

CoCoALib - Feature #1360

configure script: add flags for "only cocoalib" or "both cocoalib and cocoa5" (for boost)

01 Nov 2019 10:22 - John Abbott

<b>Status:</b>	Closed	<b>Start date:</b>	01 Nov 2019
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	John Abbott	<b>% Done:</b>	100%
<b>Category:</b>	Makefile	<b>Estimated time:</b>	14.01 hours
<b>Target version:</b>	CoCoALib-0.99850	<b>Spent time:</b>	14.00 hours
<b>Description</b>			
Considering adding configuration option(s) to specify:			
<ul style="list-style-type: none"><li>• compile just cocoalib</li><li>• compile cocoalib and also also CoCoA-5 (and GUI if poss).</li></ul>			
Which flag(s) do we want? What should they be called?			
<b>Related issues:</b>			
Related to CoCoALib - Support #1353: configure script help		<b>Closed</b>	<b>27 Oct 2019</b>
Related to CoCoALib - Design #933: Separate configure scripts for CoCoALib an...		<b>In Progress</b>	<b>30 Sep 2016</b>
Related to CoCoA-5 - Bug #1695: boost-find-lib.sh		<b>Closed</b>	<b>16 Aug 2022</b>
Related to CoCoA-5 - Design #1697: Specifying BOOST in configure		<b>Closed</b>	<b>18 Aug 2022</b>

History

#1 - 01 Nov 2019 10:28 - John Abbott

Anna suggested **--only-cocoalib**

JAA prefers that the default be "only cocoalib", and that an option must be given to build CoCoA-5 (as well).  
JAA still prefers to have two separate configuration/build structure (see [#933](#)).

Some more possible names: **--build-cocoa5**, **--no-cocoa5**

It is probably better not to have names like **--with-cocoa5** or **--without-cocoa5** since we seem to use "with" flags to refer to external libraries.

Further ideas are welcome!

#2 - 04 Nov 2019 11:01 - Anna Maria Bigatti

John Abbott wrote:

Anna suggested **--only-cocoalib**

JAA prefers that the default be "only cocoalib", and that an option must be given to build CoCoA-5 (as well).  
JAA still prefers to have two separate configuration/build structure (see [#933](#)).

Some more possible names: **--build-cocoa5**, **--no-cocoa5**

It is probably better not to have names like **--with-cocoa5** or **--without-cocoa5** since we seem to use "with" flags to refer to external libraries.

I like **--build-cocoa5**. (not with!)  
Makes a lot of sense that the default compilation of cocoalib is without cocoa5!

So readline and boost are not searched.  
For the user who actually want cocoa5 (like us) we add the extra flag in the configuration.

The default without cocoa5 goes in the direction of the separate configurations.

### #3 - 14 Nov 2019 15:28 - John Abbott

- Status changed from New to In Progress
- Target version changed from CoCoALib-0.99650 November 2019 to CoCoALib-0.99700
- % Done changed from 0 to 10

JAA prefers to delay this so there is time to make more checks...

### #4 - 09 Jan 2020 13:21 - John Abbott

- % Done changed from 10 to 40

There is the configuration option **--no-boost** which automatically inhibits building CoCoA-5 (well, it is supposed to do this -- I have not tested it).

This option effectively does what is requested (though it is not obvious from the name). So perhaps just some renaming is all that is needed?

### #5 - 09 Jan 2020 13:27 - John Abbott

It is not quite as simple as I wrote in the previous comment.

The option **--with-boost=** will automatically cause CoCoA-5 to be built.  
Not entirely sure what **--with-readline** will cause; it is anyway just optional for CoCoA-5 (whereas BOOST is required).

What should happen if the configuration options say that CoCoA-5 should not be built, but a BOOST or readline library is specified?  
Error? Warning?

In some ways I would like the options relevant only to CoCoA-5 to have names which make this clear(er). Or perhaps these options may be specified only **after** the option **--build-cocoa5**?

### #6 - 09 Jan 2020 13:29 - John Abbott

- Related to Support #1353: configure script help added

### #7 - 09 Jan 2020 13:30 - John Abbott

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

### #8 - 12 Feb 2020 18:09 - John Abbott

- Target version changed from CoCoALib-0.99700 to CoCoALib-0.99800

### #9 - 06 Oct 2020 12:27 - John Abbott

- Target version changed from CoCoALib-0.99800 to CoCoALib-0.99850

### #10 - 14 Oct 2022 10:09 - Anna Maria Bigatti

- Subject changed from configure script: add flags for "only cocoalib" or "both cocoalib and cocoa5" to configure script: add flags for "only cocoalib" or "both cocoalib and cocoa5" (for boost)

#### #11 - 12 Nov 2022 13:58 - John Abbott

- Assignee set to John Abbott

- % Done changed from 40 to 60

I think I have progress here with helpful testing & feedback from Ulrich.

It is quite a nightmare trying to make portable scripts for platforms which are similar but (not so) subtly different.

I think I am converging to a better configure...

#### #12 - 12 Nov 2022 14:21 - John Abbott

I have revised the configure options, and several auxiliary scripts (after feedback/suggestions from Anna & Ulrich)

- **(A)** New option **--only-cocoalib** (does not look for BOOST, does not compile CoCoA-5 or Qt-gui)
- **(B)** default is to compile everything (CoCoALib, CoCoA-5, Qt-gui): there is an option **--no-qt-gui**
- **(C)** if Qt is absent/insufficient then it is automatically skipped
- **(D)** if BOOST is absent/insufficient then an **error** is reported (unless --only-cocoalib was chosen)
- **(E)** all **extlibs** are now accessed via symlinks (no more "default linking")

Ulrich commented that he was disappointed that configuration did not produce autoconf.mk for CoCoALib even when BOOST is absent. JAA prefers to give error & no autoconf.mk in his case because the command as issued did fail: configuration for CoCoA-5 did not succeed. There is now a helpful error mesg.

Based on JAA's reasoning above **(C)** is not the correct behaviour: it should also give error if Qt fails. Anyway, I'll leave the configure script as it is for the moment... too many changes!!

A motivation for **(E)** was that JAA discovered that there were 2 system installations of BOOST (one in /usr/include and one in /usr/local/include), and it was not immediately obvious which would be chosen as the "system default". With symlinks it is explicitly clear.

#### #13 - 12 Nov 2022 14:23 - John Abbott

The BOOST "find" script does not give error if more than 1 installed version is found; instead is simply picks one (the first it found), and produces a INFO message.

Presumably the GMP "find" script should do the same too... (maybe it does, not sure)

#### #14 - 13 Nov 2022 02:30 - Ulrich von der Ohe

(This concerns an intermediate version of CoCoALib. The issue is not present in the current release 0.99800.)

John Abbott wrote:

Ulrich commented that he was disappointed that configuration did not produce autoconf.mk for CoCoALib even when BOOST is absent. JAA prefers to give error & no autoconf.mk in his case because the command as issued did fail: configuration for CoCoA-5 did not succeed. There is now a helpful error mesg.

Perhaps there is a misunderstanding here: The "command as issued" is a plain ./configure in the main CoCoALib directory. The (optional) CoCoA-5 front-end isn't mentioned anywhere.

Since CoCoALib doesn't require Boost I don't think the absence of Boost should cause an error when running CoCoALib's configure script.

#### **#15 - 16 Nov 2022 21:24 - John Abbott**

- % Done changed from 60 to 70

I understand Ulrich's viewpoint.

Ideally the configuration & build systems for CoCoALib and CoCoA-5 (with GUI) would be separate. After some discussion, we felt that it is likely simpler for most people to have them together. At least, those who want to use just CoCoALib likely already have enough experience to read the documentation and select the appropriate flag (namely --only-cocoalib). For this reason the default "compilation setting" is "make both CoCoALib & CoCoA-5"; if that cannot be done (*e.g.* because BOOST is missing) then the scripts should detect this, and produce a helpful error message.

Earlier we did try to implement a "make as much as possible" strategy, but this led to an excessively complicated configure script (even worse than it is now).

While the current arrangement is not perfect, it is perhaps a reasonable compromise (& does make maintenance a bit easier).

#### **#16 - 17 Nov 2022 16:12 - Ulrich von der Ohe**

John Abbott wrote:

Earlier we did try to implement a "make as much as possible" strategy, but this led to an excessively complicated configure script (even worse than it is now).

I found this statement quite surprising (based on my recent tinkering with the build scripts) and then actually compared the line counts of three recent configure scripts: 0.99800 as released, the patched version from [#1393#note-28](#) (which certainly does follow a "make as much as possible" strategy), and the latest development version that I got sent.

Whatever variation of line counting I tried (in-/excluding commented lines etc.), I didn't find much support for the latest version being much simpler than the others. (One sensible count is 1031, 1028, 1045, in the same order as above, and I'm not even sure I'd attribute the differences to "make as much as possible".)

Of course, mere line counting is an utterly simplistic approach and perhaps I'm missing something here. If so, I'd be curious (but it seems to me that "agree to disagree" is the likely outcome).

**#17 - 18 Nov 2022 16:39 - John Abbott**

Ulrich may indeed be correct to be sceptical: the revision I made to the configure script made 2 changes at more-or-less the same time:

1. handle "system installed" libs the same as "non system installed"
2. semi-automatic choice about what is actually built (= "as much as possible")

I am fairly happy with the current structure.

My view is that the arguments to configure specify what the user wants to build (just CoCoALib, or also CoCoA-5);

the Qt-GUI is still decided automatically. In principle one could have a **--build-as-much-as-possible** flag.

I currently prefer giving an error if the user asks to build CoCoA-5 but some essential ingredient is missing.

What is important is that the error message should explain what the user can try doing to fix the problem

(e.g. *BOOST too old* or *BOOST absent* ... get new version from [www.boost.org](http://www.boost.org))

Presumably a "build as much as possible" could also print out explanations for why it is not building certain parts.

Speaking of which... I'll try improving the Qt-GUI failure message.

**#18 - 23 Nov 2022 17:33 - John Abbott**

- Related to Bug #1695: *boost-find-lib.sh* added

**#19 - 23 Nov 2022 17:33 - John Abbott**

- Related to Design #1697: *Specifying BOOST in configure* added

**#20 - 23 Nov 2022 18:29 - John Abbott**

For the time being we'll not implement "build as much as possible", but instead give error if CoCoA-5 cannot be built (e.g. because of BOOST problems).

We think this may be more helpful if CoCoA-5 is compiled by some other script; if there's a problem then the error is signalled earlier.

**#21 - 07 Mar 2023 21:00 - John Abbott**

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

- % Done changed from 70 to 100

- Estimated time set to 14.01 h

I'm closing this issue -- I think it has become too long (20 comments).

The current configure script is definitely better than it used to be (but I know that the handling of readline needs to be improved -- maybe Ulrich can open a new issue about that?)