

FFmpeg in Qt Multimedia

What is Qt Multimedia, and how are we using FFmpeg to give a good media experience in Qt while solving challenges with platform maintenance?



What is Qt Multimedia?

- APIs for viewing and creating media content in Widgets and Quick apps

Features

- Media playback and display
- Capture and recording
- Sound effects
- Device discovery
- Spatial audio



Core APIs

High level media playback – *QMediaPlayer*

- Audio and video playback from file or network
- Supports subtitles, seeking, and speed control
- Decoding of common audio and video formats

Media capture/recording – *QMediaCaptureSession*

- Camera, microphone, screen, or individual windows
- Encoding to common audio and video formats

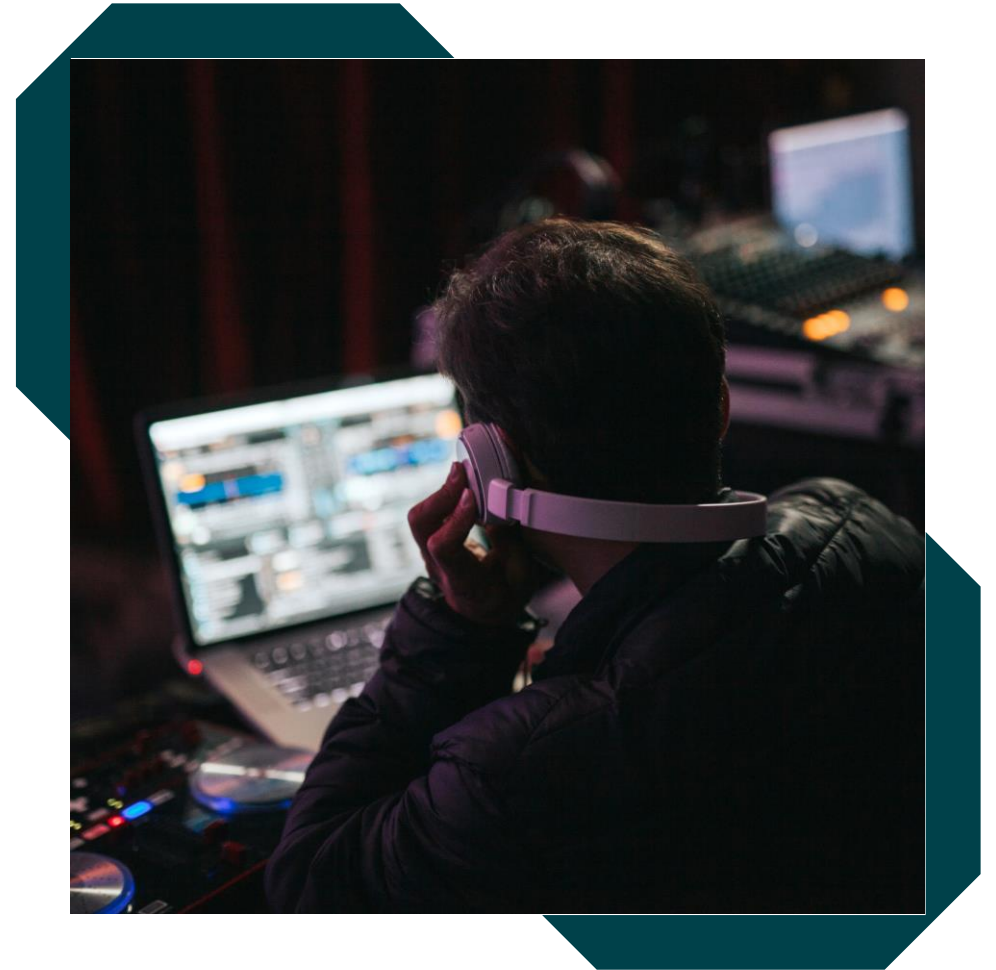
Application sound effects – *QSoundEffect*

- Available also without any media backends

Device discovery – *QMediaDevices*

- Monitors audio input/output devices and camera inputs

+ *Many more (QAudioSink, QAudioSource...)*



“Native” Media backends

Qt Multimedia is implemented using platform plugins on top of platform APIs.

- Platform plugins contribute to ~70% of the Qt Multimedia codebase.
- Maintenance and development requires expertise and hardware for all supported platforms.
- Hard to ensure uniform cross-platform experience, especially format support and signal emission.
- Cyber-security concerns must be addressed independently on each platform

OS	API
Linux	GStreamer
Windows	Windows Media Foundation
macOS/iOS	AVFoundation
Android	MediaCodec
WebAssembly	WebAudio/WebVideo

FFmpeg media backend



Qt's FFmpeg media backend is built on top of FFmpeg's Cross-platform libraries.

- Introduced on Qt 6.4, the default from 6.5 except Boot2Qt and WebAssembly. On iOS from 6.8.
- Facilitates uniform cross-platform behavior and increases Qt Multimedia's format support.
- Intends to replace the other platform backends to reduce platform-specific code, ease maintenance, and give a better developer experience.
- New Qt Multimedia features are only implemented on FFmpeg backend (Currently screen/window capturing and programmatic IO)

Considerations when switching to FFmpeg media backend

The FFmpeg media backend gives

- Better format support
- Uniform cross-platform behavior
- Cyber-security enhancements
- New feature support in upcoming Qt releases
- Simplified maintenance

Considerations

- FFmpeg's LGPL 2.1 license
 - Can complicate commercial use – vendors must assess if the license is acceptable for their products.
- Parts of FFmpeg is GPL licensed or not free
 - These are excluded from Qt Online Installer – Feature support is by default lower than FFmpeg's command line tools
 - If FFmpeg is built from sources, be careful when enabling GPL-licensed or non-free options.
- Bigger binary footprint
 - Qt Online Installer bundles ~17 MB of FFmpeg binaries
 - Can be reduced if building FFmpeg from source
- Qt involvement in FFmpeg community

Linking to FFmpeg libraries

Over the last year, we upgraded the FFmpeg media backend to support dynamic and static linkage with FFmpeg libraries.

Pre-built binaries from the Qt Online Installer link to FFmpeg as shared libraries.

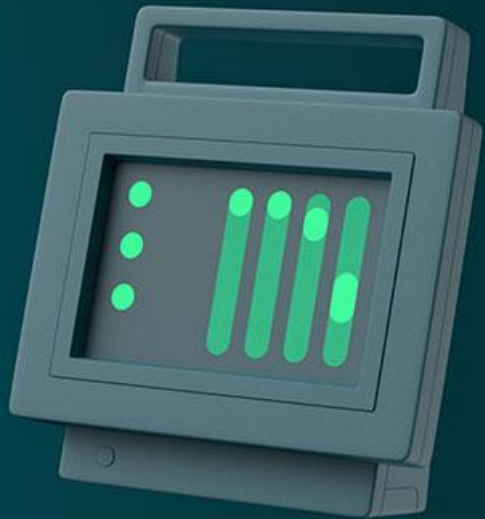
It is still possible to link the FFmpeg media backend statically to FFmpeg when building from sources.

Dynamic linkage

- FFmpeg libraries can easily be replaced/upgraded within patch versions.
- Makes it easier to adhere to LGPL 2.1's right-to-tinker requirements.
- Easiest to deal with on commercial products.
- Application must deploy FFmpeg shared libraries alongside or rely on FFmpeg provided by the platform.

Static linkage

- Easier deployment.
- Replacing FFmpeg requires re-linking Qt's FFmpeg media backend, which can be unsuitable for commercial products.



Note on licensing

- Video compression standards, such as the H.264 media compression standard, may be covered by patents and can incur royalty fees.
- This can apply to any implementation, also if the implementation is provided as an operating system service, through a third party library, or through any of Qt Multimedia's backends. The Qt licenses do not cover such fees.

See

- <https://doc.qt.io/qt-6/qtmultimedia-index.html>
- <https://ffmpeg.org/legal.html>

Roadmap

Desktop and mobile

- New features are only implemented with FFmpeg media backend.
- Only critical bugs are fixed on native backends.
- Our goal is to deprecate native backends over time.
- Android MediaCodec backend is deprecated with Qt 6.8 and will be removed with the next major Qt release.

Embedded Linux

- The FFmpeg is supported on embedded Linux; performance depends on the chip manufacturer.
- We continue to maintain and develop the GStreamer backend for embedded Linux.
- Focus will be on embedded use cases, and we may increase the minimum required version beyond what is part of mainstream Linux distros.

How can Qt Multimedia become even better?

- Discussion

- Questions