Meet Qt 6.6, and more!

Dr. Cristián Maureira-Fredes
@cmaureir
Slides available

qtinfo.dev/gnuhealthcon23
After 3 online years
it's good to be here in person
My motivation to be here
Let's talk about the Qt framework
- Cross platform framework for UI interfaces, and applications.
- Started in 1991
- With many, many modules.
- More info at qt.io
The 6.6 Release

- RC: 26.09.2023
- Final Release: 10.10.2023
Where were we? 🤔
GNUHealthCon 2022 (1/2)
Motivation
Motivation

- Python popularity 🚀
GNUHealthCon 2022 (2/2)

Motivation

- Python popularity 🚀
- Relevance of C++ 👏
GNUHealthCon 2022 (2/2)

Motivation
- Python popularity 🚀
- Relevance of C++ 👤

Features
- PyPy compatibility
- New wheel structure
- Embedded support
- Tooling
  - `pyside6-qml` (app preview)
  - `pyside6-deploy` (using nuitka)
  - `pyside6-project` (new projects)
  - `pyside6-qtpy2cpp` (converter)
A few highlights of the 6.6 release

doc-snapshots.qt.io/qt6-6.6/whatsnew66.html
Qt Graphs (TP)

QtGraphs

- New module, replacing OpenGL based DataVisualization
Qt Graphs (TP)

QtGraphs

- **New module**, replacing OpenGL based DataVisualization
- **Use cases**
  - Vis of large quantities of dynamic data
  - Creation of depth maps
New module, replacing OpenGL based DataVisualization

Use cases
- Vis of large quantities of dynamic data
- Creation of depth maps

Value
- Using QML APIs from Quick3D, using bare metal interface and acceleration through RHI
- Customizable charts with themes, items and labels
- Possibility to embed Qt Quick 3D effects in graphs
Qt Graphs (TP)  

**QtGraphs**

- **New module**, replacing OpenGL based DataVisualization
- **Use cases**
  - Vis of large quantities of dynamic data
  - Creation of depth maps
- **Value**
  - Using QML APIs from Quick3D, using bare metal interface and acceleration through RHI
  - Customizable charts with themes, items and labels
  - Possibility to embed Qt Quick 3D effects in graphs
- **How to find**
  - [https://doc-snapshots.qt.io/qt6-6.6/qtgraphs-index.html](https://doc-snapshots.qt.io/qt6-6.6/qtgraphs-index.html)
  - [https://doc-snapshots.qt.io/qt6-6.6/graphs-examples.html](https://doc-snapshots.qt.io/qt6-6.6/graphs-examples.html)
Qt Quick responsive layouts (TP)

QtQuick.Layouts

- Intuitive way to create responsive layouts

Responsive Layout Example
Qt Quick responsive layouts (TP)

QtQuick.Layouts

- Intuitive way to create responsive layouts
- Use cases
  - Re-usable GUI implementation for different window sizes
  - Dynamic handling portrait and landscape devices

Responsive Layout Example
Intuitive way to create responsive layouts

Use cases
- Re-usable GUI implementation for different window sizes
- Dynamic handling portrait and landscape devices

Value
- Foster best practices and brand consistency across devices
- Bring QML closer to web design standards and "mobile first" design approach
Qt Quick responsive layouts (TP)  

QtQuick.Layouts

- Intuitive way to create responsive layouts
- Use cases
  - Re-usable GUI implementation for different window sizes
  - Dynamic handling portrait and landscape devices
- Value
  - Foster best practices and brand consistency across devices
  - Bring QML closer to web design standards and "mobile first" design approach
- How to find

The Qt Company | Cristián Maureira-Fredes
Window capturing
QtMultimedia

- Capture videos of individual desktop applications windows

Screen capture example
Window capturing

QtMultimedia

- Capture videos of individual desktop applications windows
- Use cases
  - Record windows for specific video
  - Streaming capabilities

Screen capture example
Window capturing

QtMultimedia

- Capture videos of individual desktop applications windows
- Use cases
  - Record windows for specific video
  - Streaming capabilities
- Value
  - Better control on what to capture from a screen
    - Increase safety on sharing content
    - Reduce effort on post-recording editing
    - Optimize resources usage

Screen capture example
Window capturing

QtMultimedia

- Capture videos of individual desktop applications windows

- Use cases
  - Record windows for specific video
  - Streaming capabilities

- Value
  - Better control on what to capture from a screen
    - Increase safety on sharing content
    - Reduce effort on post-recording editing
    - Optimize resources usage

- How to find

The Qt Company | Cristián Maureira-Fredes
OpenType fonts support

- Advanced access to font shaping features

```cpp
Text {
  anchors.centerIn: parent
  text: "One divided by two is 1/2"
}
```

```cpp
Text {
  anchors.centerIn: parent
  text: "One divided by two is 1/2"
  font.features: { "frac": 1 }
}
```
OpenType fonts support

**QtFonts**

- Advanced access to font shaping features
- Use cases
  - Support OpenType features directly from Qt Framework

```cpp
Text {
  anchors.centerIn: parent
  text: "One divided by two is 1/2"
}

Text {
  anchors.centerIn: parent
  text: "One divided by two is 1/2"
  font.features: { "frac": 1 }
}
```
OpenType fonts support

QtFonts

- Advanced access to font shaping features
- Use cases
  - Support OpenType features directly from Qt Framework
- Value
  - Enable users to deliver better font experiences

```json
Text {
    anchors.centerIn: parent
    text: "One divided by two is 1/2"
}

Text {
    anchors.centerIn: parent
    text: "One divided by two is 1/2"
    font.features: { "frac": 1 }
}
```
OpenType fonts support

QtFonts

- Advanced access to font shaping features
- Use cases
  - Support OpenType features directly from Qt Framework
- Value
  - Enable users to deliver better font experiences
- How to find
  - https://doc-snapshots.qt.io/qt6-6.6/qml-qtquick-text.html#font.feature-prop

```qml
Text {
    anchors.centerIn: parent
    text: "One divided by two is 1/2"
}

Text {
    anchors.centerIn: parent
    text: "One divided by two is 1/2"
    font.features: { "frac": 1 }
}
```
New Text to Speech

QTextToSpeech

- Small improvements on text to speech module

```cpp
def _speech->synthesize(const QAudioFormat & format, const QByteArray & bytes)
    if (output->open(QIODevice::WriteOnly))
        output->open(QIODevice::WriteOnly);
        output->write(bytes);
        output->close();
    }
```
New Text to Speech

QTextToSpeech

- Small improvements on text to speech module
- Use cases
  - Text to audio synthesis
  - Audio queueing
  - Querying for available voices and engine capabilities
New Text to Speech
QTextToSpeech

- Small improvements on text to speech module
- Use cases
  - Text to audio synthetization
  - Audio queueing
  - Querying for available voices and engine capabilities
- Value
  - New abilities to produce read-aloud fluent interfaces
New Text to Speech

QTextToSpeech

- Small improvements on text to speech module
- **Use cases**
  - Text to audio synthetization
  - Audio queueing
  - Querying for available voices and engine capabilities
- **Value**
  - New abilities to produce read-aloud fluent interfaces
- **How to find**
  - [https://doc-snapshots.qt.io/qt6-6.6/qtexttospeech.html](https://doc-snapshots.qt.io/qt6-6.6/qtexttospeech.html)
Qt Creator Improvements

Developer Experience

- Github Copilot support
- Integrated terminal
Qt Creator Improvements
Developer Experience

- Github Copilot support
- Integrated terminal
- Use cases
  - Get AI help for creating code, tests or document the code
  - Run command line tasks using the terminal
Qt Creator Improvements

Developer Experience

- Github Copilot support
- Integrated terminal
- Use cases
  - Get AI help for creating code, tests or document the code
  - Run command line tasks using the terminal
- Value
  - More productive SW development with Qt Creator
Qt Creator Improvements

Developer Experience

- Github Copilot support
- Integrated terminal
- Use cases
  - Get AI help for creating code, tests or document the code
  - Run command line tasks using the terminal
- Value
  - More productive SW development with Qt Creator
- How to find
  - Qt Creator 11+

The Qt Company | Cristián Maureira-Fredes
Permission APIs are in QML now

Many modules

- New QML API for handling of permissions
Permission APIs are in QML now

Many modules

- New QML API for handling of permissions
- Use cases
  - An app for recording audio, using BLE, etc
Permission APIs are in QML now

Many modules

- New QML API for handling of permissions
- Use cases
  - An app for recording audio, using BLE, etc
- Value
  - Better UX: users get a permission request on an action
  - Permission requests just on start can be eliminated
Permission APIs are in QML now

Many modules

- New QML API for handling of permissions
- Use cases
  - An app for recording audio, using BLE, etc
- Value
  - Better UX: users get a permission request on an action
  - Permission requests just on start can be eliminated
- How to find
Discard compiled QML code from the app package

- QML code fully compiled by the Qt Quick Compiler can be discarded from the app package

```cpp
qt_standard_project_setup(REQUIRES 6.5) # <- this is needed!
qt_add_executable(appuntitled1
    main.cpp
)

set_source_files_properties(
    Main.qml
    PROPERTIES
    QT_DISCARD_FILE_CONTENTS true # <- here, you discard it
)

qt_add_qml_module(appuntitled1
    URI untitled1
    VERSION 1.0
    QML_FILES Main.qml
)
```
Discard compiled QML code from the app package

- QML code fully compiled by the Qt Quick Compiler can be discarded from the app package
- Use cases
  - Avoid access to code with sensitive IP
  - Prevent manipulations of apps

```
qt_standard_project_setup(REQUIRES 6.5) # <- this is needed!
qt_add_executable(appuntitled1
  main.cpp
)

set_source_files_properties(
  Main.qml
  PROPERTIES
  QT_DISCARD_FILE_CONTENTS true # <- here, you discard it
)

qt_add_qml_module(appuntitled1
  URI untitled1
  VERSION 1.0
  QML_FILES Main.qml
)`
Discard compiled QML code from the app package

QML

- QML code fully compiled by the Qt Quick Compiler can be discarded from the app package
- Use cases
  - Avoid access to code with sensitive IP
  - Prevent manipulations of apps
- Value
  - Better IP protection and application integrity for Qt Quick based applications

```csharp
qt_standard_project_setup(REQUIRES 6.5) # <- this is needed!
qt_add_executable(appuntitled1
  main.cpp
)

set_source_files_properties(
  Main.qml
  PROPERTIES
  QT_DISCARD_FILE_CONTENTS true # <- here, you discard it
)
qt_add_qml_module(appuntitled1
  URI untitled1
  VERSION 1.0
  QML_FILES Main.qml
)```
Discard compiled QML code from the app package

QML

- QML code fully compiled by the Qt Quick Compiler can be discarded from the app package
- Use cases
  - Avoid access to code with sensitive IP
  - Prevent manipulations of apps
- Value
  - Better IP protection and application integrity for Qt Quick based applications
- How to find
  - [https://doc-snapshots.qt.io/qt6-6.6/resources.html#discarding-the-file-contents](https://doc-snapshots.qt.io/qt6-6.6/resources.html#discarding-the-file-contents)
QML Lint plugins (TP)

- Create custom static analysis for QML code
Create custom static analysis for QML code

Use cases

- Write additional rules for QML Lint
- Post additional messages for selected languages statements in the code
QML Lint plugins (TP)

- Create custom static analysis for QML code
- Use cases
  - Write additional rules for QML Lint
  - Post additional messages for selected languages statements in the code
- Value
  - make QML code compliant with your company's coding guidelines

The Qt Company | Cristián Maureira-Fredes
Create custom static analysis for QML code

- Use cases
  - Write additional rules for QML Lint
  - Post additional messages for selected languages statements in the code

- Value
  - make QML code compliant with your company's coding guidelines

- How to find
  - Check "QML Static Analysis Tutorial"
QML Language Server is getting solid

- Can be used in Visual Studio Code, and QtC 10+
The Qt Company | Cristián Maureira-Fredes

QML Language Server is getting solid

QML

- Can be used in Visual Studio Code, and QtC 10+
- Use cases
  - Far more for developers than just a spellchecker in Word
QML Language Server is getting solid

QML

- Can be used in Visual Studio Code, and QtC 10+
- Use cases
  - Far more for developers than just a spellchecker in Word
- Value
  - 100% coverage of the actual QML language model
  - Interworking with other tools, like Qt Quick Compiler
QML Language Server is getting solid

QML

- Can be used in Visual Studio Code, and QtC 10+
- Use cases
  - Far more for developers than just a spellchecker in Word
- Value
  - 100% coverage of the actual QML language model
  - Interworking with other tools, like Qt Quick Compiler
- How to find
  - Post: https://qt.io/blog/whats-new-in-qml-language-server-qmlls-shipped-with-qt-6.6
  - qmlls --help

The Qt Company | Cristián Maureira-Fredes
Qt gRPC: overview of the ongoing works (TP) - (1/2)

QrGrpc and Qt Protobuf

- Qt 6.6
  - Added support for channel and call options
  - Integrated QML-types support in code generation
  - New QML API: in discussion and under reviews
  - Support for "oneof"
  - qmake support
Qt gRPC: overview of the ongoing works (TP) - (2/2)

QrGrpc and Qt Protobuf

- Plans for Qt 6.7
  - Improve HTTP/2 support in Qt, then Qt gRPC
  - Reduce dependencies to "gRPC" itself
  - Support more well-known types
  - Finalize QML API
  - QProtobuf conformance testing
  - gRPC interoperability testing
  - Review and extended/improve examples if needed
Qt gRPC: overview of the ongoing works (TP) - (2/2)

**QrGrpc and Qt Protobuf**

- **Plans for Qt 6.7**
  - Improve HTTP/2 support in Qt, then Qt gRPC
  - Reduce dependencies to “gRPC” itself
  - Support more well-known types
  - Finalize QML API
  - QProtobuf conformance testing
  - gRPC interoperability testing
  - Review and extended/improve examples if needed

- **Wish to fully release in 6.7. It is going to be popular modules, we better do them well**

- **How to find**
  - Since Qt 6.5, Qt gRPC and Qt Protobuf are in docs and Qt Bug Reports
  - They are listed as modules and components
API access to more privacy and security settings

QtWebEngine

- Enable/disable HTML5 Fetch API
- Read effective top-level domain
- Disable reading from canvas to prevent fingerprinting, and more
API access to more privacy and security settings

**QtWebEngine**

- Enable/disable HTML5 Fetch API
- Read effective top-level domain
- Disable reading from canvas to prevent fingerprinting, and more
- Use cases
  - Have a more control on how Chromium runtime behaves

---

The Qt Company | Cristián Maureira-Fredes
API access to more privacy and security settings

QtWebEngine

- Enable/disable HTML5 Fetch API
- Read effective top-level domain
- Disable reading from canvas to prevent fingerprinting, and more

Use cases
- Have a more control on how Chromium runtime behaves

Value
- Additional API covering more use cases
API access to more privacy and security settings

QtWebEngine

- Enable/disable HTML5 Fetch API
- Read effective top-level domain
- Disable reading from canvas to prevent fingerprinting, and more

Use cases
- Have a more control on how Chromium runtime behaves

Value
- Additional API covering more use cases

How to find
- https://doc-snapshots.qt.io/qt6-6.6/whatsnew66.html#qt-webengine-module
Dynamic Linking and Qt Loading Enhancements

WebAssembly

- Limited support for dynamic linking in Qt for WebAssembly
- New implementation for QtLoader.js
Dynamic Linking and Qt Loading Enhancements

### WebAssembly

- Limited support for dynamic linking in Qt for WebAssembly
- New implementation for `QtLoader.js`
- Use cases
  - WebAssembly app developer

---

WebAssembly is a binary instruction format for a stack-based virtual machine. It is designed as a portable compilation target for programming languages, enabling deployment on the web for client and server applications.

**Generated File** | **Brief Description**
--- | ---
app.html | HTML container
qtloader.js | JavaScript API for loading Qt apps
app.js | JS API for loading Qt apps
app.wasm | App binary
Dynamic Linking and Qt Loading Enhancements

WebAssembly

- Limited support for dynamic linking in Qt for WebAssembly
- New implementation for QtLoader.js
- Use cases
  - WebAssembly app developer
- Value
  - QtLoader used for landing the wasm app benefits of the new implementation (easier debugging/maintenance)
Dynamic Linking and Qt Loading Enhancements

WebAssembly

- Limited support for dynamic linking in Qt for WebAssembly
- New implementation for QtLoader.js
- Use cases
  - WebAssembly app developer
- Value
  - QtLoader used for landing the wasm app benefits of the new implementation (easier debugging/maintenance)
- How to find
Qt for Android
Platforms

- Improve developer experience with Qt on Android
  - Added FileProvider support when using QDesktopServices::openUrl()
  - Use AndroidX by default
  - More Qt examples are tailored for Android
  - Android 13 is max supported version
  - Support SDK level 33 to match Play Store req.
Qt for Android

Platforms

- Improve developer experience with Qt on Android
  - Added FileProvider support when using QDesktopServices: :openUrl()
  - Use AndroidX by default
  - More Qt examples are tailored for Android
  - Android 13 is max supported version
  - Support SDK level 33 to match Play Store req.

- Use cases
  - Mobile app dev with cross-platforms in focus

The Qt Company | Cristián Maureira-Fredes
Qt for Android

Platforms

- Improve developer experience with Qt on Android
  - Added FileProvider support when using QDesktopServices::openUrl()
  - Use AndroidX by default
  - More Qt examples are tailored for Android
  - Android 13 is max supported version
  - Support SDK level 33 to match Play Store req.

- Use cases
  - Mobile app dev with cross-platforms in focus

- Value
  - Better API coverage with more use cases supported
Qt for Android

Platforms

- Improve developer experience with Qt on Android
  - Added FileProvider support when using QDesktopServices::openUrl()
  - Use AndroidX by default
  - More Qt examples are tailored for Android
  - Android 13 is max supported version
  - Support SDK level 33 to match Play Store req.

- Use cases
  - Mobile app dev with cross-platforms in focus

- Value
  - Better API coverage with more use cases supported

- How to find

The Qt Company | Cristián Maureira-Fredes
Qt for Python
Asynchronous compatibility
```python
from PySide6.QtCore import (Qt, QEvent, QObject, Signal, Slot)
from PySide6.QtWidgets import (QApplication, QLabel, QMainWindow, QPushButton, QVBoxLayout, QWidget)

import outcome
import signal
import sys
import traceback
import trio

class MainWindow(QMainWindow):
    start_signal = Signal()
    def __init__(self):
        super().__init__()
        widget = QWidget()
        self.setCentralWidget(widget)
        layout = QVBoxLayout(widget)
        self.text = QLabel("The answer is 42.")
        layout.addWidget(self.text, alignment=Qt.AlignmentFlag.AlignCenter)
        async_trigger = QPushButton(text="What is the question?")
        async_trigger.clicked.connect(self.async_start)
        layout.addWidget(async_trigger, alignment=Qt.AlignmentFlag.AlignCenter)

    @Slot()
    def async_start(self):
        self.start_signal.emit()
    async def set_text(self):
        await trio.sleep(1)
        self.text.setText("What do you get if you multiply six by nine?")

if __name__ == "__main__":
    app = QApplication(sys.argv)
    main_window = QMainWindow()
    async_helper = AsyncHelper(main_window, main_window.set_text)

    main_window.show()

    signal.signal(signal.SIGINT, signal.SIG_DFL)
    app.exec()
```
Embedded wheels
Packages for aarch64

- Targeting Raspberry Pi devices mainly
- To be available on PyPI
Android
(venv) shyamnath@shyamnath-ThinkPad-X1-Carbon-Gen-9:~/qt_for_python/pyside-e-setup/examples/multimedia/camera$ pside6-android-deploy --wheel-pside=/home/shyamnath/qt_for_python/pyside-setup/dist/PySide6-6.6.0a1-6.6.0-cp37-abi3-android_x86_64.whl --wheel-shiboken=/home/shyamnath/qt_for_python/pyside-setup/dist/shiboken6-6.6.0a1-6.6.0-cp37-abi3-android_x86_64.whl --name=camera --keep-deployment-files
For license holders

Installation

Maintenance Tool
For license holders

Installation

Qt Maintenance Tool

qtpip
Towards a Qt flavored Python IDE

QtCreator

- Interpreter selection
- Virtual env creation
- Automatic PySide installation
- More to come
Silent features and fixes

PySide and Shiboken
How can Qt contribute to Libre and Open Health initiatives?
Let's be in touch!
Communication channels

- Telegram [https://t.me/qtforpython](https://t.me/qtforpython)
- IRC/Matrix #qt-pyside on Libera.chat
- Mailing list [lists.qt-project.org/mailman/listinfo/pyside](https://lists.qt-project.org/mailman/listinfo/pyside)

More platforms at [wiki.qt.io/Qt_for_Python#Community](https://wiki.qt.io/Qt_for_Python#Community)
Don't forget the documentation
Qt for Python

Qt for Python offers the official Python bindings for Qt, which enables you to use Python to write your Qt applications. The project has two main components:

- **PySide6**, so that you can use Qt6 APIs in your Python applications, and
- **Shiboken**, a binding generator tool, which can be used to expose C++ projects to Python, and a Python module with some utility functions.

**Porting from PySide2 to PySide6** provides information on porting existing PySide2 applications.

This project is available under the LGPLv3/GPLv3 and the Qt commercial license.

Quick Start

You can obtain the latest stable version by running `pip install pyside6`. If you want to build it yourself, check the getting started guide.

To learn how to use it, check out write your first application, and to learn what is installed with the pyside6, check the package content, structure, and tools page.

Documentation

- Write your first Qt application.
- Install and build from source.
- PySide API reference.
Meet Qt 6.6, and more!

Dr. Cristián Maureira-Fredes
@cmaureir

The Qt Company | Cristián Maureira-Fredes