QGIS Application - Bug report #1950
Python binary choice on OS X (10.6) / Building bug

2009-09-24 10:32 AM - vince -

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:**	invalid
Crashes QGIS or corrupts data:		**Copied to github as #:**	12010

Description

As usual, and after a summer break, I'm back on porting Qgis to Macports. I selected the new release 1.3.0.

First, the build fails in spatialite.c. It appears that OS X 10.6, or my configuration, does not like the call to "locale_charset ()", which results in an undefined symbol at link time. However, bypassing the #ifdef APPLE, thus doing the same as LINUX/UNIX, works fine.

Next, I would like to know how to override PYTHON_LIBRARY; I've tried to force it with a suitable Cmake define, but it results as a mess, and I don't understand exactly with what I should set it (a path? a flag?). This relates to the next inquiry: how can I force Qgis to use a specific Python binary? When I launch it from the Finder, it chooses system Python, which causes an incompatibility; when I launch it within a terminal window with a correct PATH env variable, it works.

Otherwise, I don't get the Grass plugin, but maybe it's because I lack gdal-grass.

Thanks

History

#1 - 2009-09-24 11:35 AM - William Kyngesburye

> *First, the build fails in spatialite.c. It appears that OS X 10.6, or my configuration, does not like the call to "locale_charset ()", which results in an undefined symbol at link time. However, bypassing the #ifdef APPLE, thus doing the same as LINUX/UNIX, works fine.*

Works fine in the Xcode project build. locale_charset() is a part of the iconv in OS X. Maybe there are variations of libiconv (OSX uses GNU iconv) and you're using a different one than provided by Apple that uses the linux method? In any case, this is really an issue with spatialite, and is not something that can be fixed in Qgis.

> *Next, I would like to know how to override PYTHON_LIBRARY; I've tried to force it with a suitable Cmake define, but it results as a mess, and I don't understand exactly with what I should set it (a path? a flag?). This relates to the next inquiry: how can I force Qgis to use a specific Python binary?*

Qgis detects Python from your PATH, so add your prefered python bin folder to the front of your PATH before configuring. Also, Qgis uses an internal python interpreter and links directly to the python framework, so it happens at linking and you can't change it at runtime.

> *Otherwise, I don't get the Grass plugin, but maybe it's because I lack gdal-grass.*

Qgis uses the GRASS libraries directly, not thru GDAL. Maybe it's not autodetecting GRASS? On OSX, cmake defaults to looking for an application build of GRASS in a default install path, so if you have a Unix build or have customized the install location, you must specify the GRASS_INCLUDE_DIR and

GRASS_PREFIX when configuring Qgis.

#2 - 2009-09-24 12:40 PM - vince -

Thanks for that quick answer.

Replying to [comment:1 kyngchaos]:

> *Works fine in the Xcode project build. locale_charset() is a part of the iconv in OS X. Maybe there are variations of libiconv (OSX uses GNU iconv) and you're using a different one than provided by Apple that uses the linux method? In any case, this is really an issue with spatialite, and is not something that can be fixed in Qgis.*

Ok. [[MacPorts]] uses its own libiconv. That's why the link fails. I tried to link against system libiconv, but it raises other errors, so the patch is fine and justified (adapt a build to Macports).

> *Qgis detects Python from your PATH, so add your prefered python bin folder to the front of your PATH before configuring. Also, Qgis uses an internal python interpreter and links directly to the python framework, so it happens at linking and you can't change it at runtime.*

Okay, so let me put it another way: I have two Python frameworks. One is the Apple default, the other is the new 2.6.2 Python release compiled under Macports and located elsewhere. How can I force Qgis to link with the latter rather than the former?

> *Qgis uses the GRASS libraries directly, not thru GDAL. Maybe it's not autodetecting GRASS? On OSX, cmake defaults to looking for an application build of GRASS in a default install path, so if you have a Unix build or have customized the install location, you must specify the GRASS_INCLUDE_DIR and GRASS_PREFIX when configuring Qgis.*

That's what I did, and it is fine:

```
--->  Configuring qgis
-- The C compiler identification is GNU
-- The CXX compiler identification is GNU
-- Check for working C compiler: /usr/bin/gcc
-- Check for working C compiler: /usr/bin/gcc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/g++
-- Check for working CXX compiler: /usr/bin/g++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Found Iconv: /usr/pkg/lib/libiconv.dylib
-- Found Proj: /usr/pkg/lib/libproj.dylib
-- Found Expat: /usr/pkg/lib/libexpat.dylib
-- Using GSL from /usr/pkg
-- Found GEOS: /usr/pkg/lib/libgeos_c.dylib
-- Found GDAL: /usr/pkg/lib/libgdal.dylib
-- Found [[PostgreSQL]]: /usr/pkg/lib/postgresql84/libpq.dylib
-- Found GRASS: /usr/pkg/apps/GRASS-6.4.app/Contents/MacOS (6.4.0svn)
-- Looking for openpty
-- Looking for openpty - found
-- Found [[PythonLibs]]: -framework Python
-- Found [[PythonInterp]]: /usr/pkg/bin/python2.6
-- Python libraries found
-- Python bindings enabled
-- Looking for Q_WS_X11
-- Looking for Q_WS_X11 - not found.
-- Looking for Q_WS_WIN
-- Looking for Q_WS_WIN - not found.
-- Looking for Q_WS_QWS
-- Looking for Q_WS_QWS - not found.
-- Looking for Q_WS_MAC
-- Looking for Q_WS_MAC - found
-- Looking for QT_MAC_USE_COCOA
-- Looking for QT_MAC_USE_COCOA - found
-- Found Qt-Version 4.5.2
-- Configuring done
```

but it does not seem to show up in the extensions. The libqgisgrass.dylib are present, BTW.

#3 - 2009-09-24 01:29 PM - vince -

> > *Otherwise, I don't get the Grass plugin, but maybe it's because I lack gdal-grass.*
>
> *Qgis uses the GRASS libraries directly, not thru GDAL. Maybe it's not autodetecting GRASS? On OSX, cmake defaults to looking for an application build of GRASS in a default install path, so if you have a Unix build or have customized the install location, you must specify the GRASS_INCLUDE_DIR and GRASS_PREFIX when configuring Qgis.*

Ok, I've got that. The LC_ID_DYLIB field in the various Grass libs is wrong (points towards the temporary build directory) so that any further lib or executable built against them searches in a non-existent directory (and not in the directory they actually stay). I'll fix that in my grass port.

#4 - 2009-09-24 05:34 PM - William Kyngesburye

> *Okay, so let me put it another way: I have two Python frameworks. One is the Apple default, the other is the new 2.6.2 Python release compiled under Macports and located elsewhere. How can I force Qgis to link with the latter rather than the former?*

As I said: add your Python bin folder to the PATH before configuring Qgis.

#5 - 2009-09-24 10:53 PM - vince -

Replying to [comment:4 kyngchaos]:

> *As I said: add your Python bin folder to the PATH before configuring Qgis.*

That's what I did. In my .cshrc, /usr/pkg/bin is leading in the path, so:

```
-> which python
/usr/pkg/bin/python
```

However, when I launch Qgis from the Finder, by clicking on the app icon, it finds the system Python instead and pops an error informing me that it cannot find sip and the Python bindings will be disabled. That does not happen if I lauch Qgis from a terminal.

#6 - 2009-09-25 05:26 AM - vince -

Replying to [comment:3 vince]:

> *Ok, I've got that. The LC_ID_DYLIB field in the various Grass libs is wrong (points towards the temporary build directory) so that any further lib or executable built against them searches in a non-existent directory (and not in the directory they actually stay). I'll fix that in my grass port.*

It is done. I've fixed the library ID problem in Grass64 and it is now fine, I get the Grass plugin. Now, only the Python problem remains.

#7 - 2009-09-25 06:11 AM - William Kyngesburye

> *That's what I did. In my .cshrc, /usr/pkg/bin is leading in the path, so:*

But, is it *configured* and compiled with your python in the PATH? At runtime, Qgis doesn't use the python executable, but the python library it was linked to when compiled. (Anyways, shell init files don't affect applications.)

And speaking of SIP/PyQt, the sip and pyqt programs need to be in your PATH also during configuration and compilation.

#8 - 2009-09-25 06:24 AM - vince -

Replying to [comment:7 kyngchaos]:

> *But, is it configured* and compiled with your python in the PATH? At runtime, Qgis doesn't use the python executable, but the python library it was linked to when compiled. (Anyways, shell init files don't affect applications.)

Yes, qgis is build from a terminal with the correct PATH. If you look at the output of the cmake, it correctly spots it:

```
-- Found [[PythonInterp]]: /usr/pkg/bin/python2.6
```

which is not Apple default Python (located in /usr/bin).

> *And speaking of SIP/PyQt, the sip and pyqt programs need to be in your PATH also during configuration and compilation.*

That's also the case (sip-2.6, pyuic4-2.6 and pyrcc4-2.6).

#9 - 2009-09-25 06:42 AM - vince -

I got it, and you were right:

```
otool -L libqgispython.dylib
libqgispython.dylib:
  @executable_path/lib/libqgispython.1.3.0.dylib (compatibility version 1.3.0, current version
    [...]
  /System/Library/Frameworks/Python.framework/Versions/2.6/Python (compatibility version 2.6.0, current version 2.6.1)
```

That is where it is wrong. If, with install_name_tool, I change that, it is fine (and it works now).

So, question: how shall I instruct the build process to link to the alternate framework rather than the System one?

#10 - 2009-09-25 06:59 AM - William Kyngesburye

I looked into the cmake python detection, and it's two separate steps - the library and the executable. The executable is simple, and it should easily find your python. The library is different in that for OSX, it *always* looks for the python framework in the standard locations, and there is no way to tell it to look elsewhere.

So while cmake may report finding your python executable, it finds a different framework and *that* is what is linked into Qgis (Qgis only uses the executable during compilation to run the setup.py).

It's a cmake problem, nothing that can be fixed in Qgis.

... maybe you can short-circuit cmake - add some cmake vars to your configuration:

```
-D Python_FRAMEWORKS="/path/to/your/Python.framework"
-D CMAKE_FIND_FRAMEWORKS_INCLUDED=1
```

#11 - 2009-09-25 09:35 AM - vince -

Replying to [comment:10 kyngchaos]:

> *I looked into the cmake python detection, and it's two separate steps - the library and the executable. The executable is simple, and it should easily find your python. The library is different in that for OSX, it always* looks for the python framework in the standard locations, and there is no way to tell it to look elsewhere.
> So while cmake may report finding your python executable, it finds a different framework and *that* is what is linked into Qgis (Qgis only uses the executable during compilation to run the setup.py).
> It's a cmake problem, nothing that can be fixed in Qgis.
>
> ... maybe you can short-circuit cmake - add some cmake vars to your configuration:

```
> -D Python_FRAMEWORKS="/path/to/your/Python.framework"
> -D CMAKE_FIND_FRAMEWORKS_INCLUDED=1
```

I'll try that. On the other hand, similarly to what I did with Grass, I can use install_name_tool to change the reference inside the library.

Anyway, thanks for the help. Now, I'm trying to figure out why the Grass plugin won't open any mapset but just tells me they are already in use, even if they are new...

Cheers !

#12 - 2009-09-25 09:49 AM - vince -

BTW, there is maybe another solution to the Python framework problem. With -framework Python, ld accepts also -F {FRAMEWORK_PATH} that specify alternate frameworks path searched before the standard path. Is there a way to add an additional flag to be passed to ld?

#13 - 2009-09-25 10:00 AM - William Kyngesburye

> *On the other hand, similarly to what I did with Grass, I can use install_name_tool to change the reference inside the library.*

Except that there is no guarantee that the same version of Python will be found by the default python detection as is installed in [[MacPorts]]. Linking to v2.5 and then telling Qgis to load v2.6 will likely cause trouble.

> *With -framework Python, ld accepts also -F {FRAMEWORK_PATH} that specify alternate frameworks path searched before the standard path.*

The way cmake works with libraries, it probably links the framework binary directly, so -F will have no effect. ie instead of:

```
-F /Library/Frameworks -framework Python
```

it will link:

```
/Library/Frameworks/Python.framework/Versions/2.5/Python
```

This is so that (I'm guessing) cmake can make sure the exact library binary that you want, or an internal library binary, is linked and -L/-F paths don't confuse things.

#14 - 2009-09-25 10:31 AM - vince -

Replying to [comment:13 kyngchaos]:

```
> -F /Library/Frameworks -framework Python
```

> *it will link:*

```
> /Library/Frameworks/Python.framework/Versions/2.5/Python
```

> *This is so that (I'm guessing) cmake can make sure the exact library binary that you want, or an internal library binary, is linked and -L/-F paths don't confuse things.*

In that case, may be another patch should work: redefine the loader LD to be a script shell that strips out /Library/Framework/... and replace it by the correct flags. I did that to patch scipy that had issues with the Fortran compiler, and it worked like a charm.

#15 - 2009-09-26 02:13 AM - vince -

- Resolution set to fixed

- Status changed from Open to Closed

Well, I found the trick. There was two problems at stake:

1. The python Framework installed by Macports is incomplete and is not recognized by ld, thus always defaulting to the system one. This is an internal bug that should be fixed by the Macports team;

2. The file to patch is build/src/python/CMakeFiles/qgispython.dir/link.txt. Now, it works fine. I close the bug.

#16 - 2009-09-26 07:43 AM - William Kyngesburye

- Status changed from Closed to Feedback

- Resolution deleted (fixed)

#17 - 2009-09-26 07:45 AM - William Kyngesburye

- Status changed from Feedback to Closed

- Resolution set to invalid

I'm just reclosing this to apply a more meaningful resolution, as the issues were all outside of Qgis.