Audacious - Feature #876

Automatically reconnect when HTTP server closes connection

February 27, 2019 17:06 - CJ Kucera

Status:	Rejected	**Start date:**	February 27, 2019
Priority:	Minor	**Due date:**	
Assignee:		**% Done:**	0%
Category:	plugins/neon	**Estimated time:**	0.00 hour
Target version:			
Affects version:	3.10.1		

Description

I've been on the road for a few weeks and have been listening to music primarily via .m3u playlists which point to HTTP URLs to mp3s which live on my home server. I've noticed that if I pause a track, leave it for sufficiently long, and then hit play again, the song will continue playing for awhile but then stop suddenly and proceed to the next track in the playlist. The error which gets generated is this:

```
ERROR neon.cc:662 [fill_buffer]: &lt;0x7f37d8027430&gt; Error while reading from the network
    ERROR neon.cc:698 [reader]: &lt;0x7f37d8027430&gt; Error while reading from the network. Termi
nating reader thread
    ERROR neon.cc:745 [try_fread]: &lt;0x7f37d8027430&gt; No request to read from, seek gone wrong
?
```

I assume what's happening is that audacious is just using a single HTTP stream, and pausing it results in no further TCP activity for awhile, and the server eventually times out the connection, so Audacious is just left with whatever data has been pre-buffered. What I've taken to doing is just re-playing from the beginning when I get back to the laptop and then seeking to the point at which I'd paused, which lets me continue with the file, but it'd be nice to have pause/play work as expected in this scenario.

I realize this might be difficult - could an HTTP URL be a "live" stream which isn't properly seekable? In my case these MP3s are all just being served from a pretty standard Apache HTTPD, which supports requesting by range - the relevant HTTP headers sent back are:

```
HTTP/1.1 200 OK
    Accept-Ranges: bytes
    Content-Length: 11507449
    Content-Type: audio/mpeg
```

An alternative in this case could be locally caching the entire file and playing from that instead, though I admit that again in cases where the HTTP URL might be a "proper" stream, that would be infeasible, and I'm not sure how you'd distinguish between the two (perhaps via the presence of Content-Length and Accept-Ranges?). On the server logs, it looks like a total of four HTTP requests ends up getting generated, including a couple which result in 206es - I assume at least one of these requests is for the purposes of extracting tags:

```
[27/Feb/2019:11:03:34 -0600] "GET /Electronic/Aphex_Twin/Syro/10-PAPAT4_pineal_mix.mp3 HTTP/1.1" 2
00 10363286
    [27/Feb/2019:11:03:35 -0600] "GET /Electronic/Aphex_Twin/Syro/10-PAPAT4_pineal_mix.mp3 HTTP/1.
1" 206 128
    [27/Feb/2019:11:03:35 -0600] "GET /Electronic/Aphex_Twin/Syro/10-PAPAT4_pineal_mix.mp3 HTTP/1.
1" 200 10363286
    [27/Feb/2019:11:03:36 -0600] "GET /Electronic/Aphex_Twin/Syro/10-PAPAT4_pineal_mix.mp3 HTTP/1.
1" 206 128
```

Anyway, let me know if I can provide any more info, etc! Not really sure what Category this belongs in. Running this on 3.10.1, on Arch.

History

#1 - February 27, 2019 21:09 - John Lindgren

- Category set to plugins/neon

- Subject changed from Pausing HTTP-streamed mp3 for long enough prevents resuming to Automatically reconnect when HTTP server closes connection

- Tracker changed from Bug to Feature

#2 - May 03, 2019 16:20 - Artem S. Tashkinov

I asked for the same thing in the past and it was swiftly rejected: #564

Weird.

#3 - August 03, 2022 18:41 - John Lindgren

- Status changed from New to Rejected

Closing feature requests that have seen no activity in 3 years.