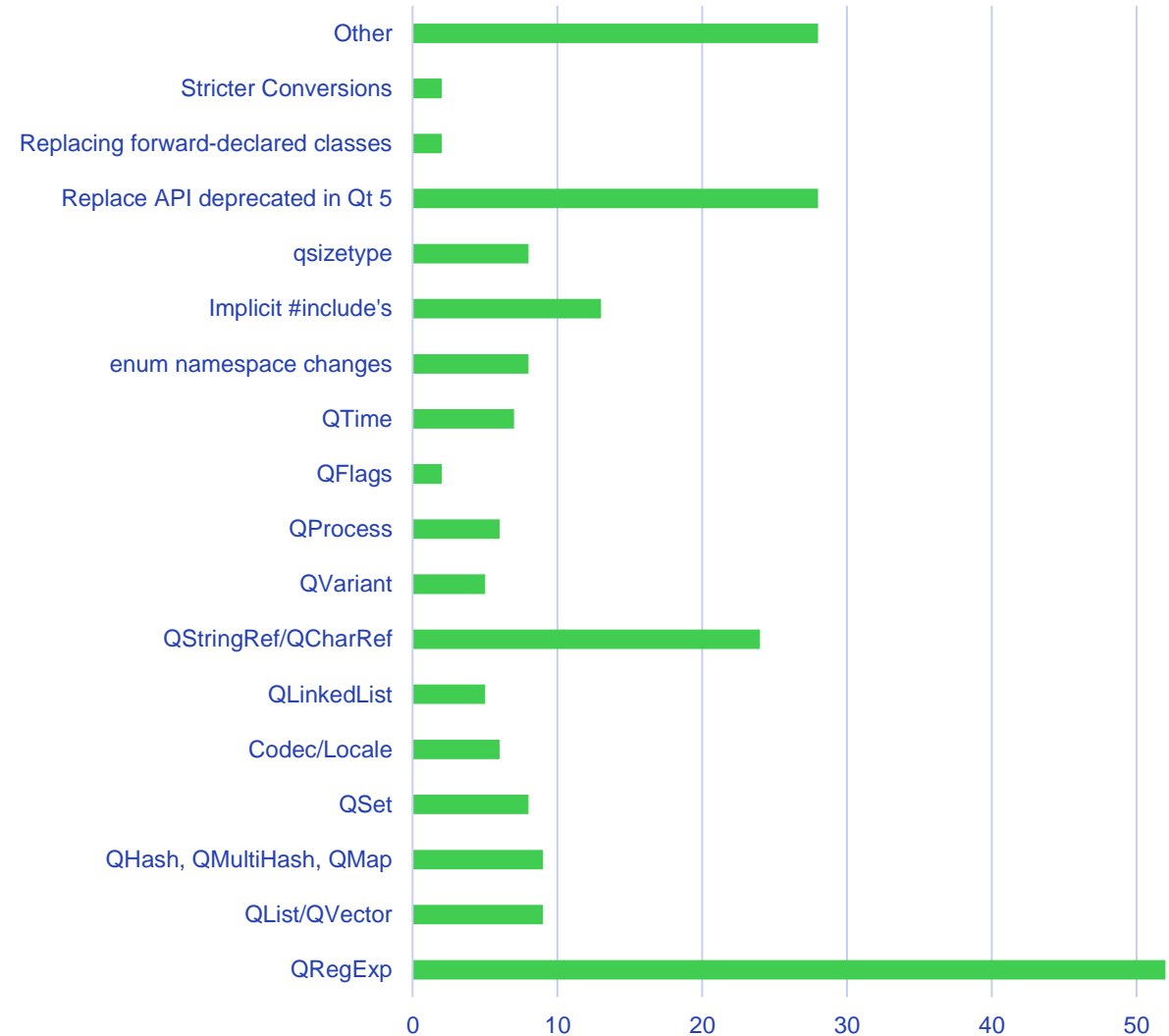
C++ API Changes

Case Study – Qt Creator

> Qt Creator

- > QWidget based application
- > Porting started in spring 2020 (following early Qt 6 work)
- > Work tracked in <https://bugreports.qt.io/browse/QTCREATORBUG-24098>
- > Looked at ~180 patches

Qt 6 Related Patches in Qt Creator



Fix API deprecated in Qt 5.15

```
add_compile_definitions(QT_DISABLE_DEPRECATED_BEFORE=0x050F00)
```

QRegExp

- › QRegExp is EOL
 - › Class itself moved to **Core5Compat**
 - › Methods taking QRegExp were removed
- › Replacement: QRegularExpression
 - › Available since Qt 5.0
 - › Typical patch: <https://codereview.qt-project.org/c/qt-creator/qt-creator/+/303620/3/src/libs/extensionsystem/pluginmanager.cpp>
- › Gotchas
 - › Changes in regular expression syntax
 - › Wildcard handling (slashes)
 - › Return value of QRegExp::matchedLength(), QRegularExpressionMatch::capturedLength() ...

QStringRef

- › QStringRef is EOL
 - › Class itself moved to Qt5Compat
 - › API in Qt exposing or accepting QStringRef were removed
- › Replacement: QStringView
 - › Available since Qt 5.10 (but only feature complete since Qt 5.15)
- › Gotcha
 - › Changing the string might invalidate QStringView

QHash, QMultiHash, QSet

- › QHash, QMultiHash, QSet
 - › QHash and QMultiHash are not distinct types
 - › Implementation changed from node-based approach to a two-stage lookup table
- › Gotcha: Reference (in)stability
 - › References to elements may become invalid if container changes
 - › For instance: Changing container while iterating over it
 - › Fixes
 - › Rethink your design 😊
 - › Switch to `std::unordered_set`, `std::unordered_map`


QList, QVector, QListedList

- › QList memory model is simplified
 - › Always continuous memory
 - › QVector is now a typedef of QList
- › Gotcha: Reference (in)stability
 - › References to elements may become invalid if container changes
 - › Use `std::list`
- › QListedList
 - › Use `std::list`

Summary

- › *Porting is mostly straightforward!*
- › Interim step: Port to Qt 5.15, fix deprecations
- › Required applications depend a lot on application and coding style
 - › Check out <https://doc.qt.io/qt-6/modulechanges.html>



KDE/clazy 


Qt-oriented static code analyzer based on the Clang framework

PR 54
Contributors

0
Issues

438
Stars

65
Forks



The screenshot shows the GitHub repository page for KDE/clazy. It features the repository name, a description, and statistics for pull requests, issues, stars, and forks. The KDE logo is also present.

Use crazy!

- › Latest crazy supports
 - › qt6-deprecated-api-fixes
 - › qt6-fwd-fixes
 - › qt6-header-fixes
 - › qt6-qhash-signature

- › Feedback welcome 😊

Qt

5

Questions