

Robotics Systems Commons - Tasks #1346

Patch request: use either libboost 1.46 or 1.48 on debian systems?

01/22/2013 11:32 AM - S. Wrede

Status:	Rejected	Start date:	01/22/2013
Priority:	Normal	Due date:	
Assignee:	J. Wienke	% Done:	0%
Category:	cmake	Estimated time:	0.00 hour
Target version:	rsb-0.9		

Description

Original request from Alexandre Tuleu:

It would be nice so we could use either libboost 1.46 or libboost 1.48 on precise system.

He proposed the patch attached to this issue. Thx for that!

History

#1 - 01/22/2013 11:35 AM - S. Wrede

Just a question that worried us when briefly thinking about this: Do you think this will work given the header-only implementations contained in RSC and other dependant libs? Consider that RSC was build against libboost 1.46 and this Debian package is now used on a system with 1.48? Would that work, ABI-wise?

#2 - 01/22/2013 11:50 AM - A. Tuleu

Actually I did not thought that RSC was not header only, but does have some precompiled binaries. (librsc.so.0.9 is installed)

Then ABI compatibilty is not insured between 1.46 and 1.48 (that's why package does not have same name). However the big difference between this two version is C++ support.

Anyway since nemomath is header only, it should allow this conditionnal dependency (see #1341) for precise (There is no limitation, and it leave a weight for other user). Then you should choose between using 1.46 or 1.48 with RSC. The biggest difference with the two version is C++0x and C++11 standard support if I am right. Or maybe we could have two line of packages version, for example librsc0.9 and librsc0.9-boost1.48, but that may just be too much of an hassle for just clang support.

However I recompiled without problem RSC and nemomath with libboost1.46 and clang v3.1 and v3.2 with no problem. Only RCI show some compilation errors. Maybe this is some bug that only show up with clang in RCI.

#3 - 01/22/2013 12:46 PM - J. Wienke

Alexandre Tuleu wrote:

Actually I did not thought that RSC was not header only, but does have some precompiled binaries. (librsc.so.0.9 is installed)

Then ABI compatibilty is not insured between 1.46 and 1.48 (that's why package does not have same name). However the big difference between this two version is C++ support.

RSC is mostly implemented as a shared library with the least amount of header implementations. However, we are using boost, where a big fraction of code is implemented in the headers. Hence, a version compiled against 1.46 cannot correctly work with 1.48, because many parts of the boost implementation 1.46 are actually compiled into RSC due to the header implementations. We would have to provide a second package of RSC capable of working with 1.48. But this then ripples through all downstream dependencies, which results in a maintenance nightmare for us. So I see no real chance to do this.

Anyway since nemomath is header only, it should allow this conditionnal dependency (see #1341) for precise (There is no limitation, and it leave a weight for other user). Then you should choose between using 1.46 or 1.48 with RSC. The biggest difference with the two version is C++0x and C++11 standard support if I am right. Or maybe we could have two line of packages version, for example librsc0.9 and librsc0.9-boost1.48, but that may just be too much of an hassle for just clang support.

As explained above, this ripples through all downstream dependencies like RSB etc. I don't think we can manage this additional effort. Moreover, all ubuntu packages depend on 1.46. If we add the 1.48 version, linking against ubuntu packages, which internally also use boost, is not possible anymore without conflicts. I would like to keep the common baseline for ubuntu to preserve this compatibility.

#4 - 01/29/2013 08:51 AM - J. Wienke

Alexandre? Any more thoughts on this or would it be ok to close this issue?

#5 - 01/30/2013 03:25 PM - A. Tuleu

No it is ok to close it

#6 - 01/30/2013 04:10 PM - J. Wienke

- Status changed from New to Rejected

Files

fixes_rsc_for_multipleboost_version.patch	897 Bytes	01/22/2013	S. Wrede
---	-----------	------------	----------