

CoCoALib - Design #999

configuration: include -std=c++03 by default?

18 Jan 2017 18:01 - John Abbott

Status:	Closed	Start date:	18 Jan 2017
Priority:	Normal	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	Portability	Estimated time:	2.22 hours
Target version:	CoCoALib-0.99560	Spent time:	2.15 hours
Description After having compile CoCoALib on several newish Linux boxes, I have noticed that there are many warnings about auto_ptr. These go away if compilation is with the flag -std=c++03. Perhaps we should use this flag by default (until we update the CoCoALib sources)?			
Related issues:			
Related to CoCoALib - Feature #1010: C++11: Mario's Hilbert scheme code		New	20 Feb 2017
Related to CoCoA-5 - Design #83: C++11 compatibility questions		In Progress	26 Jan 2012
Related to CoCoALib - Feature #82: C++11 compatibility questions		Closed	26 Jan 2012

History

#1 - 19 Jan 2017 17:05 - John Abbott

The compilation flag **-std=c++03** is not recognized by the old g++ (version 4.2.1) on my old MacBook Pro, so causes compilation failure.

Options are:

- **(A)** hard luck me, I'll have to specify CXXFLAGS manually;
- **(B)** do an automatic test to see whether -std=c++03 is accepted as a flag (if not, get rid of it);
- **(C)** have a configuration option saying whether to use the -std=c++03 flag.

I do not much like option **(C)** because there are already too many configuration options.

Option **(A)** is tempting but it does make life awkward for people with old platforms -- that said, we are only a small project and cannot afford too much effort toward "backward compatibility" for out-of-date platforms.

Perhaps option **(B)** is a fair compromise; it should not be too hard to implement (I hope), and ought to make life easy for old-platformers. Of course, when we switch to C++11 (when??? soon?) then old platforms will be automatically excluded :-/

If we do plan to switch soon to C++11 (c'mon, it's already 2017) then it may not make much sense to invest time in keeping compatibility with old platforms just for an extra few months... (sigh!)

#2 - 20 Jan 2017 14:11 - John Abbott

- Description updated

- Status changed from New to In Progress

- % Done changed from 0 to 10

I note that the compilation flag **-std=c++03** will cease to be relevant (and may even become "harmful") once we update the source code to C++11 (or later).

So the question is: should I invest the time to add `-std=c++03`, or is the time better spent updating the code to(wards) C++11?
In this case it would suffice to replace **auto_ptr** with **shared_ptr** (at least as a quick fix).

#3 - 21 Jan 2017 18:01 - John Abbott

- Status changed from *In Progress* to *Resolved*

- Assignee set to *John Abbott*

- % Done changed from 10 to 60

After thinking about having to compile+install CoCoA/CoCoALib on numerous recent Linux boxes for the upcoming CoCoALib minicourse, I decided it was better to put in the `-std=c++03` flag.

Getting it to work nicely also with my old MacBook Pro, I quickly opted for the idea of making a new script which checks whether the option `-std=c++03` is accepted by the compiler.

It works on the old MacBook as well as the new linux portable (Fedora 25), so I'm about to check in.

Let's see if it works for Anna too.

NOTE Damn! I can't VPN from the MacBook -- must be some wrong setting somewhere :-/

#4 - 20 Feb 2017 13:20 - John Abbott

- Related to Feature #1010: C++11: Mario's Hilbert scheme code added

#5 - 08 Nov 2017 17:08 - John Abbott

- Status changed from *Resolved* to *Closed*

- % Done changed from 60 to 100

- Estimated time set to 2.22 h

I have not experienced any problem with compiling CoCoALib over the last 10 months, so presumably whatever I did is OK.

Closing, but will need to revise when we move to C++11.

#6 - 08 Nov 2017 17:08 - John Abbott

- Related to Design #83: C++11 compatibility questions added

#7 - 08 Nov 2017 17:08 - John Abbott

- Related to Feature #82: C++11 compatibility questions added