Qt 5 for OS/2 roadmap

v2018-03-08

Overview of Qt 5 components

Sources of Qt version 5 have a modular structure (using git submodules) where each submodule encapsulates a broad set of sources implementing some fundamental function. There are around 50 submodules in Qt 5. Each submodule consists of one or more logical Qt API modules representing different aspects of functionality. Essential Qt API modules are listed here: https://doc-snapshots.qt.io/qt5-5.9/qtmodules.html. Most of these logical modules are part of the **qtbase** git submodule which provides the core Qt functionality which is used by all other Qt submodules. This submodule is what needs to be ported first.

The **qtbase** submodule consists of the following key API modules:

Module	**Source path**	**Description**
Qt Core	src/corelib	Core non-graphical classes used by other modules.
Qt GUI	src/gui	Base classes for graphical user interface (GUI) components. Includes OpenGL.
Qt Network	src/network	Classes to make network programming easier and more portable.
Qt SQL	src/sql	Classes for database integration using SQL.
Qt Test	src/testlib	Classes for unit testing Qt applications and libraries.
Qt Widgets	src/widgets	Classes to extend Qt GUI with C++ widgets.

Note that although Qt supports rendering 2D GUI windows with OpenGL, this part is about to be skipped in the OS/2 port as much as possible due to the lack of hardware OpenGL support on OS/2. In 2D, OpenGL is used speed up composition of multiple UI layers in to the top-level application window. Even though there are OS/2 libraries that emulate OpenGL in software, it makes no sense to use them as it would only slow things down due to an additional layer between Qt and the underlying OS/2 graphical subsystem.

There are also other submodules that contain essential components. The key ones are **qtdeclarative**, **qtquickcontrols**, **qtquickcontrols2** and **qtmultimedia**.

The **qtdeclarative** submodule encapsulates the core of Qt scripting functionality (including the JavaScript engine) and consists of the following key API modules:

Module	Source path	Description
Qt QML	src/qml	Classes for QML and JavaScript languages.
Qt Quick	src/quick	A declarative framework for building highly dynamic applications with custom user interfaces.

This submodule is needed for Qt applications written in Qt's own declarative language with JavaScript support called QML rather than in C++. This is a secondary target of the OS/2 port.

The **qtquickcontrols** and **qtquickcontrols2** submodules both provide two different versions of the **Qt Quick Controls**, **Qt Quick Dialogs** and **Qt Quick Layout** modules used to write GUI applications in QML. The second version of Qt Quick Controls is a redesign of the first version which was mostly oriented to desktop platforms. The second version's primary focus is mobile devices. Both submodules are also secondary targets of the OS/2 port.

The **qtmultimedia** submodule contains **Qt Multimedia** and **Qt Multimedia Widgets** modules that provide classes for audio, video, radio and camera functionality and widget-based classes for implementing multimedia functionality, respectively. These are also secondary targets of the OS/2 port — mostly, because of weak hardware audio/video support on OS/2 which requires additional work (outside of the scope of this project) to make them useful.

There are also two other special and important submodules: **qtwebkit** and **qtwebengine**. Both allow to embed web content in Qt applications. The **qtwebkit** submodule uses the Apple's WebKit engine for that purpose. A special WebKit version is built on top of Qt that does rendering of web content using Qt graphical primitives. This submodule is deprecated since Qt version 5.6. The newer **qtwebengine** submodule uses the Chromium browser project's engine which performs complete rendering of web content on its own and the result is then composed into a Qt window using either OpenGL or a built-in software renderer (available in latest versions starting from Qt 5.9).

Our goal is to stay current so the **qtwebengine** submodule is to be ported first to OS/2. Given that web access is a very important part of any modern application, this submodule is also a primary task of the OS/2 port. This task basically includes porting the Chromium browser to OS/2 which is a project of its own with many different components such as Skia (a 2D graphics library Chromium uses to render pages).

The **qtwebkit** submodule is still considered to be ported one day as it should not require too much work because of heavy use of Qt graphical components (which we have to port anyway). Moreover, we already succeeded in doing that in Qt 4 for OS/2. Compared to **qtwebengine,** this submodule will give us a much more lightweight (yet less functional and standards-compliant) web browser inside a Qt application which is useful in cases like the built-in help viewer, the installation wizard and so on. This is not as important as the full-featured browser component, so it's a secondary target of the OS/2 port.

The rest of submodules contain optional add-on components. Some of them are per-platform and not applicable to OS/2 at all. The remaining ones are third-level targets for the OS/2 platform due to their low importance.

OS/2 port primary target roadmap

The table below lists the project milestones and estimated time frames for each milestone. Note that the order of subtasks with is not necessarily chronological as some components intermix so that they should be worked on in parallel.

Task	*Time*
1. Port qtbase submodule This consists of the following parts:	12 months
1.1. Port basic tools This includes qmake, moc, uic, rcc and some other console tools.	1 month
1.2. Port Qt Core Basic classes like file and console I/O, various utility functions.	2 months
1.3. Port GUI (excluding OpenGL) Classes for basic 2D drawing in desktop windows and GUI application management.	3 months
1.4. Port Qt Network Only IPv4 parts will be ported to OS/2 due to the lack of IPv6 support.	1 month
1.5. Port Qt SQL This includes porting a number of SQL drivers (mysql, sqlite).	2 weeks
1.6. Port Qt Test	2 weeks
1.7. Port Qt Widgets This module contains the most code for the **qtbase** submodule and requires most effort. Not all classes of this module are to be ported to OS/2 due to limitations of the platform.	4 months
2. Port qtwebengine submodule This includes porting not only Qt wrappers for Chromium but also individual Chromium components such as Skia and much more.	6 months

The total time to reach the primary target is **18 months**.