



Published on *cisTEM* (<https://cistem.org>)

[Home](#) > How to compile self-contained cisTEM binaries?

How to compile self-contained cisTEM binaries?

Tue, 03/10/2020 - 14:41

#1

dxs2007

How to compile self-contained cisTEM binaries?

Hello, I am trying to create cisTEM binaries such as reconstruct3d and merge3d that I can move around and run from any folder.

Currently, I'm creating the binaries from cistem-1.0.0-beta using the docker container here https://hub.docker.com/r/arohou/cistem_dev_env on my HPC cluster.

The binaries run fine within the docker container, but they give this error when I run them on the cluster: `./merge3d: error while loading shared libraries: libwx_baseu-3.0.so.0: cannot open shared object file: No such file or directory`

I tried running `./configure` with the `--disable-shared` and `--enable-staticmode` but the error remains.

Would a developer be willing to assist? Thank you.

alexis

Hi - this is not something I

Hi - this is not something I've done recently, and Tim may have better ideas, but as a first attempt, I guess I would check whether static wx libraries are even available for linking in my docker container. Is it possible that it attempts to link against static wx, but that goes wrong, and then it defaults back to linking against dynamic wx? I'm sorry that I don't remember those details from when I put this Docker together - bad memory!

Cheers,
Alexis

dxs2007

Hi Alexis, thank you for your

Hi Alexis, thank you for your reply. I believe there are static wx libraries as you can see in the .a files here

```
Singularity> cd /usr/local/lib/  
Singularity> ls
```

```
...  
libwx_baseu-3.0.a          libwx_gtk2u_ribbon-3.0.so  
libwx_baseu-3.0.so        libwx_gtk2u_ribbon-3.0.so.0  
libwx_baseu-3.0.so.0      libwx_gtk2u_ribbon-3.0.so.0.4.0  
libwx_baseu-3.0.so.0.4.0  libwx_gtk2u_richtext-3.0.a  
...
```

I went over the "make" command log and they look something like this (the -static flag is present, but there's also this "/usr/local/lib/libtiff.so" file in the command, which might messing things up):

```
libtool: link: g++ -I/usr/local/lib/wx/include/gtk2-unicode-3.0 -  
I/usr/local/include/wx-3.0 -D_FILE_OFFSET_BITS=64 -DWXUSINGDLL -  
D__WXGTK__ -DwxUSE_GUI=0 -DNDEBUG -funroll-loops -O3 -Wall -  
pipe -fexpensive-optimizations -D_FILE_OFFSET_BITS=64 -  
D_LARGEFILE_SOURCE -o refine2d programs/refine2d/refine2d-  
refine2d.o -pthread -L/usr/local/lib libcore.a -lwx_baseu-3.0 -  
lwx_baseu_net-3.0 /usr/local/lib/libtiff.so -lm -ldl -lfftw3f -pthread
```

I checked the /usr/local/lib folder and there's also the static libtiff.a file, hence might the linking against "/usr/local/lib/libtiff.so" instead of "/usr/local/lib/libtiff.a" be the problem?

```
ls libtiff*  
libtiff.a libtiff.so libtiff.so.5.4.0 libtiffxx.la libtiffxx.so.5  
libtiff.la libtiff.so.5 libtiffxx.a libtiffxx.so libtiffxx.so.5.4.0
```

Thank you.

dxo2007

Apologies, the full output

Apologies, the full output for one binary should look something like this:

```
/bin/bash ../libtool --tag=CXX --mode=link g++ -  
I/usr/local/lib/wx/include/gtk2-unicode-3.0 -I/usr/local/include/wx-3.0 -  
D_FILE_OFFSET_BITS=64 -DWXUSINGDLL -D__WXGTK__ -  
DwxUSE_GUI=0 -DNDEBUG -funroll-loops -O3 -Wall -pipe -fexpensive-  
optimizations -D_FILE_OFFSET_BITS=64 -D_LARGEFILE_SOURCE -  
L/usr/local/lib -static -o refine2d programs/refine2d/refine2d-refine2d.o  
libcore.a -L/usr/local/lib -pthread -lwx_baseu-3.0 -lwx_baseu_net-3.0 -  
ltiff -ldl -lfftw3f
```

```
libtool: link: g++ -I/usr/local/lib/wx/include/gtk2-unicode-3.0 -  
I/usr/local/include/wx-3.0 -D_FILE_OFFSET_BITS=64 -DWXUSINGDLL -  
D__WXGTK__ -DwxUSE_GUI=0 -DNDEBUG -funroll-loops -O3 -Wall -  
pipe -fexpensive-optimizations -D_FILE_OFFSET_BITS=64 -  
D_LARGEFILE_SOURCE -o refine2d programs/refine2d/refine2d-  
refine2d.o -pthread -L/usr/local/lib libcore.a -lwx_baseu-3.0 -  
lwx_baseu_net-3.0 /usr/local/lib/libtiff.so -lm -ldl -lfftw3f -pthread
```


Tue, 03/10/2020 - 17:02 (Reply to #4)

alexis

OK, I also am unable to

OK, I also am unable to produce completely static binaries right now. Obviously it's possible to do so, because Tim makes such binaries for releases etc. But I don't have much time to investigate right now. Tim should be able to comment, though he is not online very frequently at the moment, so it may take him a little while. Thanks for your patience.

Tue, 03/10/2020 - 19:42 (Reply to #5)

dxs2007

Sure Alexis thank you for

Sure Alexis thank you for your help. I'll wait for Tim's reply.

timgrant

Hi,

Hi,

The way that I do this is to also compile wxWidgets myself, and configure it with the --disable-shared option. If you are using the released cisTEM code then you need to use wxWidgets 3.0.2 (<https://sourceforge.net/projects/wxwindows/files/3.0.2/>).

To be specific, this is my configure line for wxWidgets :-

```
CXX=icpc CC=icc ./configure --disable-precomp-headers --  
prefix=/directory_to_install_to --with-libnotify=no --disable-shared --  
without-gtkprint --with-libjpeg=builtin --with-libpng=builtin --with-  
libtiff=builtin --with-zlib=builtin --with-expat=builtin --disable-compat28 --  
without-liblzma --without-libjbig
```

Obviously you need to replace "--prefix=/directory_to_install_to" with the actual directory. Then do make, make install

After this, configure cisTEM with "--with-wx-config=path_to_your_wx_config" and the --enable-staticmode.

Thanks,

Tim

Mon, 03/23/2020 - 18:33 [\(Reply to #7\)](#)

Thanks Tim!

As I'm compiling cisTEM on a HPC, there are certain modules that I don't have access to, such as the icc compiler. Hence, I wasn't able to follow your compilation method.

However, I used another method to compile cisTEM and it seems to be working

1. Run Alexis' VM (https://hub.docker.com/r/arohou/cistem_dev_env) on HPC
2. On the VM, compile cisTEM using
 1. `./configure --prefix=/hpc/home/xd24 --enable-staticmode`
 2. `make`
3. Manually compile the specific binaries that I need (reconstruct3d and merge3d)
 1. `cd` into /src folder
 2. `g++ -DPACKAGE_NAME=\"cisTEM\" -DPACKAGE_TARNAME=\"cistem\" -DPACKAGE_VERSION=\"1.0.0-beta\" -DPACKAGE_STRING=\"cisTEM1.0.0-beta\" -DPACKAGE_BUGREPORT=\"\" -DPACKAGE_URL=\"\" -DPACKAGE=\"cistem\" -DVERSION=\"1.0.0-beta\" -DSTDC_HEADERS=1 -DHAVE_SYS_TYPES_H=1 -DHAVE_SYS_STAT_H=1 -DHAVE_STDLIB_H=1 -DHAVE_STRING_H=1 -DHAVE_MEMORY_H=1 -DHAVE_STRINGS_H=1 -DHAVE_INTTYPES_H=1 -DHAVE_STDINT_H=1 -DHAVE_UNISTD_H=1 -DHAVE_DLFCN_H=1 -DLT_OBJDIR=\".libs/\" -Dwx_is_available=1 -I. -I/usr/local/lib/wx/include/gtk2-unicode-3.0 -I/usr/local/include/wx-3.0 -D_FILE_OFFSET_BITS=64 -DWXUSINGDLL -D__WXGTK__ -DwxUSE_GUI=0 -DNDEBUG -funroll-loops -O3 -Wall -pipe -fexpensive-optimizations -D_FILE_OFFSET_BITS=64 -D_LARGEFILE_SOURCE -I/usr/local/lib/wx/include/gtk2-unicode-3.0 -I/usr/local/include/wx-3.0 -D_FILE_OFFSET_BITS=64 -DWXUSINGDLL -D__WXGTK__ -DwxUSE_GUI=0 -DNDEBUG -funroll-loops -O3 -Wall -pipe -fexpensive-`

```
optimizations -D_FILE_OFFSET_BITS=64 -  
D_LARGEFILE_SOURCE -MT programs/merge3d/merge3d-  
merge3d.o -MD -MP -MF programs/merge3d/.deps/merge3d-  
merge3d.Tpo -c -o programs/merge3d/merge3d-merge3d.o  
`test -f 'programs/merge3d/merge3d.cpp' || echo  
`.` programs/merge3d/merge3d.cpp
```

3. /bin/bash ../libtool --tag=CXX --mode=link g++ -
l/usr/local/lib/wx/include/gtk2-unicode-3.0 -
l/usr/local/include/wx-3.0 -D_FILE_OFFSET_BITS=64 -
DWXUSINGDLL -D__WXGTK__ -DwxUSE_GUI=0 -
DNDEBUG -funroll-loops -O3 -Wall -pipe -fexpensive-
optimizations -D_FILE_OFFSET_BITS=64 -
D_LARGEFILE_SOURCE -L/usr/local/lib -all-static -o
merge3d programs/merge3d/merge3d-merge3d.o libcore.a -
L/usr/local/lib -pthread -lwx_baseu-3.0 -lwx_baseu_net-3.0 -
ltiff -ldl -lfftw3f

olibclarke

Related to this, are there

Related to this, are there any binaries available for Mac OS X? I have tried compiling with no luck - I can't get the specific version of wxWidgets required to compile.

I'm not going to run any heavy duty processing on my mac, but I would like to have some of the smaller utilities like `sharpen_map` and `resample_map` available - if wxWidgets is only needed for the GUI is there any way to compile in a headless mode?

Cheers

Oli

Mon, 06/01/2020 - 12:53 [\(Reply to #9\)](#)

timgrant

Hi Oli,

Hi Oli,

You need wxwidgets to compile the text based program also. I don't currently have a mac, but I have done this before and it should work for the text programs. The gui can be compiled, but does not run properly.

If you run ./configure - does it give you an error?

Tim

Wed, 06/03/2020 - 11:50 (Reply to #10)

olibclarke

Hi Tim,

Hi Tim,

Thanks for the reply! When I run ./configure I get an error saying my Wxwidgets version (provided by homebrew) is too new:

```
checking wxWidgets version... 3.0.3  
configure: error: "Only wxWidgets version 3.0.2 is currently supported."
```

There are no older versions provided by homebrew, and I have had no luck compiling wxwidgets separately.

Cheers

Oli

Wed, 06/03/2020 - 12:01 [\(Reply to #11\)](#)

timgrant

Hi Oli,

Hi Oli,

Well, if you only want to use the console programs, you can probably remove this restriction. if you edit the file `configure.ac` and change line 144 from :-

```
if test -n "$vers" && test "$vers" -eq 3000002; then
```

to

```
if test -n "$vers" && test "$vers" -ge 3000002; then
```

(basically change the `eq` to `ge`) then run `autoconf` in that directory and rerun `./configure` I think it should work.

Cheers,

Tim

Wed, 06/03/2020 - 12:26 [\(Reply to #12\)](#)

olibclarke

Hi Tim,

Hi Tim,

Still no go unfortunately. I did as you said, and after installing autoconf (also had to install automake, m4 and perl, and changing the version numbers of aclocal and automake in the Makefile to match the ones provided by brew (1.16 for both), it fails during make with the following error:

```
In file included from programs/projectx/projectx.cpp:2:  
programs/projectx/../../core/gui_core_headers.h:1:10: fatal error:  
'wx/artprov.h' file not found  
#include <wx/artprov.h>  
      ^~~~~~  
1 error generated.
```

Any way around this one?

Thanks for your help!

Cheers

Oli

Wed, 06/03/2020 - 12:33 [Reply to #13](#)

timgrant

Hi Oli,

Hi Oli,

You can try editing the `src/Makefile.am` and editing line 195
"`bin_PROGRAMS =` " to remove `cisTEM` (or even to remove everything
but the programs you want). Then run `automake` and try again.

Cheers,

Tim

Wed, 06/03/2020 - 12:42 [\(Reply to #14\)](#)

olibclarke

Thanks Tim - I tried that,

Thanks Tim - I tried that, but I still get this error for the command line programs:

```
programs/sharpen_map/../../core/core_headers.h:38:10: fatal error:  
'wx/wx.h' file not found
```

```
#include <wx/wx.h>
```

```
    ^~~~~~
```

```
1 error generated.
```

```
make[1]: *** [programs/sharpen_map/sharpen_map-sharpen_map.o]  
Error 1
```

Any workaround for this one? sorry for the hassle!

Oli

Wed, 06/03/2020 - 12:45 [\(Reply to #15\)](#)

olibclarke

The only other thing I notice

The only other thing I notice in the config log is:

```
./configure: line 3991: AC_LIBTOOL_DLOPEN: command not found
```

```
./configure: line 3992: AC_PROG_LIBTOOL: command not found
```

But I have libtool installed...

Wed, 06/03/2020 - 12:50 (Reply to #16)

timgrant

if you run `autoreconf` and

if you run `autoreconf` and `libtoolize` in the main directory, it will hopefully fix those errors.

As for not finding `wx.h` that is weird, as it is the main `wxWidgets` header, and so it implies your `wxWidgets` is not found or setup correctly. Can you confirm you can find the `wx-config` executable, and maybe run `configure` pointing at it directly `./configure --with-wx-config=<path_to_your_wx_config>`

Tim

Wed, 06/03/2020 - 12:56 [\(Reply to #17\)](#)

olibclarke

Hi Tim,

Hi Tim,

I have wx-config:

```
`MacBook-Pro:cistem-1.0.0-beta_2 olibclarke1$ which wx-config  
/Applications/ccp4-7.1/bin/wx-config`
```

But I still see the same error after re-running configure and make with the flags you suggest...

Wed, 06/03/2020 - 12:57 [\(Reply to #18\)](#)

olibclarke

and yes, re-running

and yes, re-running autoreconf removed the errors pertaining to libtool -
just the one relating to wx remaining

Wed, 06/03/2020 - 13:01 [\(Reply to #19\)](#)

olibclarke

Ah, works now. CCP4 was

Ah, works now. CCP4 was providing some version of wx that was fighting with the version installed by brew. Running in a shell where I removed CCP4 from my bashrc, it now runs correctly and the programs are compiled!

Thanks so much for your help :)

Wed, 06/03/2020 - 13:14 [\(Reply to #20\)](#)

olibclarke

In case it is useful to

In case it is useful to anyone else, here is a link to the standalone mac binaries:

https://www.dropbox.com/s/3ymatkium0fhioh/cistem_mac_binaries.tar.gz?dl=0

GUI doesn't work but the command line programs seem to function.

Cheers

Oli

Wed, 06/03/2020 - 13:33 [\(Reply to #21\)](#)

timgrant

Great that is works - and

Great that is works - and thank you for posting the binaries!

Tim

olibclarke

Thanks Tim - I tried that,

Thanks Tim - I tried that, but I still get this error for the command line programs:

```
programs/sharpen_map/../../core/core_headers.h:38:10: fatal error:  
'wx/wx.h' file not found
```

```
#include <wx/wx.h>
```

```
    ^~~~~~
```

```
1 error generated.
```

```
make[1]: *** [programs/sharpen_map/sharpen_map-sharpen_map.o]
```

```
Error 1
```

Any workaround for this one? sorry for the hassle!

Oli

olibclarke

Any tips on compiling for

Any tips on compiling for apple silicon (ARM64)?

I get through configure ok, but once I get to `make` I run into trouble, which I suspect is an architecture issue:


```

core/matrix.cpp:208:5: error: invalid output constraint '=t' in asm
    MAKE_ROTATION_f(x, y, z);
    ^
core/matrix.cpp:16:5: note: expanded from macro 'MAKE_ROTATION_f'
    FLOATSINCOS(x, sin_x, cos_x);          \
    ^
core/matrix.cpp:8:31: note: expanded from macro 'FLOATSINCOS'
#define FLOATSINCOS(x, s, c) _AL_SINCOS((x) * AL_PI / 128.0, s, c)
    ^
core/matrix.cpp:7:51: note: expanded from macro '_AL_SINCOS'
#define _AL_SINCOS(x, s, c) __asm__ ("fsincos" : "=t" (c), "=u" (s) : "0"
(x))
    ^
core/matrix.cpp:208:5: error: invalid output constraint '=t' in asm
core/matrix.cpp:17:5: note: expanded from macro 'MAKE_ROTATION_f'
    FLOATSINCOS(y, sin_y, cos_y);          \
    ^
core/matrix.cpp:8:31: note: expanded from macro 'FLOATSINCOS'
#define FLOATSINCOS(x, s, c) _AL_SINCOS((x) * AL_PI / 128.0, s, c)
    ^
core/matrix.cpp:7:51: note: expanded from macro '_AL_SINCOS'
#define _AL_SINCOS(x, s, c) __asm__ ("fsincos" : "=t" (c), "=u" (s) : "0"
(x))
    ^
core/matrix.cpp:208:5: error: invalid output constraint '=t' in asm
core/matrix.cpp:18:5: note: expanded from macro 'MAKE_ROTATION_f'
    FLOATSINCOS(z, sin_z, cos_z);          \
    ^
core/matrix.cpp:8:31: note: expanded from macro 'FLOATSINCOS'
#define FLOATSINCOS(x, s, c) _AL_SINCOS((x) * AL_PI / 128.0, s, c)
    ^
core/matrix.cpp:7:51: note: expanded from macro '_AL_SINCOS'
#define _AL_SINCOS(x, s, c) __asm__ ("fsincos" : "=t" (c), "=u" (s) : "0"
(x))
    ^
3 errors generated.
make[1]: *** [core/libcore_a-matrix.o] Error 1
make: *** [all-recursive] Error 1

```


timgrant

Yes, I think those functions

Yes, I think those functions use some assembly which would be X86 specific. If you replace them with standard C functions, it would probably fix it.

Tim

[Log in](#) or [register](#) to post comments

Source URL: <https://cistem.org/how-compile-self-contained-cistem-binaries>