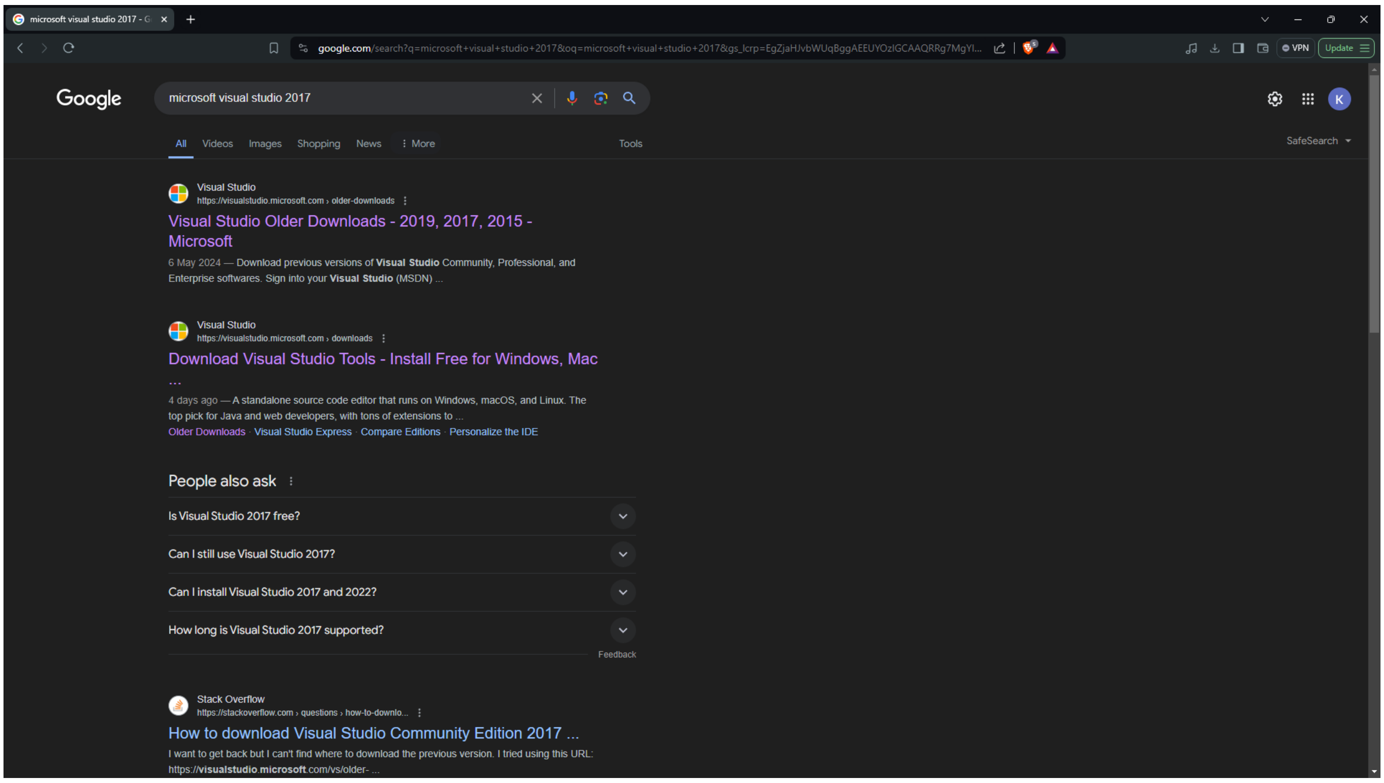
I Installation

The OSRLUM package works on the simulation build on Visual Studio code and Opensim API. Hence, we start by installing the visual studio 2017, to run our Simulation package defined in C++.

II Visual Studio 2017

For the Opensim 4.2, the supported Microsoft Visual Studio Instance is Microsoft Visual Studio 2017 which is supported by Microsoft Visual Studio. The steps for downloading the Visual Studio 2017 is as follows:

A Step 1



google.com/search?q=microsoft+visual+studio+2017&oq=microsoft+visual+studio+2017&gs_lcrp=EgZjaHJvbWUqBggAEEUYOzIGCAAQRRg7MgYI...

microsoft visual studio 2017

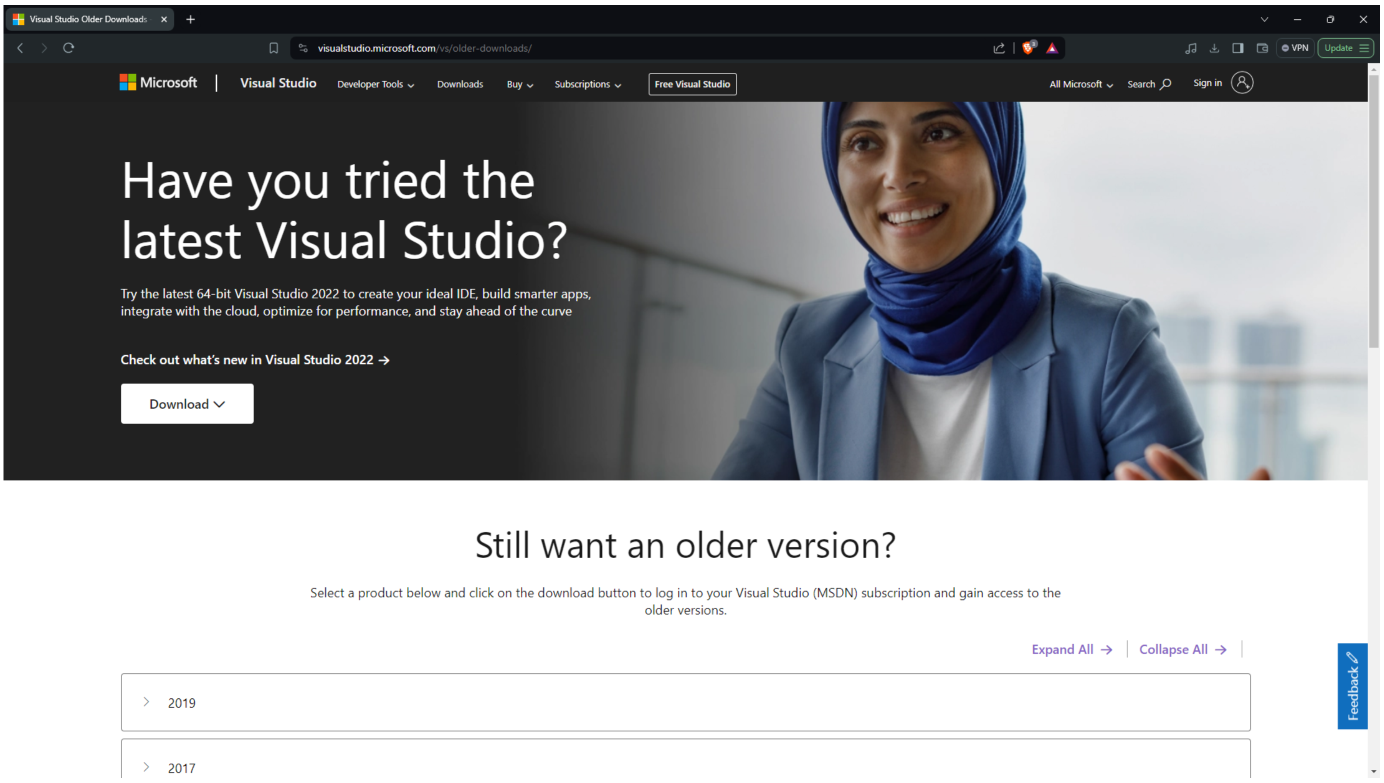
Visual Studio
https://visualstudio.microsoft.com › older-downloads
Visual Studio Older Downloads - 2019, 2017, 2015 - Microsoft
6 May 2024 — Download previous versions of **Visual Studio** Community, Professional, and Enterprise softwares. Sign into your **Visual Studio** (MSDN) ...

Visual Studio
https://visualstudio.microsoft.com › downloads
Download Visual Studio Tools - Install Free for Windows, Mac ...
4 days ago — A standalone source code editor that runs on Windows, macOS, and Linux. The top pick for Java and web developers, with tons of extensions to ...
Older Downloads · Visual Studio Express · Compare Editions · Personalize the IDE

People also ask
- Is Visual Studio 2017 free?
- Can I still use Visual Studio 2017?
- Can I install Visual Studio 2017 and 2022?
- How long is Visual Studio 2017 supported?

Stack Overflow
https://stackoverflow.com › questions › how-to-downlo...
How to download Visual Studio Community Edition 2017 ...
I want to get back but I can't find where to download the previous version. I tried using this URL: https://**visualstudio**.**microsoft**.com/vs/older- ...

B Step 2



Have you tried the latest Visual Studio?

Try the latest 64-bit Visual Studio 2022 to create your ideal IDE, build smarter apps, integrate with the cloud, optimize for performance, and stay ahead of the curve

Check out what's new in Visual Studio 2022 →

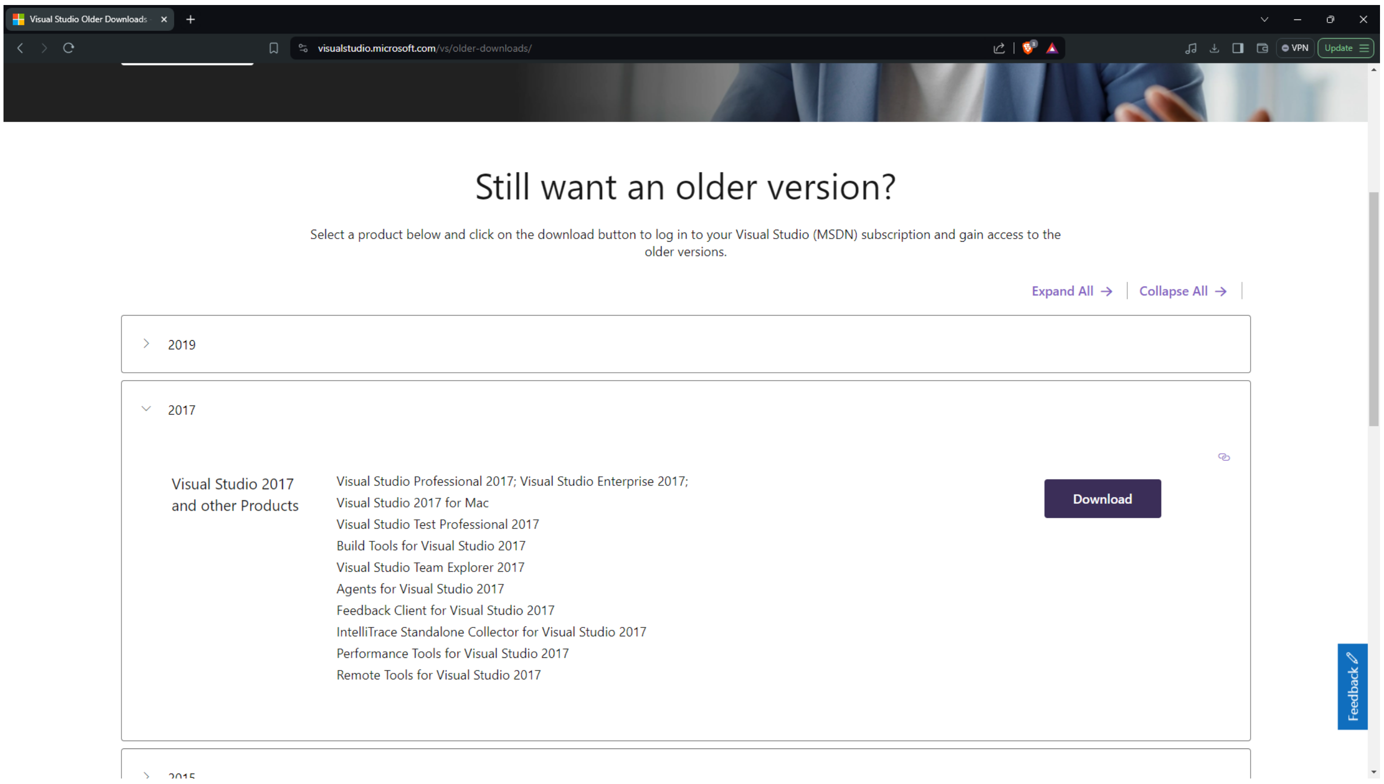
Still want an older version?

Select a product below and click on the download button to log in to your Visual Studio (MSDN) subscription and gain access to the older versions.

2019

2017

C Step 3


Visual Studio Older Downloads
visualstudio.microsoft.com/vs/older-downloads/
Expand All
Collapse All
Download
Feedback

Still want an older version?

Select a product below and click on the download button to log in to your Visual Studio (MSDN) subscription and gain access to the older versions.

2019

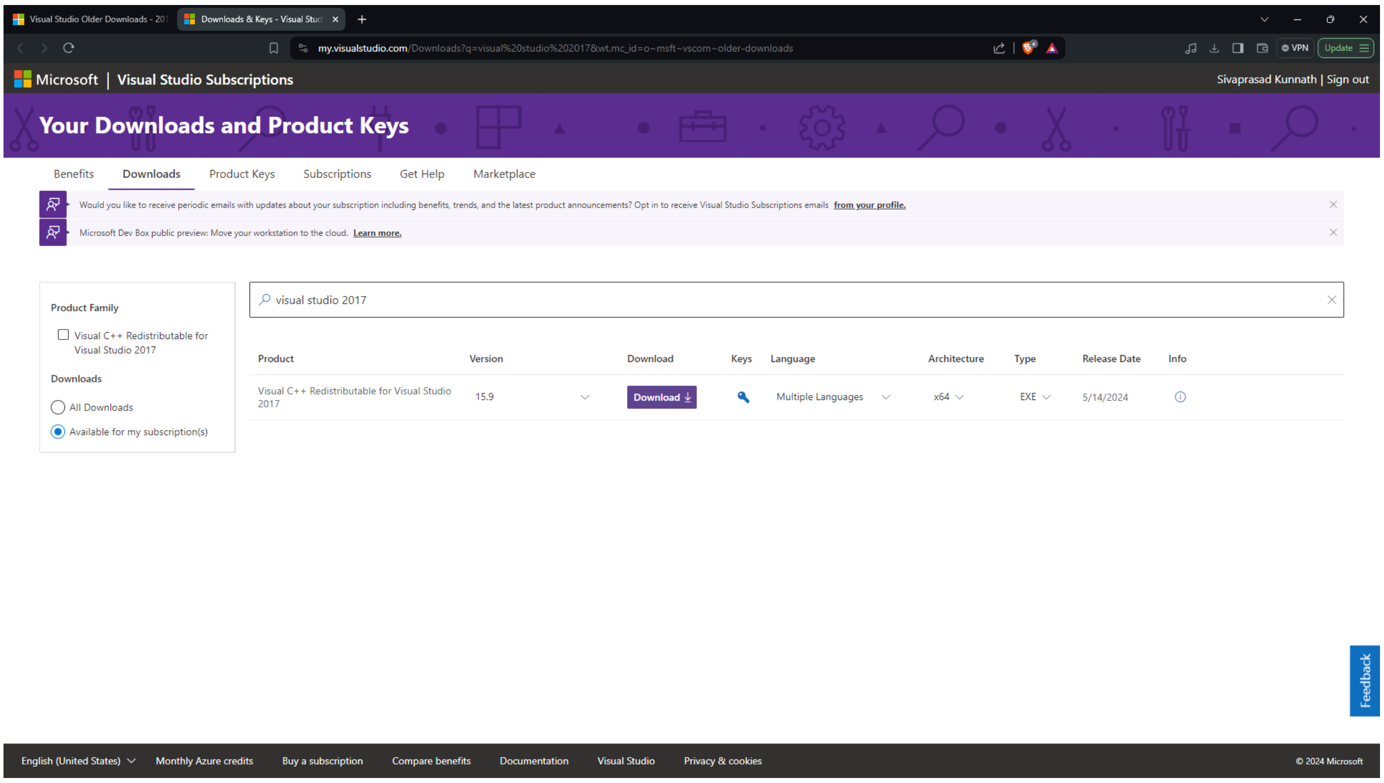
2017

Visual Studio 2017 and other Products

Visual Studio Professional 2017; Visual Studio Enterprise 2017;
Visual Studio 2017 for Mac
Visual Studio Test Professional 2017
Build Tools for Visual Studio 2017
Visual Studio Team Explorer 2017
Agents for Visual Studio 2017
Feedback Client for Visual Studio 2017
IntelliTrace Standalone Collector for Visual Studio 2017
Performance Tools for Visual Studio 2017
Remote Tools for Visual Studio 2017

2015

D Step 4


my.visualstudio.com/Downloads?q=visual%20studio%202017&wt.mc_id=o~msft~vscom~older-downloads

Microsoft | Visual Studio Subscriptions — Sivaprasad Kunnath | Sign out

Your Downloads and Product Keys

Benefits | Downloads | Product Keys | Subscriptions | Get Help | Marketplace

Would you like to receive periodic emails with updates about your subscription including benefits, trends, and the latest product announcements? Opt in to receive Visual Studio Subscriptions emails from your profile.

Microsoft Dev Box public preview: Move your workstation to the cloud. Learn more.

visual studio 2017

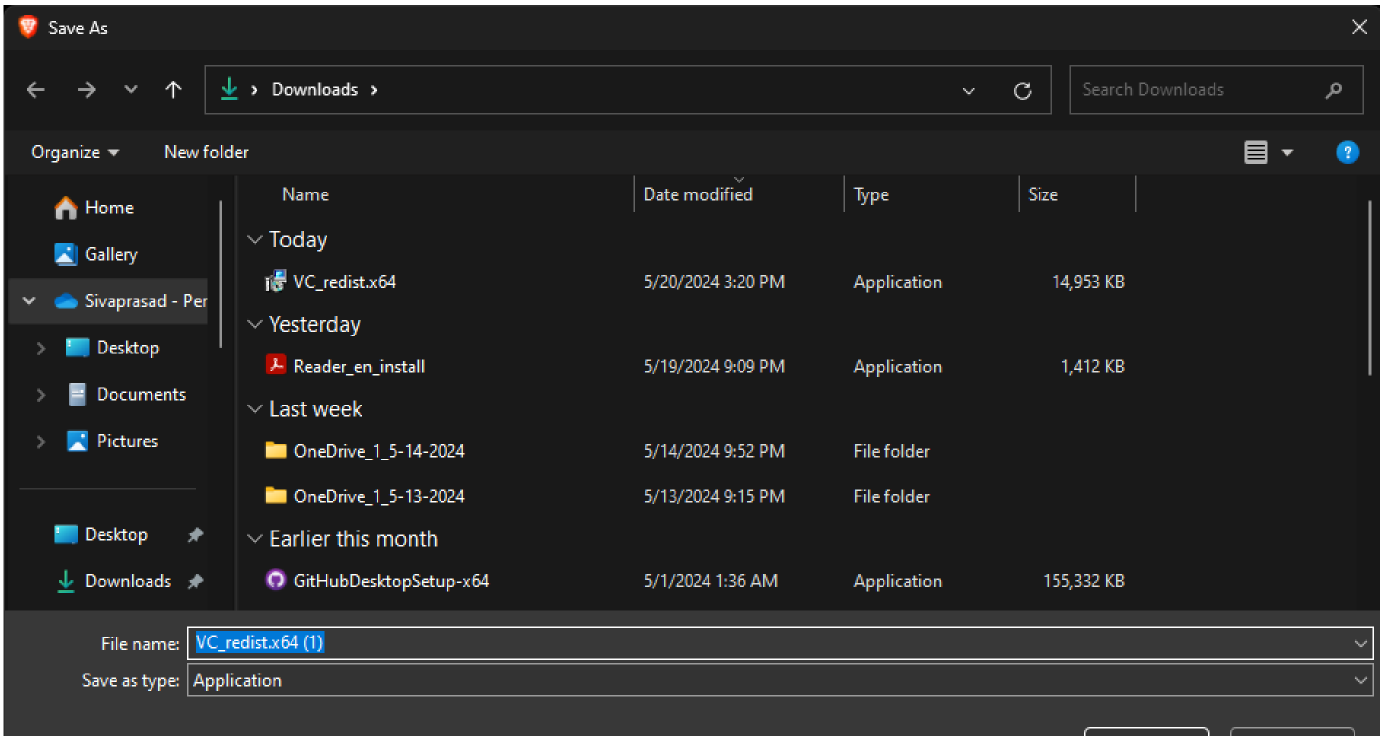
Product Family
- [ ] Visual C++ Redistributable for Visual Studio 2017

Downloads
- All Downloads
- [✓ Available for my subscription(s)]

Product	Version	Download	Keys	Language	Architecture	Type	Release Date	Info
Visual C++ Redistributable for Visual Studio 2017	15.9	Download	🔑	Multiple Languages	x64	EXE	5/14/2024	ⓘ

English (United States) | Monthly Azure credits | Buy a subscription | Compare benefits | Documentation | Visual Studio | Privacy & cookies | © 2024 Microsoft

E Step 5


Save As

Downloads

Name	Date modified	Type	Size
Today			
VC_redist.x64	5/20/2024 3:20 PM	Application	14,953 KB
Yesterday			
Reader_en_install	5/19/2024 9:09 PM	Application	1,412 KB
Last week			
OneDrive_1_5-14-2024	5/14/2024 9:52 PM	File folder	
OneDrive_1_5-13-2024	5/13/2024 9:15 PM	File folder	
Earlier this month			
GitHubDesktopSetup-x64	5/1/2024 1:36 AM	Application	155,332 KB

File name: VC_redist.x64 (1)

Save as type: Application

III Opensim 4.2

For this simulation package to work, we are heavily dependent on the Opensim 4.2 package, which has the following:

1. Musculoskeletal Models.

2. Example CPP Codes.

A Step 1

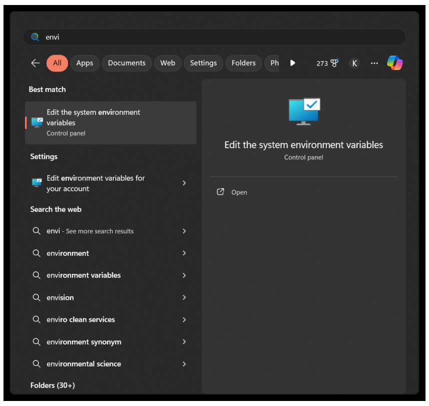
IV Import OpenSim Libraries in C++

We will be using Microsoft Visual Studio as the debugger for our simulation package and to build projects. Hence, it is crucial to import OpenSim API for C++.

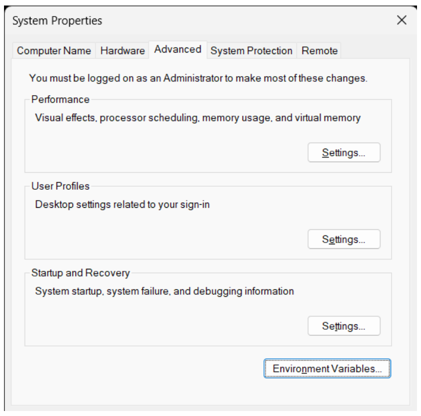
A Setting Path Variables in Windows

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 below to add `C:\OpenSim 4.x\bin` to your PATH environment variable.

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` as displayed in Figure 1a and 1b. 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.



(a) Search window


System Properties

You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles

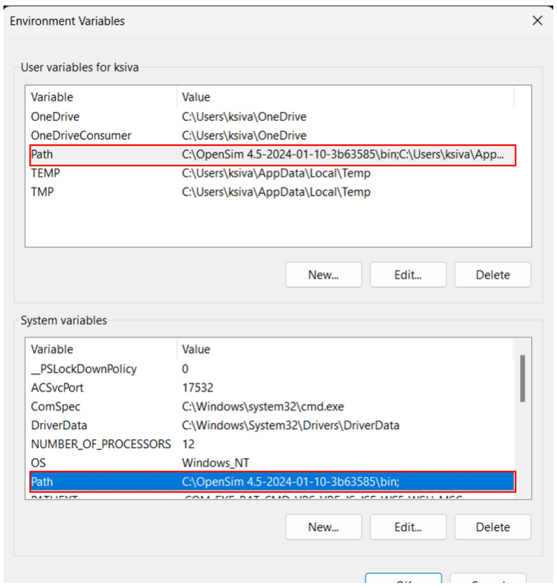
Desktop settings related to your sign-in

Startup and Recovery

System startup, system failure, and debugging information

(b) System Properties window

Figure 1: Setting up System variables - Step 1


Environment Variables

User variables for ksiva

Variable	Value
OneDrive	C:\Users\ksiva\OneDrive
OneDriveConsumer	C:\Users\ksiva\OneDrive
Path	C:\OpenSim 4.5-2024-01-10-3b63585\bin;C:\Users\ksiva\App...
TEMP	C:\Users\ksiva\AppData\Local\Temp
TMP	C:\Users\ksiva\AppData\Local\Temp

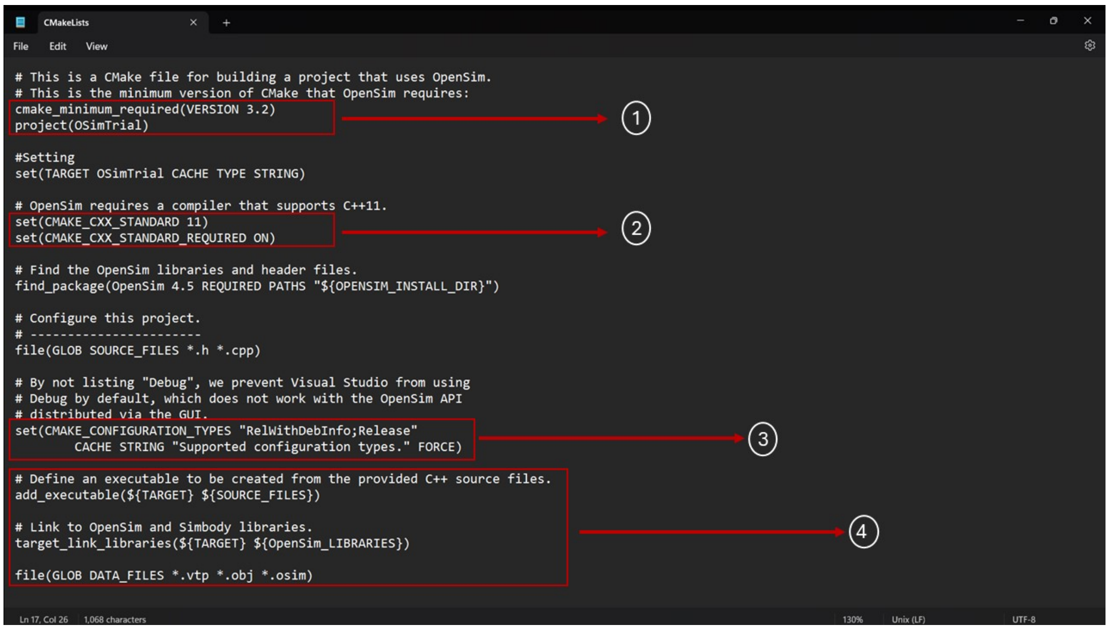
System variables

Variable	Value
__PSLockDownPolicy	0
ACSvcPort	17532
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	12
OS	Windows_NT
Path	C:\OpenSim 4.5-2024-01-10-3b63585\bin;

Figure 2: Path variables that needs update marked in red

B CMakeLists

CMake is an open-source, cross-platform family of tools designed to build, test, and package software. It is primarily used to manage the build process in a compiler-independent manner.



```
# This is a CMake file for building a project that uses OpenSim.
# This is the minimum version of CMake that OpenSim requires:
cmake_minimum_required(VERSION 3.2)
project(OSimTrial)

#Setting
set(TARGET OSimTrial CACHE TYPE STRING)

# OpenSim requires a compiler that supports C++11.
set(CMAKE_CXX_STANDARD 11)
set(CMAKE_CXX_STANDARD_REQUIRED ON)

# Find the OpenSim libraries and header files.
find_package(OpenSim 4.5 REQUIRED PATHS "${OPENSIM_INSTALL_DIR}")

# Configure this project.
# ----------------------
file(GLOB SOURCE_FILES *.h *.cpp)

# By not listing "Debug", we prevent Visual Studio from using
# Debug by default, which does not work with the OpenSim API
# distributed via the GUI.
set(CMAKE_CONFIGURATION_TYPES "RelWithDebInfo;Release"
        CACHE STRING "Supported configuration types." FORCE)

# Define an executable to be created from the provided C++ source files.
add_executable(${TARGET} ${SOURCE_FILES})

# Link to OpenSim and Simbody libraries.
target_link_libraries(${TARGET} ${OpenSim_LIBRARIES})

file(GLOB DATA_FILES *.vtp *.obj *.osim)
```

Figure 3: CMake List explanation

The CMake file has the following sections:

1.

V Building the Simulation with CMake

The following steps are to be taken to build the solution.

1. Launch the CMake application GUI. **Windows:** Go to `Start -> CMake (cmake-gui)`.
2. For the field **Where is the source code**: Browse to the correct folder where you stored your code, such as `<WorkSpace>/OSimTrial`.
3. For the field **Where to build the binaries**: Copy-paste the same folder into this field, but add a build folder. For example, `<WorkSpace>/OSimTrial/build`. (It's okay if you use all backslashes instead of all forward slashes.)
4. Click **Configure**.
   - If the `build` directory doesn't exist yet, a message box will pop up asking if you want to create this directory. Click **Yes**.
5. **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.
   - **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**.

   - **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.

6. 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.
7. Check that the `TARGET` is specified. This specifies the name of the executable file that will be generated. In this case, it is `"OSimTrial"`.
8. 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. Click the **Open Project** button to open the example in Visual Studio.
12. Close CMake.