

## Audacious - OLD, PLEASE USE GITHUB DISCUSSIONS/ISSUES - Support #1150

### XSPF playlists shouldn't rely on duration entry

January 28, 2022 02:45 - Luis Ferreira

<b>Status:</b>	Closed	<b>Start date:</b>	January 28, 2022
<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>			
According to XSPF standard:			
A user-agent MUST NOT use this value to determine the rendering duration, since the data will likely be low quality.			
Audacious is currently relying on `duration` to determine the rendering duration, which is wrong.			

#### History

##### #1 - January 28, 2022 19:09 - John Lindgren

I am not aware that any of our decoders pay attention to the XSPF duration. Can you clarify what you mean? What is the issue you are seeing?

##### #2 - March 05, 2022 14:56 - John Lindgren

- Tracker changed from Bug to Support

##### #3 - March 15, 2022 18:31 - Luis Ferreira

John Lindgren wrote:

I am not aware that any of our decoders pay attention to the XSPF duration. Can you clarify what you mean? What is the issue you are seeing?

I generate the playlist by myself, and if don't provide or provide a shorter duration, the music player make it appears shorter. The player doesn't update the real duration of the music, when I hit play but the music apparently play as expected. This issue affects, e.g. the slider, since it is dependent of that metadata.

##### #4 - March 15, 2022 19:54 - John Lindgren

Displaying the XSPF duration in the UI is by design, and doesn't affect the "rendering" (playback) duration. I don't think Audacious is doing anything wrong here.

Why are you generating XSPF playlists with inaccurate duration values, anyway?

##### #5 - March 16, 2022 01:02 - Luis Ferreira

John Lindgren wrote:

Displaying the XSPF duration in the UI is by design, and doesn't affect the "rendering" (playback) duration. I don't think Audacious is doing anything wrong here.

What I mean is that audacious should try to read the file metadata or try to figure out the real duration. The XSPF spec says the value is only a hint and user-agent should not rely on such value for rendering. Look, e.g. at the behaviour of VideoLAN Player (VLC). This also happens to other metadata fields. If a certain music doesn't have author on XSPF, but on the music source, Audacious doesn't show that info.

Why are you generating XSPF playlists with inaccurate duration values, anyway?

As the standard says about that field: data will likely be low quality. In my case, I'm getting the duration from an external metadata source, which can be unreliable and sometimes non-existent. The cases where it is non-existent, I can't go back and forward on the music, even though, the music has metadata about the duration. Some formats can provide a reliable way to calculate duration (without relying on, e.g. exif metadata).

#### **#6 - March 16, 2022 12:47 - John Lindgren**

- *Status changed from New to Closed*

You can avoid the issue by not using playlists with incomplete/inaccurate info. If you want Audacious to read all the info from the files themselves, there is no reason to use XSPF to start with. Just dump the filenames to a simple M3U playlist.

Closing.

#### **#7 - March 16, 2022 14:20 - Luis Ferreira**

John Lindgren wrote:

You can avoid the issue by not using playlists with incomplete/inaccurate info. If you want Audacious to read all the info from the files themselves, there is no reason to use XSPF to start with. Just dump the filenames to a simple M3U playlist.

Closing.

Audacious is literally off-spec. `duration` field is said to be inaccurate, so relaying on it for rendering is not spec conformant. I'm conforming with XSPF, and in the perspective of a generator, point 5. Requirements for XSPF generators, this is off-spec. A generator may have incomplete information too, the spec doesn't enforce those fields to be present. It is the responsibility of the player to render the information accurately as it is played.

The situation is not as simple as generating reliable information. I'm extracting the resources from a 3rd-party that has generic metadata about a certain music in an album and the location URIs are remote addresses. The playlist also has a high number of tracks, so extracting remote metadata is impractical.

Furthermore, the reason to use XSPF, is the fact that XSPF can specify author, title, and a bunch of other fields that, M3U can't. M3U doesn't has a formal spec, so I can't reliably create a portable format, as other players might read it differently. Audacious doesn't even support M3U extensions, so M3U is off the table here.