

Packages - Bug #1984

Clementine 1.3.1 crashes when it changes track or suddenly.

2018-09-12 02:34 PM - iacobus

Status:	fixed	% Done:	100%
Priority:	bug		
Assignee:	freemor		
Category:			
Description			
This is the output in the terminal:			
<pre>11:22:37.458 WARN unknown libpng warning: iCCP: known incorrect sRGB pro file 11:22:37.778 WARN unknown QPainter::begin: Paint device returned engine == 0, type: 2 11:22:39.260 ERROR unknown JSonScanner::yylex - error while reading from io device 11:22:39.261 ERROR unknown json_parser - syntax error found, forcing abo rt, Line 1 Column 1 (clementine:14579): GStreamer-CRITICAL **: 11:23:30.108: gst_element_get_static_pad: assertion 'GS T_IS_ELEMENT (element)' failed Violación de segmento (`core' generado)</pre>			

History

#1 - 2018-09-12 04:29 PM - freemor

Can confirm this behaviour:

```
(clementine:2886): GStreamer-CRITICAL **: 13:23:25.799:
Trying to dispose element typefind, but it is in PLAYING instead of the NULL state.
You need to explicitly set elements to the NULL state before
dropping the final reference, to allow them to clean up.
This problem may also be caused by a refcounting bug in the
application or some element.

(clementine:2886): GStreamer-CRITICAL **: 13:23:25.799:
Trying to dispose element decodebin45, but it is in PLAYING instead of the NULL state.
You need to explicitly set elements to the NULL state before
dropping the final reference, to allow them to clean up.
This problem may also be caused by a refcounting bug in the
application or some element.

(clementine:2886): GStreamer-CRITICAL **: 13:23:25.799:
Trying to dispose element uridecodebin-586, but it is in PLAYING instead of the NULL state.
You need to explicitly set elements to the NULL state before
dropping the final reference, to allow them to clean up.
This problem may also be caused by a refcounting bug in the
application or some element.

(clementine:2886): GStreamer-CRITICAL **: 13:23:25.799: gst_element_set_state: assertion 'GST_IS_ELEMENT (elem
ent)' failed

(clementine:2886): GStreamer-CRITICAL **: 13:23:25.799: gst_bin_remove: assertion 'GST_IS_ELEMENT (element)' f
ailed
Segmentation fault (core dumped)
```

x86_64/systemd/fully up-to-date

#2 - 2018-09-12 04:49 PM - freemor

Seeing if it just need a rebuild against recent gst updates....

A rebuild does not fix this problem...

Note: Sometime one has to switch tracks rather aggressively to get the fail.

Scrubbing to close to the end of the song seems to be a more reliable way to trigger this.

#3 - 2019-02-03 06:13 AM - CommodoreCrunch

Oh hey, I'm not the only one having this issue. I seem to have found a fix. The issue was in "remove-nonfree-artwork-and-spotify.patch", at the part where it patches "src/engines/gstenginepipeline.cpp". In the process of deblobbing, it removes an if statement entirely, which I guess puts it into a "transition into the next song" state when there isn't a next song and it gets confused.

Line 673 of the patch file:

```
@@ -923,17 +891,7 @@ GstPadProbeReturn GstEnginePipeline::EventHandoffCallback(GstPad*,
void GstEnginePipeline::SourceDrainedCallback(GstURIDecodeBin* bin,
                                              gpointer self) {
    GstEnginePipeline* instance = reinterpret_cast<GstEnginePipeline*>(self);
-
-    if (instance->has_next_valid_url() &&
-        // I'm not sure why, but calling this when previous track is a local song
-        // and the next track is a Spotify song is buggy: the Spotify song will
-        // not start or with some offset. So just do nothing here: when the song
-        // finished, EndOfStreamReached/TrackEnded will be emitted anyway so
-        // NextItem will be called.
-        !(instance->url_.scheme() != "spotify" &&
-            instance->next_url_.scheme() == "spotify")) {
-        instance->TransitionToNext();
-    }
+    instance->TransitionToNext();
}
```

It should be something like this:

```
@@ -923,16 +891,8 @@ GstPadProbeReturn GstEnginePipeline::EventHandoffCallback(GstPad*,
void GstEnginePipeline::SourceDrainedCallback(GstURIDecodeBin* bin,
                                              gpointer self) {
    GstEnginePipeline* instance = reinterpret_cast<GstEnginePipeline*>(self);
-
-    if (instance->has_next_valid_url() &&
-        // I'm not sure why, but calling this when previous track is a local song
-        // and the next track is a Spotify song is buggy: the Spotify song will
-        // not start or with some offset. So just do nothing here: when the song
-        // finished, EndOfStreamReached/TrackEnded will be emitted anyway so
-        // NextItem will be called.
-        !(instance->url_.scheme() != "spotify" &&
-            instance->next_url_.scheme() == "spotify")) {
-        instance->TransitionToNext();
+    if (instance->has_next_valid_url()) {
+        instance->TransitionToNext();
+    }
}
```

Full modified patch file I used, for reference: <https://ptpb.pw/Arfx>

I was having this problem consistently on multiple Parabola installs, one systemd and one OpenRC. Now I've been listening to music for about an hour, one track at a time, without any of them dying early. As soon as I switch back to the unmodified package, whether from the repo or a copy built myself, that behaviour returns. It ends about five seconds too soon and the track is greyed out, or in some cases Clementine crashes entirely.

#4 - 2019-02-03 06:37 AM - bill-auger

- Assignee set to freemor

that is very good news CommodoreCrunch

could you try aggressively scrubbing around the edge as freemor suggested, or otherwise find any way to make it crash again - i will assign this issue to freemor maybe he can verify and adopt your patch

#5 - 2019-02-03 07:10 AM - CommodoreCrunch

bill-auger wrote:

that is very good news CommodoreCrunch

could you try aggressively scrubbing around the edge as freemor suggested, or otherwise find any way to make it crash again - i will assign this issue to freemor maybe he can verify and adopt your patch

Try as I might, I can't trigger it no matter how much I navigate around, change songs, let songs play though, or anything. Trying the same thing on

the repo package, it happens every time.

#6 - 2019-02-03 09:16 PM - freemor

- *Status changed from open to in progress*

Thanks [CommodoreCrunch](#) looking at it and testing it now.. If all goes well hopefully this will get closed in a short while

#7 - 2019-02-04 12:38 AM - freemor

Definitely looks good pushing x86_64 build
Starting other builds

#8 - 2019-02-04 12:45 AM - freemor

- *% Done changed from 0 to 80*

- *Status changed from in progress to fixed*

#9 - 2019-02-04 12:20 PM - freemor

- *% Done changed from 80 to 100*

All builds now pushed