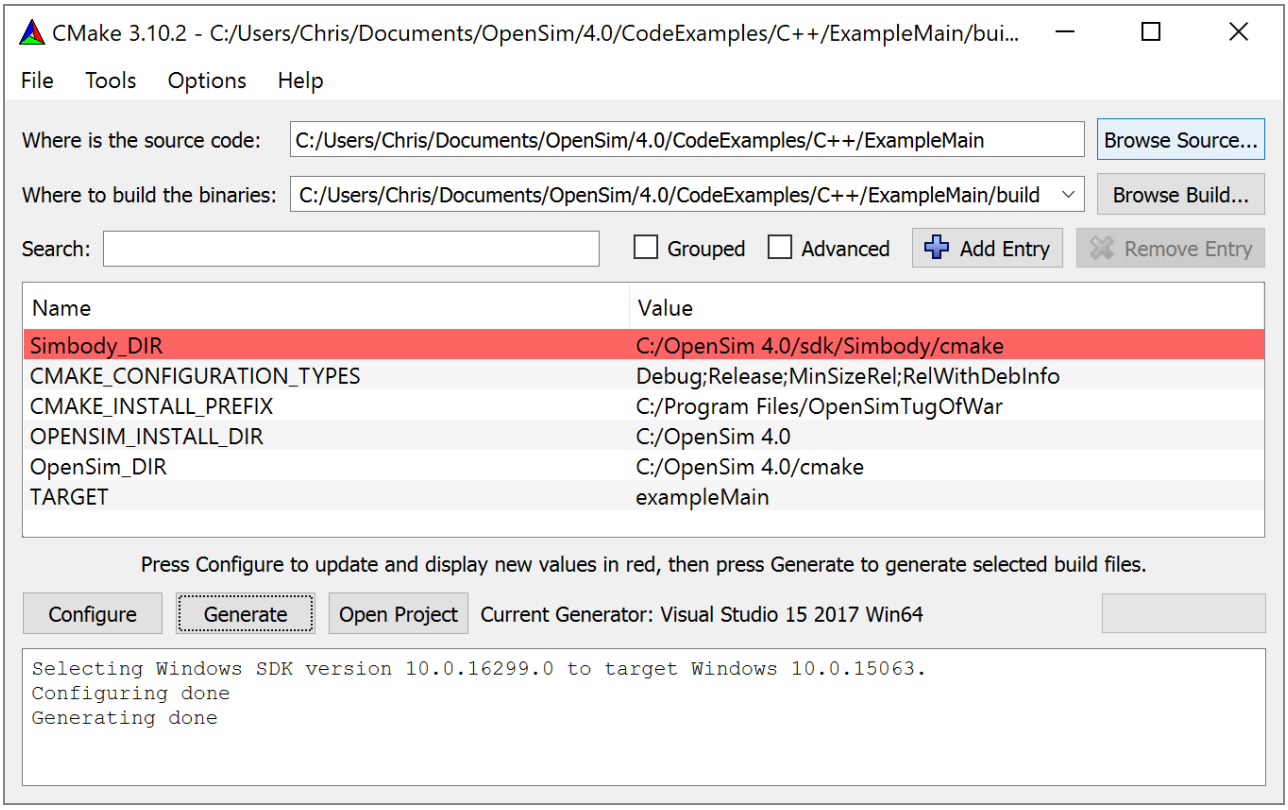
How to Build a C++ Example

In this first example, you will go step-by-step through the entire process of setting up your build folder using CMake, opening and viewing the C++ code in Visual Studio, compiling your first executable, running it and viewing the results in OpenSim. The topics covered in this section include:

Find the C++ Examples

In File Explorer (Windows) or Finder (Mac), navigate to where you installed OpenSim's resources (C:\Users\<name>\Documents\OpenSim\4.x on Windows, ~/Documents/OpenSim/4.x on Mac). We will call this <OpenSimResourcesDir>. Now, navigate to <OpenSimResourcesDir>/CodeExamples /C++. This will be referred to as <WorkSpace> below.

Running CMake



1. Launch the CMake application GUI. On Windows, go to Start -> CMake (cmake-gui). on Mac, go to Launchpad and click CMake).
2. For the field "Where is the source code": Browse to the correct folder where you stored your code such as <WorkSpace>/ExampleMain.
3. For the field "Where to build the binaries": Copy-paste the same folder into this field, but add a build folder. Eg. <WorkSpace>/ExampleMain/build (It's ok if you use all backslashes instead of all forward slashes)
4. Click Configure.
5. If the "build" directory doesn't exist yet, a message box will pop up asking if you want to create this directory. Click Yes.
6. **Choosing a Generator**. Another dialog box will open. The drop-down menu provides a list of *generators*, which, means the type of project files you want to use to compile your code.
    a. **Windows**: For Visual Studio the generator is a combination of the Visual Studio version (e.g., "15 2017") and whether you want to build 32-bit or 64-bit ("Win64") executables with your CMake project. For OpenSim 4.0, you should select "Visual Studio 15 2017 Win64" (starting with 4.0, only 64-bit OpenSim is available for download). Leave "Use default native compilers" selected in the option menu below the drop-down box. Click Finish.
    b. **Mac**: Choose "Xcode" if you want to use a graphical interface to compile the code, and choose "Unix Makefiles" if you are comfortable using the Terminal. The remaining instructions assume you chose Xcode.
7. One of the pink fields that shows up is called OPENSIM_INSTALL_DIR. Make sure the value for this variable is your <OpenSimInstallDir>. If you previously had OpenSim 3.x or earlier installed, this field might already have a value like C:\OpenSim 3.3. **Make sure to edit this variable to point to your 4.x installation**.
8. Check that the TARGET is specified. This specifies the name of the executable file that will be generated. In this case it is "exampleMain".

You can view the description of the CMake variables by hovering over the variable and waiting for a pop-up with info to appear.

9. Click Configure. The variable fields should no longer be pink, and the message window at the bottom of the CMake window should say "Configuring done".
10. Click Generate. The message window should now display "Configuring done Generating done"
11. In CMake 3.8 and later, you can click the Open Project button to open the example in Visual Studio or Xcode.
12. Close CMake.

Compiling and running the example

If your CMake is older than 3.8, then you must open the Visual Studio "solution" or Xcode project manually. In File Explorer or Finder, navigate to the build directory you just created using CMake, e.g., <WorkSpace> /ExampleMain/build. Double-click on OpenSimTugOfWar.sln (Windows) or OpenSimTugOfWar.xcodeproj (Mac).

Windows: Visual Studio

1. The Solution Configuration (drop-down in the toolbar) should be Release. If you choose Debug, your code will not run unless you built the OpenSim API in Debug (the OpenSim API that comes with the GUI is built in Release). You can also choose RelWithDebInfo (Release with debug info), which will allow you to debug your code.
2. Open the TugOfWar1_CreateModel.cpp file, by browsing the libraries in your Solution Explorer.
3. Compile the .cpp file(s): right-click on ALL_BUILD and choose Build from the drop-down menu. In the bottom Output window you can see whether or not your file compiled correctly.
4. Run the program you just created, using one of two methods:
    a. In File Explorer, navigate to the build directory and then to the Release directory. Open the executable that you just created: exampleMain.exe
    b. Run the program from Visual Studio. In the Solution Explorer, right-click exampleMain, and select Set as StartUp Project. Then open the DEBUG menu and select Start Without Debugging. If you used RelWithDebInfo instead of Release and if you select Start With Debugging (from the DEBUG menu), you can pause the program at any line you want by adding a breakpoint. This you can do by clicking anywhere in the left grey column.
5. You should see a console that says "OpenSim example completed successfully."
6. If you had issues running the program, it is likely because C:\OpenSim 4.x\bin is not on your Windows PATH; see the previous page for more information.

Mac: Xcode

1. Click the button that says "ALL_BUILD > My Mac" and select the exampleMain "scheme." When you click the "play" button (right arrow in the upper left), the selected scheme ("exampleMain") will compile and run.
2. You must edit the Configuration used in this scheme. From the menu bar, select Product > Scheme > Edit Scheme... and change Build Configuration from Debug to Release. If you leave the Configuration as Debug, your code will not run unless you built the OpenSim API in Debug (the OpenSim API that comes with the GUI is built in Release).
3. Compile and run the example by clicking the "play" button. You should see a console that says "OpenSim example completed successfully."

Next: C++ API Examples

Previous: Prerequisites

Home: Scripting and Development | Developer's Guide