Audacious - OLD, PLEASE USE GITHUB DISCUSSIONS/ISSUES - Bug #1218

PipeWire and volume level

July 12, 2023 22:39 - grayich grayich

Status:	Closed	**Start date:**	July 12, 2023
Priority:	Minor	**Due date:**	
Assignee:		**% Done:**	0%
Category:	plugins/pipewire	**Estimated time:**	0.00 hour
Target version:	4.4		
Affects version:	4.3.1		

Description

When outputting via (pulse, alsa, etc.), the volume changes in a linear relationship.

When audio is output via **PipeWire**, the volume changes in a logarithmic relationship.
It's inconvenient, you need to either make it linear like it's always been, or optional.

History

#1 - July 20, 2023 09:49 - Michael Schwendt

There is nothing in the source code of Audacious' pipewire output plugin that confirms your assumption. Yet Pipewire itself may work differently than Pulseaudio or direct ALSA output due to handling per app volume in different ways. More trouble-shooting will be needed.

#2 - July 21, 2023 05:14 - John Lindgren

I suspect this is a valid defect.

ALSA volume levels are typically percentages (0-100%) that translate linearly to decibel levels. It's been that way as long as I can remember, though it beats me if that's documented anywhere. On my system, each 5% up/down translates to a 3 dB change. (Nowadays, alsamixer uses some different, incomprehensible scheme, but that's just at the UI level; the underlying API calls haven't changed.)

On the other hand, the PipeWire documentation for SPA_PROP_channelVolumes says "0.0 is silence, 1.0 is without attenuation". It's exceptionally vague wording, but that suggests to me that the 0-1 range is meant to translate linearly to some reference voltage range. If that's the case, we should probably do some math in the get_volume/set_volume() methods so that the volume level in Audacious still translates to a decibel level.

I'm not interested in PipeWire personally, but if someone else wants to take a stab at a patch, the general rule is: a +20 dB increase means the voltage multiplies by 10x. You can also find a sample calculation in the audio_amplify() function (libaudcore/audio.cc).

#3 - July 21, 2023 05:14 - John Lindgren

- Target version deleted (4.3.1)

- Category set to plugins/pipewire

#4 - August 09, 2023 07:48 - Artem S. Tashkinov

This is what MPV/audio/out/ao_pulse.c (https://github.com/mpv-player/mpv/blob/master/audio/out/ao_pulse.c) uses for PulseAudio:

```
#define VOL_PA2MP(v) ((v) * 100.0 / PA_VOLUME_NORM)
#define VOL_MP2PA(v) lrint((v) * PA_VOLUME_NORM / 100)
```

Not sure if it's relevant, sorry if it's not.

Though for audio/out/ao_pipewire.c (https://github.com/mpv-player/mpv/blob/master/audio/out/ao_pulse.c) it's quite different.

#5 - November 04, 2023 11:58 - Thomas Lange

I have asked the PipeWire developer(s), their answer is that PipeWire uses a linear scale for the volume.

https://gitlab.freedesktop.org/pipewire/pipewire/-/issues/3623

@John: Do you have further questions for them? I'm not sure how to proceed now.

#6 - November 04, 2023 20:05 - John Lindgren

I think we have all the information we need. We just need to implement the decibel to linear translation now.

#7 - December 17, 2023 23:07 - Thomas Lange

Please see/check my pull request:
https://github.com/audacious-media-player/audacious-plugins/pull/146

#8 - December 18, 2023 17:33 - Thomas Lange

- Status changed from New to Closed

- Subject changed from pipewire and volume level to PipeWire and volume level

Fixed with https://github.com/audacious-media-player/audacious-plugins/commit/1f02fb1e277e340c47a74391b2db047570b21a0c.

#9 - January 21, 2024 19:03 - Thomas Lange

- Target version set to 4.4