

Audacious - OLD, PLEASE USE GITHUB DISCUSSIONS/ISSUES - Support #514

Suggestions, tips, advice for developing new plugin

March 03, 2015 20:47 - Jim Turner

<b>Status:</b>	Closed	<b>Start date:</b>	March 03, 2015
<b>Priority:</b>	Minor	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>Affects version:</b>			
<b>Description</b>			
Greetings,			
I did this myself years ago to audacious 1.5.1 (when the code was much simpler and smaller), but now would like to add to latest version and submit the final patch for possible inclusion. I've spent some time trying, but to no avail. What I want to do is add a plugin? to allow Audacious to play audio data from STDIN, ie:			
<pre>\$&gt;cat myfile.mp3   audacious <del>E</del>-file://.mp3</pre>			
This example seems trivial, but in certain situations, I need it for something like:			
<pre>\$&gt;cat /tmp/fifo.wav   audacious <del>ncg</del>-file://.wav &amp;     \$&gt;mplayer -idle -slave volume=42 -nojoystick -nolirc \$1 -msglevel all=-1 -benchmark -vc null -vo null -ao pcm:waveheader:file=/tmp/fifo.wav &amp;</pre>			
Thanks!			
 Jim			

History

#1 - March 05, 2015 22:11 - John Lindgren

- Affects version deleted (3.7)

The problem is that Audacious will want to open any input file more than once--otherwise, you could just run "audacious /dev/stdin". In theory, you could create a special transport plugin to split stdin into multiple streams (which would require a lot of caching); but instead of writing a plugin, why not just dump the stream to a temporary file and play that instead?

#2 - March 06, 2015 08:19 - Jim Turner

Well, I went and did it, getting it to work perfectly w/MP3s with essentially 1 line of code in gio.cc (but added a cpl. more to pretty it up! The reason I used "stdin://.wav" instead of /dev/stdin (though it gets converted to that in gio.cc) is that Audacious needs (I think) to know the file type being streamed in w/o trying to "rewind" stdin, and this seems easiest, as I know what I'm piping in. Anyway, here goes:

```
diff gio.cc.org gio.cc
37c37
< static const char * const gio_schemes[] = {"ftp", "sftp", "smb"};
---
static const char * const gio_schemes[] = {"ftp", "sftp", "smb", "stdin"};
```

110a111

```
#ifdef _WINDOWS
```

111a113,116

```
#endif
#ifdef _WINDOWS
m_file = (strstr(filename, "://-.") ? g_file_new_for_path ("/dev/stdin") : g_file_new_for_uri (filename); /* JWT:ADDED TO HANDLE INPUT FROM
stdin */
#endif
```

257a263,264

```
if (strstr(m_filename, "://-.") return 0; /* JWT: CAN'T SEEK ON STDIN!! */
```

HOWEVA!-- The other two (ogg and wav) that I wanted required changes to the input plugins themselves to only "open" the file once and never close it, etc. I had hoped to NOT modify the plugins this time around, but could not figure out how (NOTE, I'm completely wet behind the ears w/C++ pgming!). I've also added some useful (at least to me) command-line options:

```
audacious36hacked --help
```

```
...
-c, --clear      Clear Playlist // open up w/an empty playlist
-D, --delete     DELETE all playlists!
-n, --new        New Instance // force opening a new instance
-P, --pause-mute  Pause mutes instead of pausing
-z, --no-equalizer Force Equalizer Off // useful when streaming talkshow, etc. streams
```

Also added a simple "-" stdin file to pipe in a list of files to play, useful for linuxy-stuff like:

```
$>find ~/Music -name "**dancin*" -print | audacious36hacked -
```

I'm currently testing everything out. Any suggestions on how/where to send my patches for consideration?

Thanks,

Jim

### #3 - March 06, 2015 13:29 - John Lindgren

You can attach patches to this ticket. But it may be that some of the changes for this one use case have undesirable side effects for general use.

### #4 - March 06, 2015 19:39 - Jim Turner

- File *gio\_v3.6jwt.cc.diff* added
- File *sndfile.plugin\_v3.6jwt.cc.diff* added
- File *vorbis\_v3.6jwt.cc.diff* added
- File *gio\_v3.6jwt.cc* added
- File *sndfile.plugin\_v3.6jwt.cc* added
- File *vorbis\_v3.6jwt.cc* added

I totally agree a/b the potential of side-effects since in a relative sense, I'm a Perl programmer by trade and don't really know much a/b what I'm doing here (pbly just enough to be dangerous)! I've already found 1 side-effect in that the Vorbis hack doesn't seem to work w/MPRIS2 enabled - seems to change the order of plugin activation. That's why I'm attempting to reach out for collaboration, so that any changes I can contribute will ultimately be integrated and done right!

The biggest issue I've encountered is that input plugins seem to open the input stream using their own very abstracted methods. I was thinking, after making the simple change in the Transport plugin *gio.cc* (that I chose to use) that worked for mp3 that that was where the functionality of stream IO is handled and abstracted to the input plugins, Is this true? More light is definitely needed!

The changes for this feature (the "input from stdin" option) are all in *gio.c* (see patch above), *sndfile/plugin.cc* (for wav) and *vorbis.cc* for ogg, which I'm including here.

I'll send the core changes for the other features, which were fairly trivial in a separate followup.

### #5 - March 06, 2015 20:07 - Jim Turner

- File *main\_v3.6jwt.cc.diff* added
- File *main\_v3.6jwt.cc* added
- File *runtime\_v3.6jwt.h.diff* added
- File *runtime\_v3.6jwt.h* added
- File *runtime\_v3.6jwt.cc.diff* added
- File *runtime\_v3.6jwt.cc* added
- File *output\_v3.6jwt.cc.diff* added
- File *output\_v3.6jwt.cc* added

These are the patches for all the other features (command-line options and song-file names input via stdin) discussed earlier in the thread, in the Audacious core; Thanks for your consideration.

### #6 - March 07, 2015 06:51 - John Lindgren

Thanks, I'll look at these as I have time. Some initial comments though:

1. I would rather not confuse "pause" and "mute". It might be useful to add a "mute" command, but aren't most systems nowadays set up with a mute button on the keyboard already?

2. Speaking without having looked at the patches yet: running a new instance reliably involves more than might be apparent at first glance. The main issue is to keep the two instances from fighting over the config files, one overwriting the other's changes.

I'm working on some changes to the VFS layer that **might** bring "audacious /dev/stdin" closer to a reality. I'd like to avoid changing individual input plugins if at all possible. Just because from past experience: any additional burden/restriction placed on input plugins means that some obscure plugin won't be updated correctly, and feature X, Y, or Z will be broken as soon as that plugin is used.

#### #7 - March 07, 2015 21:57 - Jim Turner

1) The reason I replace "Pause" with "Mute" in this option vs. a separate "Mute" option is that I use the Winamp skin interface generally and don't know how to add another button to the skins for "mute", but wanted to use the existing "Pause" button. The reason for wanting to "Mute" is that simply muting via the keyboard mutes the sound card itself, whereas my primary goal for this option is this option is to be able to mute just THIS instance of Audacious whilst still having the sound card unmuted so I can temporarily listen to something else (ie. while the "muted" stream is playing commercials, etc.)

2) Good point. This may require further work/testing. (My use of the new instance option was to be able to stream 2 different stations, switching now and then between them, "muting" one while listening to the other, while both continue stream away.

3) Awesome! I totally agree ! It would be ideal if we don't have to modify ANY of the input plugins, but could handle the stdin file-handle within the core. For example, I spent several hours trying to get ".wav" to work thru it's regular plugin ffaudio to no avail due to my inability to get the VFS layer to play nice w/stdin and modified sndfile to grab it. I basically see the main issue to making stdin "input plugin-independent" is the need to avoid closing, re-opening, and attempting to seek on stdin and the likely need to determine the input format solely by extension (provided in my model for handling stdin) vs "by content" which, for some formats, seems to cause opening and reading parts of the stream then rewinding it before reopening it to play. I assume that the "neon" plugin somehow avoids this, since it has to deal w/un-seekable streams?

On a slightly different subject, could you consider adding a [Close] button to the "Playlist Manager" screen? I use the "Afterstep" window manager and it considers it a "transient" window, thus no "window decorations", therefore no easy way to "close" that window! If you want, I can file a separate "bug" for this issue.

Thanks,

Jim

#### #8 - March 08, 2015 03:49 - John Lindgren

Try Git master and see how it works for you with stdin (use "audacious -"). I tested MP3 and Ogg Vorbis files; both seemed to play okay (though the MP3 tag gets ignored). Since a few releases ago, we've already had some buffering magic that allows the input plugins to think we're rewinding the file without actually doing so; I just massaged the code a bit so that we can continue to use that same buffered data when we actually start playing the file.

#### #9 - March 09, 2015 06:53 - Jim Turner

Thanks John for the quick work! Tell me what I'm doing wrong - I started with a clean untar from the current production v3.6 (which I had originally used to build my hacks), replaced the 9 respective source files with the raw versions in Git.master, but could not get them to compile, tried a few things, and gave up. I didn't see any later source trees, so I assumed, since diff-ing the effected files and that matched with the diffs shown. It said "probe-buffer.h" was missing, I found it and tried again, no joy. I haven't used Git before, so I may be missing something else. The 1st error I got (after finding probe-buffer.h on Git.master) was:

```
playlist.cc: In function 'void scan_finish(ScanRequest*)':
playlist.cc:559:81: error: 'struct ScanRequest' has no member named 'file'
if (playlist_playing_playlist && entry_playlist->position && request->file)
^
playlist.cc:560:48: error: 'struct ScanRequest' has no member named 'file'
playback_set_file (std::move (request->file));
^
playlist.cc:560:53: error: 'playback_set_file' was not declared in this scope
```

```
playlist_set_file (std::move (request->file));
```

```
^
```

Failed to compile playlist.cc (lib)!

I then guessed that you had meant "request->filename", after looking at the struct. Then got:

```
playlist.cc: In function 'void scan_finish(ScanRequest*)':
```

```
playlist.cc:560:57: error: 'playlist_set_file' was not declared in this scope
```

```
playlist_set_file (std::move (request->filename));
```

```
^
```

At this point I gave up and posted this.

#### #10 - March 09, 2015 13:14 - John Lindgren

There were more than 9 files updated. You had better use "git clone" rather than trying to update them one by one. See [1] or [2].

1. <http://audacious-media-player.org/developers>

2. <http://redmine.audacious-media-player.org/boards/1/topics/788>

#### #11 - March 09, 2015 20:04 - Jim Turner

Thanks, followed your instructions & got it working as you described. It seems to work fine for ogg and mp3 as you said. wav does not work yet, however. Without an extension on the stdin filehandle, you have it searching thru all the input plugins trying to match by content. It seems that sndfile and vorbis are reading off some of the stream before determining that it's not their format, which then seems to cause ffmpeg to fail when trying wav. After having spent a lot of time with this several years ago and now again recently, it seems to me that we're most likely going to end up having to specify content by extension somehow (ie. "-.ext"), but if you can get it work for at least ogg, mp3, and wav w/o modifying the input plugins, I'm good, but ideally, as far as possible, in a input-format-independent way.

Note: I had also successfully used stdin for double-duty (without extension) for inputting song filenames into the active playlist, ie:

```
$>find . -name "**dancin*" -print | audacious -
```

This was done in my patch to main.cc. Either way, I want to thank you for your time and effort so far in working w/me on this feature! I want to continue to assist in anyway I can on this.

Jim

#### #12 - March 09, 2015 21:05 - John Lindgren

Jim Turner wrote:

Note: I had also successfully used stdin for double-duty (without extension) for inputting song filenames into the active playlist, ie:

```
$>find . -name "**dancin*" -print | audacious -
```

Wouldn't audacious `find . -name "\*\*dancin\*"` be simpler?

I'll look into the issue with .wav.

#### **#13 - March 09, 2015 21:57 - Jim Turner**

Well hot damn, that works!! Thanks! Guess I need to bone up on my shell features too! :D

Jim

#### **#14 - March 10, 2015 05:07 - John Lindgren**

.wav from stdin should be working now. It will play in the sndfile plugin, not ffaudio.

#### **#15 - March 10, 2015 05:30 - John Lindgren**

- *Status changed from New to Closed*

I'm going to go ahead and close this support ticket since the main topic (playing from stdin) is solved.

For the other ideas mentioned:

- Multiple instances would be nice to have in a future version, but more thought needs to be put into how to manage the configuration correctly between them. Separate configuration "profiles" might be the way to go.
- Since most of the other command-line options are duplicates of actions that can already be done when Audacious is running, I wonder if these might be unnecessary with separate configuration profiles. For example, you could keep a separate configuration with only one playlist and the equalizer disabled specifically for streaming talk shows.

#### **#16 - March 10, 2015 22:51 - Thomas Lange**

Jim Turner wrote:

On a slightly different subject, could you consider adding a [Close] button to the "Playlist Manager" screen? I use the "Afterstep" window manager and it considers it a "transient" window, thus no "window decorations", therefore no easy way to "close" that window! If you want, I can file a separate "bug" for this issue.

There is an easy way to close the window: the Escape key. ;)

## #17 - March 12, 2015 23:18 - Jim Turner

1) Separate config files would be fine w/me for these. What did you have in mind, something like an option to specify? ie:

```
$>audacious -c ~/.config/audacious/myspecialconfigfile
```

2) Did you change sndfile/plugin.cc to get .wav to work from stdin or were you able to make it work in the core? Reason I'm asking is that I found and tried an ".m4a" file, which fails via stdin, but works otherwise. I don't really care myself if that works or not, but thought you might if you are aiming to do it properly in core. Both methods use ffmpeg for input, but stdin appears to fail trying to do illegal seeks.

3) I really DO want that "Pause-does-Mute" option though, to be able to mute commercials, etc. in one stream whilst keeping my sound-card unmuted for other uses! - it's a pretty simple hack to zero out the output to the speakers if the user specifies this option - in summary:

```
$>diff output.cc output.hacked.cc
32a33
> #include <stdio.h>
65a67,68
> short iamPaused = 0; /* JWT: ADDED FOR ALLOWING PAUSE TO CONTINUE READING STREAMS, ONLY PAUSING OUTPUT! */
/
>
116,117c119,133
< cop->pause (s_paused);
< vis_runner_start_stop (true, s_paused);
---
> /* JWT: IF PauseMute SET, PAUSE SHOULD JUST SUSPEND OUTPUT
> THIS IS SO WE CAN "MUTE" COMMERCIALS FROM RADIO WHILST PLAYING
> SOMETHING ELSE! THIS WAY, WE CAN RESUME AT THE CURRENT POINT vs
> WHERE WE PAUSED! THIS HACK WILL KEEP THE VISUALIZATION RUNNING
> PERMITTING VISUAL CONFIRMATION THAT THE STREAM IS STILL STREAMING,
> ONLY NO SOUND GOES TO THE SOUND CARD!
>
> */
> if (aud_get_pausemute_mode ())
> {
>     iamPaused = s_paused;
>     vis_runner_start_stop (true, false); /* NOT SURE WHY "s_paused" CAN'T BE USED HERE????! */
> }
> else
> {
>     cop->pause (s_paused);
>     vis_runner_start_stop (true, s_paused);
> }
230c246
< const void * out_data = data.begin ();
---
> void * out_data = data.begin ();
242a259,262
> /* JWT: ZERO-OUT (SILENCE) STREAM GOING TO OUTPUT WHILST PAUSING IF PAUSE-IS-MUTE OPTION SET: */
> if (iamPaused)
>     memset (out_data, '\0', out_bytes_held);
```

Regards,

Jim

## Files

gio_v3.6jwt.cc.diff	488 Bytes	March 06, 2015	Jim Turner
sndfile.plugin_v3.6jwt.cc.diff	2.48 KB	March 06, 2015	Jim Turner
vorbis_v3.6jwt.cc.diff	2.03 KB	March 06, 2015	Jim Turner
gio_v3.6jwt.cc	9.1 KB	March 06, 2015	Jim Turner
sndfile.plugin_v3.6jwt.cc	12 KB	March 06, 2015	Jim Turner
vorbis_v3.6jwt.cc	14.3 KB	March 06, 2015	Jim Turner
main_v3.6jwt.cc.diff	6.38 KB	March 06, 2015	Jim Turner
main_v3.6jwt.cc	14.4 KB	March 06, 2015	Jim Turner
runtime_v3.6jwt.h.diff	124 Bytes	March 06, 2015	Jim Turner
runtime_v3.6jwt.h	3.3 KB	March 06, 2015	Jim Turner
runtime_v3.6jwt.cc.diff	267 Bytes	March 06, 2015	Jim Turner
runtime_v3.6jwt.cc	8.75 KB	March 06, 2015	Jim Turner
output_v3.6jwt.cc.diff	963 Bytes	March 06, 2015	Jim Turner
output_v3.6jwt.cc	14 KB	March 06, 2015	Jim Turner