The bug can be reproduced with all sorts of unmarked files – in Ogg Vorbis, WAV and MP3 format. The bug causes greater inconvenience as some unmarked files play roughly 9 dB louder than correctly marked files.

Analysis:

From looking at src/libaudcore/output.cc it seems that default_gain (the value entered by the user to amplify unmarked files) is only used when s_gain is true.

The variable s_gain is set to false in the function output_open_audio.

In the function output_set_replay_gain, s_gain is set to true, if s_input is true, i.e. when the input plugin is connected. This seems wrong. This should always be the case when sound can be heard, so under normal circumstances, s_gain will never be false.

I'm a bit puzzled as the ReplayGainInfo structure does not contain a field that tells us whether there is a RG tag at all. :-|

Happy new year!

Thomas

History

#1 - January 04, 2018 02:55 - Thomas Uwe Grüttmüller
Okay, here is the fix.

---------------------------------------------------------------------
audacious-3.9/src/libaudcore/audio.h
1. BEFORE ###
struct ReplayGainInfo {
float track_gain; /* dB */
float track_peak; / 0-1 /
float album_gain; / dB /
float album_peak; / 0-1 */
};
1. AFTER ###
struct ReplayGainInfo {
float track_gain; /* dB */
float track_peak; / 0-1 /
float album_gain; / dB /
float album_peak; / 0-1 /
int track_gain_is_set; / true or false /
int album_gain_is_set; / true or false */
};
---------------------------------------------------------------------
audacious-3.9/src/libaudcore/tuple.cc
1. BEFORE ###
EXPORT ReplayGainInfo Tuple::get_replay_gain () const {
ReplayGainInfo gain {};
[...] return gain;
}
1. AFTER ###
EXPORT ReplayGainInfo Tuple::get_replay_gain () const {
    ReplayGainInfo gain {};
    [...]  
gain.album_gain_is_set = (data->is_set (AlbumGain)) ? 1 : 0;
gain.track_gain_is_set = (data->is_set (TrackGain)) ? 1 : 0;

    return gain;
}

---------------------------------------------------------------------
audacious-3.9/src/libaudcore/output.cc

1. BEFORE ###
static bool s_gain; /* replay gain info set */

1. AFTER ###
// line removed

1. BEFORE ###
static void apply_replay_gain (Index<float> & data) {
    if (! aud_get_bool (0, "enable_replay_gain"))
        return;

    float factor = powf (10, aud_get_double (0, "replay_gain_preamp") / 20);
    if (is_gain) {
        float peak;

        auto mode = (ReplayGainMode) aud_get_int (0, "replay_gain_mode");
        if ((mode == ReplayGainMode::Album) ||
            (mode == ReplayGainMode::Automatic &&
             (! aud_get_bool (0, "shuffle") || aud_get_bool (0, "album_shuffle")))
            && gain_info.album_gain_is_set) {
            factor *= powf (10, gain_info.album_gain / 20);
            peak = gain_info.album_peak;
        } else {
            factor *= powf (10, gain_info.track_gain / 20);
            peak = gain_info.track_peak;
        }

        if (aud_get_bool (0, "enable_clipping_prevention") && peak * factor > 1)
            factor = 1 / peak;
    } else{
            factor *= powf (10, aud_get_double (0, "default_gain") / 20);
    }

    if (factor < 0.99 || factor > 1.01)
        audio_amplify (data.begin (), 1, data.len (), & factor);
}

1. AFTER ###
static void apply_replay_gain (Index<float> & data) {
    if (! aud_get_bool (0, "enable_replay_gain"))
        return;

    float factor = powf (10, aud_get_double (0, "replay_gain_preamp") / 20);
    float peak;
    auto mode = (ReplayGainMode) aud_get_int (0, "replay_gain_mode");
    if ((mode == ReplayGainMode::Album) ||
        (mode == ReplayGainMode::Automatic &&
         (! aud_get_bool (0, "shuffle") || aud_get_bool (0, "album_shuffle"))) &&
        (gain_info.album_gain_is_set)) {
        factor *= powf (10, gain_info.album_gain / 20);
        peak = gain_info.album_peak;
        if (aud_get_bool (0, "enable_clipping_prevention") && peak * factor > 1)
factor = 1 / peak;
} else {
if (gain_info.track_gain_is_set) {
    factor *= powf (10, gain_info.track_gain / 20);
    peak = gain_info.track_peak;
    if (aud_get_bool (0, "enable_clipping_prevention") && peak * factor > 1)
        factor = 1 / peak;
} else
    factor *= powf (10, aud_get_double (0, "default_gain") / 20);
}

if (factor < 0.99 || factor > 1.01)
    audio_amplify (data.begin (), 1, data.len (), & factor);
}

---

1. BEFORE ###
s_gain = s_paused = s_flushed = false;

1. AFTER ###
s_paused = s_flushed = false;

1. BEFORE ###
s_gain = true;

1. AFTER ###
// line removed

---------------------------------------------------------------------
src/vorbis/vorbis.cc

1. BEFORE ###
static bool update_replay_gain (OggVorbis_File * vf, ReplayGainInfo * rg_info) {
    const char *rg_gain, *rg_peak;
    vorbis_comment * comment = ov_comment (vf, -1);
    if (! comment)
        return false;

    rg_gain = vorbis_comment_query(comment, "REPLAYGAIN_ALBUM_GAIN", 0);
    if (!rg_gain) rg_gain = vorbis_comment_query(comment, "RG_AUDIOPHILE", 0); /* Old */
    rg_info->album_gain = (rg_gain != nullptr) ? str_to_double (rg_gain) : 0.0;
    AUDDBG ("Album gain: %s (%f)
", rg_gain, rg_info->album_gain);

    rg_gain = vorbis_comment_query(comment, "REPLAYGAIN_TRACK_GAIN", 0);
    if (!rg_gain) rg_gain = vorbis_comment_query(comment, "RG_RADIO", 0); /* Old */
    rg_info->track_gain = (rg_gain != nullptr) ? str_to_double (rg_gain) : 0.0;
    AUDDBG ("Track gain: %s (%f)
", rg_gain, rg_info->track_gain);

    rg_peak = vorbis_comment_query(comment, "REPLAYGAIN_ALBUM_PEAK", 0);
    rg_info->album_peak = rg_peak != nullptr ? str_to_double (rg_peak) : 0.0;
    AUDDBG ("Album peak: %s (%f)
", rg_peak, rg_info->album_peak);

    rg_peak = vorbis_comment_query(comment, "REPLAYGAIN_TRACK_PEAK", 0);
    if (!rg_peak) rg_peak = vorbis_comment_query(comment, "RG_PEAK", 0); /* Old */
    rg_info->track_peak = rg_peak != nullptr ? str_to_double (rg_peak) : 0.0;
    AUDDBG ("Track peak: %s (%f)
", rg_peak, rg_info->track_peak);

    return true;
}

1. AFTER ###
static bool update_replay_gain (OggVorbis_File * vf, ReplayGainInfo * rg_info) {
    const char *rg_gain, *rg_peak;
    vorbis_comment * comment = ov_comment (vf, -1);
    if (! comment)
        return false;

    rg_gain = vorbis_comment_query(comment, "REPLAYGAIN_ALBUM_GAIN", 0);
    if (!rg_gain) rg_gain = vorbis_comment_query(comment, "RG_AUDIOPHILE", 0); /* Old */
    rg_info->album_gain = (rg_gain != nullptr) ? str_to_double (rg_gain) : 0.0;
AUDDBG ("Album gain: %s (%f)\n", rg_gain, rg_info->album_gain);
rg_info->album_gain_is_set = (rg_gain != nullptr) ? 1 : 0;

rg_gain = vorbis_comment_query(comment, "REPLAYGAIN_TRACK_GAIN", 0);
if (!rg_gain) rg_gain = vorbis_comment_query(comment, "RG_RADIO", 0); /* Old */
rg_info->track_gain = (rg_gain != nullptr) ? str_to_double (rg_gain) : 0.0;
AUDDBG ("Track gain: %s (%f)\n", rg_gain, rg_info->track_gain);
rg_info->track_gain_is_set = (rg_gain != nullptr) ? 1 : 0;

rg_peak = vorbis_comment_query(comment, "REPLAYGAIN_ALBUM_PEAK", 0);
rg_info->album_peak = rg_peak != nullptr ? str_to_double (rg_peak) : 0.0;
AUDDBG ("Album peak: %s (%f)\n", rg_peak, rg_info->album_peak);

rg_peak = vorbis_comment_query(comment, "REPLAYGAIN_TRACK_PEAK", 0);
if (!rg_peak) rg_peak = vorbis_comment_query(comment, "RG_PEAK", 0); /* Old */
rg_info->track_peak = rg_peak != nullptr ? str_to_double (rg_peak) : 0.0;
AUDDBG ("Track peak: %s (%f)\n", rg_peak, rg_info->track_peak);

return true;
}

#2 - January 04, 2018 02:58 - Thomas Uwe Grüttmüller
- File Patch added

Arrgh, the bugtracker has messed up the formatting.

#3 - January 04, 2018 03:43 - John Lindgren
Can you please re-upload the patch in diff -ur format (or simply git diff if you have the repository cloned locally)?

#4 - January 04, 2018 05:01 - Thomas Uwe Grüttmüller
- File audacious_git.diff added
- File audacious-plugins_git.diff added

All right…

#5 - January 04, 2018 06:29 - John Lindgren
- Status changed from New to Closed
- Assignee deleted (Thomas Uwe Grüttmüller)
- Target version set to 3.10
- % Done changed from 0 to 100
Never mind, I committed a slightly different fix that doesn't break the plugin ABI.

Apparently this has been broken for a long time, at least back to 2010 and possibly before. Anyway, thank you for the report.

Files

<table>
<thead>
<tr>
<th>Patch</th>
<th>6.66 KB</th>
<th>January 04, 2018</th>
<th>Thomas Uwe Grüttmüller</th>
</tr>
</thead>
<tbody>
<tr>
<td>audacious_git.diff</td>
<td>3.81 KB</td>
<td>January 04, 2018</td>
<td>Thomas Uwe Grüttmüller</td>
</tr>
<tr>
<td>audacious-plugins_git.diff</td>
<td>1.06 KB</td>
<td>January 04, 2018</td>
<td>Thomas Uwe Grüttmüller</td>
</tr>
</tbody>
</table>