QGIS Application - Bug report #1548

Python gui.so links to wrong QGIS libs when more than one version of QGIS is installed

2009-02-18 12:05 AM - Tim Sutton

Status:	Closed		
Priority:	Low		
Assignee:	Martin Dobias		
Category:	Build/Install		
Affected QGIS version:		**Regression?:**	No
Operating System:	Linux	**Easy fix?:**	No
Pull Request or Patch supplied:		**Resolution:**	invalid
Crashes QGIS or corrupts data:		**Copied to github as #:**	11608

Description

Scenario:

I install my development builds to /home/timlinux/apps/

Since we adopted versioned lib names (e.g. libqgis_core.so.1.1) it should be possible to have more than one version of QGIS coexsisting in the same CMAKE_INSTALL_PREFIX. This does indeed work and all libs and apps link to the correct libs except the python generated libs under /share/qgis/python/qgis/core.so and <prefix>/share/qgis/python/qgis/gui.so, as the listings that follow show:

```
<pre>
 ldd gui.so | grep qgis
    libqgis_core.so.1.1 => /home/timlinux/apps/lib/libqgis_core.so.1.1 (0x00007f598b893000)
    libqgis_gui.so.1.0 => /home/timlinux/apps/lib/libqgis_gui.so.1.0 (0x00007f598aa47000)
    libqgis_core.so.1.0 => /home/timlinux/apps/lib/libqgis_core.so.1.0 (0x00007f5983624000)
</pre>
```

and

```
<pre>
[qgis] ldd core.so | grep qgis
    libqgis_core.so.1.0 => /home/timlinux/apps/lib/libqgis_core.so.1.0 (0x00007fbce8b98000)
</pre>
```

You can see some strange things are going on here.

- Firstly they should all be linking to 1.1 libs since this build is a 1.1 build.
- Secondly, instead of being called core.so, these libs should be core.so.1.0 and core.so.1.1 libs so that the appropriate python libs are loaded based on the version of QGIS running (1.0 or 1.1)
- thirdly gui.so seems to be linked to both 1.0 and 1.1 of libqgis_core

It will be great if you can resolve this so that the core and gui libs are versioned and linking to the appropriate version of qgis core and gui libs.

Many thanks

Tim

History

#1 - 2009-02-18 02:37 AM - Martin Dobias

Hi Tim,

⍰
installing two version of qgis in one install prefix doesn't make much sense to me. The "sonames" we're using for libraries are meant to help the compatibility, e.g. if there's an app requiring library with soname 1.1 and you have the library in version 1.3.2 with soname 1.1, the linker knows this will work.

⍰
While this allows different versions of libraries to coexist in one path, we must think also of other parts of QGIS: binary "qgis" is not versioned, thus overwritten with a newer install, plugins are not versioned (overwritten too), resources can exist also only in one version... The same is for python bindings.

⍰
But imagine that it would be possible to have pyqgis modules core.so.1.0 and core.so.1.1 in the same path. When you open python console and type: import qgis.core - how should the interpreter know which module are you asking for? To my knowledge, python fails to load the module if you try to rename it from core.so to core.so.X.Y.

⍰
The problem with the linking (pyqgis modules linking to both 1.0 and 1.1 libs) is surely caused by the mixture of two different builds in your build directory. Doing a clean compilation will fix the problem. I don't know how to force the linker to choose library with right soname. As far as I understand it, the linker checks for soname of the shared object you're linking against and then dynamic linker loads the library with correct soname.

⍰
I hope I have convinced you that keeping two different versions of QGIS in one path is a bad idea and this ticket is invalid :-)

⍰
Martin

#2 - 2009-02-18 01:14 PM - Tim Sutton

Hi Martin

⍰
Reading back to my original post, I see I didnt explain something well. I am not trying to **run** two concurrent versions from the same location (as you say some things obviously wont work in this context like the fact that the QGIS binary is being overwritten). I am however trying to **install** two versions to the same location (using only the last installed at any time though). Let me explain better:

I have two QGIS source directories, one for trunk and one for stable. Each one has its own build dir. And both install to the same prefix. When I want to work with QGIS 1.0.x I run a make install in the stable source dir build directory and then run QGIS. Same thing for QGIS 1.1. I have followed this practice for some time without issues. Recently things have changed - I guess since we started 1.1 lib name versioning and the possibility arrived for linking to the two libs.

"

But imagine that it would be possible to have pyqgis modules core.so.1.0 and core.so.1.1 in the same path. When you open python console and type: import qgis.core - how should the interpreter know which module are you asking for?"

My expectation was that if core.so is linked to only one qgis_core lib, the interpreter should use the one linked to.

"To my knowledge, python fails to load the module if you try to rename it from core.so to core.so.X.Y. "

Ok just having gui.so and core.so link to only one lib would be a satisfactory solution here.

"The problem with the linking (pyqgis modules linking to both 1.0 and 1.1 libs) is surely caused by the mixture of two different builds in your build directory. Doing a clean compilation will fix the problem. I don't know how to force the linker to choose library with right soname. As far as I understand it, the linker checks for soname of the shared object you're linking against and then dynamic linker loads the library with correct soname. "

This is the heart of the matter. Note that I am working with a clean build directory i.e. it contains no old object files from pre 1.1. For some reason in recent weeks something has changed such that overwriting installs between two different QGIS version causes odd linking in the python generated libs.

I appreciate that my useage is an edge case however as I mentioned above, my approach used to work fine until the last 3 or 4 weeks (guessing on timeframe here). I just deleted my build dir in trunk sources and built again into my standard prefix to double / triple check and I still get the same cross linking side effect:

```
[qgis] ldd gui.so  | grep qgis
   libqgis_core.so.1.1 => /home/timlinux/apps/lib/libqgis_core.so.1.1 (0x00007feb02f1a000)
```

```
    libqgis_gui.so.1.0 => /home/timlinux/apps/lib/libqgis_gui.so.1.0 (0x00007feb020ce000)
    libqgis_core.so.1.0 => /home/timlinux/apps/lib/libqgis_core.so.1.0 (0x00007feafacaa000)
```

As such I am not ready to close this bug as invalid at this point as there is definately something odd going on...

Best regards

Tim

#3 - 2009-02-18 02:28 PM - Jürgen Fischer

Does the following patch help?

```
Index: CMakeLists.txt
===================================================================
--- CMakeLists.txt    (revision 10172)
+++ CMakeLists.txt    (working copy)
@@ -319,6 +319,7 @@
 INSTALL(FILES ${CMAKE_BINARY_DIR}/qgsconfig.h DESTINATION ${QGIS_INCLUDE_DIR})

 INCLUDE_DIRECTORIES(${CMAKE_BINARY_DIR})
+LINK_LIBRARIES(${CMAKE_BINARY_DIR}/src/core ${CMAKE_BINARY_DIR}/src/gui)

 #############################################################
 # create qgssvnversion.h
```

I applied this to both Version_1.0 and trunk and your problem doesn't occur, but I didn't try to reproduce the problem without it.

#4 - 2009-02-18 03:19 PM - Martin Dobias

Replying to [comment:2 timlinux]:

> *I have two QGIS source directories, one for trunk and one for stable. Each one has its own build dir. And both install to the same prefix. When I want to work with QGIS 1.0.x I run a make install in the stable source dir build directory and then run QGIS. Same thing for QGIS 1.1. I have followed this practice for some time without issues. Recently things have changed - I guess since we started 1.1 lib name versioning and the possibility arrived for linking to the two libs.*

Ah ok, now I go it :-)

> *''*
> *But imagine that it would be possible to have pyqgis modules core.so.1.0 and core.so.1.1 in the same path. When you open python console and type: import qgis.core - how should the interpreter know which module are you asking for?''*
>
> *My expectation was that if core.so is linked to only one qgis_core lib, the interpreter should use the one linked to.*

That's correct expectation.

> *I appreciate that my useage is an edge case however as I mentioned above, my approach used to work fine until the last 3 or 4 weeks (guessing on timeframe here). I just deleted my build dir in trunk sources and built again into my standard prefix to double / triple check and I still get the same cross linking side effect:*

```
> [qgis] ldd gui.so  | grep qgis
>     libqgis_core.so.1.1 => /home/timlinux/apps/lib/libqgis_core.so.1.1 (0x00007feb02f1a000)
>     libqgis_gui.so.1.0 => /home/timlinux/apps/lib/libqgis_gui.so.1.0 (0x00007feb020ce000)
>     libqgis_core.so.1.0 => /home/timlinux/apps/lib/libqgis_core.so.1.0 (0x00007feafacaa000)
```

Ok so the linker picks for some reason the wrong libs. I don't have an advice for you what exactly to do. My ideas are: go to (build_dir)/python/core and (build_dir)/python/gui and check the generated makefiles - especially the LIBS variable to see what paths are used when searching for libraries.

If you won't be able to find the reason, we can take a look at it on the hackfeast.

Martin

#5 - 2009-02-19 03:49 PM - Tim Sutton

Jefs patch above (though using LINK_DIRECTORIES instead of LINK_LIBRARIES) partially fixes the issue in that the core and gui libs are no longer double linked to two different versions of QGISCORE/QGISGUI. However on starting QGIS I still get an error

```
Couldn't load [[PyQGIS]].
Python support will be disabled.

Traceback (most recent call last):
  File "", line 1, in
[[ImportError]]: No module named qgis.core

Python version:
2.5.2 (commit:c5cd2930 (SVN r253):60911, Oct  5 2008, 19:49:54)
[GCC 4.3.2]

Python path:
['/python', '/home/timlinux/.qgis//python/plugins', '/python/plugins', '/usr/lib/python2.5', '/usr/lib/python2.5/plat-linux2', '/usr/lib/python2.5/lib-tk',
'/usr/lib/python2.5/lib-dynload', '/usr/local/lib/python2.5/site-packages', '/usr/lib/python2.5/site-packages',
'/usr/lib/python2.5/site-packages/Numeric', '/usr/lib/python2.5/site-packages/PIL', '/usr/lib/python2.5/site-packages/gst-0.10',
'/var/lib/python-support/python2.5', '/usr/lib/python2.5/site-packages/gtk-2.0', '/var/lib/python-support/python2.5/gtk-2.0',
'/usr/lib/python2.5/site-packages/wx-2.6-gtk2-unicode']
```

#6 - 2009-02-23 12:07 PM - Martin Dobias

This will be caused by C++ plugins for older version of QGIS. I still don't know why exactly this happens, but if you leave in (prefix)/lib/qgis some plugins for older version of QGIS (i.e. they link to some other version of qgis_core and qgis_gui libs), [[QgsApplication]] returns empty strings instead of returning correct paths. You can see from you error message that the python path is set to directories like /python or /python/plugins which simply don't include the prefix.

You can work this around by checking the plugins in the install prefix and delete those for the older version (e.g. gridmaker?).

Martin

#7 - 2009-07-16 01:48 AM - Paolo Cavallini

I reclass this, as it does not seem critical. Please change it again if I was wrong

#8 - 2009-11-07 03:51 PM - Paolo Cavallini

- Resolution set to invalid

- Status changed from Open to Closed