The Quest for Realtime Desktop Audio

Nirbheek Chauhan, Centricular Ltd.

(*nirbheek* on GitLab / GitHub / Matrix / Twitter)

https://nirbheek.in

Subject Matter

The Needs of a Desktop App

Consequences of Needing Realtime Audio

New Perspective about the Usability of GStreamer

About the app

For groups of musicians to make music together while in different cities

Audio + video, but we only care about audio latency

Time audio packets spend inside the app: <5ms

Focus on not requiring any special hardware or configuration

macOS + Windows

Desktop is a different beast

Target platform: hardware, operating system

Consumer-grade: cannot restrict hardware much

Highly hardware-dependent

Desktop is a different beast

Consumer devices can have all kinds of hardware

- Built-in sound card
- Webcam mics
- USB mic, DAC, audio interface
- HDMI/DP output
- Laptop microphone arrays
- Bluetooth devices

Desktop is a different beast

Lots of edge cases

- Zero audio devices by default!
- Audio jack plugged-in detection
- Plugged-in device type detection

Desktop is a different beast

Edge-cases are not the biggest problem!

Long tail of small "not-uncommon" issues

Test, test, test

Lots of fixes needed: macOS + Windows

Lots of fixes needed: macOS + Windows

- Actually obey latency-time setting

Lots of fixes needed: macOS + Windows

- Actually obey latency-time setting
- Device switching & HW type (BT, USB, Built-in, etc)

Lots of fixes needed: macOS + Windows

- Actually obey latency-time setting
- Device switching & HW type (BT, USB, Built-in, etc)
- Persistent and unique ID

Lots of fixes needed: macOS + Windows

- Actually obey latency-time setting
- Device switching & HW type (BT, USB, Built-in, etc)
- Persistent and unique ID
- Unpositioned multi-channel support

Lots of fixes needed: macOS + Windows

- Actually obey latency-time setting
- Device switching & HW type (BT, USB, Built-in, etc)
- Persistent and unique ID
- Unpositioned multi-channel support
- Clock provider accuracy

Lots of fixes needed: macOS + Windows

- Actually obey latency-time setting
- Device switching & HW type (BT, USB, Built-in, etc)
- Persistent and unique ID
- Unpositioned multi-channel support
- Clock provider accuracy
- Audio thread priority

Lots of fixes needed: Windows Only

- Lots of improvements in `gstwasapi2`

Lots of fixes needed: Windows Only

- Lots of improvements in `gstwasapi2`
  - Missing devices in the winrt device provider

Lots of fixes needed: Windows Only

- Lots of improvements in `gstwasapi2`
  - Missing devices in the winrt device provider
  - Tighter usage, less unnecessary buffering

Lots of fixes needed: Windows Only

- Lots of improvements in `gstwasapi2`
  - Missing devices in the winrt device provider
  - Tighter usage, less unnecessary buffering
  - Select IAudioClient3 automatically in more cases

Lots of fixes needed: Windows Only

- Lots of improvements in `gstwasapi2`
  - Missing devices in the winrt device provider
  - Tighter usage, less unnecessary buffering
  - Select IAudioClient3 automatically in more cases
  - Exclusive mode support, incl device provider

Lots of fixes needed: Windows Only

- Lots of improvements in `gstwasapi2`
  - Missing devices in the winrt device provider
  - Tighter usage, less unnecessary buffering
  - Select IAudioClient3 automatically in more cases
  - Exclusive mode support, incl device provider
  - Query all available caps, format negotiation

Lots of fixes needed: Windows Only

- Lots of improvements in `gstwasapi2`
  - Missing devices in the winrt device provider
  - Tighter usage, less unnecessary buffering
  - Select IAudioClient3 automatically in more cases
  - Exclusive mode support, incl device provider
  - Query all available caps, format negotiation
- Ultimately WASAPI is a fallback option
  - Drivers are awful, firmware is conservative, vendors do the minimum
  - Microsoft seems to have low interest in latency-sensitive applications

Lots of fixes needed: Windows Only

- Lots of improvements in `gstwasapi2`
  - Missing devices in the winrt device provider
  - Tighter usage, less unnecessary buffering
  - Select IAudioClient3 automatically in more cases
  - Exclusive mode support, incl device provider
  - Query all available caps, format negotiation
- Ultimately WASAPI is a fallback option
  - Drivers are awful, firmware is conservative, vendors do the minimum
  - Microsoft seems to have low interest in latency-sensitive applications
- ASIO is the king, works great

Lots of fixes needed: Windows Only

- Lots of improvements in `gstwasapi2`
  - Missing devices in the winrt device provider
  - Tighter usage, less unnecessary buffering
  - Select IAudioClient3 automatically in more cases
  - Exclusive mode support, incl device provider
  - Query all available caps, format negotiation
- Ultimately WASAPI is a fallback option
  - Drivers are awful, firmware is conservative, vendors do the minimum
  - Microsoft seems to have low interest in latency-sensitive applications
- ASIO is the king, works great
  - *when it works*

Nothing is Easy on Windows

- Can't trust device-reported audio period values
- Exclusive mode volume is too low
- Using 44.1KHz in exclusive mode makes device render future 48KHz streams at 44.1KHz
- Output devices add 2-4x period latency just for fun
- Some devices on plug will expose a device then remove it and expose a differently-named device
- Totally different behaviour when using WASAPI vs ASIO on the same device
- ASIO devices abusing the specification, disallowing configuration
- Realtek's enshittified ASIO driver

Three Platforms, Four Audio Servers

Linux: Pipewire

Windows: WASAPI, ASIO

macOS: CoreAudio

Three Platforms, Four Audio Servers

Linux: Pipewire

Windows: WASAPI, ASIO

macOS: CoreAudio

`ASIO : WASAPI :: JACK : Pulseaudio`

The message

Being useful is the only thing that matters

Product Mindset

Thank you!

Jan, Seungha, Piotr, Amy