Wt - Support #2713

Probably a linking error

02/24/2014 09:38 PM - Tor Arne Fallingen

Status:	Feedback	**Start date:**	02/24/2014
Priority:	Normal	**Due date:**	
Assignee:	Koen Deforche	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:	3.3.1		

Description

Ive been playing around a bit with Wt now, but ever so often I encounter this silly little problem, and Im not certain where the error stems from:

Undefined symbols for architecture x86_64:

"Wt::WLink::WLink(Wt::WLink::Type, std::string const&)", referenced from:

ControlExample::page1() in ControlSkeleton.o

"Wt::WObject::setObjectName(std::string const&)", referenced from:

vtable for ControlExample in ControlSkeleton.o

"Wt::WString::WString(std::string const&, Wt::CharEncoding)", referenced from:

ControlExample::ControlExample(Wt::WEnvironment const&) in ControlSkeleton.o

ControlExample::header() in ControlSkeleton.o

"Wt::WApplication::makeAbsoluteUrl(std::string const&) const", referenced from:

vtable for ControlExample in ControlSkeleton.o

ld: symbol(s) not found for architecture x86_64

clang: error: linker command failed with exit code 1 (use -v to see invocation)

make: * [ControlSkeleton] Error 1

Both my Boost and my Wt install are done to custom folders, so I had to run: export DYLD_LIBRARY_PATH=/custom/directory/lib:$DYLD_LIBRARY_PATH for Boost. I did not do this for Wt, but I have added DYLD_LIBRARY_PATH to my Run environment in Qt Creator. And Ive added the LIBS -lwt -lwthttp -lboost_signals (including their paths) to my *.pro file. I wonder if I lack some libraries in my LIBS settings, and Ive searched for what libraries the functions are dependent on, but I would not be surprised if someone provide me a link to a site I most definitively should have looked at. Im running this on my MacBook. Earlier today I ran the same program on my Raspberry Pi, no problems at all.... its so strange. Im trying to find somewhere I can see what DYLD_LIBRARY_PATH's are set on this MacBook, so if anyone knows..

History

#1 - 02/25/2014 10:35 PM - Koen Deforche

- Status changed from New to Feedback

- Assignee set to Koen Deforche

Hey,

I can pinpoint it like that. It will help to see the actual link command:

```
VERBOSE=1 make
```

A mac usually has an empty DYLD_LIBRARY_PATH by default, in my experience. You can check with

```
echo $DYLD_LIBRARY_PATH
```

But it's important to understand that this path is not involved in the linking problem. Linking ignores the DYLD_LIBRARY_PATH; only running an application may consult this path.

Regards,

koen

#2 - 02/25/2014 11:25 PM - Tor Arne Fallingen

Hey koen.

Im not sure I follow, but I think you want to see more of the make commands? Well, I copied the whole thing from the compile output. It was not like a million miles long. Im using Qt Creator.

```
22:08:19: Starting: "/usr/bin/make"

/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/clang -headerpad_max_install_names
-Wl,-syslibroot,/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.8.sdk
-mmacosx-version-min=10.6 -o ControlSkeleton main.o -F/Users/taf/Qt5.2.1/5.2.1/clang_64/lib -L/Users/taf/Wt/lib -lwt -lwthttp -L/Users/taf/Boost/lib
-lboost_signals -lboost_system -lboost_thread -lboost_filesystem -lboost_regex -framework QtCore

Undefined symbols for architecture x86_64:

"Wt::WLink::WLink(Wt::WLink::Type, std::string const&)", referenced from:

ControlSkeleton::page1() in main.o

"Wt::WObject::setObjectName(std::string const&)", referenced from:

vtable for ControlSkeleton in main.o

"Wt::WString::WString(std::string const&, Wt::CharEncoding)", referenced from:

ControlSkeleton::ControlSkeleton(Wt::WEnvironment const&) in main.o

ControlSkeleton::header() in main.o

"Wt::WApplication::makeAbsoluteUrl(std::string const&) const", referenced from:

vtable for ControlSkeleton in main.o

ld: symbol(s) not found for architecture x86_64

clang: error: linker command failed with exit code 1 (use -v to see invocation)

make: * [ControlSkeleton] Error 1

22:08:20: The process "/usr/bin/make" exited with code 2.

Error while building/deploying project ControlSkeleton (kit: Desktop Qt 5.2.1 clang 64bit)

When executing step 'Make'

22:08:20: Elapsed time: 00:01.
```

Then if I understand you correctly, $DYLD_LIBRARY_PATH is one single entry (one path) and its temporary? Because yes I tried the "echo $DYLD_LIBRARY_PATH", and that gives me nada. And since I installed both Boost and Wt in custom locations, I need to link both the lib and the include directories manually when I use something dependent on them. Is there a way to do this more permanent? Like entering a path in a system file to include in the systems default directories to look for libraries and such? I feel green when I ask these things, but Im learning, and setting up RPi 50-100 times surely helps in starting to understand a little more how things are working.

#3 - 03/04/2014 10:19 AM - Koen Deforche

Hey,

The reason may be that you haven't built Wt the library with the same toolchain as your application:

- gcc versus clang
- libc versus stdlibc
- C+11 versus C+98

On MacOSX, the toolchains are a bit in a state of flux (moving away from gcc, towards CLang) but that requires some attention when trying to build applications and libraries.

Regards,

koen