

## Doomsday Engine - Bug #2133

### Building on Fedora, CMake issues (linking minizip in Assimp target)

2015-11-27 21:00 - Calinou

<b>Status:</b>	Closed	<b>Start date:</b>	2015-11-27
<b>Priority:</b>	Normal	<b>% Done:</b>	100%
<b>Assignee:</b>	skyjake		
<b>Category:</b>	Defect		
<b>Target version:</b>			
<b>Description</b>			
Hello,			
I cloned latest Doomsday Git from GitHub (on 2015-11-27) on Fedora 22 64-bit. I have the required dependencies, and attempted to generate build files using CMake GUI (CMake command-line complains about in-source builds not being allowed, even if building from a directory called "bin" and using "cmake .."),			
This is the error:			
<pre>@Configuring doomsday... Configuring sdk... Configuring libcore... Revision: build1791-0-gb7eef70 Wrote sdk/libcore/net.dengine.stdlib.pack (contains 4 files). Configuring libshell... Configuring liblegacy... CMake Error at sdk/libgui/CMakeLists.txt:17 (add_subdirectory): The source directory  /home/hugo/Git/doomsday-engine/doomsday/external/assimp  does not contain a CMakeLists.txt file.  CMake Error at sdk/libgui/CMakeLists.txt:18 (set_property): set_property could not find TARGET assimp. Perhaps it has not yet been created.  Configuring libgui... Wrote sdk/libgui/net.dengine.stdlib.gui.pack (contains 2 files). Configuring libappfw... Configuring tools... Configuring doomsdayscript... Configuring md2tool... Configuring savegametool... Configuring shell... Configuring shell-text... Configuring texc... Configuring wadtool... Configuring apps... Configuring libdoomsday... Wrote apps/libdoomsday/net.dengine.base.pack (contains 6 files). Configuring plugins... Configuring example... Configuring idtech1converter... Configuring savegameconverter... Configuring dehread... Configuring doom... Compiling legacy PK3s... Configuring heretic... Configuring hexen... Configuring doom64... Configuring fmod... Configuring openal...</pre>			

Configuring fluidsynth...  
Configuring server...  
Configuring client...  
Wrote apps/client/net.dengine.client.pack (contains 91 files).  
Configuring tests...  
Configuring incomplete, errors occurred!  
See also `"/home/hugo/Git/doomsday-engine/doomsday/bin/CMakeFiles/CMakeOutput.log".@`

I do have Assimp installed on my system. It looks like the bundled one isn't building properly though.

---

## Associated revisions

### Revision e1bb1ee6 - 2016-05-02 19:31 - skyjake

Unix|CMake|libgui: Link with zlib

IssueID #2133

### Revision 4631b967 - 2016-05-06 10:17 - skyjake

Fixed|CMake|Assimp: Linking Assimp's system dependencies

libassimp is a static library, however it may have system libraries as dependencies. The libassimp interface target must take these into account.

IssueID #2133

---

## History

### #1 - 2015-11-27 21:34 - Calinou

**Partially solved:** I had to initialize submodules using `git submodule update --init`. Alternatively, the repository can be cloned using the `--recursive` option.

However, I now get this when linking (at the end of the build process):

```
[ 0%] Automatic moc and rcc for target libcore
[ 0%] Built target libcore_automoc
[ 10%] Built target libcore
[ 10%] Automatic moc and rcc for target libshell
[ 10%] Built target libshell_automoc
[ 11%] Built target libshell
[ 11%] Automatic moc and rcc for target liblegacy
[ 11%] Built target liblegacy_automoc
[ 13%] Built target liblegacy
[ 13%] Automatic moc and rcc for target libgui
[ 13%] Built target libgui_automoc
[ 13%] Automatic moc and rcc for target assimp
[ 13%] Built target assimp_automoc
[ 23%] Built target assimp
[ 26%] Built target libgui
[ 27%] Automatic moc and rcc for target libappfw
[ 27%] Built target libappfw_automoc
[ 30%] Built target libappfw
[ 30%] Automatic moc and rcc for target doomsdayscript
[ 30%] Built target doomsdayscript_automoc
[ 31%] Built target doomsdayscript
[ 31%] Automatic moc and rcc for target md2tool
[ 31%] Built target md2tool_automoc
[ 31%] Built target md2tool
[ 31%] Automatic moc and rcc for target savegametool
[ 31%] Built target savegametool_automoc
[ 31%] Automatic rcc for target lzss
[ 31%] Built target lzss_automoc
[ 32%] Built target lzss
[ 33%] Built target savegametool
[ 33%] Automatic moc and rcc for target shell
[ 33%] Built target shell_automoc
[ 34%] Built target shell
[ 34%] Automatic moc and rcc for target shell-text
[ 34%] Built target shell-text_automoc
[ 35%] Built target shell-text
[ 35%] Automatic moc and rcc for target texc
[ 35%] Built target texc_automoc
```

```

[ 35%] Built target texc
[ 36%] Automatic moc and rcc for target wadtool
[ 36%] Built target wadtool_automoc
[ 36%] Built target wadtool
[ 36%] Automatic moc and rcc for target libdoomsday
[ 36%] Built target libdoomsday_automoc
[ 40%] Built target libdoomsday
[ 40%] Automatic moc and rcc for target example
[ 40%] Built target example_automoc
[ 40%] Built target example
[ 41%] Automatic moc and rcc for target idtech1converter
[ 41%] Built target idtech1converter_automoc
[ 41%] Built target idtech1converter
[ 41%] Automatic moc and rcc for target savegameconverter
[ 41%] Built target savegameconverter_automoc
[ 41%] Built target savegameconverter
[ 41%] Automatic moc and rcc for target dehread
[ 41%] Built target dehread_automoc
[ 41%] Built target dehread
[ 41%] Automatic moc and rcc for target doom
[ 41%] Built target doom_automoc
[ 49%] Built target doom
[ 49%] Automatic moc and rcc for target heretic
[ 49%] Built target heretic_automoc
[ 57%] Built target heretic
[ 57%] Automatic moc and rcc for target hexen
[ 57%] Built target hexen_automoc
[ 66%] Built target hexen
[ 66%] Automatic moc and rcc for target doom64
[ 66%] Built target doom64_automoc
[ 74%] Built target doom64
[ 74%] Automatic moc and rcc for target audio_openal
[ 74%] Built target audio_openal_automoc
[ 75%] Built target audio_openal
[ 75%] Automatic moc and rcc for target audio_fluidsynth
[ 75%] Built target audio_fluidsynth_automoc
[ 75%] Built target audio_fluidsynth
[ 75%] Automatic moc and rcc for target server
[ 75%] Built target server_automoc
[ 82%] Built target server
[ 83%] Automatic moc and rcc for target client
[ 83%] Built target client_automoc
[ 83%] Linking CXX executable doomsday
../../sdk/libgui/libdeng_gui.so.2.0.0: undefined reference to `unzGoToFirstFile'
../../sdk/libgui/libdeng_gui.so.2.0.0: undefined reference to `unzClose'
../../sdk/libgui/libdeng_gui.so.2.0.0: undefined reference to `unzReadCurrentFile'
../../sdk/libgui/libdeng_gui.so.2.0.0: undefined reference to `unzOpen'
../../sdk/libgui/libdeng_gui.so.2.0.0: undefined reference to `unzOpenCurrentFile'
../../sdk/libgui/libdeng_gui.so.2.0.0: undefined reference to `unzGetCurrentFileInfo'
../../sdk/libgui/libdeng_gui.so.2.0.0: undefined reference to `unzOpen2'
../../sdk/libgui/libdeng_gui.so.2.0.0: undefined reference to `unzGoToNextFile'
../../sdk/libgui/libdeng_gui.so.2.0.0: undefined reference to `unzCloseCurrentFile'
collect2: error: ld returned 1 exit status
apps/client/CMakeFiles/client.dir/build.make:7262: recipe for target 'apps/client/doomsday-2.0.0' failed
make2: * [apps/client/doomsday-2.0.0] Error 1
CMakeFiles/Makefile2:2903: recipe for target 'apps/client/CMakeFiles/client.dir/all' failed
make1: [apps/client/CMakeFiles/client.dir/all] Error 2
Makefile:149: recipe for target 'all' failed
make: ** [all] Error 2

```

## #2 - 2015-11-28 10:30 - skyjake

- Tags set to Fedora, Linux, CMake
- Status changed from New to In Progress
- Assignee set to skyjake
- Target version set to 49

## #3 - 2015-11-28 11:45 - skyjake

I tried a clean build of the master on Fedora 21, and the compilation was successful. I'll see if I can try a newer version of Fedora, but the error messages hint that Assimp wasn't built quite right (it has the unzip functions built-in). Did you try cleaning your CMake config files (CMakeCache.txt, CMakeFiles/, any build files under external/assimp/), and running a fresh build?

#### #4 - 2015-11-28 14:42 - Calinou

After a fresh re-clone with the latest commits, I still get the same error at the end of compilation.

The stable branch of the Git repository compiles and runs fine.

#### #5 - 2015-11-28 20:07 - skyjake

It's possible that the build configuration needs adjusting on the newer Fedora. I'll upgrade and see what happens.

#### #6 - 2015-11-29 00:04 - skyjake

I tried a clean install of Fedora Workstation 22 and managed to build everything like this: [http://wiki.dengine.net/w/Compilation\\_\(Fedora\\_22\)](http://wiki.dengine.net/w/Compilation_(Fedora_22))

#### #7 - 2016-03-23 13:42 - skyjake

- Status changed from In Progress to Feedback

- Assignee changed from skyjake to Calinou

#### #8 - 2016-03-27 07:50 - skyjake

- Tags changed from fedora, linux, CMake to linux, CMake, Fedora

#### #9 - 2016-04-18 00:24 - sermayen

I can confirm the undefined reference to unz\* errors on Gentoo. I was using revision cc51eea of the doomsday git. My cmake commandline :

```
cmake -DCMAKE_INSTALL_PREFIX=/usr/local/games/doom/doomsday -DDENG_FLUIDSYNTH_EMBEDDED=on  
-DFMOD_DIR=../fmodapi44457/linux -DCMAKE_BUILD_TYPE=Release ../Doomsday-Engine/doomsday/
```

output of make :

```
[ 82%] Linking CXX executable doomsday  
../sdk/libgui/libdeng_gui.so.2.0.0: undefined reference to `unzGoToFirstFile'  
../sdk/libgui/libdeng_gui.so.2.0.0: undefined reference to `unzClose'  
../sdk/libgui/libdeng_gui.so.2.0.0: undefined reference to `unzReadCurrentFile'  
../sdk/libgui/libdeng_gui.so.2.0.0: undefined reference to `unzOpen'  
../sdk/libgui/libdeng_gui.so.2.0.0: undefined reference to `unzOpenCurrentFile'  
../sdk/libgui/libdeng_gui.so.2.0.0: undefined reference to `unzGetCurrentFileInfo'  
../sdk/libgui/libdeng_gui.so.2.0.0: undefined reference to `unzOpen2'  
../sdk/libgui/libdeng_gui.so.2.0.0: undefined reference to `unzGoToNextFile'  
../sdk/libgui/libdeng_gui.so.2.0.0: undefined reference to `unzCloseCurrentFile'  
collect2: error: ld returned 1 exit status  
apps/client/CMakeFiles/client.dir/build.make:7729: recipe for target 'apps/client/doomsday-2.0.0' failed  
make2: * [apps/client/doomsday-2.0.0] Error 1  
CMakeFiles/Makefile2:2220: recipe for target 'apps/client/CMakeFiles/client.dir/all' failed  
make1: [apps/client/CMakeFiles/client.dir/all] Error 2  
Makefile:149: recipe for target 'all' failed  
make: ** [all] Error 2
```

#### #10 - 2016-05-02 19:35 - skyjake

[e1bb1ee](#) now links with zlib to hopefully fix these errors. Let me know how it goes... (builds correctly on my system)

#### #11 - 2016-05-02 19:35 - skyjake

- Status changed from Feedback to In Progress

- Assignee changed from Calinou to skyjake

- % Done changed from 0 to 50

#### #12 - 2016-05-02 23:11 - sermayen

Nope, absolutely no change, same error message. I did a complete cleanup of my Doomsday local git repo, recloned it (with recursive switch for libassimp), cleaned up my build directory and even cleaned my ccache, no change whatsoever.

#### #13 - 2016-05-03 11:41 - skyjake

Hmm, could you attach your CMakeCache.txt, perhaps something there can explain the issue. (For me the build is working on Ubuntu 14.04, 16.04 and Fedora Core 23.)

#### #14 - 2016-05-03 22:21 - sermayen

- File CMakeCache.txt added

Ok, here it is

#### #15 - 2016-05-04 11:29 - skyjake

Thanks, that looks pretty much as expected. There is another file that I'm also interested in: {build-dir}/sdk/libgui/CMakeFiles/libgui.dir/link.txt

Could you also attach that? That should tell us if zlib is not being linked at all or if there is another problem with the linker options.

#### #16 - 2016-05-04 12:15 - sermayen

- File link.txt added

Here it is. As far as i can see libz gets linked (-lz)

#### #17 - 2016-05-05 16:20 - skyjake

The order of the options may also have significance. Could you try what happens if you manually edit the link.txt so that assimp/code/libassimp.a is moved in front of -lz?

If that helps, I'll have to figure out how to coax CMake to output the flags in the correct order...

#### #18 - 2016-05-05 20:37 - sermayen

Tried that, no change. The strange thing is, when i do ldd sdk/libgui/libdeng\_gui.so.2.0.0, it shows that libz.so.1 is in fact linked. I have a feeling that my libz.so.1.2.8 is really missing those symbols, but thats a wild guess.

#### #19 - 2016-05-05 20:39 - sermayen

Here's the output of ldd sdk/libgui/libdeng\_gui.so.2.0.0 :

```
linux-vdso.so.1 (0x00007ffdf2f3e000)
  libdeng_core.so.2.0 => /home/chris/sources/doomsday-build/sdk/libcore/libdeng_core.so.2.0 (0x00007f817226b000)
  libQt5OpenGL.so.5 => /usr/lib64/libQt5OpenGL.so.5 (0x00007f81721c3000)
  libSM.so.6 => /usr/lib64/libSM.so.6 (0x00007f8171fbb000)
  libICE.so.6 => /usr/lib64/libICE.so.6 (0x00007f8171d9e000)
  libX11.so.6 => /usr/lib64/libX11.so.6 (0x00007f8171a5b000)
  libXext.so.6 => /usr/lib64/libXext.so.6 (0x00007f8171849000)
  libQt5X11Extras.so.5 => /usr/lib64/libQt5X11Extras.so.5 (0x00007f8171845000)
  libXrandr.so.2 => /usr/lib64/libXrandr.so.2 (0x00007f8171639000)
  libXxf86vm.so.1 => /usr/lib64/libXxf86vm.so.1 (0x00007f8171433000)
  libz.so.1 => /lib64/libz.so.1 (0x00007f817121d000)
  libQt5Network.so.5 => /usr/lib64/libQt5Network.so.5 (0x00007f81710bf000)
  libGLU.so.1 => /usr/lib64/libGLU.so.1 (0x00007f8170e3d000)
  libGL.so.1 => /usr/lib64/opengl/nvidia/lib/libGL.so.1 (0x00007f8170bae000)
  libQt5Widgets.so.5 => /usr/lib64/libQt5Widgets.so.5 (0x00007f8170524000)
  libQt5Gui.so.5 => /usr/lib64/libQt5Gui.so.5 (0x00007f8170069000)
  libQt5Core.so.5 => /usr/lib64/libQt5Core.so.5 (0x00007f816fbde000)
  libstdc++.so.6 => /usr/lib/gcc/x86_64-pc-linux-gnu/4.9.3/libstdc++.so.6 (0x00007f816f8cd000)
  libm.so.6 => /lib64/libm.so.6 (0x00007f816f5d6000)
  libgcc_s.so.1 => /usr/lib/gcc/x86_64-pc-linux-gnu/4.9.3/libgcc_s.so.1 (0x00007f816f3bf000)
  libc.so.6 => /lib64/libc.so.6 (0x00007f816f022000)
  libuuid.so.1 => /lib64/libuuid.so.1 (0x00007f816eed000)
  libbsd.so.0 => /usr/lib64/libbsd.so.0 (0x00007f816ec07000)
  libxcb.so.1 => /usr/lib64/libxcb.so.1 (0x00007f816e9e3000)
  libdl.so.2 => /lib64/libdl.so.2 (0x00007f816e7df000)
  libXrender.so.1 => /usr/lib64/libXrender.so.1 (0x00007f816e5d5000)
  libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f816e3b8000)
  libssl.so.1.0.0 => /usr/lib64/libssl.so.1.0.0 (0x00007f816e13d000)
  libcrypto.so.1.0.0 => /usr/lib64/libcrypto.so.1.0.0 (0x00007f816dcfb000)
  libGLX.so.0 => /usr/lib64/opengl/nvidia/lib/libGLX.so.0 (0x00007f816dac8000)
  libGLdispatch.so.0 => /usr/lib64/opengl/nvidia/lib/libGLdispatch.so.0 (0x00007f816d7e0000)
  libpng16.so.16 => /usr/lib64/libpng16.so.16 (0x00007f816d5ab000)
  libharfbuzz.so.0 => /usr/lib64/libharfbuzz.so.0 (0x00007f816d344000)
  libicui18n.so.55 => /usr/lib64/libicui18n.so.55 (0x00007f816ceda000)
  libicuuc.so.55 => /usr/lib64/libicuuc.so.55 (0x00007f816cb47000)
  libpcre16.so.0 => /usr/lib64/libpcre16.so.0 (0x00007f816c8e1000)
  libglib-2.0.so.0 => /usr/lib64/libglib-2.0.so.0 (0x00007f816c5a8000)
  librt.so.1 => /lib64/librt.so.1 (0x00007f816c3a0000)
  libsystemd.so.0 => /usr/lib64/libsystemd.so.0 (0x00007f816c30e000)
  /lib64/ld-linux-x86-64.so.2 (0x000055cbc9e5e000)
  libXau.so.6 => /usr/lib64/libXau.so.6 (0x00007f816c10a000)
  libXdmpc.so.6 => /usr/lib64/libXdmpc.so.6 (0x00007f816bf03000)
  libkrb5.so.3 => /usr/lib64/libkrb5.so.3 (0x00007f816bc2b000)
  libk5crypto.so.3 => /usr/lib64/libk5crypto.so.3 (0x00007f816b9f9000)
```

```
libfreetype.so.6 => /usr/lib64/libfreetype.so.6 (0x00007f816b74c000)
libgraphite2.so.3 => /usr/lib64/libgraphite2.so.3 (0x00007f816b51e000)
libcudata.so.55 => /usr/lib64/libcudata.so.55 (0x00007f8169a66000)
liblz4.so.1 => /usr/lib64/liblz4.so.1 (0x00007f816985d000)
libcap.so.2 => /lib64/libcap.so.2 (0x00007f8169657000)
libcom_err.so.2 => /lib64/libcom_err.so.2 (0x00007f8169452000)
libkrb5support.so.0 => /usr/lib64/libkrb5support.so.0 (0x00007f8169246000)
libkeyutils.so.1 => /lib64/libkeyutils.so.1 (0x00007f8169042000)
libresolv.so.2 => /lib64/libresolv.so.2 (0x00007f8168e2a000)
libbz2.so.1 => /lib64/libbz2.so.1 (0x00007f8168c1a000)
libattr.so.1 => /lib64/libattr.so.1 (0x00007f8168a15000)
```

#### #20 - 2016-05-06 08:34 - skyjake

I'm using the same version of zlib (1.2.8) so it should have the right symbols.

However, I notice in the earlier comment that the unz\* linker errors come when building the client (doomsday). ~~Could you check the client's link.txt (apps/client/CMakeFiles/client.dir/link.txt) and add -lz there if it's missing? Or if it is there, move the -lz before all the .so files.~~

#### #21 - 2016-05-06 08:43 - skyjake

I'm checking the Assimp sources and it does actually look like the unz\* functions are not coming from libz at all, but another library called "unzip".

#### #22 - 2016-05-06 08:50 - skyjake

Further investigation shows that Assimp checks for a library called "minizip" via pkg-config, and if that is found, it does not compile its own unzip functions. Does "pkg-config --libs minizip" indicate that the library is available on your system? It wasn't on mine, which could be the reason why this has been working for me. I've now installed it and will check if the build fails.

#### #23 - 2016-05-06 10:20 - skyjake

- % Done changed from 50 to 100

Please try again with [7adcf4736](#). I'm reasonably confident I've now fixed the underlying cause of the issue (linking of the minizip library).

#### #24 - 2016-05-07 00:12 - sermayen

Great, its working with 7adcf4736 now :) Thank you !

#### #25 - 2016-05-07 09:44 - skyjake

- Subject changed from Building on Fedora, CMake issues to Building on Fedora, CMake issues (linking minizip in Assimp target)

- Status changed from In Progress to Closed

Great! :)

#### #26 - 2017-02-25 12:02 - skyjake

- Target version deleted (49)

### Files

CMakeCache.txt	55.2 KB	2016-05-03	sermayen
link.txt	2.82 KB	2016-05-04	sermayen