

## CoCoALib - Bug #975

### configure: possibly ignore GMP compilation flags

17 Nov 2016 19:23 - John Abbott

<b>Status:</b>	Closed	<b>Start date:</b>	17 Nov 2016
<b>Priority:</b>	High	<b>Due date:</b>	
<b>Assignee:</b>	John Abbott	<b>% Done:</b>	100%
<b>Category:</b>	Portability	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	CoCoALib-0.99560	<b>Spent time:</b>	4.20 hours
<b>Description</b>			
<p>Gereon had a strange problem. He was compiling on a cloud VM, and got strange error messages about the platform-specific compilation flags.</p> <p>The computer used MacOSX, and GMP had been installed via "brew". This left gmp.h with flags saying that it had been compiled on a "snadybridge" machine, but the actual host had an older processor. The compiler complained.</p> <p>While this is not really CoCoA's fault, it will probably happen again.</p> <p>Perhaps CoCoA could attempt to verify the acceptability of the flags taken from gmp.h by compiling a simple program; if that fails, compilation can be reattempted without the flags; if that fails, configuration should report a problem with the GMP installation. Otherwise it returns with GMP's platform flags, or an empty list.</p>			
<b>Related issues:</b>			
Related to CoCoALib - Bug #593: Temporary directories used during configuration		<b>Closed</b>	<b>20 Jul 2014</b>

### History

#### #1 - 17 Nov 2016 19:25 - John Abbott

I hope to resolve this while I'm here, and can see what happens on Gereon's cloud.

Of course, it is tedious having to do this because someone else was lazy... :-/

#### #2 - 17 Nov 2016 21:16 - John Abbott

- Status changed from New to In Progress

- Assignee set to John Abbott

- % Done changed from 0 to 50

I have modified the script gmp-cxx-flags.sh. A quick test on my machine passed (well, the first few tests didn't pass, but now it does).

Tomorrow I hope to check on Gereon's VM.

#### #3 - 18 Nov 2016 18:21 - John Abbott

- Related to Bug #593: Temporary directories used during configuration added

#### #4 - 18 Nov 2016 18:28 - John Abbott

If I recall well, with the clang compiler there is a compile-time error with the platform-specific flags from the "rogue" GMP header. However, with g++, the compiler generates executables, some of which may generate **Illegal instruction** errors at run-time... many CoCoALib tests failed, but many also passed.

The simple solution is not to use the compilation flags from the GMP header; but that won't work on my old MacBook (since the compiler defaults to 32-bits, but GMP is compiled with 64-bits).

Maybe I could add yet another configuration option to enable/disable getting the platform flags from gmp.h?

Past experience suggests that there are not normally such problems, so perhaps the default should be to extract the flags from gmp.h?

It could be tricky to write a script to decide automatically: I could do it for Gereon's machine, but then probably another similar problem will arise for a different platform and my gereon-specific test may not detect it... :-/

**#5 - 22 Nov 2016 12:42 - John Abbott**

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

- % Done changed from 50 to 80

I have added (yet another) configuration option `--ignore-gmp-cflags`.

Combining this with `--with-cxxflags=` should allow full control of the compilation flags.

Really Gereon should make the sysadmin install GMP properly, but I fear many people are in a similar situation.

Another approach would be to use "autotools" which is capable of detecting the system type. But I have heard that autotools is not so easy to use...

**#6 - 15 Jan 2017 17:47 - John Abbott**

Another email (from Matthias Koeppel) reported much the same problem as Gereon had.

In replying to him, I wondered "aloud" whether it would not be better to make the default behaviour be not to extract compilation flags from `gmp.h`, and to have a configuration flag which says explicitly to take flags from `gmp.h`. This will mean that I have to specify the `"get g++ flags from gmp"` option when compiling on my old MacBook, but I've known for a long time that my old MacBook was a peculiar platform (so it seems reasonable that I should use special flags on it).

Such a change would not be "backward compatible", but the incompatibility would be noticed only by those using peculiar old computers like mine.

Any comments? Other ideas?

**#7 - 17 Jan 2017 15:13 - Anna Maria Bigatti**

John Abbott wrote:

Such a change would not be "backward compatible", but the incompatibility would be noticed only by those using peculiar old computers like mine.

I admit I haven't understood this problem, nor the solution.  
Would this also affect me, as those with a local compilation of `gmp`?

**#8 - 17 Jan 2017 18:20 - John Abbott**

Anna: you can check whether you would have problems by configuring with the option `*--ignore-gmp-cflags@*` and then trying to compile. If compilation succeeds then you won't have any problem if I change the default behaviour to be "ignore GMP CFLAGS".

Let me know the outcome of your test.

**#9 - 18 Jan 2017 14:49 - Anna Maria Bigatti**

John Abbott wrote:

Anna: you can check whether you would have problems by configuring with the option `*--ignore-gmp-cflags@*` and then trying to compile. If compilation succeeds then you won't have any problem if I change the default behaviour to be "ignore GMP CFLAGS".

```
./configure --with-boost=/Users/bigatti/0.99/boost_1_61_0 --with-libgmp=/Users/bigatti/0.99/gmp-6.1.0/.libs/libgmp.a --ignore-gmp-cflags
make
.....
=====
Good news: all CoCoALib tests passed
=====
```

**#10 - 18 Jan 2017 15:05 - John Abbott**

OK, then I should try to change the default behaviour... this evening's "entertainment"? :-/

**#11 - 21 Jan 2017 18:07 - John Abbott**

- Status changed from Resolved to Feedback

- % Done changed from 80 to 90

I have changed it on my machines, but cannot currently check in (because cannot make VPN connection).

**#12 - 21 Jan 2017 22:25 - John Abbott**

One thing which puzzles me is that CoCoALib compiles fine and all tests pass on my old MacBook even without `-m64` flag... I found this out by forgetting to specify the `--acquire-gmp-cflags` configuration option. Moreover, the time taken to run all the tests is essentially the same... in other words, it appears that the "fancy" platform-specific flags do not produce any measurable gain in speed (as far as the CoCoALib tests are concerned).

**#13 - 23 Jan 2017 15:16 - John Abbott**

At this point I do not see any purpose in keeping even the option to extract compilation flags from `gmp.h`, so I now propose simply eliminating the relevant parts of the configure script (and any relevant auxiliary scripts).

Comments? Objections?

**#14 - 06 Nov 2017 16:43 - John Abbott**

I have made a preliminary version of configure and various auxiliary scripts which do not expect extra `CXXFLAGS` for GMP. Seems to work fine.

**#15 - 15 Dec 2017 15:08 - John Abbott**

- *Status changed from Feedback to Closed*

- *% Done changed from 90 to 100*

It has been working fine for several months with the extraction of GMP flags disabled by default.  
I still have to delete the relevant code from configure.