Blog Inventic.eu

Notes and comments from ORM Designer development

Qt on OSX Maverick – Undefined symbols for architecture x86_64

The problem is that starting with OSX 10.9 Apple changed default standard c++ library from libstdc++ to libc++.

> *Qt binary distribution compile with -stdlib=libstdc++ to be compatible with 10.6, Xcode 5 on 10.9 will select -stdlib=libc++ by default (for OS X 10.7 and better only). So symbol using classes from the standard library (like std::string in this case) will not resolve correctly at link time.*

```
Undefined symbols for architecture x86_64:
"boost::filesystem::path_traits::convert(char const*, char const*, std::basic_st
boost::filesystem::path::wstring(std::codecvt&lt;wchar_t, char, __mbstate_t&gt;
boost::filesystem::path::wstring(std::codecvt&lt;wchar_t, char, __mbstate_t&gt;
boost::filesystem::path::wstring(std::codecvt&lt;wchar_t, char, __mbstate_t&gt;
```

So it’s necessary to compile all libraries with one type of libstdc++. Because I need to keep 10.6 compatibility, it’s necessary to compile boost and other libraries with libstdc++ dependency.

To check which library is used, use otool tool:

```
otool -L library.dylib
```

As result you will get something like this (check /libc++1.dylib):

```
otool -L boost/lib/libboost_filesystem.dylib
boost/lib/libboost_filesystem.dylib:
    boost/lib/libboost_filesystem.dylib (compatibility version 0.0.0, current ve
    boost/lib/libboost_system.dylib (compatibility version 0.0.0, current versio
    /usr/lib/libc++.1.dylib (compatibility version 1.0.0, current version 120.0.
    /usr/lib/libSystem.B.dylib (compatibility version 1.0.0, current version 119
```

How to fix it for boost

it’s necessary to recompile boost (and don’t forget to remove ./bin.v2 directory) with these params:

```
./b2 cxxflags="-stdlib=libstdc++" linkflags="-stdlib=libstdc++" ...
```

and run otool again:

```
boost/lib/libboost_filesystem.dylib:
    boost/lib/libboost_filesystem.dylib (compatibility version 0.0.0, current ver
    boost/lib/libboost_system.dylib (compatibility version 0.0.0, current versior
    /usr/lib/libstdc++.6.dylib (compatibility version 7.0.0, current version 60.0
    /usr/lib/libSystem.B.dylib (compatibility version 1.0.0, current version 1197
```

How to fix it for CMake libraries

In case you’re using library which is built by CMake system, you need to add following flag:

```
cmake -DCMAKE_CXX_FLAGS="-stdlib=libstdc++"
```

How to fix other libraries

For any other library it’s necessary to pass libstdc++ flag in any available way, for example modify makefile and add :

```
CXXFLAGS = -stdlib=libstdc++
```

And that’s all

Hope this post saves you a lot of time I have to spent by searching these answers 😉

External links:

- https://qt-project.org/forums/viewthread/35646

This entry was posted in Mac OS, Programming and tagged clang, libstdc++, mac, maverick, Qt on 2014/01/05 [http://blog.inventic.eu/2014/01/qt-on-osx-maverick-undefined-symbols-for-architecture-x86_64/] .

One thought on “Qt on OSX Maverick – Undefined symbols for architecture x86_64”

soumya
2014/01/29 at 03:20

You could just use the variant of the port to install the boost with required compiler like

```
port variants boost
port install boost +gcc49
```