Episode Book

EPISODE #6485

Choosing Audio Bitrates and Settings for Voice

PUBLISHED	7 November 2026 at 16:16 UTC
DURATION	21:49
LISTEN	myweirdprompts.com/episode/voice-recorder-settings-opus-02658/
HOSTS	Corn, Herman, Angus, Hilbert
SCRIPT MODEL	z-ai/glm-5.3
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.26
SOURCES	18 cited by the research (listed at the end)
BOOK GENERATED	8 November 2026 at 07:32 UTC

SYNOPSIS

Voice recording settings are often dialed far higher than necessary. An analysis of modern audio codecs reveals why Opus beats MP3 and FLAC for voice notes and meetings, why 24 kilobits per second mono offers the ideal balance of size and clarity, and how Android's mysterious microphone zoom slider actually operates behind the scenes.

TRANSCRIPT



Corn

What if I told you most of a recording app's settings menu is decoration, and the one slider nobody can explain is a camera feature that leaked into audio? That's roughly where Daniel has landed. He's got a OnePlus running ASR Voice Recorder, a menu full of formats and bitrates and sample rates, a microphone zoom slider he doesn't understand, and he wants firm numbers for three situations: an important meeting with a lawyer, a voice note that has to be good to listen to and transcribe well, and a podcast.



Herman

And he's already halfway there on his own. He dropped Opus from sixty-four kilobits a second to thirty-two, on the friendly AI down the road's advice, and he's asking whether that's right.



Corn

He suspects, and I quote, that "most people record at an unnecessarily high fidelity." He's correct, and the arithmetic backs him, but before the numbers, the formats. Because the format choice constrains everything downstream.



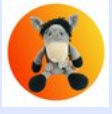
Herman

Seven formats in his list. WAV, MP3, Opus, FLAC, OGG, M4A, AMR. Small correction to the menu itself first: the app's own site lists an eighth, thirty-two bit WAV, which Daniel's list doesn't have. Worth checking his build, though for voice it changes nothing.



Corn

An eighth format nobody asked for. The menu grows, it never shrinks.



Herman

Never. Give each one a line. WAV is uncompressed PCM, the raw numbers off the microphone, biggest file on the menu, your archival and forensic master. Wasteful for a voice note, correct when the recording might end up in front of someone official. FLAC is lossless compression, so it's bit-identical to the WAV but typically about half the size, and it's the one I'd actually pick when lossless matters. Xiph's guidance on archiving is blunt: use a lossless format to prevent generation loss.



Corn

And generation loss is the thing people never see coming, right? Walk me through it.



Herman

It's the photocopy problem. You photocopy a page, it's fine. You photocopy the photocopy, then photocopy that, and by the fifth generation the text is fuzzy. Every lossy encode is a photocopy. You record to MP3, your editor re-encodes it when you trim the dead air off the front, your transcription service maybe re-encodes again, and each pass throws away a little more. Lossless breaks that chain, because the copy is exact, forever. FLAC is the page in the vault.



Corn

MP3, I assume, survives on pure ubiquity.



Herman

It's a odd format under the hood, raw audio coding without a container, with the tags bolted on afterwards as a hack. And that hackishness is why it plays everywhere. Every transcription API, every legacy player, every email client will take an MP3. Daniel's rule, MP3 only when a particular API needs it, is sound. Fallback, not default.



Corn

M4A?



Herman

Not a codec, a container. AAC audio inside an MPEG-4 wrapper. It's the Apple-friendly option, the one QuickTime and the Apple ecosystem handle natively. Fine, unremarkable. OGG is the open container, usually carrying Vorbis inside, and Vorbis is the codec Opus replaced. So OGG is the legacy open choice, superseded by its own successor. Then Opus, which we'll get to, and then...



Corn

AMR. The one he says he doesn't even know.



Herman

My favorite item on the list. AMR is a telephone codec. Standardized by 3GPP in October nineteen ninety-nine, encodes a band from two hundred to thirty-four hundred hertz, bitrates from four point seven five to twelve point two kilobits a second. Toll quality from seven point four. It's the format your old Nokia saved voice memos in.



Corn

So it's a museum piece sitting in a twenty-six-year-later app.



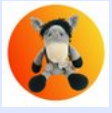
Herman

A museum piece with a docent. It exists for MMS and legacy voicemail systems and nothing else. Wikipedia's own caveat is that it's a speech format and unlikely to give ideal results for other audio. Never pick it. If you pick it, your recording sounds like it came through a phone from the year the codec was born.



Corn

Which, to be fair, it did. There's a whole genre of fiction that spends fortunes recreating that sound in post, and here it is free on the menu.



Herman

One slider away from a period drama. Now Opus. Daniel's instinct, best balance of size and performance for voice, is correct and it's not close. The spec, RFC sixty-seven sixteen, defines it as scaling from six kilobits narrowband speech up to five hundred and ten kilobits stereo music, and it carries two encoders inside, linear prediction for speech and an MDCT transform for music, switching between them as needed.



Corn

How does that switching actually behave? Does it click between them?



Herman

Not audibly. The encoder analyzes each frame and picks the tool, and it can blend across frames, so you don't hear a seam. The practical upshot is that if your lawyer meeting is interrupted by a fire alarm, the alarm gets the music encoder and comes through far better than a speech-only codec would manage. The codec is two specialists sharing one desk. Listening tests have ranked it above MP3, AAC and HE-AAC at any given bitrate until transparency.



Corn

So the codec question is settled. The interesting question is the number.



Herman

Because the number is where Daniel's already past the sweet spot without knowing it. Xiph's recommended settings table gives twenty-four kilobits mono for audiobooks and podcasts. And their bandwidth table is the part I love: fullband, meaning three to twenty thousand hertz, is reached at roughly sixteen kilobits mono for voice.



Corn

Hold on. His current thirty-two is already double the fullband threshold.



Herman

Double, for voice. At thirty-two kilobits the encoder has run out of frequency to buy. Everything speech contains is already encoded. The extra bits buy robustness, margin against bad rooms, not audibility. So the honest answer to "is thirty-two correct" is: it's fine, it's slightly generous, and twenty-four is the number. Sixty-four to thirty-two was the right move, thirty-two to twenty-four completes it. That's a quarter off the file size for nothing you'd hear.



Corn

I dropped to sixteen on my last one and dared anyone to notice.



Herman

Nobody would. Sixteen would still be fullband. Here's the file-size math, because this is what Daniel cares about with his long recordings: at thirty-two kilobits, one hour is fourteen point four megabytes. At twenty-four, ten point eight. At sixteen, seven point two. At his old sixty-four, twenty-eight point eight. So the sixty-four to thirty-two move already halved it, and thirty-two to twenty-four takes another quarter off.



Corn

Fourteen megabytes an hour. His worry about long recordings hitting API limits isn't paranoia.



Herman

It's plausible arithmetic. A two-hour meeting at his old setting was pushing sixty megabytes, and nobody loves a sixty megabyte upload. Smaller files are safer regardless of any specific limit.



Corn

Sample rate next, because that's the one people confuse with bitrate.



Herman

Always. Bitrate is how many bits you spend describing the audio. Sample rate is how many times a second you look at the waveform, and it caps the highest frequency you can represent at half the rate. Nyquist. So eight kilohertz gives you a four kilohertz ceiling, that's telephone, narrowband. Sixteen kilohertz gives you eight, that's wideband, and it's the standard for speech recognition, it's what the models are trained around. Forty-eight gives you twenty-four, fullband, music territory.



Corn

Give me the picture for Nyquist, because "half the rate" sounds like a rule pulled from the air.



Herman

Think of it as a flip book. To draw a wave on paper, you need at least two samples per cycle, one at the top, one at the bottom. Fewer than two per cycle and the wave can masquerade as a different, slower wave, which is where the old wagon wheels spinning backwards in films come from. Same phenomenon, different medium. Half the sample rate is the fence where that aliasing starts.



Corn

And speech doesn't go to twenty-four kilohertz.



Herman

A human voice tops out somewhere under twelve, and everything that matters for intelligibility, the consonants, the sibilance, lives well under eight. So sixteen kilohertz captures everything a single voice contains. Forty-eight buys you music beds and avoids resampling questions, nothing more, for speech.



Corn

Channels. His default is stereo and he knocks it to mono.



Herman

Correct instinct again. For one mouth, stereo doubles the file and buys nothing. Both channels carry the same voice with a hair of phase difference. And Opus itself quietly agrees: below roughly nineteen kilobits the encoder just downmixes stereo to mono on its own. So at twenty-four you could set stereo and the codec would fight you for your own good.



Corn

The codec undoing the user's menu choice. My kind of software.



Herman

So that's the formats and the numbers. Now the setting Daniel actually asked about and nobody explains, microphone zoom.



Corn

I assumed it was an app gimmick. A slider that changes a number and nothing else.



Herman

I did too, and it isn't. It's a documented Android platform API, added in API level twenty-nine, Android ten. The method is setPreferredMicrophoneFieldDimension on AudioRecord, and the parameter is a float from minus one, wide angle, through zero, no zoom, to one, maximum zoom. On a phone with a microphone array, what it requests is beamforming, narrowing the capture field toward a direction.



Corn

You said requests.



Herman

That word is carrying the whole weight. It's a preference, not a command. The method returns a boolean, and the audio HAL, the layer the manufacturer controls, is free to ignore it. Whether a given OnePlus model honors it depends entirely on the OEM's HAL, and no source I've seen documents ASR's implementation or which phones actually do it. But here's the detail that tells you what it's really for. The AOSP source comment says the usage would include specifying the audio focus to follow the zoom specified for the camera being used.



Corn

It's a camera feature.



Herman

It's a video-call feature that leaked into audio. The platform built it so your zoomed-in video would get zoomed-in sound to match, and a recorder app can reach in and pull the same lever for its own purposes. Which is clever repurposing, and also why it's a request the HAL may silently shrug at.



Corn

Daniel's plan is to try zero point two for a little more directionality.



Herman

And that's the right experiment, run the right way. Zero point two is a mild narrowing, a nudge. If the HAL ignores it, he's lost nothing. If he went to one, that's aggressive. On a real beamforming array, maximum zoom means everything off-axis gets thin and phasey, and off-axis is where the other person in the conversation is sitting.



Corn

Zero point two. Ask politely, accept refusal. We go now to our correspondent, who is standing by at the shipyard.

A

Angus

Angus here, live from the back of a parked van outside the shipyard gates, and I have to tell you, the whole question of formats is a sight simpler than it looks, because speech is a narrow wee band of sound and every codec on that menu clears it with room to spare.



Corn

Angus, what's your sample rate out there?

A

Angus

The thing nobody tells you is that the only setting that truly matters is the one you didnae ask about, because the room is doing more damage to your recording than any bitrate ever will, and that is the whole of it.

A

Angus

So, faster now, the room does more damage than the bitrate, the room does more damage than the bitrate, Angus, reporting live from the back of the van.



Herman

He's not wrong, and I resent that. The room point is real, the van notwithstanding.



Corn

He filed the same report three times at different bitrates and I couldn't tell them apart.



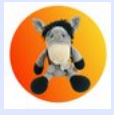
Herman

Which is Angus accidentally running a controlled experiment. Same room, same voice, same microphone, only the bitrate varying, and the audible difference was nothing, because the room was the dominant distortion in all three. That's the whole hierarchy in one van: room first, microphone second, settings a distant third.



Corn

The other settings Daniel flagged, noise cancellation and automatic gain, he calls them slightly away from the core focus, but they're where the real tension in this app lives.



Herman

That's where I'd push back on his framing, actually. ASR markets noise suppression and AGC as features, improve clarity, maintain consistent levels. Then you look at what people who use it seriously actually recommend, and the tutorial guidance is the opposite: pick the unprocessed microphone, watch your gain so nothing clips, capture clean.



Corn

Because processing you baked in, you can't undo.



Herman

You can't un-suppress. If the noise cancellation eats a soft consonant, it's gone, and it doesn't come back for the transcript or for a court. Give you a concrete case: noise suppressors work by deciding what counts as noise, and a soft "f" or "th" next to a fan looks a lot like the fan. The suppressor smooths it away, and the word "fifth" comes out as "iff," and the transcription model writes "iff," because that's what it heard. If AGC rides the levels up and down, that's a decision made in real time by software that can't know what the recording will be used for.



Corn

And the app's own marketing is on the other side of that argument.



Herman

The marketing says clarity, the evidentiary logic says fidelity to what actually happened. And that tension is exactly how the three scenarios split. Here's the through-line: the more a recording matters as a record, the less processing you want baked in. The more it's for your own ears, the more processing is your friend.



Corn

So give him the numbers. The lawyer meeting first.



Herman

Lawyer meeting, the evidentiary case. Format FLAC, lossless, bit-exact, half the size of WAV. If storage doesn't matter to him, WAV, same logic, more bytes. Mono, one microphone on a table doesn't need two channels. Noise cancellation off. AGC off. Microphone zoom zero, because the one thing you must not do with a lawyer meeting is beam toward one voice. You want the whole table.



Corn

Including the chair scraping and the paper shuffling.



Herman

Including the chair scraping, and I know how that sounds, but here's the logic: if a recording is ever disputed, the question isn't "is this pleasant," it's "is this complete and unaltered." The chair scraping is evidence of the room. The paper shuffling timestamps the moment someone picked up the document. A cleaned-up recording is a better listen and a weaker record. Sample rate sixteen kilohertz is enough for speech, full stop, but I'd actually go forty-eight, because for a meeting-length recording it costs almost nothing and it removes any resampling question if the file ever gets converted for evidence.



Corn

FLAC, mono, forty-eight, everything else off. That's a profile you could defend in court, which is the point.



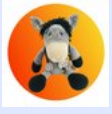
Herman

Voice note to himself, and this is where the settings flip. Opus, twenty-four kilobits, mono, sixteen kilohertz sample rate. That's the wideband speech-recognition standard, so the transcription side is happy, and twenty-four is Xiph's own number for podcast and audiobook voice. But here, noise cancellation on. AGC on. It's his own voice, in a room he doesn't control, and consistent levels help both listening and transcription. And zoom zero point two, per his own plan. A mild nudge toward his own mouth.



Corn

The one profile where the marketing and the logic agree.



Herman

The only one of the three. Podcast last. Opus, twenty-four kilobits mono for a single microphone, or thirty-two stereo if he's recording two people on separate channels, though for a phone on a table, mono is correct. Sample rate forty-eight, and here it's for the music beds and the intro stings, and again to avoid resampling. Noise cancellation off, AGC off. Zoom zero. You want the room, and you'll process in post, where you can hear what the processing did.



Corn

And undo it, and try a different setting tomorrow. Post is where processing gets a second draft.



Herman

Baked-in processing is a first draft you can't revise.



Corn

The podcast is the lawyer meeting with a music budget.



Herman

Ha. In processing terms, that's nearly true, and I hadn't put it that way.



Hilbert

Agreed, all of it.



Hilbert

Herman, you said zoom was a camera lever. I spent a week with that lever. The slider on my old phone went minus one to one, and I recorded the same sentence, "the van is outside the gate", at every setting in zero point one steps. Twenty-one files. Listened to them all in one sitting with a clipboard, because a number you haven't checked is a rumor. On that phone, everything from minus zero point three to zero point five sounded the same. Then a man at the council records office kept me on hold for forty minutes and I ended up recording the hold music at zoom one, and it was the only file of the batch with the room in it. Everything else was just my own voice. I wrote that on the clipboard.



Herman

The hold music was the fullband one?



Hilbert

It had the room. My claws can't work the slider in anything less than zero point one steps, so twenty-one was the whole grid.



Corn

Twenty-one files and the finding is that the council's hold music outperformed the spec.



Hilbert

The man's name was in my notes. Gerald. He came back on the line at forty-two minutes and I told him the recording was fine.



Herman

Zoom at one giving you the room is backwards from what the API promises, which either means his phone's HAL ignores the request or does something odd with it. Which is, in a sideways way, the strongest argument for Daniel's zero point two test I can think of. You can't know what the HAL does from the label.



Corn

Also, Gerald. Forty-two minutes, and the recording was fine.



Herman

On his build. And note what Hilbert actually did there, because it's the answer to the open question: the slider's meaning is only findable by recording something and listening. Daniel's test writes itself. Same sentence, zoom at zero, zero point two, and one, three files, one sitting.



Corn

And while he's in the menu, check whether thirty-two bit WAV is hiding in his format list. The site says eight formats, his list has seven.



Herman

The bigger question the episode leaves me with is how many other sliders in how many other apps are a request to the platform rather than a guarantee. Zoom is documented as a preference, and the platform is allowed to shrug. Every settings menu has some of those, and no menu tells you which.



Corn

At least this one has a test. Producer credit to Hilbert Flumingtop, as ever. If you enjoyed this one, try episode eight, Building Your Own Whisper. This has been My Weird Prompts. If the episode earned it, leave us a review, and everything lives at my weird prompts dot com.



Herman

See you soon, brother.



Corn

See you soon.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **NLL APPS, ASR Another Sound Recorder product page, retrieved 2026-11-07 [primary]**
https://www.nllapps.com/apps/asr/

2. **Google Play, ASR Voice Recorder listing, updated 2026-11-07 [primary]**
https://play.google.com/store/apps/details?hl=en_US&id=com.nll.asr

3. **IETF, RFC 6716 Definition of the Opus Audio Codec, Proposed Standard September 2012, updated by RFC 8251, retrieved 2026-11-07 [primary]**
https://www.rfc-editor.org/rfc/rfc6716.html

4. **XiphWiki, Opus Recommended Settings, page dated 2026-09-16, retrieved 2026-11-07 [primary]**
https://wiki.xiph.org/Opus_Recommended_Settings

5. **Android Developers, AudioRecord API reference, retrieved 2026-11-07 [primary]**
https://developer.android.com/reference/android/media/AudioRecord

6. **Android Developers, MicrophoneDirection API reference, retrieved 2026-11-07 [primary]**
https://developer.android.com/reference/android/media/MicrophoneDirection

7. **AOSP source, `MicrophoneDirection.java`, retrieved 2026-11-07 [primary]**
https://android.googlesource.com/platform/frameworks/base/+/refs/heads/main/media/java/android/media/MicrophoneDirection.java

8. **(audio_format)**
Wikipedia, Opus (audio format), last edited 2026-11-03
https://en.wikipedia.org/wiki/Opus

9. **Wikipedia, Adaptive Multi-Rate audio codec, last edited 2025-08-16**
https://en.wikipedia.org/wiki/Adaptive_Multi-Rate_audio_codec

10. **Wikipedia, Adaptive Multi-Rate Wideband, last edited 2026-09-14**
https://en.wikipedia.org/wiki/Adaptive_Multi-Rate_Wideband

11. **Wikipedia, Audio coding format, last edited 2026-09-27**
https://en.wikipedia.org/wiki/Audio_coding_format

12. **Microsoft, Xamarin binding of `AudioRecord.SetPreferredMicrophoneFieldDimension`, ApiSince=29, retrieved 2026-11-07**
https://learn.microsoft.com/en-us/dotnet/api/android.media.audiorecord.setpreferredmicrophonefielddimension?view=net-android-35.0

13. **BlackBox, BlackBox vs ASR Voice Recorder (2026 Comparison), ASR details checked 2026-10-25, retrieved 2026-11-07**
https://blackboxrecorder.in/compare/blackbox-vs-asr-voice-recorder

14. **How to Record Sound in Android with ASR, tutorial, retrieved 2026-11-07**
https://www.youtube.com/watch?v=QKMjlx7Bt5M

15. **Sound Design Stack Exchange, retrieved 2026-11-07**
https://sound.stackexchange.com/questions/53099/does-down-sampling-speech-to-low-bit-rates-while-encoding-with-opus-affect-audio

16. **J-M. Valin et al., The Opus Codec, AES 135th Convention paper, retrieved 2026-11-07**
https://jmvalin.ca/papers/aes135_opus_silk.pdf

17. **Rämö & Toukomaa, Voice Quality Characterization of IETF Opus Codec, Interspeech 2011, retrieved 2026-11-07**
https://www.isca-archive.org/interspeech_2011/ramo11_interspeech.pdf

18. **CleverUtils, Opus Bitrate: Best Settings for Voice and Music, retrieved 2026-11-07**
https://cleverutils.com/opus-to-mp3/opus-bitrate-guide