Prerequisites

The topics covered in this section include:

Overview

The requirements and examples in this guide are targeted for Windows or Mac, because the OpenSim software package and software development kit (SDK) is distributed for these platforms. (For those who are familiar with CMake and the compilers for their operating system, it is possible to build the OpenSim libraries from source, including on Linux; see https://github.com/opensim-org/opensim-core for instructions.) To run the examples provided in the developer's guide, you will need:

1. CMake 3.2 or later (http://www.cmake.org/cmake/resources/software.html)
2. A C++ Integrated Development Environment and/or compiler.
    a. Windows: Visual Studio Community 2017 (https://www.visualstudio.com/vs/).
    b. Mac: Xcode (install from the Mac App Store).
3. The latest version of OpenSim installed.

Each of these programs is described below in more detail.

We also recommend the following tools:

- An XML Editor for editing model and setup files, for example:
    - Windows: Notepad++ (http://notepad-plus.sourceforge.net/uk/site.htm)
    - Mac: TextMate (http://macromates.com)

CMake

CMake is a cross-platform open-source build system that will set up the build environment in a compiler-independent manner. Simple configuration files placed in each source directory (called CMakeLists.txt files) are used to generate standard build files (e.g., Makefiles on Unix, projects/solution in Windows MSVC, Xcode projects on Mac) which are then loaded into a compiler such as Visual Studio, for programming and compiling. In the OpenSim API examples, you will use CMake to generate the correct build "solution" files for Visual Studio.

Documentation for CMake is available from: https://cmake.org/cmake/help/latest/.

Here is a fantastic slideshow, in PDF format, that describes why one would use CMake, what it does, and the basic CMake commands: learning_cmake.pdf

Windows: Visual Studio

Visual Studio is a free Windows "Integrated Developer Environment" that developers at any level can use to create applications for the Windows operating system using C, C++, C# and other languages. In the OpenSim API examples, Visual Studio is used to view, edit and add new C++ code, and compile the resulting program to an executable or plug-in.

Download Visual Studio Community 2017 from https://www.visualstudio.com/vs/. The installer will ask you to choose which "workloads" to install; make sure to check the **Desktop development with C++** workload.

Although there is a version of Visual Studio for Mac, it doesn't support C++ and thus can't be used for OpenSim's API examples.

Mac: Xcode or Terminal

Xcode is the "Integrated Development Environment" that Apple provides for developing applications for Macs using C, C++, Objective-C, and Swift. You can install Xcode from the Mac App Store. You can use Xcode to compile and run the OpenSim API examples.

Alternatively, you can compile and run OpenSim API examples in the Terminal application, using Unix Makefiles as you would on Linux. If you prefer this, you must install the Command-line Developer Tools: open a Terminal window and run **xcode-select --install**.

OpenSim

To install OpenSim, which uses the OpenSim API, download the installer for your platform from https://simtk.org/frs/?group_id=91 and run the installer. The API is accessible with installation of the OpenSim application. Remember to make a note of where you installed OpenSim, which will be referred to as <OpenSimInstallDir>, which by default is "C:\OpenSim 4.x" on Windows and "/Applications/OpenSim 4.x/" on Mac.

Windows

To be able to run the programs you built on top of OpenSim, you need to add the OpenSim libraries to your PATH. Follow the instructions here to add C:\OpenSim 4.x\bin to your PATH environment variable.

Warning: Earlier installations of OpenSim will continue to be accessible but only through the GUI (the GUI does not use the PATH variable to find OpenSim libraries).

Make sure your **system PATH** and **user PATH** contain only your <OpenSimInstallDir>\bin (e.g., "C:\OpenSim 4.x\bin"). You can check this by going to Start->search "environment"->select "Edit the system environment variables"->system variables->Path. If the correct OpenSim\bin path is not included, then add this to your PATH. Also, make sure that no other OpenSim "\bin" directory is in your PATH. If you've ever built OpenSim from source code, make sure no directory containing .lib or .dll files for OpenSim are present in your PATH either. If you don't do this, OpenSim may get confused and possibly use .dll and .lib files from other/older versions of OpenSim instead of the files from the current version of OpenSim and you will likely experience run-time errors.

Advanced Users: Building from Source

Rather than installing from pre-built libraries, advanced users can build Simbody and OpenSim from the source code. OpenSim source files can be downloaded from the SimTK downloads page or from GitHub. Detailed instructions for building from source are in the README.md file in the source code.

Obtaining the Example Programs

The examples for the OpenSim C++ API are installed when you open the OpenSim GUI the first time, usually in:

- Windows: C:\Users\<name>\Documents\OpenSim\4.x\Code\C++
- Mac: ~/Documents/<name>/OpenSim/4.x/Code/C++

Note that you may have chosen a custom location for these resources when you opened the OpenSim GUI the first time.

Next: How to Build a C++ Example

Previous: Getting Started as a Developer

Home: Scripting and Development | Developer's Guide