

CoCoALib - Feature #573

Use symbolic links for external libraries

18 Jun 2014 12:03 - John Abbott

Status:	Closed	Start date:	18 Jun 2014
Priority:	Normal	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	Portability	Estimated time:	9.90 hours
Target version:	CoCoALib-0.99550 spring 2017	Spent time:	9.30 hours
Description <p>To avoid problems with "awkward" path names to external libraries, I suggest creating symbolic links (with nice names!) inside the CoCoALib subtree to these libraries; this way all the "nastiness" is hidden inside the symbolic link.</p> <p>The problem arose because Christof wanted test-normaliz1 to depend on libnormaliz.a, so it is automatically rebuilt if libnormaliz.a is updated. The obvious solution did not work because the full path to libnormaliz.a is enclosed in double quotes (for safety), but make does not strip double quotes when interpreting dependencies, and reported that "libnormaliz.a" could not be found.</p> <p>Is this idea a good one? Can it be made to work? Comments, ideas, etc?</p>			
Related issues:			
Related to CoCoA-5 - Bug #712: External Libs: missing dependencies in Makefiles		In Progress	18 May 2015
Related to CoCoALib - Design #764: ExternalLib: simplify configuration		Closed	02 Sep 2015
Related to CoCoALib - Design #1704: Symlink directory for external files need...		New	21 Oct 2022

History

#1 - 19 Jun 2014 15:15 - John Abbott

- Status changed from New to In Progress
- Assignee set to John Abbott
- % Done changed from 0 to 10

I now think that the correct place for the external library directory is in configuration/.

It may as well be called ExtLibs.

Probably we should put GMP there too -- some experimenters may want to use their own private version of GMP (whose path could contain "nasty" characters).

I first trial editing autoconf.mk by hand (and creating the symlinks by hand) worked fairly well; there are still a couple of problems to iron out...

#2 - 19 Jun 2014 15:27 - Anna Maria Bigatti

John Abbott wrote:

I now think that the correct place for the external library directory is in configuration/.

It may as well be called ExtLibs.

I would call it ExternalLibs because other .C .H files are called ExternalLib...

Probably we should put GMP there too -- some experimenters may want to use their own private version of GMP (whose path could contain "nasty" characters).

I agree, and also boost: all linked libraries

#3 - 19 Jun 2014 17:00 - John Abbott

- % Done changed from 10 to 30

Here's a quite summary of the current situation (created entirely manually so far).

There's a new directory configuration/ExternalLibs/ which contains two subdirectories: include/ and lib/.

The subdirectory include/ contains a symlink to frobby.h, and 3 symlinks to directories (called boost, gsl, libnormaliz). Here is the relevant part of ls -l:

```
boost -> /usr/local/include/boost
frobby.h -> /Users/abbott/Work/FROBBY/frobby_v0.9.0/src/frobby.h
gsl -> /usr/local/include/gsl
libnormaliz -> /Users/abbott/Work/NORMALIZ/Normaliz2.11/source/libnormaliz
```

The subdirectory lib/ contains many symlinks to the relevant libraries: here's ls -l

```
libboost_filesystem.a -> /usr/local/lib/libboost_filesystem.a
libboost_system.a -> /usr/local/lib/libboost_system.a
libboost_thread.a -> /usr/local/lib/libboost_thread.a
libfrobby.a -> /Users/abbott/Work/FROBBY/frobby_v0.9.0/bin/libfrobby.a
libgmp.a -> /usr/local/lib/libgmp.a
libgmpxx.a -> /usr/local/lib/libgmpxx.a
libgsl.a -> /usr/local/lib/libgsl.a
libnormaliz.a -> /Users/abbott/Work/NORMALIZ/Normaliz2.11/source/libnormaliz/libnormaliz.a
```

#4 - 19 Jun 2014 17:53 - John Abbott

I thought I could be clever in autoconf.mk and use -lgsl, -lfrobby and so on.

But the linker does some strange acrobatics, and chooses shared libraries further down the search path than the static libraries I had placed at the start. :(

IDEA Maybe I could include the last 2 components of the path? **NO** -lExtLib/gsl does not work :(

IDEA2 I could change the names of the symlinks so that they differ from the true library names; it worked with e.g. libgmp-symlink.a meaning the ld

flag is -lgmp-symlink

#5 - 23 Jun 2014 17:34 - John Abbott

- % Done changed from 30 to 70

The symbolic link approach now seems to be working -- I even tried with a nasty directory name apparently without mishap.

The configure file and the script still need to be cleaned.

I'm also undecided about where temporary files (e.g. for compilation tests) should be put. An obvious choice is /tmp (which is automatically cleaned by the system every so often), but in some cases it is convenient to have them near configuration/ExternalLibs/.

#6 - 20 May 2015 17:24 - John Abbott

- Status changed from In Progress to Feedback

- % Done changed from 70 to 90

What is the current status? Has the cleaning been done? It has been working fine for several months, so changing status to feedback.

#7 - 20 May 2015 19:13 - Anna Maria Bigatti

John Abbott wrote:

What is the current status? Has the cleaning been done? It has been working fine for several months, so changing status to feedback.

you should ask Christof if it works for him now

#8 - 28 Apr 2017 11:31 - Anna Maria Bigatti

- Project changed from CoCoA to CoCoALib

- Category changed from Portability to Portability

- Target version set to CoCoALib-0.99550 spring 2017

I think this is well tested now.

This issue was under "CoCoA" instead of "CoCoALib".

I'm recovering these old and forgotten issues, so we reconsider them.

#9 - 28 Apr 2017 13:37 - John Abbott

- Status changed from Feedback to Closed

- % Done changed from 90 to 100

- Estimated time changed from 8.00 h to 9.90 h

This has been in feedback for ages (more than 2 years). I have tested on various different Linux platforms, and Anna must have tested it on different MacOSX versions without problem.

Closing.

#10 - 21 Oct 2022 16:01 - Anna Maria Bigatti

- *Related to Design #1704: Symlink directory for external files needed by CoCoA-5 added*