Audacious - OLD, PLEASE USE GITHUB DISCUSSIONS/ISSUES - Bug #975

Segfault/leak on exit with streamtuner enabled

April 22, 2020 06:40 - Jim Turner

Status:	New	**Start date:**	April 22, 2020
Priority:	Minor	**Due date:**	
Assignee:	Ariadne Conill	**% Done:**	0%
Category:	plugins/streamtuner	**Estimated time:**	0.00 hour
Target version:			
Affects version:	4.0.2		

Description

Fresh install of 4/21 GIT (to test proposed patch), ran (control run) pristine b4 making any changes several times, often getting a segfault OR a "string leak" when exiting with radio stream from StreamTuner.iheart playing (and the streamtuner plugin window active and embedded in the main window). Otherwise seem to work as expected. Then applied patch (for unrelated issue), tested, same results. Saved core each time and ran gdb. I've since cleared playlist, re-added a StreamTuner.iheart station (don't remember what the one that failed was). Anyway, got the same backtrace from the cores I looked at (see attached). I'm running Qt, version 5.71, system is 32-bit Linux (uname -a: Linux integra 4.20.12-antix.1-486-smp #1 SMP Mon Feb 25 10:34:04 EET 2019 i686 GNU/Linux)

- I can't seem to reproduce IF I close the StreamTuner plugin and restart, but if I re-enable the StreamTuner plugion, I CAN easily reproduce repeatedly (alternating between either the "string leak" or the "segfault"). Both are included in the attachment.

Normal way to reproduce (StreamTuner loaded): start Audacious from command line, no args, either start playing something or not, and just quickly exit.

Cheers,

Jim

History

#1 - April 22, 2020 13:15 - John Lindgren

- Category set to plugins/streamtuner

- Subject changed from Segfault/leak from 4/21 GIT (see Feature#909) to Segfault/leak on exit with streamtuner enabled

- Affects version 4.0.2 added

- Affects version deleted (4.0.3)

#2 - April 22, 2020 14:00 - John Lindgren

- Assignee set to Ariadne Conill

With streamtuner enabled, I can reproduce this easily. I also saw a segmentation fault when disabling the plugin, with Audacious still running.

I will let Ariadne speak to how easy/difficult this will be to fix. It may make sense to mark streamtuner as still experimental for the 4.0.x series.

Backtrace from the segfault here:

```
Thread 1 "audacious" received signal SIGSEGV, Segmentation fault.
0x0000000000000020 in ?? ()
(gdb) bt
#0  0x0000000000000020 in  ()
#1  0x00007ffff750c547 in QMetaObject::invokeMethod(QObject*, char const*, Qt::ConnectionType, QGenericReturnA
rgument, QGenericArgument, QGenericArgument, QGenericArgument, QGenericArgument, QGenericArgument, QGenericArg
ument, QGenericArgument, QGenericArgument, QGenericArgument, QGenericArgument) () at /usr/lib/x86_64-linux-gnu
/libQt5Core.so.5
#2  0x00007ffff74ab24f in QAbstractItemModel::endResetModel() () at /usr/lib/x86_64-linux-gnu/libQt5Core.so.5
#3  0x00007fffdf76d8a9 in IcecastTunerModel::<lambda(char const*, const Index<char>&)>::operator()(const Index
<char> &, const char *) (buf=..., __closure=<optimized out>) at icecast-model.cc:94
#4  0x00007ffff7f83e58 in std::function<void (char const*, Index<char> const&)>::operator()(char const*, Index
<char> const&) const (__args#1=..., __args#0=<optimized out>, this=0x5555564d2ba8) at /usr/include/c++/9/bits/
std_function.h:683
```

```
#5  send_data(void*) () at vfs_async.cc:56
#6  0x00007ffff7f66d73 in QueuedFuncHelper::run() (this=0x7fffe00485d0) at mainloop.cc:153
#7  QueuedFuncHelper::run() (this=0x7fffe00485d0) at mainloop.cc:145
#8  0x00007ffff7528cf5 in QObject::event(QEvent*) () at /usr/lib/x86_64-linux-gnu/libQt5Core.so.5
#9  0x00007fffee2cca66 in QApplicationPrivate::notify_helper(QObject*, QEvent*) () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
#10 0x00007fffee2d60f0 in QApplication::notify(QObject*, QEvent*) () at /usr/lib/x86_64-linux-gnu/libQt5Widgets.so.5
#11 0x00007ffff74fc93a in QCoreApplication::notifyInternal2(QObject*, QEvent*) () at /usr/lib/x86_64-linux-gnu/libQt5Core.so.5
#12 0x00007ffff74ff5b8 in QCoreApplicationPrivate::sendPostedEvents(QObject*, int, QThreadData*) () at /usr/lib/x86_64-linux-gnu/libQt5Core.so.5
#13 0x00007ffff7554f67 in  () at /usr/lib/x86_64-linux-gnu/libQt5Core.so.5
#14 0x00007ffff7c06fbd in g_main_context_dispatch () at /usr/lib/x86_64-linux-gnu/libglib-2.0.so.0
#15 0x00007ffff7c07240 in  () at /usr/lib/x86_64-linux-gnu/libglib-2.0.so.0
#16 0x00007ffff7c072e3 in g_main_context_iteration () at /usr/lib/x86_64-linux-gnu/libglib-2.0.so.0
#17 0x00007ffff7554565 in QEventDispatcherGlib::processEvents(QFlags<QEventLoop::ProcessEventsFlag>) () at /usr/lib/x86_64-linux-gnu/libQt5Core.so.5
#18 0x00007ffff74fb4db in QEventLoop::exec(QFlags<QEventLoop::ProcessEventsFlag>) () at /usr/lib/x86_64-linux-gnu/libQt5Core.so.5
#19 0x00007ffff7503246 in QCoreApplication::exec() () at /usr/lib/x86_64-linux-gnu/libQt5Core.so.5
#20 0x00007ffff7f6639e in interface_run() () at interface.cc:162
#21 0x00007ffff7f7db26 in aud_run() () at runtime.cc:321
#22 0x000055555556020d in main(int, char**) (argc=<optimized out>, argv=<optimized out>) at main.cc:393
```

#3 - April 22, 2020 14:03 - Ariadne Conill

It should not be that difficult to fix if I port the network accesses to use QNetworkAccessManager. The problem is that we need to cancel the VFS requests, but there is no support for cancellation. I can do it this weekend.

#4 - April 22, 2020 16:11 - John Lindgren

Awesome, thanks. I'll plan on tagging 4.0.3 mid-next week and pulling in the fix.

#5 - April 29, 2020 22:10 - Ariadne Conill

Sorry, some shit went down and wasn't able to do it last weekend. Should have it for 4.0.4 though.

#6 - May 01, 2020 00:17 - John Lindgren

No worries. Completely up to you whether you want to slip it into 4.0.4 or wait for 4.1. I keep hoping each 4.0.x will be the last ...

Files

auddebug_data.txt	975 Bytes	April 22, 2020	Jim Turner
config.log	40.2 KB	April 22, 2020	Jim Turner