

TPAC 2020: Audio WG Minutes

Visibility: **Public**

Status: Current

Last Update: 10/22/2020

Oct 12-15, 19-22

Agenda

[Meeting agenda](#)

Minutes

Oct 12

Attendees

Christopher Needham, Jeff Switzer, Matthew Paradis, Raymond Toy, Hongchan Choi, Paul Adenot, Chris Wilson, Takashi Toyoshima, Chris Lilley

Minutes

- Introductions
 - Raymond Toy, Chrome, co-chair
 - Matt Paradis BBC, co-chair
 - Hongchan Choi, Chrome
 - Paul Adenot, Mozilla
 - Christopher Needham, BBC,
 - Jeff Switzer, CG member
- Status of V1
 - rtoy@: Summarizes what Chris Lilley said last meeting
 - <https://github.com/WebAudio/web-audio-api/issues/1933>
 - rtoy@: we also have a dozen, but they are not controversial. We just need to get them done.
 - padenot@: they are spec-technical, not the computer-technical. :)
 - padenot@: The next round of security review should be okay, because we already passed it twice in the past. So not a blocker? but we'll see.

- matt@: the impact on the test suite when we advance to REC? when some impl fail, and others pass?
- padenot@: if there's a clear intent to implement (from all vendors) then we are fine.
- hoch@: can W3C do something when the inconsistencies are noticeable?
- all@: `\\_(\`)/`
- cwiso@: hello
- rtoy@: <https://github.com/WebAudio/web-audio-api/issues/2248>
- Web MIDI (cwilso@, toyoshim@)
 - [Agenda](#) by toyoshim@
 - toyoshim@: Implementation updates from Chrome Web MIDI
 - we plan to prompt the permission dialog always.
 - updated the Windows MIDI backend to the recent one (work in progress)
 - padenot@: we have all the codes! working and tested. No shipping date yet.
 - back-end for Mac OS -> cross platform MIDI library
 - hongchan@: Perhaps notes and CC can be shipped first?
 - padenot@: positive about it, but no solid timeframe yet
 - toyoshim@: do we need to change the spec? anything to improve?
 - padenot@: no. fine as it is.
 - cwilso@: we don't need to worry about V2 until there's an actual device supports it. The biggest unsolved issue in V1 spec is the "backpressure". that can be solved with the stream.
 - hongchan@: so the stream is the solution?
 - (more thoughts on backpressure from cwilso@)
 - cwilso@: midi message as a user gesture => not a good idea (any connected Midi devices can send a message)
 - toyoshim@: new issue <https://github.com/WebAudio/web-midi-api/issues/215>
 - hongchan@: in worker instead?
 - padenot@, hongchan@: SAB is better in this case
 - /* hongchan@ couldn't scribe... */
 - // traiging/resolving the issues on the tracker...
 - Additional notes from toyoshim@:
 - https://docs.google.com/document/d/1wqdYBCy3p_r3lx2CqU5JoC-HM37CFY0FqMxcxRdEdsA/edit?usp=sharing
 - Updates:
 - Implementation updates for Chrome
 - No visible progress due to lack of engineering resource now, but there are several internal design changes to be aligned with current chrome design
 - Remaining works for the current latest spec
 - Permission prompt for non-sysex requests

- MIDIOutput.clear()
 - Windows backend API change from Win32 MMIO to WinRT does not happen yet, and still behind a flag
 - Can not obtain equivalent device names that reflects product names via the WinRT API
 - Implementation updates for Firefox
 - Sysex is only security concern
 - Keep working on, but no timeline for now
 - May have feedback on the spec, but will file issues when things matter
- Spec
 - Remaining works for V1
 - (<https://github.com/WebAudio/web-midi-api/issues>)
 - Major topics
 - Back pressure, probably using Streams
 - Handling MIDI messages as a user gesture -> not good ideas
 - Visited V1 targeting open issues, and took actions
 - Revisit the goal or the next step
 - MIDI 2.0
 - (<https://www.midi.org/specifications-old/category/midi-2-0-specifications-v1-1>)
 - Relevant issues
 - <https://github.com/WebAudio/web-midi-api/issues/211>
 - <https://bugs.chromium.org/p/chromium/issues/detail?id=1047087>
 - <https://discourse.wicg.io/t/web-midi-api-support-for-midi-2-0/4208>
 - Conclusion: It isn't mature enough to discuss here
 - Meeting Adjourned (10:54AM)

Oct 13

Attendees

Peng Liu, Philippe Milot, Philippe Le Hegaret, Jeff Switzer, Hongchan Choi, Matt Paradis, Raymond Toy, Chris Lillley, Jack Schaedler, Paul Adenot, Vinateya Koratkar

Minutes

- 9:07: Meeting started
- Introduction
 - PLH: W3C project management
 - P Milot: Audio Kinetic
 - Peng Liu: WebKit media Apple
 - Matt: BBC, co-chair
 - Jeff Switzer: dev, CG member
- Agenda
 - Spec V2 logistics
 - rtoy@: i had a PR for v2 spec, a boilerplate
 - rtoy@: V2 - it will be little changes, how do we work on that?
 - rtoy@: noise generator is pretty worked out
 - matt@: going from V1 to V2, including changes in V2 new nodes how do we make it neat/tidy way?
 - rtoy@: AudioParam power law - V1 addendum?
 - chrislilly@: if make a major change, we can copy over to V2.
 - rtoy@: new phase oscillator? we copy the entire oscillator section?
 - clilley@: yeah we should copy over.
 - rtoy@: once we go to REC, we can copy the entire V1 to v2.
 - clilley@:
 - hongchan@: Will file a new issue on how to write V2.
 - Priorities for V2
 - Hongchan presents [survey results](#)..
 - padenot@: Matches my expectations
 - Jack: Wonders if render size responses would be different if the question were asked differently to indicate that Android devices can only use 50% of the CPU with the current 128 frame size.
 - Dev tool support
 - padenot@: ECMAScript has added finalizers allowing implementing large parts of the dev tool support. The other parts are probably doable by writing a cross-browser extension. Firefox usage was rather small, but those users really liked it a lot.
 - chrislilly@: asks about DTS streaming.
 - hongchan@: Not really webaudio
 - padenot@: HTML audio stream seems the right place
 - matt@: Let's follow the proposed prioritization.
 - Output device selection
 - WASM integration
 - Output device change

- Configure audio device
 - CPU usage
- Output device selection
 - padenot@: Long history for this. See the following for some background:
 - <https://github.com/w3c/mediacapture-output/issues/50>
 - <https://github.com/w3c/mediacapture-output/issues/49>
Supported today when device output changes
 - <https://github.com/w3c/mediacapture-output/issues/46>
 - Philippe Milot: What is needed to move this forward?
 - padenot@: Writing spec and implementing. Some complications with multiple AudioContexts connected by a media stream.
 - hongchan@: What's the history here?
 - Paul: Was told this is where output device selection was supposed to be done.
 - Paul: Permissions and enumeration handled for us with mediacapture.
 - Hongchan: Not much configurability? Can't select channel counts, etc.
 - Hongchan: Also
<https://w3c.github.io/mediacapture-output/#mediadevices-extensions>
 - Paul: There is a move in webrtc to expose less things that can be fingerprinted. Old style is enumeration with list of devices and IDs. Now you request capability and get one device that matches.
 - Philippe Milot: Seems good for our use case as long as the info contains all of the desired info.
 - Hongchan: need to extend this
 - Paul: We can do a constraint pattern to select the device.
 - Hongchan: Related question: Can you determine what kind of output is connected? Like headset or speakers. Useful for games.
 - Philippe Milot: Yes, it's useful. Platform dependent but a boolean is good enough to say it's headphones or not. See
<https://github.com/WebAudio/web-audio-api-v2/issues/89>
 - Hongchan: This is relevant
<https://developer.mozilla.org/en-US/docs/Web/API/MediaDevices/ondevicechange>, but we need to provide more information about the device.
 - Paul: Also relevant:
<https://github.com/w3c/mediacapture-main/issues/730>
 - Hongchan; For configurability, what else do we need to request besides sample rate and number of channels.
 - Phillipe Milot: Wyse would want to present a list to user to select the desired output device. Then use the device to get the parameters.
 - Paul: Constraints don't have to be exact.
 - Hongchan: How to use channel mask?

- Philippe Milot: Mostly to determine where the LFE channel is. Typically WAV format. Complicated code to get vorbis and opus(?) channels to WAV because the ordering isn't the same for the codecs.
- Paul: We use SMPTE, it's in our spec
- Hongchan: How do we work on this? In Audio WG or in MediaCapture?
- Paul: Ideal to use MediaCapture to support what we need.
- Matt: Do we want this?
- Hongchan: We do want this but might want to piggyback on MediaCapture
- Paul: Same for us.
- Matt: Ready to move on to the next item on the list.

Oct 14

Attendees

Jack Schaedler, Philippe Milot, Raymond Toy, Hongchan Choi, Paul Adenot, Matt Paradis

Agenda

- Use WASM within Audio Worklet more efficiently. ([issue](#), [issue](#))
- <https://github.com/whatwg/html/pull/6056>
- Programmatically inspect the audio rendering CPU usage in JS. ([issue](#))
- More AudioNodes: Noise generator, Noise gate, Expander, and new Oscillators

Minutes

- WASM with AudioWorklet
 - Hongchan: <https://github.com/WebAudio/web-audio-api-v2/issues/5#issuecomment-673589793>
 - Raymond: Yes, knowing the cost would be beneficial. I vaguely remember Jack saying he was interested in doing this. My guess is the cost of copying is small, but if the worklet is a gain node, it's a significant fraction.
 - Paul: Lots of non-linearity. Depends on cache effects depending on the memory used. Will take some care to measure to get meaningful results.
 - Philippe: Closely related to BYOB issue?
 - Paul: Yes.
 - Hongchan: Can we do this without help from WASM people.
 - Paul: Yes, but we can use functions instead of class.
 - Hongchan: How does the worklet know about these functions?

- Paul: Can use C-style to pass state and functions (pointers) to WASM. Need to ensure minimal overhead. WASM can only call functions.
- Raymond: Can developers already do this?
- Paul: implement register function to register the process function with state and function.
- Hongchan: Sounds like animation worklet.
- Paul: Yes, but kind of the opposite.
- [Paul updates the issue #5]
- Paul: May want to have a simple example with this idea to start with. Will write one up later.
- Hongchan: How is #4 related? It's about registering memory
- Paul: Having #4 and #5 is best, but just one is ok.
- Hongchan: Memory needs to be owned by the node, not wasm. Perhaps done in the constructor?
- [Hongchan summarizes discussion in #4] - [link](#)
- Hongchan: What happens if the channel count changes requiring more memory
- Paul: Could have a callback saying more memory is needed. Lots of memory could be allocated and a pointer to the block to be used by the node.
- Hongchan: Will write a simple example for #4.
- Worklet
 - [Worklets are being moved to the HTML standard](#)
 - Paul: This is a welcome change.
 - Hongchan: Just need to be aware of the upcoming changes to make sure AudioWorklets are fine.
- CPU usage
 - Hongchan: Paul says to allow performance.now() in the worklet.
 - Hongchan: How does this apply to the entire graph?
 - Paul: It doesn't really.
 - Hongchan: Would like this for the entire graph.
 - Hongchan: Is performance.now() heavy?
 - Paul: Can be fast, depending on the OS. (Very cheap on mac. More complicated on Win). Probably used during development, not production.
 - Raymond: Is performance.now() resolution limited?
 - Paul: Yes. Happens on non-isolated machines with spectre issues. Or if anti-fingerprinting enabled (in Firefox).
 - Raymond: Probably not a show-stopper. Just call performance.now() every 10 or 100 renders.
 - Paul: Let's proceed as if performance.now() is useful for audio profiling.
 - Raymond: Still want to add API for the entire graph?
 - Hongchan: Yes. Something like AudioContext.foo to get usage.

- Jack: I think most devs who are asking for this, are mainly wanting to be able to answer the question, "Should I turn off the Reverb on my app because I'm on some mobile device that is having trouble keeping up?"
- Raymond: Does providing the CPU usage for Webaudio work?
- Jack: Yes. Something more "chunky" is totally fine.
- Raymond: Already designed most of the API
- Hongchan: Want to talk to privacy people
- Paul: Definitely, since it says something about the machine.
- Hongchan: Come up with a rough API design.
- Paul: Date.now() does work in an AudioWorklet.
- Render size
 - Philippe: Proposes discussing this tomorrow
 - [General agreement]
- Audio Nodes
 - Noisegate/Expander ([issue](#))
 - Raymond: Paul mentioned in the issue that a prototype would be really useful.
 - Echo cancellation?
 - Hongchan: Could we implement echo cancellation with webaudio?
 - Paul: Should work. But capturing all other audio is difficult or impossible (Mac).
 - Raymond: Yeah, exposing other audio to the web seems scary. Capturing the mic input and audio output from a tab seems ok and echo cancellation should work. Just can't cancel other audio from other sources.
 - Raymond: Maybe the only thing missing is an API to allow access to all other audio sources. I never have any other audio when on a call.
- Adjourned. We'll start with render size tomorrow.

Oct 15

Agenda

- Select the render block size of the Web Audio renderer ([issue](#))

Attendees

Philippe Milot, Raymond Toy, Hongchan Choi, Paul Adenot, Ruth John

Minutes

- Render size
 - Paul: There's agreement to do this. Do we want to discuss the API?
 - Raymond: Yes.
 - Paul: Default is 128. There should be a way to ask for optimal. And to be able to specify a number.
 - Paul: Is the number a hint?
 - Raymond: Yes. I want browsers not to have to support all possible values. The spec should require powers of two, and leave the rest to the browser
 - Philippe: Can the spec mandate certain values?
 - Paul: Yes. Multiples of 128 should not be a problem.
 - Raymond: What about 384? Chrome currently wants convolvers to use a power of 2. Might work by buffering the data for the convolvers to operate only on power-of-two sizes. Makes the convolver processing bursty.
 - Raymond: On windows 440 size could be done with 512, losing about 15% CPU.
 - Paul: Yeah. But definitely a problem on Android.
 - Paul: Pixel (4?) is 96.
 - Raymond: Would powers of two be good enough?
 - Jack: Yes
 - Raymond: Proposal: It's a hint; only powers of two are required; an attribute is needed on the context to indicate the actual size.
 - [Updated issue with proposal]
 - Raymond: Paul's last comment about device changes
 - Paul: Yeah, devices changes can be detected.
 - Raymond: But device changes don't affect these;
 - Paul: Yeah. Things should continue as is with buffering/resampling. Devs need to redo the graph if needed.
 - Raymond: Paul: what does "being able to know the right size"
 - Paul: Something in media devices to get the HW size. But not really needed now. Just request the optimum size.
 - Raymond: Do we want to expose the HW size in addition to the actual size?
 - Paul: No, not necessary
 - Philippe: Not necessary
- Other P1 issues ([issues](#))
 - NoiseGate (<https://github.com/WebAudio/web-audio-api-v2/issues/12>)
 - taking the modular approach: build envelop follower
 - Raymond: Is providing envelope follower enough to build everything with other native nodes?
 - Paul./ChrisL: Yes. Just need gain, waveshaper, and delay.
 - Paul: What's the latest in subclassing AudioNode?
 - Hongchan: Would be great to have. Have seen people do this. AudioWorkletNodes support this and it works.

- Raymond: My guess is that people who want this want a node to throw in and not have to build it themselves.
- Hongchan: Curious about this approach. Will it be fast and efficient?
- Paul: Could mix and match with native nodes and worklet.
- Raymond: My take is that either we should provide the node (no envelope follower) or let devs create their own with a worklet. Providing a follower isn't enough.
- Raymond: What's the conclusion here?
- Paul: It's easy with an envelope follower. Just a couple of lines.
- Raymond: I'm ok with that if the spec has examples showing how to do a nice noisegate/expander.
- Raymond: Conclusion is that community input is really important?
- [Playback position](#)
 - Paul: I like the term "position" instead of "time".
 - Raymond: Works for me.
 - Jack: Just need to update once per frame (animation frame)
 - Paul: Let me write a proposal in the issue.
 - Raymond: Looks good.
- [Change outputChannelCount dynamically](#)
 - Paul: Firefox only allows changes between renders
 - Raymond: Chrome is the same, except the may not happen in the next render because it can't get the locks needed to make the change.
 - Raymond: Use case with a convolver is a good example. Mono input with mono response is mono. But change the response to stereo, and the output is stereo. Can't do that currently with an AudioWorklet.
 - Paul: Let's provide an API to set the output channel count. But also need a way to unset this change.
 - [Paul updates the issue]
- [Handling of output parameter for process\(\)](#)
 - Raymond: The original issue is no longer relevant because the parameters are frozen now. (Discussion moved into different directions)
 - Paul: This is an edge case that makes things hard.
 - Paul: Example: Delay node with 4 s delay, and a stereo input that changes to mono. The delay needs to preserve stereo for 4 more sec.
 - Raymond: Yeah, I don't know how to handle that in a worklet. Could have process send a message to the main thread to request changing the channel count after 4 sec. Not terrible but ok. (This could have been handled nicely if we had made process() return the output data instead of passing in the output arrays. Too late for that.)
- Adjourned. Continue next time with handling of output parameter.

Oct 19

Attendees

Jack Schaedler, Lora Friedenthal, Raymond Toy, Chris Lilley, Paul Adenot, Jer Noble, Peng Liu, Matt Paradis

Minutes

- [Handling of output parameter for process\(\)](#)
 - Raymond: Reiterates we should close this because this isn't possible anymore. It should be taken up in #37.
 - Paul: Way forward is changing the channel in the RT thread is out-of-scope so you can't change the number of output channels in the same process call. One way out is if we move to BYOB to change the channel output.
 - Raymond: Yes, but that's really part of #37 and the BYOB issue.
 - Paul: Yes.
 - [Raymond updates and closes #42]
 - [Paul updates #37]
- [Update the README.md](#)
 - [Paul updates the issue]
- [High res timer in AudioWorkletGlobalScope](#)
 - Paul: Need performance.now() and cpu load factor (issue #). Should solve the problems with devs knowing how to profile and adjust the graph appropriately.
 - Jer: [Mic not working]. Do you need a specific clock or whether a monotonically increasing counter is needed.
 - Paul: It's complicated. Talked to a lot more people and they want to know load measure to see if they're about to glitch. For this you need to have some real-time value. The engine could compute the load and expose that.
 - Jer: Could give quantized load factor, say 5% level. Not sure about exposing perf.now() given spectre. Maybe ok with enough cpu mitigations.
 - Jack: I think for the real nitty gritty profiling, using the browser tools is actually preferable to doing something with performance.now(). For the case of computing load and responding on a machine you might not have control over, I think most devs would be fine with a really coarse quantization of the load, ~10% or so.
 - Raymond: Agrees that you need dev tools for detailed profiling.
 - [Paul summarizes in the issue]
- **Update from Apple**
 - Jer: Last release concentrated more broadly on supporting major websites. Major pain point was the lack of AudioWorklet. This is now available (or will be soon). Includes unprefixing. Safaris 114 is the current version, and worklet

should be in the next one. Should be cross-platform. iOS will have it in the next full release.

- Jack: yeah, hard to overstate how exciting this is!
- [Input only/muted AudioContext](#)
 - Jer: For analysis?
 - Raymond: Yes.
 - Jer: Use case: users on iOS devices with hearing aids via bluetooth. Media isn't heard.
 - Paul: We do something like this.
 - Paul: Start in input-only mode and change output device to hear audio, glitch-free. This worked in Firefox OS.
 - Raymond: That seems really complicated.
 - Jer: In context of the issue, this seems really just analysis only.
 - Raymond: Yes. WebRTC asked us about this exact ability some time ago to bypass the user-gesture when only doing analysis.
 - Raymond: This particular use case seems straightforward, but the other cases Paul mentioned about switching outputs and enabling audio seem really complicated.
 - Paul: Yes. I don't have a solution.
 - [Other discussion that I didn't capture]
 - Raymond: This use-case is simple. The other use-cases should perhaps be a different issue?
 - Paul: Need more input on uses cases and CG help here.
 - [Paul updates issue]
- Priority 1 issues are done; moving on to priority 2
- [Allow OfflineAudioContext to render smaller chunks repeatedly](#) (issue #6) and [incremental delivery of gargantuan data from OfflineAudioContext](#) (issue #66)
 - Raymond: This (and #66) came up many TPACs ago and Jer's suggestion about calling startRendering() again would work. Question is how to specify how much data to render.
 - Paul: Oftentimes we think streams would be the solution, but sometimes it's not.
 - Jack: I like where Raymond is going with the idea there. The intention, "Give me the next N samples" from the context is pretty nice for many reasons.
 - Jer: What's the proposal? startRendering(nsamples)?
 - Paul: That's one approach. Streaming would work.
 - Jer: Use case involves raw audio or encoded? Presumably encoded to be uploaded somewhere?
 - Jer: Minimal AP change is startRendering(n), allowing startRendering to be called more than once.
 - Paul: I can see using streams
 - Jer: Yes, and you can write your own stream handler.

- Paul: WebRTC is moving towards streams. So is WebCodecs. Very elegant to be able to interact with this.
- Raymond: Streams are fine with me; just need an API.
- Paul: Would be nice to see what #66 would look like with a stream.
- Paul: Need to re-read streams section.
- Raymond: Do we want to make a proposal?
- Jack: Callback with a buffer is just so simple and easy, but I definitely see the appeal of the stream based approach
- Raymond: Proposal: close #6 in favor of #66 and make sure #66 handles the termination case.
- [Paul writes summary in issue]
- [Raymond updates #6 and closes it]

Oct 20

Attendees

Jack Schaedler, Paul Adenot, Raymond Toy, Hongchan Choi

Minutes

- [Storing AudioBuffers in different bit depths](#)
 - Paul: Not sure what to do here. Streaming decode is already available. Do people really need in-memory assets? Probably.
 - Raymond: I think people maybe just want decodeAudioData to have 16-bit data, all done transparently, like Firefox does today.
 - Paul: I'm looking up someone's blog about free algorithms for in-memory compression of audio. <https://www.bitsnbites.eu/high-quality-dpcm-attempts/>
 - Paul: I'm envisioning some method to allow compressing audio (decodeAudioData) using DDPCM.
 - Raymond: But it's lossy.
 - Paul: I wonder how it compares to mp3.
 - Raymond: What about FLAC? License is free
 - Paul: It's pretty complex. Indexing into the file to get the right data is expensive. And looping makes it hard.
 - Raymond: Yeah, looping is a problem.
 - Raymond: No objections as long as users have to ask for lossy compression
 - Paul: Of course.
 - Raymond: Not sure where to go from here. We've asked for proposals on what this should look like.
 - Raymond: We all agree this is useful; we just don't know what it would look like and what it should do.
 - [Paul writes summary]

- [Real-time pitch adjustment](#)
 - Paul: I've heard that students have implemented pitch-shift in AudioWorklet. But failed to ask where the code is.
 - <https://github.com/olvb/phaze/> This is pitch shifter, not time shifter.
 - Raymond: Doesn't seem to be all that big
 - Paul: Yeah, it's really tiny.
 - Jack: I'd be very interested to try this implementation out!
 - Hongchan: Is this a source node or processing node?
 - Jack: This definitely feels like a case where you'd want to prototype a bunch of options using AW, and work through all the kinks. Consider playing backwards, etc. A source would perhaps make more sense
 - Jack: Crossfading and looping cleanly might also require having access to the source.
 - Jack: Yeah, for a pitch-shifter, or a paulstretch kind of thing, a node is fine.
 - Jack: However, people are going to want to build things like warped clips in Live/Bitwig, and that's going to be tricky with a node.
 - Raymond: I guess the next step is to get the example.
 - [Raymond summarizes in issue]
- [Worker support for BaseAudioContext](#)
 - Jack: The pain of scheduling contesting with the main thread is very real :)
 - Hongchan: Seems really useful, but how useful would it be without media streams and media elements. We'll lose access to getUserMedia.
 - Paul: Found the proposal from Intel and Mozilla: <https://github.com/w3c/mediacapture-worker/> (draft)
 - Hongchan: Maybe split this into two parts: AudioContext in worker, and media stream in worker?
 - Paul: Maybe we can transfer the streams? [Pings colleague on this, asking for info on this].
 - Paul: Splitting this in two is fine. But what's missing is use-cases. There are people who care about this who don't need media stream. Sounds like a good plan to split.
 - Raymond: Someone want to summarize this and create the new issue?
 - Paul: Yes, I'll do it.
- [Customizable windowing function for Analyser](#)
 - Raymond: I think we have the API designed, but need to decide what happens when you switch the window type. I find it unlikely people would want to change it after construction. Maybe not allow changes?
 - Jack: The only case I can imagine where that's important is someone making a DSP tutorial or something where they want to compare the effect of the windowing functions. They can work around that by just making N nodes
- [ConvolverNode limited to two channels](#)
 - Raymond: Mostly historical. Not sure how to handle 4-channel responses.

- Raymond: Maybe add option to specify response is for individual channels, no matrixing operations. Number of outputs is the number of responses.
- Raymond: Then only need to describe how to upmix the inputs.
- Raymond: But do we really want to do this?
- Paul: One of those things where if you're not doing complex surround sound, you don't need it.
- Jack: Yotam Mann would be good to talk to Yotam uses WebAudio for big (32) channel student projects
- Raymond: Yes, we should ask him.
- [Raymond updates issue]
- [loadHRTFDatabase for SpatialPanner](#)
 - Paul: Talk to Matt about this.
 - Raymond: Let's assign to Matt to get more information on what's needed here.
 - [Raymond updates issue]
- [Configurable sample rate quality](#)
 - Paul: What is the best way to measure this?
 - Raymond: The quality?
 - Paul: Was told that high quality with SIMD is faster than linear. But didn't verify the claim.
 - Raymond: Seems hard to imagine, but maybe.
 - Raymond: I see it useful for final output of AudioContext
 - Paul: Firefox does it carefully for ABSN, hiding the latency of the resampler.
 - Jack: No, but I could make one if you want something similar to the karplus tester thing if you want.
 - Raymond: What do you want to do?
 - Paul: To understand the cost of resampling.
 - Jack: I could just make a page with tons of buffers playing back in an amusing way
 - Paul: I'll get some data on this.

Oct 21

Attendees

Paul Adenot, Raymond Toy, Hongchan Choi, Jack Schaedler, Matt Paradis, Chris Lilley

Minutes

- [Configurable sample rate](#) (cont'd)
 - Raymond: Is there more to say about this?
 - Paul: This is really about final output?
 - Raymond: Mostly

- Paul: Took a look and output resampling on Windows is different between voice and general audio. In the latter, we do it more carefully. Output latency is about 70 frames (depends on sample rates).
- Raymond: 70 frames is pretty good.
- Hongchan: This should be configurable at runtime
- Paul: I was testing yesterday with playback rate between 0 and 1 exclusive. Switched between high quality resampler and linear interpolation. (High quality is very optimized with SIMD, etc.) No results yet.
- Paul: Could be playing back 8 kHz audio, low quality source, so low quality resampler is not terrible.
- Hongchan: How important is this?
- Jack: I'm interested in any config settings to trade capacity for quality. Especially important once render capacity is available.
- Hongchan: Maybe start with final output
- Paul: Devs may choose quality depending on asset. Long background audio may not need high quality.
- Hongchan: What's the next step?
- Paul: Output should be done. The rest could be important with significant savings.
- Raymond: Changing quality in ABSN while running would sound funny
- Paul: Jack, how would you do this? Recreating the source?
- Jack: That would work, but we might just reload the whole page
- Paul: If we go per node, static could be enough.
- Paul: Could be ABSN option
- Raymond: What does FF use for resampling when playback rate is not 1?
- Paul: Use the resampler from Opus, formerly from speex.
- [Paul updates issue]
- [Setting convolution buffer asynchronously](#)
 - Hongchan: We should just do it? HRTF should be a separate issue
 - Paul: Yes, let's do convolver here.
 - Raymond: Do we deprecate the buffer setter? Of course it won't go away.
 - Paul: I was thinking about just this the other day.
 - [Paul updates issue]
- [Add method to cancel ABSN.stop\(\)](#)
 - Raymond: Calling stop with a large number works for me. Or maybe with a negative number?
 - Raymond: But either way, the API would be weird to the OP.
 - Hongchan: No strong opinions; never encountered this situation.
 - Hongchan: Is this really priority-2?
 - Raymond: It was decided at the last F2F meeting; I don't remember the reasoning.
 - Raymond: I'd prefer not to add an API for this; stop(huge) works.

- Raymond: What should we do?
 - Jack: Agree that it's not priority-2.
 - [No conclusion]
- [When does stop\(time\) stop?](#)
 - Raymond: Since Chrome and Firefox do the same thing already, we just need to make it clearer in the spec when stop actually happens. Not sure what to say in the spec though.
 - Paul: What Karl says is right, but would prefer the way things are now
 - Raymond: Yes, Karl is right, but having audio before start and after stop would be weird to developers.
 - [Raymond updates issue]
- [Use SharedArrayBuffer for getChannelData](#)
 - Chris: Not sure what the primary concern is.
 - Paul: The buffer could be read from a different thread while it's being modified. Not something we want.
 - Jack: We ran into this same issue and used WASM to work around this issue. We had a WebWorker and doing the copy. Conceptually want a const ref to pass the data around. It's a real problem.
 - Paul: It's a bit of a problem. It's another way to get shared memory. We need to be careful about what to do here, especially when it's not available.
 - Jack: Seems like the most common case is loading a big file and drawing a waveform.
 - Paul: Agree, but we want to be careful about enabling shared memory.
 - Raymond: Would having a sharedArrayBuffer have helped you Jack?
 - Jack: If you can get the data to a webworker without a copy, that would have helped us. Copying the data took more time than the analysis on the worker.
 - Paul: One of those things where I wish there was a way to freeze a portion of memory.
 - Raymond: What should we say? Further study?
 - Paul: I think it's fine.
 - Raymond: Can you write up a summary?
 - [Paul updates issue]
- [Informing AudioWorklet if output is not connected](#)
 - [Raymond summarizes the issue]
 - Raymond: Could be a useful optimization so worklet doesn't have to compute the output if the output isn't connected.
 - Paul: Could be useful. Are there precedents in other frameworks?
 - Paul: What would it look like?
 - Paul: Don't think changing the channel count is the right way.
 - Paul: People probably don't check the output
 - Raymond: Yes, this will definitely break things
 - Paul: It's useful though.

- Raymond: No one is asking for this except me, and I don't write things.
 - Paul: Maybe ask people.
 - [Paul updates issue]
- [Support Q and S for shelf filters](#)
 - [Raymond links to [demo](#)]
 - Raymond: Demos shows that we can't make them backward compatible. Need new types.
 - Paul: Yeah, just get a name and specify it.
 - [Raymond updates issue]

Oct 22

Attendees

Lora Friedenthal, Jack Schaedler, Matthew Paradis, Raymond Toy, Hongchan Choi, Paul Adenot

Minutes

- [Remove compileStreaming from AudioWorkletGlobalScope.WebAssembly](#)
 - Hongchan: Does compileStreaming involve multi-threading?
 - Paul: It works with fetch to start a streaming compile, or anything else that can be streamed.
 - Paul: is Response only from fetch? Yes.
 - Hongchan: Seems quite useful.
 - Paul: But we don't want to compile on the audio thread.
 - Hongchan: But this is a whole package coming from WASM.
 - Paul: That's what I was asking but I don't know if we can. Just need to find the right person to ask the question.
 - Hongchan: [Action item to ask]
 - Hongchan: If the answer is no, what should we do?
 - Paul: Well it exists but is basically useless.
 - Paul: I guess we just need an answer from people who know.
 - [Paul updates issue]
- [Multithreading](#)
 - Paul: Any new developments on this?
 - Hongchan: What about [comment on Jun 13 \(wytrych\)](#)
 - Raymond: I think he means two contexts get separate threads in Chrome.
 - Paul: Firefox is thinking of changing its implementation to have separate threads.
 - Hongchan: This is about multiple threads for all processing?
 - Hongchan: Seems interesting, but would need a rewrite. May take several years.
 - Raymond: But would we expose this to the user?
 - Hongchan: Problem is unsynced clocks and lack of single clock.

- Hongchan: Could be up to the UA.
- Hongchan; Multiple pthreads in WASM with workers solves this.
- Paul: Then we're back to the problem of thread priorities.
- Raymond: Can't this be done inside a worklet?
- Hongchan: I think so. Oh. Not in chrome. Pthreads not supported.
- Paul: Probably don't want to allow this either.
- Hongchan; Ask developers if they want multiple threads of nodes, etc.
- Paul: I linked the mailing list with more info.
- Jack: This is also the use case I imagine most devs will want addressed.
- Hongchan: Even if you could spawn pthreads, they won't be realtime.
- Paul: That's why I suggested adding a flag to say audio contexts run on a separate thread. Basically opt-in to multiple threads.
- Hongchan: Needs more clarification from devs. Insufficient detail in the issue.
- Paul: I think in the end, the F2F comment still holds, since Chrome does this.
- Jack: This topic is really complicated. I'm going to go back do some homework, then write into the comments outlining the exact use cases that are most likely.
- Hongchan: I think the main point is "When running multiple contexts in parallel we of course run into problems like unsynced clocks and the lack of a single sink so it's impossible to have one master gain for example"
- Jack: I think the main/common desire is this: "I have a big WASM app... I do some analysis of my custom audio graph, and realize that I can (in theory) render two portions of my graph in parallel... In a DAW context this might be two 'tracks' which don't interact in any way. If I could create two threads from my worklet, that would be helpful." But as you say, if I can't create threads at the right priority and everything... then
- Hongchan: If this is an issue with pthreads and WASM, then this is a problem to be solved in WASM.
- Jack: +1
- Paul: Firefox allows 200 ms in a worklet (instead of the normal 3 ms). Then worklet errors out.
- Hongchan: Want to ask WASM people if it's possible to enable realtime pthread.
- Paul: There are native apps that allow this.
- Hongchan: I'll ask WASM people about this.
- [Paul updates issue]
- Next steps
 - Discussion on how to get more feedback on issues
 - Sorting by thumbs up is somewhat useful, but there are too many different emoji (heart, smiley face, etc.)
 - Ask CG members to thumbs up the issues that are important to them.
- [Headphone detection](#)
 - Hongchan: self-assigned
 - Moving to priority-1

- [Channel layout detection for output device](#)
 - Hongchan: Should be handled by media output devices
 - Hongchan: Not sure if layout detection is possible. Also exposes more fingerprinting.
 - Paul: I can tell you but it will take a long time. Stereo output could actually be channels 7 and 8 on a fancy audio card.
 - [Paul updates issue]
 - Hongchan: Ask Phillipe how this works in Wyse.
 - [See also discussion above about [output device selection](#).]
- Meetings adjourned for TPAC 2020.
 - We'll resume our regularly scheduled meetings next week at the usual time and place.
 - Thanks to everyone who participated!