

## CoCoALib - Design #932

### CoCoALib configuration: BOOST dependency

30 Sep 2016 10:55 - John Abbott

<b>Status:</b>	Closed	<b>Start date:</b>	30 Sep 2016
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	John Abbott	<b>% Done:</b>	100%
<b>Category:</b>	Portability	<b>Estimated time:</b>	1.66 hour
<b>Target version:</b>	CoCoALib-0.99700	<b>Spent time:</b>	1.60 hour
<b>Description</b>			
Currently we have a single configuration process (script) for both CoCoALib and CoCoA-5. This is not ideal!			
Gereon was surprised that a CoCoALib program he had written needed to link in BOOST code before it could run. I, too, was surprised when he told me (since I believed CoCoALib to be BOOST-free).			
What happens is that the configure script looks for BOOST automatically; and, if found, takes note of where it is. If BOOST is present then inside CoCoALib the ThreadsafeCounter code will switch to using BOOST code (even if CoCoA_THREADSafe_HACK is not set).			
What to do? Discuss!			
<b>Related issues:</b>			
Related to CoCoALib - Feature #319: BOOST -- how it could help in CoCoALib		<b>Closed</b>	<b>21 Feb 2013</b>
Related to CoCoALib - Design #933: Separate configure scripts for CoCoALib an...		<b>In Progress</b>	<b>30 Sep 2016</b>

#### History

##### #1 - 30 Sep 2016 10:56 - John Abbott

- Related to Feature #319: BOOST -- how it could help in CoCoALib added

##### #2 - 30 Sep 2016 11:01 - John Abbott

As far as I can see, the code in ThreadsafeCounter will probably become much simpler when we switch to C++11 (since atomic types are part of the language).

ThreadsafeCounter is used for creating anonymous symbols (see symbol.H) and ring IDs (see ring.C).

##### #3 - 30 Sep 2016 11:06 - John Abbott

A reasonable solution might simply to be to remove the BOOST dependent code in ThreadsafeCounter. This would mean that threadsafety is not possible with a pre-C++11 compiler (though you would probably have to create very rapidly many anonymous symbols or many rings to trigger a problem).

##### #4 - 30 Sep 2016 11:20 - John Abbott

- Related to Design #933: Separate configure scripts for CoCoALib and CoCoA-5 added

##### #5 - 29 Nov 2016 19:24 - John Abbott

- Status changed from New to In Progress

- % Done changed from 0 to 10

The CoCoA configure script complains if someone specifies "with Normaliz" but "without BOOST". The code in CoCoALib does not depend on BOOST at all.

Bruns reported by email (2016-11-29) some problems with CoCoALib wanting to use BOOST (perhaps as an attempt to make Normaliz and cocoa5(?) communicate with Sage).

#### #6 - 29 Nov 2016 19:30 - John Abbott

I have added a `--no-boost` option to the configure script; this disables automatic searching for the BOOST headers and libraries.

configure used to give an error if `--with-normaliz` was specified and BOOST was absent; however, a quick check showed that there is no true dependency: *i.e.* disabling the check in the configure script, and then compiling and running everything worked fine.

I do note that nm finds some BOOST symbols in the executables involving Normaliz, but they have probably come from just the BOOST header files... so absence of BOOST is not a problem.

I shall comment out the check in configure for BOOST being required if `--with-normaliz` has been specified. I prefer not to delete it yet in case there is some nasty surprise waiting for me... :-/ **[fine after 2 years: deleted the cruft 2019-03-25]**

#### #7 - 27 Jan 2020 15:25 - John Abbott

- Status changed from *In Progress* to *Closed*
- Assignee set to *John Abbott*
- Target version changed from *CoCoALib-1.0* to *CoCoALib-0.99700*
- % Done changed from *10* to *100*
- Estimated time set to *1.66 h*

Since we moved to C++11/C++14 we can use an atomic C++ counter instead of the threadsafe code; so I have already deleted the threadsafecounter code (some time ago).

This issue is now completely resolved (I hope!)