Audacious - OLD, PLEASE USE GITHUB DISCUSSIONS/ISSUES - Bug #568

Absolute paths in M3U playlists are not converted from ISO-8859-1 on UTF-8 systems

August 19, 2015 01:12 - Larry Hastings

Status:	Closed		**Start date:**	August 19, 2015
Priority:	Minor		**Due date:**	
Assignee:			**% Done:**	100%
Category:	plugins/m3u		**Estimated time:**	0.00 hour
Target version:	3.7			
Affects version:	3.5			

Description

Currently audacious reads and writes M3U playlists using the utf-8 encoding. utf-8 is wonderful, but it's the wrong encoding for M3U files. M3U files were invented on Windows in 1997 and thus use a dumb stupid encoding of CP-1252, an awful Windows-specific thing. I gather that latin-1 (8859-1) is an acceptable, reasonably-compatible substitute.

If I encode my m3u playlists in latin-1, filenames with non-ASCII characters in them are silently ignored.

Using Audacious 3.5 on Ubuntu 15.04 64-bit.

History

#1 - August 19, 2015 04:11 - John Lindgren

Audacious has support for legacy character encodings out-of-the-box. You most likely have it misconfigured somehow. If we actually had no support for Latin-1 as you claim, people would have noticed years ago. Some questions you can answer to help us help you find the real problem:

1. What is the console output?
2. What are your character encoding settings? ISO-8859-1 should be there as a default; did you remove it?

#2 - August 19, 2015 07:22 - Larry Hastings

Thank you for the quick reply!

You're right, Audacious does have support for legacy character encodings, and in many scenarios it works great. But there is a bug here--and I'm happy to report that after some experimentation I can be much more specific.

The bug:

> If an m3u playlist is encoded in latin-1, and an entry in that playlist is an **absolute path** containing non-ASCII characters, Audacious will fail to load that entry.

The entry in the m3u playlist is:

> /home/media/flac/booster/Tangerine Dream - Booster/Ça Va - Ça Marche - Ça Ira Encore (Tangerine Dream).flac

The error message printed to the console is:

> Cannot convert from locale (UTF-8): /home/media/flac/booster/Tangerine Dream - Booster/ Va - Marche - Ira Encore (Tangerine Dream).flac

Notes:

- If I convert the m3u playlist into utf-8, Audacious loads the entry correctly.
- If the entry in the m3u playlist is a **relative** path containing non-ASCII characters encoded in latin-1, Audacious loads the entry correctly.
- If I drag the file into an empty playlist in Audacious, and export the playlist in m3u format, Audacious writes all entries as URL-encoded utf-8 file:/// URLs. Therefore the encoding of the file itself is ASCII, which prevents me from answering the question "what encoding does Audacious

itself prefer".

- For testing purposes, I renamed my "~/.config/audacious" directory to something else, so my configuration should be using all default settings.
- I don't know specifically what "character encoding settings" you mean. But: if I click on the "Playlist" entry in the category list on the left side of the Settings dialog, the text field for "Fallback character encodings:" contains the entry "ISO-8859-1". (m3u playlists appear to be handled with a plugin, but the plugin does not furnish a Settings dialog.)

Meanwhile, I'm happy to work around the problem by converting my m3u playlists to relative paths, which I prefer anyway. Or I could use .m3u8 files which are explicit about their correct encoding.

#3 - August 19, 2015 21:28 - John Lindgren

- Subject changed from audacious uses the wrong encoding when reading/writing M3U playlists to Absolute paths in M3U playlists are not converted from ISO-8859-1 on UTF-8 systems

- Category set to plugins/m3u

Hmm, for absolute paths, we do assume that they are in the local filename character encoding in use on the system (UTF-8 for most modern Linux systems). We could change this, but does it make sense to do so? I'm struggling to come up with a convincing use case. Using absolute paths means your M3U playlist is inherently non-portable, so your old Windows software isn't going to be able to read it anyway.

What software were you using to generate the M3U playlist in the first place? I'm curious why the developers of that software chose to convert the filenames to ISO-8859-1. Did they consider that they are limiting their potential audience to Western Europe and the Americas?

#4 - August 21, 2015 07:14 - John Lindgren

- Status changed from New to Closed

- Target version set to 3.7

- % Done changed from 0 to 100

I pushed a fix for this Git master. If an Latin-1 absolute file name is used on a UTF-8 system, Audacious will first try opening the file name as-is. If that doesn't work, it will then try to convert the file name to UTF-8 (provided ISO-8859-1 is configured as a fallback encoding).

I've tried the following combinations:

System locale	Path from M3U	fopen() attempts
Latin-1	Latin-1 (relative)	Latin-1, UTF-8
Latin-1	Latin-1 (absolute)	Latin-1, UTF-8
Latin-1	UTF-8 (relative)	Latin-1, UTF-8
Latin-1	UTF-8 (absolute)	Latin-1, UTF-8
UTF-8, Latin-1 fallback	Latin-1 (relative)	UTF-8
UTF-8, Latin-1 fallback	Latin-1 (absolute)	Latin-1, UTF-8
UTF-8, Latin-1 fallback	UTF-8 (relative)	UTF-8
UTF-8, Latin-1 fallback	UTF-8 (absolute)	UTF-8
UTF-8, no fallback	Latin-1 (relative)	none
UTF-8, no fallback	Latin-1 (absolute)	Latin-1
UTF-8, no fallback	UTF-8 (relative)	UTF-8
UTF-8, no fallback	UTF-8 (absolute)	UTF-8