

QGIS Application - Bug report #768

A few lib paths wrong in OSX

2007-09-26 05:30 PM - William Kyngesburye

Status:	Closed	
Priority:	Low	
Assignee:	nobody -	
Category:	Build/Install	
Affected QGIS version:		Regression?: No
Operating System:	OS X	Easy fix?: No
Pull Request or Patch supplied:		Resolution: fixed
Crashes QGIS or corrupts data:		Copied to github as #: 10827
Description		
<p>msexport, qgis_help and the qgis python libs (core.so and gui.so) link to the source copies of libqgis_core and libqgis_gui. Probably an install_name_tool step missed?</p> <p>The python qgis libs' linking is a bit tricky, and @executable_path is probably not a good method to use. It would have to be relative to the python executable in /Library/Frameworks which is outside the Qgis bundle, which would make Qgis.app non-relocatable. @loader_path would be better (ie @loader_path/../../lib/libqgis_core.dylib) - it's relative to the library or program loading the library referenced. BUT, this is a Tiger-only feature, so this would make Python Qgis features Tiger+ -only.</p> <p>AND, the Qgis plugins would then also have to use @loader_path. Thus making all of Qgis Tiger+ -only. For a basic Qgis build, with all Qt frameworks and [[PyQt]]/SIP <i>outside</i> the Qgis bundle, here's what I came up with:</p> <ul style="list-style-type: none">- bin<ul style="list-style-type: none">- msexport.app/Contents/MacOS/msexport<ul style="list-style-type: none">- qgis_core -> @executable_path/../../lib/- qgis_gui -> @executable_path/../../lib/- qgis_help.app/Contents/MacOS/qgis_help<ul style="list-style-type: none">- qgis_core -> @executable_path/../../lib/- lib<ul style="list-style-type: none">- libqgis_gui<ul style="list-style-type: none">- qgis_core -> @loader_path/- libqgisgrass<ul style="list-style-type: none">- qgis_core -> @loader_path/- qgis [plugins]<ul style="list-style-type: none">- all<ul style="list-style-type: none">- qgis_core -> @loader_path/..- qgis_gui -> @loader_path/..- qgis [app]<ul style="list-style-type: none">- qgis_core -> @executable_path/lib/- qgis_gui -> @executable_path/lib/- share/qgis/python/qgis<ul style="list-style-type: none">- core, gui<ul style="list-style-type: none">- qgis_core, qgis_gui -> @loader_path/../../lib/ <p>Executables (qgis, msexport, qgis_help) should use @executable_path. Libraries should use @loader_path.</p>		

History

#1 - 2008-01-31 09:26 AM - pcavallini -

has this been fixed? please check and close it if appropriate

#2 - 2008-03-28 09:55 AM - John Tull

This has not been fixed.

#3 - 2008-06-20 08:55 AM - William Kyngesburye

As reported recently on the mailing list, this seems to be affecting the operation of some (all?) python plugins.

Other info - it looks like the `@loader_path` method is not required. Fixing the python core.so and gui.so to use `@executable_path` (as the rest of Qgis currently uses) works.

#4 - 2008-06-20 09:05 AM - John Tull

Here is the workaround for the python plugin failure from the dev list. This manual fix can be applied by those who are looking for a short-term fix. This workaround was provided by William (kyngchaos), and it may help whoever works on the code:

The paths all depend on where you have your source and where you install Qgis, so I won't give explicit paths here. After installing, open the qgis app package (right-click, Show Package Contents). Dig into Contents/MacOS/share/qgis/python/qgis to find core.so and gui.so. (I do this to easily get the path so I don't have to type it in the Terminal)

In a Terminal, type

```
otool -L
```

(that's a space after the L) and drag gui.so to the Terminal window. Press return.

Towards the top should be a line for `libqgis_gui.0.11.dylib` and a line for `libqgis_core.0.11.dylib`. Type:

```
sudo install_name_tool -change [/path/to/libqgis_gui.dylib] @executable_path/lib/libqgis_gui.dylib -change [/path/to/libqgis_core.dylib] @executable_path/lib/libqgis_core.dylib [drag gui.so to Terminal]
```

for the `/path/to/libqgis_*.dylib` above, copy and paste what you found for the full path to each library in `otool -L`.

For example **this is just for mine** :

```
sudo install_name_tool -change /Users/Shared/src/qgis/svn/qgis_r8652/build-leo/src/gui/libqgis_gui.0.11.dylib @executable_path/lib/libqgis_gui.dylib -change /Users/Shared/src/qgis/svn/qgis_r8652/build-leo/src/core/libqgis_core.0.11.dylib @executable_path/lib/libqgis_core.dylib /Applications/qgis0.11.0.app/Contents/MacOS/share/qgis/python/qgis/gui.so
```

Do the same for core.so. You can just up-arrow to get the previous command and change gui.so to core.so at the very end of the command. It doesn't matter that core.so doesn't have `libqgis_gui.dylib` in it, it will ignore that.

Run `otool -L` on them again to verify the change.

#5 - 2008-10-10 01:31 PM - Tim Sutton

Tom / William can you offer a permanent fix for this or can we otherwise close this somehow.

Regards

Tim

#6 - 2008-10-11 09:31 AM - William Kyngesburye

I tried to make sense of the cmake setup. The progress output is no help, even with verbose.

It looks like src/helpviewer/cmake_install.cmake is where the install_name_tool step happens for help_viewer, and it looks right. So maybe it's skipping the install_name_tool step for some reason?

It looks like python/cmake_install.cmake is where the python qgis modules are installed, but I see no install_name_tool steps there.

#7 - 2008-10-11 09:51 AM - William Kyngesburye

... And the cmake_install.cmake's are generated from cmakelists, of course...

for the python modules, which are missing the instal_name_tool... as an example, the main qgis gui library, it has:

```
INSTALL(TARGETS qgis_gui
  RUNTIME DESTINATION ${QGIS_BIN_DIR}
  LIBRARY DESTINATION ${QGIS_LIB_DIR}
  ARCHIVE DESTINATION ${QGIS_LIB_DIR})
```

I'm guessing that the cmake INSTALL LIBRARY and RUNTIME keywords tell cmake to generate the install_name_tool steps on OSX. The python cmakelists.txt only uses the FILES keyword:

```
INSTALL(FILES +init+.py ${CMAKE_CURRENT_BINARY_DIR}/qgisconfig.py ${BINDINGS_LIBS} DESTINATION ${SITE_PKG_PATH}/qgis)
```

#8 - 2009-01-31 02:31 AM - Paolo Cavallini

Has this been fixed in 1.0? If so, please close it.

#9 - 2009-01-31 09:32 AM - William Kyngesburye

qgis_help is OK now.

But the Qgis Python libs are still a problem.

No word from Tom - I wonder if he's seen this? I added him to the CC for the ticket just in case.

#10 - 2009-02-08 04:40 PM - William Kyngesburye

Correction - help_viewer now has a different linking problem. it links @executable_path/lib/libqgis_core.1.0.dylib, and since it's not in [[MacOS]] folder, but in [[MacOS]]/bin/qgis_help.app..., it doesn't work.

The relative path here should be @executable_path/../../../../lib/libqgis_core.1.0.dylib.

#11 - 2009-07-15 09:09 AM - William Kyngesburye

I have no idea how to *fix* this in the cmake files. But now that there is the Xcode (in trunk only) project that's working out well, the cmake build is not critical to me. For 1.0.x releases I can post-process the build.

If someone can figure this out, great.

#12 - 2009-07-15 09:34 AM - William Kyngesburye

Ah, see part of patch submitted in #1658.

#13 - 2009-09-04 06:20 PM - William Kyngesburye

As I noted in #1658, the install_name_tool step should be done at install like the rest of Qgis, so it can't happen in the python compilation.

#14 - 2009-09-04 07:48 PM - William Kyngesburye

- *Resolution set to fixed*
- *Status changed from Open to Closed*

Ooh. The key is the BUILD_WITH_INSTALL_RPATH cmake target property. That was easy. This setting shouldn't be a problem with an OSX build because the application is not executable in the source until assembled in the application package.

Fixed in trunk and 1.0