Audacious - Feature #638

Bypass crossfade settings when playing a file.

April 28, 2016 12:31 - Chris Laurie

Status:	Closed	**Start date:**	April 28, 2016
Priority:	Minor	**Due date:**	
Assignee:		**% Done:**	100%
Category:	audtool	**Estimated time:**	0.00 hour
Target version:	3.8		
Affects version:	3.7.2		

Description

My use case in our community theatre group is this: I play a playlist of music before and after a show and also during interval. For this I use the crossfade and remove silence plugins and it works really well. Once the show starts I now use the same machine and Audacious player to play individual files as sound effects, show music, backing tracks etc - this is done via commandline calls from my lighting control software. The problem is that when I now play these files they have a fade out at the end that matches the crossfade settings.

This is problem because for a show one needs the clips to play at the set volume for their entirety. Currently I try to switch to Audacious to turn crossfade off, then switch back to my other software to start running the show. This can get quite difficult in terms of timing. Sometimes there are a number of silent seconds where the audience gets fidgety (unpopular with critics) or the band starts playing (or other show activity) without the technical operator being ready.

When the show is over I would like to be able to go back to my crossfade settings.

I see a few possible solutions to this:

- Add a command line switch (say -x) when playing a file that means that Audacious will ignore all crossfade settings for playing the particular track.
- Since the show tracks are played on the "Now playing" playlist, saving the settings per playlist would also work. Then I can set-up Audacious according to how I need it to run.
- Add an audtool api that changes the active status of a plug-in. Something like: audtool crossfade-set-active 1.

I suspect the audtool api to be the simplest.

History

#1 - April 28, 2016 14:45 - John Lindgren

Perhaps multiple instances (with separate settings) would be helpful.

#2 - April 28, 2016 15:20 - Chris Laurie

I'm not sure how the multiple instances would work. If it can remember the settings on a per instance then maybe. How would command line scripts and audtool calls know which instance to use as target for the api request?

An api to enable/disable a plug-in at runtime seems a simpler solution.

#3 - April 28, 2016 20:35 - John Lindgren

Something like audacious -2 would start a second instance (which would store its own settings in, say, ~/.config/audacious-2), and audtool -2 <command> would control it remotely. It would actually be fairly simple to implement.

This approach might not have any advantage for your particular needs, but I think it's more generic and could be useful in other scenarios as well. For example, some time ago we had another user, also running sound for a theater, who wanted to run a playlist of background music in one instance, while at the same time playing individual sound effects from a playlist in another instance (using the "No Playlist Advance" setting).

#4 - April 28, 2016 20:54 - Chris Laurie

I can totally see the value of this, If the 2 instances are feeding separate sound outputs. In my case the sound is output on the same device. So the only one should play at a time.

I suggest being able to enable a plug-in through the command line is a very useful functionality anyway, for many other uses, not only my own.

My solution for the playlist no advance described above is to have the lighting software (QLC+) call audacious from the command line to play a track, so it creates a single entry in the now playing playlist. That way I do not have to actually administer the Audacious instance manually, it all happens via the command line. In fact I could run it headless, but I don't because in live theatre it is good to have some insurance.

Being able to control more parameters from the command line also allows me to create and thoroughly test the control scripts and then it becomes easier to give the full production to a less experienced and knowledgeable operator.

#5 - April 29, 2016 01:22 - John Lindgren

Chris Laurie wrote:

> I suggest being able to enable a plug-in through the command line is a very useful functionality anyway, for many other uses, not only my own.

Yes, I agree, provided it worked for all plugins, not just crossfade.

#6 - April 29, 2016 02:52 - John Lindgren

- Category set to audtool

- Status changed from New to Closed

- Target version set to 3.8

- % Done changed from 0 to 100

- Affects version 3.7.2 added