Robot Control Interface - Bug #1343
librci0.4-dev debian packaged should mark libboost-regex-dev as a dependency

01/21/2013 04:41 PM - A. Tuleu

Status:	Resolved	**Start date:**	01/21/2013
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	rci0.4		

Description

compiling liboncilla (that depends on rci), needs the file libboost_regex-mt.so to be present, and packaged in package libboost-regex-dev

I would suggest to put :

```
Depends: libboost-regex-dev | libboost-regex1.48-dev
```

in the dependency of the package, or in the dependency of upstream dependencies if librci does not actually depends on that package, but one of its deps actually depend on it.

Associated revisions

Revision 645 - 01/21/2013 07:48 PM - anordman

Added missing debian dependency to libboost-regex-dev

fixes #1343

Revision 647 - 01/22/2013 11:13 AM - anordman

Allow dependency to libboost-regex1.48-dev also, refs #1343

History

#1 - 01/21/2013 08:13 PM - Anonymous

- Status changed from New to Resolved

- Assignee set to Anonymous

- % Done changed from 0 to 100

Updated. Thanks for reporting

#2 - 01/22/2013 09:44 AM - A. Tuleu

Why not putting libboost-regex1.48-dev as an alternative dependency ? It would allow users to install and use boost 1.48 instead of only boost 1.46 on precise ? Actually clang and boost 1.46 does not work at all, but 1.48 does.

And I would very much like use boost 1.48 and clang on precise ....

And as for quantal, 1.49 is the default, so if you continously integrate over quantal you should get it.

#3 - 01/22/2013 11:13 AM - Anonymous

I patched the package accordingly. Should roll-out within the next minutes.

#4 - 01/22/2013 02:51 PM - A. Tuleu

Actually I read more about all this stuff, and it seems that boost prior to 1.48.0-3 has a lot of issue with c++0X and c++11, even with just smart pointers.

- http://stackoverflow.com/questions/7964360/using-stdshared-ptr-with-clang-and-libstdc
- http://comments.gmane.org/gmane.comp.compilers.clang.devel/15926

The issue is that the old GCC 4.6 does not implement the complete C++11 standard, and let not this bug show up. But newer compiler does....

So Using RSC allone is OK, using RSC + NemoMath (need to use c++11), will always break with a more recent compiler.

However, for that stupid Roboard platform I need to support lucid (it stays at boost 1.40).... so we could not enforce on that platform boost >= 1.48.0-3

This whole issue drives me crazy .... I think I would even try to backport to lucid libboost1.48 (no really, I would like to be closer than the standard ....)

#5 - 01/22/2013 02:55 PM - A. Tuleu

- Status changed from Resolved to In Progress

Sorry for lot of comments on that issue on reopening

I think that we should rather forbids compilation of rci of boost version is less than 1.48 and the compiler is either GCC 4.7 or Clang

And we should dismiss that conditionnal install, as it could lead to API break with libboost_regex and librsc

I would prepare some patch for you if you agree

#6 - 01/22/2013 03:20 PM - A. Tuleu

- File check_for_right_boost_version_if_decent_c 11_compiler.patch added

The added patch would check at configuration time if someone is trying to use a bad compiler / boost / C++11 mix and report an error

Should be extended to see if gcc 4.7 (full C++11 support) should be tested too for boost >> 1.48.0 (1.48.0-3 is debian patched version of boost, this issue is fixed only for >> 1.48.0 in non debian world)

Maybe you could add a module in RSC for that. I will add the same with liboncilla.

#7 - 01/22/2013 04:13 PM - Anonymous

Thanks for diving into this. Note, that Nemomath has the option to work without C++11 (cmake option USE_CPP_0X). Would it help to set this flag, in the incompatible cases you mentioned?

#8 - 08/12/2013 02:05 PM - Anonymous

- Status changed from In Progress to Resolved

- Target version set to rci0.4

Files

check_for_right_boost_version_if_decent_c 11_compiler.patch	1.5 KB	01/22/2013	A. Tuleu