

QGIS Application - Bug report #3999

[OsX] GRASS provider doesn't load on OS X

2011-06-19 09:57 AM - William Kyngesburye

Status:	Closed	
Priority:	High	
Assignee:		
Category:	GRASS	
Affected QGIS version:	1.8.0	Regression?: No
Operating System:	Mac OS X	Easy fix?: No
Pull Request or Patch supplied:	No	Resolution: fixed
Crashes QGIS or corrupts data:	No	Copied to github as #: 13987
Description		
<p>In commit:0b2317e8 the GRASS provider was changed from a module to a shared library (because the grass plugin uses functions in the provider), which is then linked by the GRASS plugin. This causes the provider module to not load on OS X, then the plugin fails to load.</p> <p>On OS X, shared libraries have an extension .dylib, while modules use the *nix .so extension. The plugin loading code in QGIS looks for .so files on OS X.</p> <p>If there are things in the provider that are needed by both the provider and the plugin, they should be moved to the grass library, or another way to handle this must be figured out.</p>		

Associated revisions

Revision 9e57e20f - 2012-04-02 10:23 PM - Jürgen Fischer

move grassprovider to grass library (might fix #3999)

History

#1 - 2011-06-28 07:55 PM - William Kyngesburye

- Must fix set to yes

Maybe special linking options can be used for the grass plugin on OS X to dynamically lookup symbols (from the provider as a module) at runtime so linking doesn't fail in compilation.

But that's just a hack, and may fail anyways because (I think) QGIS loads plugins and providers alphabetically (or at least with no guaranteed order), so the "grassprovider" module won't be loaded yet when it attempts to load "grassplugin". If it could be guaranteed that providers load before plugins, it should work.

#2 - 2011-06-30 08:10 PM - William Kyngesburye

... dynamic_lookup seems to work. Maybe the load order is different. Maybe symbols are not looked up at load time, but at usage time.

I'd like to have a definite answer about other solutions before applying this fix.

#3 - 2011-07-01 10:34 AM - Tom Elwertowski

William, you explain the solution clearly and concisely in the last paragraph of the bug report. The shared code needs to be moved to a library. I'm baffled

why others are no longer willing to use cross-platform design patterns.

I solved this once for GRASS several years ago and until now, the design pattern remained in use. Relying upon special options or load order tends to result in the problem reappearing anytime something changes.

The basic issue is that on Linux, libraries and plugins are interchangeable and on the Mac they are not. In CMake terms, MODULE and SHARED produce the same type of output on Linux but produce different types of output on a Mac. Linux can use a MODULE as if it were a SHARED; a Mac cannot.

A Mac plugin is a leaf executable. It can only be loaded on demand and not linked as a library. A Mac library is a non-leaf executable. It is automatically loaded because an app, plugin or another library is linked to it and cannot be loaded on demand. Perhaps some or all of this is overridable by special options but that makes the system less straightforward.

The cross-platform design pattern is that share code must always be in a library. If code originally written just for a plugin is found to be useful elsewhere, it should be moved either to an existing library or to a new library.

#4 - 2011-11-06 02:00 PM - William Kyngesburye

- % Done changed from 0 to 50
- Pull Request or Patch supplied set to No

Haven't heard of any problems with my dynamic lookup fix (it's in my 1.7.1 and dev packages). So, added to master 595eef0 until a better solution can be found. I'll leave this open just in case, marked 50% done.

#5 - 2011-11-16 07:32 AM - William Kyngesburye

- % Done changed from 50 to 0
- Priority changed from High to 6

OK, so the dynamic lookup only partially works. Loading GRASS data works, but running GRASS commands crashes QGIS with an error about a missing grassLayer symbol.

It may be related to linking namespaces (not C++ namespaces). The default linking on OS X uses a 2-level namespace setup, which stores the library name + symbol name for lookup, or something like that. The error says it's looking for the grassLayer symbol in flat namespace, which makes sense because with dynamic lookup you don't know where the symbol will be.

I would rather not change everything to a flat namespace, as that will probably introduce other headaches.

This needs to be fixed, and as suggested, by moving shared code to the GRASS library. That's what libraries are for.

#6 - 2011-12-24 12:59 AM - Giovanni Manghi

- Crashes QGIS or corrupts data set to No
- Subject changed from GRASS provider doesn't load on OS X to [OsX] GRASS provider doesn't load on OS X
- Affected QGIS version set to master

#7 - 2012-01-31 01:51 PM - John Helly

I'd like to add that QGIS 1.9.90-Alpha fails immediately when I enable the GRASS plugin from the plugin manager on OSX 10.6.8 on a MacBook Pro. This is the same behavior exhibited by Wrocław (don't have version number available).

#8 - 2012-02-07 06:45 PM - John Helly

This problem does cause a crash everytime.

#9 - 2012-03-09 07:23 AM - William Kyngesburye

- Target version changed from Version 2.0.0 to Version 1.8.0
- Affected QGIS version changed from master to 1.8.0

#10 - 2012-04-02 01:29 PM - Jürgen Fischer

- Status changed from Open to Closed

Fixed in changeset commit:"9e57e20f61d025132f5b44f60a892e7263043661".

#11 - 2012-04-02 01:30 PM - Jürgen Fischer

- Status changed from Closed to Feedback

does commit:9e57e20f61d help on OSX?

#12 - 2012-04-02 05:01 PM - William Kyngesburye

That --no-undefined flag is causing trouble. The OS X ld does not have that option (OSX ld is a custom job), but it does have "-undefined error" (or "-undefined warning", and others, similar to my temporary fix for the provider). Note that the default is error.

Strangely, it only appears in module linking (so far). CMAKE_SHARED_LINKER_FLAGS doesn't seem to do anything - I don't see the flag in any library linking. Haven't got as far as the QGIS app to see if it appears from CMAKE_EXE_LINKER_FLAGS.

#13 - 2012-04-03 12:27 AM - Jürgen Fischer

William Kyngesburye wrote:

That --no-undefined flag is causing trouble. The OS X ld does not have that option (OSX ld is a custom job), but it does have "-undefined error" (or "-undefined warning", and others, similar to my temporary fix for the provider). Note that the default is error.

Well, that's just to produce warnings - if it doesn't work on OSX just disable it:

```
diff --git a/CMakeLists.txt b/CMakeLists.txt
index f7bf497..9449b66 100644
--- a/CMakeLists.txt
+++ b/CMakeLists.txt
@@ -399,11 +399,11 @@ IF (WIN32)
   SET(DLLIMPORT "__declspec(dllimport)")
   SET(DLLEXPORT "__declspec(dllexport)")
 ELSE (WIN32)
- IF(PEDANTIC)
+ IF(PEDANTIC AND NOT APPLE)
   SET(CMAKE_SHARED_LINKER_FLAGS "${CMAKE_SHARED_LINKER_FLAGS} -Wl,--no-undefined")
```

```
    SET(CMAKE_MODULE_LINKER_FLAGS "${CMAKE_MODULE_LINKER_FLAGS} -Wl,--no-undefined")
    SET(CMAKE_EXE_LINKER_FLAGS "${CMAKE_EXE_LINKER_FLAGS} -Wl,--no-undefined")
-   ENDIF(PEDANTIC)
+   ENDIF(PEDANTIC AND NOT APPLE)

SET(DLLIMPORT "")
SET(DLLEXPORT "")
```

But that might also be an issue for different versions of gcc on Linux - waiting for the nightly builds to see - so I didn't apply that yet.

#14 - 2012-04-03 02:34 PM - Jürgen Fischer

Jürgen Fischer wrote:

| *But that might also be an issue for different versions of gcc on Linux - waiting for the nightly builds to see - so I didn't apply that yet.*

Apparently not a problem on Linux - applied in commit:d2fbcfeb

#15 - 2012-04-15 08:46 AM - Giovanni Manghi

- *Priority changed from 6 to High*

#16 - 2012-04-16 05:59 AM - Giovanni Manghi

is this fixed now?

#17 - 2012-04-16 06:38 AM - William Kyngesburye

Seems to work, with some simple testing.

#18 - 2012-04-17 05:31 AM - Jürgen Fischer

- *Status changed from Feedback to Closed*

- *Resolution set to fixed*