

Audacious - OLD, PLEASE USE GITHUB DISCUSSIONS/ISSUES - Bug #242

Playlist and equalizer window in classic winamp mode don't get the correct window to be transient for if they are enabled on startup

January 14, 2013 15:13 - Lars Mueller

Status:	Rejected	Start date:	January 14, 2013
Priority:	Minor	Due date:	
Assignee:		% Done:	0%
Category:	plugins/skins	Estimated time:	0.00 hour
Target version:			
Affects version:	3.2.4		
Description			
<p>When starting Audacious using the classic winamp mode with playlist and/or equalizer enabled, playlist and equalizer do not correctly react to minimizing and window switching.</p> <p>The playlist and equalizer windows are published to X11 and the window manager before the main window. That results in an incorrect transient relationship between those windows, which can result in bad minimizing and window switch behavior (e.g. when alt-tabbing through windows, only the main window is raised, not playlist/equalizer, because the window manager cannot know the correct ancestor window for the transient playlist/equalizer).</p> <p>This is either a bug in Audacious or in Gtk. A possible solution on Audacious' side could be to publish the main window to X11 before publishing the playlist and equalizer windows.</p> <p>I also commented on issue http://redmine.audacious-media-player.org/issues/203 (which was closed 2 months ago), including some code information. This issue was reported to Cinnamon here: https://github.com/linuxmint/Cinnamon/issues/1576 (I'm one of the cinnamon developers)</p>			

History

#1 - January 14, 2013 23:44 - John Lindgren

I see two calls to XSetTransientForHint when starting Audacious in the Winamp interface with the playlist and equalizer enabled. What more is Audacious/GTK+ supposed to do?

#2 - January 14, 2013 23:47 - John Lindgren

```
Breakpoint 1, 0x00007ffff37ca850 in XSetTransientForHint ()
    from /usr/lib/libX11.so.6
(gdb) bt
#0  0x00007ffff37ca850 in XSetTransientForHint () from /usr/lib/libX11.so.6
#1  0x00007ffff4d231a0 in g_closure_invoke () from /usr/lib/libgobject-2.0.so.0
#2  0x00007ffff4d345b0 in ?? () from /usr/lib/libgobject-2.0.so.0
#3  0x00007ffff4d3c51c in g_signal_emit_valist ()
    from /usr/lib/libgobject-2.0.so.0
#4  0x00007ffff4d3c6b2 in g_signal_emit () from /usr/lib/libgobject-2.0.so.0
#5  0x00007ffff79f119b in gtk_widget_realize (widget=widget@entry=0x818000)
    at gtkwidget.c:4464
#6  0x00007ffff79fb918 in gtk_window_show (widget=0x818000) at gtkwindow.c:4873
#7  0x00007ffff4d231a0 in g_closure_invoke () from /usr/lib/libgobject-2.0.so.0
#8  0x00007ffff4d33ed3 in ?? () from /usr/lib/libgobject-2.0.so.0
#9  0x00007ffff4d3c51c in g_signal_emit_valist ()
    from /usr/lib/libgobject-2.0.so.0
#10 0x00007ffff4d3c6b2 in g_signal_emit () from /usr/lib/libgobject-2.0.so.0
#11 0x00007ffff79f0739 in gtk_widget_show (widget=0x818000) at gtkwidget.c:4042
#12 gtk_widget_show (widget=0x818000) at gtkwidget.c:4019
#13 0x00007ffff4d231a0 in g_closure_invoke () from /usr/lib/libgobject-2.0.so.0
#14 0x00007ffff4d345b0 in ?? () from /usr/lib/libgobject-2.0.so.0
#15 0x00007ffff4d3c51c in g_signal_emit_valist ()
    from /usr/lib/libgobject-2.0.so.0
#16 0x00007ffff4d3c6b2 in g_signal_emit () from /usr/lib/libgobject-2.0.so.0
#17 0x00007ffff77c46e8 in _gtk_action_emit_activate (action=0x72a440)
    at gtkaction.c:801
#18 0x00007fffe9d1edf5 in skins_init () at plugin.c:129
#19 0x000000000041022f in interface_load (plugin=plugin@entry=0x6a1aa0)
```

```
    at interface.c:45
#20 0x0000000000410745 in iface_plugin_set_current (plugin=0x6a1aa0)
    at interface.c:243
#21 0x000000000041ad2d in start_single (type=7) at plugin-init.c:96
#22 start_plugins (type=7) at plugin-init.c:141
#23 0x000000000040ab71 in init_two (p_argv=0x7fffffffea30, p_argc=
    0x7fffffffea3c) at main.c:491
#24 main (argc=1, argv=0x7fffffffeb68) at main.c:566
```

#3 - January 14, 2013 23:53 - John Lindgren

The above backtrace is missing some levels due to tail-call optimization, but [#12](#) `gtk_widget_show()` is being called to show the main window, which confirms that GTK+ is correctly delaying the calls to `XSetTransientForHint()` until the main window is shown.

#4 - January 15, 2013 00:13 - John Lindgren

Just to be abundantly clear that there is no bug here:

```
xwininfo | grep "Window id" (click on main window): xwininfo: Window id: 0x1c0001c "Audacious"
xprop | grep TRANSIENT (click on playlist window): WM_TRANSIENT_FOR(WINDOW): window id # 0x1c0001c
xprop | grep TRANSIENT (click on equalizer window): WM_TRANSIENT_FOR(WINDOW): window id # 0x1c0001c
```

#5 - January 19, 2013 00:09 - Grant Cloete

I don't really understand the above code or anything, but are you saying that the bug is with the OS, and not with Audacious?

#6 - January 19, 2013 16:04 - John Lindgren

- Status changed from New to Rejected

Grant Cloete wrote:

I don't really understand the above code or anything, but are you saying that the bug is with the OS, and not with Audacious?

The specific bug Lars reported does not exist, as a simple test with `xwininfo` and `xprop` shows. The real bug is almost certainly in the window manager. That is nothing new. Some of the most common window managers such as Compiz, Metacity, and Openbox all have been known to fail with Audacious in various and sundry ways. Audacious is not buggy, it simply requires more of the window manager than most programs and thus tends to expose window manager bugs that are not seen with other programs.