

## CoCoA-5 - Bug #1713

### Windows: Qt-Gui does not link correctly

22 Nov 2022 23:44 - Nico Mexis

<b>Status:</b>	In Progress	<b>Start date:</b>	22 Nov 2022
<b>Priority:</b>	Low	<b>Due date:</b>	
<b>Assignee:</b>	Nico Mexis	<b>% Done:</b>	50%
<b>Category:</b>	GUI	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	CoCoA-5.?.?	<b>Spent time:</b>	0.85 hour
<b>Description</b>			
<p>As discussed in <a href="#">#1712</a>, problems with the Qt-Gui should be handled in another issue.</p> <p>When trying to link the Qt-Gui, there are many undefined references such as this one:</p> <pre>/usr/lib/gcc/x86_64-pc-cygwin/11/../../../../x86_64-pc-cygwin/bin/ld: /cygdrive/f/Misc/SelfBuilt/CoCoA5/ExtLibs/normaliz-3.9.4/source/libnormaliz/cone.cpp:2105: undefined reference to `omp_get_max_threads'</pre> <p>I actually have not tested linking the Qt-Gui without normaliz support enabled. Most (if not all) of the OpenMP-related linker issues seem to be related to normaliz and can be fixed by adding the <code>-lgomp</code> flag inside <code>C5.pro.in</code>. However, it might be best to use <code>-fopenmp</code> instead (I have not tested that one yet either).</p> <p>When adding <code>-lgomp</code> to <code>C5.pro.in</code>, I get another bunch of linker issues like this one:</p> <pre>/usr/lib/gcc/x86_64-pc-cygwin/11/../../../../x86_64-pc-cygwin/bin/ld: GUI-release/C5.o:C5.C:(.text+0x3d): undefined reference to `__imp__ZN19QEditorInputBindingD2Ev'</pre> <p>This time, they seem to be related to <code>QCodeEdit</code>. However, <code>C5.pro.in</code> and the generated <code>C5.pro</code> files both try to link against <code>-LQCodeEdit-lqcodeedit</code> already. So, I have no clue what is going wrong there. All the linker errors after adding <code>-lgomp</code> can be found in the attachments.</p> <p>It should also be mentioned that I am forced to use Qt 5.9.4 since the Qt distributions in CygWin are apparently unmaintained and have not been updated since then.</p>			
<b>Related issues:</b>			
Related to CoCoALib - Bug #1712: Windows: New configuration script		<b>Closed</b>	<b>16 Nov 2022</b>

### History

#### #1 - 23 Nov 2022 10:09 - John Abbott

Below is a copy of my notes about building libnormaliz from source -- I have version 3.9.4. I use a "personal installation": so I specified the `--prefix` option when configuring. It seems that `--disable-shared` is not important (at least on my old MacBook)

```
# Website https://www.normaliz.uni-osnabrueck.de/download/
```

```
cd normaliz-3.9.4
export GMP_INSTALLDIR=/usr/local
./configure --disable-shared --disable-openmp # --prefix=/home/jabbott/XYZ (installation DIR)
```

```
make # took about 4 mins' CPU
--> result is in source/libnormaliz.la or source/.libs/libnormaliz.a
make check # took about 1 minute
```

```
make install # Went into /usr/local/include/, /usr/local/lib/ and /usr/local/bin (normaliz prog)
```

## #2 - 23 Nov 2022 10:57 - Nico Mexis

John Abbott wrote:

Below is a copy of my notes about building libnormaliz from source -- I have version 3.9.4.  
I use a "personal installation": so I specified the --prefix option when configuring.  
It seems that --disable-shared is not important (at least on my old MacBook)

[...]

Thank you very much. After disabling OpenMP on normaliz's side, the OpenMP errors no longer occur. However, the other QCodeEdit-related ones sadly persist.

## #3 - 30 Nov 2022 15:57 - John Abbott

- Status changed from New to In Progress
- Target version set to CoCoA-5.?.?
- % Done changed from 0 to 10

I am not sure who would be able to help with this (Qt-gui on Microsoft)

I have set target version to "sooner or later".

## #4 - 14 Mar 2023 10:07 - John Abbott

- Related to Bug #1712: Windows: New configuration script added

## #5 - 17 Feb 2024 12:05 - Nico Mexis

- File GUI-finally.png added

I did not give up that easily on this issue: Today, I was finally able to compile the Qt5-Gui on Windows!

Recently, in another research project which was written in C++, I came across a very similar issue and so I started searching for these errors I got when building the Qt5-Gui for CoCoA again.

This is what I found: <https://stackoverflow.com/a/15727160/5894824>

And actually, that did the trick!

So, what was wrong?

When compiling QCodeEdit, the QCodeEdit.pro file contains the following line: CONFIG += qt release warn\_off staticlib thread

This tells the build system to compile QCodeEdit as a **static** library where function names can get a bit mangled (as seen above with the \_\_imp\_\_[...] prefixes).

When removing this specifier (i.e., changing lines 7 and 8 in QCodeEdit/QCodeEdit.pro.in and removing both staticlib specifiers), QCodeEdit gets linked and included correctly (as a **shared** library).

So, as long as this does not break Linux builds, I would suggest changing these two lines.

I will try to write up everything I needed to do in order to make the C5 GUI compile and add it to my guide mentioned in [#1661](#).

It's mainly about packages one needs to install - everything else worked pretty much out of the box (except for changing these two lines mentioned above).

Oh, and also: starting the GUI is not very straightforward... Qt5 applications compiled using CygWin only work using a XServer. This is why I needed to use xstart together with VcXsrv installed on my host machine. I shall also document that!

P.S.: I have included a screenshot of the freshly compiled GUI running on my Windows machine (the errors from the console on the right side occur when resizing the C5 GUI window...) :-)

**#6 - 17 Feb 2024 15:41 - Nico Mexis**

- % Done changed from 10 to 50

I have updated my documentation file. Actually, only three CygWin packages are needed in order to successfully build the GUI. Now, the only thing left is to change lines 7 and 8 in QCodeEdit/QCodeEdit.pro.in and remove both staticlib specifiers.

**#7 - 17 Feb 2024 21:07 - John Abbott**

- Assignee set to Nico Mexis

Great! Thanks for sorting this out. I shall test it (on Linux) shortly -- let's hope it works!

**#8 - 17 Feb 2024 21:27 - John Abbott**

Partial success: compilation succeeded, but attempting to run the resulting executable produced an error **Cannot find shared object...** presumably because libqcodeedit.so is not in an approved directory for shared libraries.

I must think about what to do. Also there could be a problem if we want to distribute the C5 executable, since we would have to include the shared library as well (together with instructions on what to do with it).

Or maybe I make the contents of QCodeEdit.pro.in depend on the platform...?

**#9 - 17 Feb 2024 22:58 - Nico Mexis**

These differences between the linkers on Windows and Linux are really unfortunate sometimes...

In that case, I would leave it as-is for now. If there is a quick way to make the contents of QCodeEdit.pro.in depend on the current platform, that would obviously be even better.

For now, I have included another step in the instructions to change these two lines when being on a Windows machine.

And yes, the resulting library for QCodeEdit will need to be included in the binary distribution when linking it "un-static-ally".

I have looked a bit further into what I could do on the Windows side, but everything else didn't work sadly.

When I find something else, I will try it out, but for now I am out of ideas regarding this

**#10 - 24 Feb 2024 21:00 - Ulrich von der Ohe**

John Abbott wrote:

Partial success: compilation succeeded, but attempting to run the resulting executable produced an error **Cannot find shared object...** presumably because libqcodeedit.so is not in an approved directory for shared libraries.

One way to solve this problem on unixoid systems is to adjust the RUNPATH in the header of the C5 binary so that the library can be found by the loader.

One simple way of achieving that here is to insert the line

QMAKE\_RPATHDIR += QCodeEdit

into src/CoCoA-5/C5.pro.in. On several common platforms, qmake will then take appropriate precautions (likely set RUNPATH to \$ORIGIN/QCodeEdit), so that the necessary shared object files are found in the subdirectory QCodeEdit.

This should be a good solution on platforms where qmake supports setting relative RUNPATHs, GNU/Linux and Darwin among them (see [here](#)). Lower level solutions are possible on other platforms, should anyone wish to run the GUI on them.

**#11 - 25 Feb 2024 12:39 - John Abbott**

Thanks Ulrich! There's a lot I do not know about Qt (and many other things too). I hope to try out your suggestion soon: with my "new" job in Kaiserslautern, I do not have so much time for CoCoA any longer :-)

**#12 - 25 Feb 2024 19:04 - John Abbott**

It seems to have worked on my laptop (Ubuntu). I'll do a few more checks, but hope to include the change in the next interim release.

**#13 - 25 Feb 2024 20:42 - Nico Mexis**

I can confirm that adding QMAKE\_RPATHDIR += QCodeEdit does not break the build on Windows. Thank you @Ulrich von der Ohe for finding out about that and mentioning it!

**Files**

---

linker.txt	10.8 KB	22 Nov 2022	Nico Mexis
GUI-finally.png	697 KB	17 Feb 2024	Nico Mexis