CUSTOMER SUPPORT NOTE

Using Python with LUSAS Modeller

Note Number:	**CSN/LUSAS/1037**

This support note is issued as a guideline only. Valid as of 07/05/2024.

LUSAS

Forge House, 66 High Street, Kingston upon Thames, Surrey, KT1 1HN, UK
Tel: +44 (0)20 8541 1999 Fax: +44 (0)20 8549 9399
Email: info@lusas.com www.lusas.com

Table of Contents

1. Introduction

This document explains how to install Python and test that it works with Modeller. It also briefly summarizes some Component Object Model (COM) aspects.

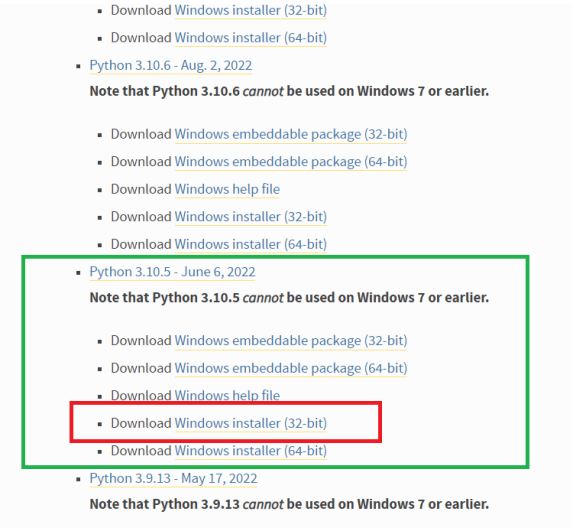
2. Installation

These working instructions have been written using Python 3.10.5, and that version has been tested for use with LUSAS. However, Python has frequent releases, and there will be several more recent releases available to you. If the most modern version is different to the one recommended in this document (currently 3.11.8) please feel free to test the most modern version and update this document if it works.

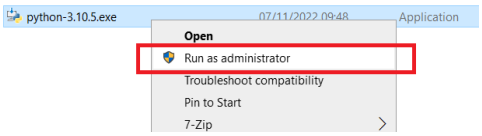
Pywin version 306 (the latest version, last updated Mar. 25, 2023) combined with Python 3.12.x 32-bit versions **do not** work with Modeller. The latest version of Python that works with Pywin 306 to run Modeller is currently version 3.11.8 32-bit (released Feb. 6, 2024).

To install Python, it is necessary to:

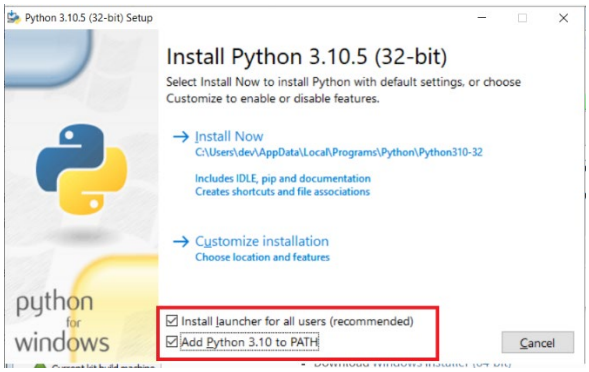
1. Go to www.python.org/downloads/windows.
2. Scroll down to find the section for your chosen version (3.11.2 is recommended and shown highlighted in green below). Within that section, click on the link "Download Windows Installer (32bit)" (highlighted red).



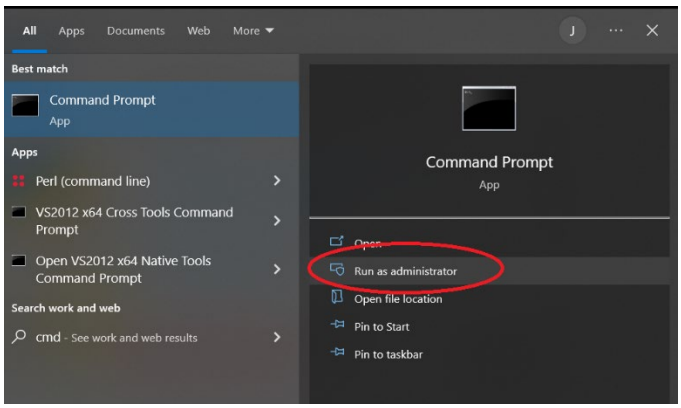
3. Find the file you downloaded in your downloads folder. Right click on it and "run as administrator"



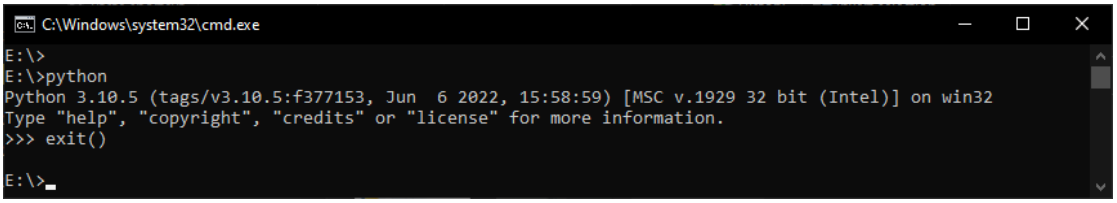
4. Ensure both checkboxes are switched ON as shown below



5. Press “Install Now”
6. Press the Windows logo key to open the Start menu, type CMD, and then click “run as administrator”



7. Within the Windows Command Prompt window, type `python`. You should see a message like the one shown below.

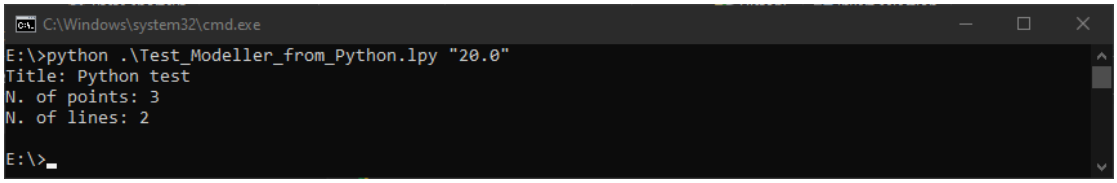


8. Type `exit()` to exit the Python interpreter and return to the Windows Command Prompt.
9. Type `pip install pywin32`.
10. You may see a message telling you that a more recent version is available, this is not important.
11. Locate the following two Python scripts (they are kept alongside this document in SVN):
    a. Test_Modeller_from_Python.lpy
    b. Test_Python_from_Modeller.lpy
12. Make or locate a temporary folder, for example C:\temp.
13. Copy the two scripts to that folder.
14. Change the working folder of the Windows Command Prompt window to that folder, e.g. type `C:` and then `cd temp`.
15. Identify any properly installed version of LUSAS. Any version will do, but you should use the most recent you have. For example, if you have V20.0 installed, type this command:
    `python .\Test_Modeller_from_Python.lpy "20.0"`

Whereas if you only have V19.1 installed, type this command:
`python .\Test_Modeller_from_Python.lpy "19.1"`

Or similar if you have some other version installed (you only need to do this once, even if you have many versions of LUSAS installed).

16. The script takes around 30 seconds to run. If it ran successfully, it will have created Python_test.mdl in your temporary folder and it should show a message like the one shown below.



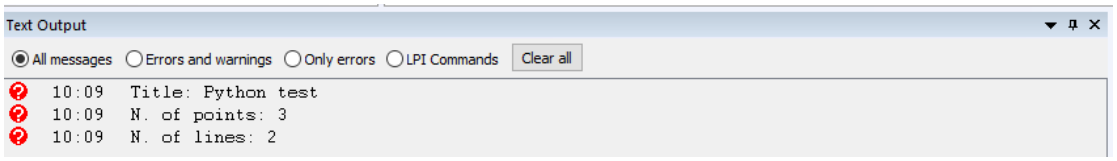
17. Within the cmd prompt window, change directory to this folder: C:\Users**<your username>**\AppData\Local\Programs\Python\Python311-32\Lib\site-packages\win32comext\axscript\client[1]

18. Run the command `python pyscript.py`. You should see the message "Registered: Python".

19. Start an instance of your chosen version of LUSAS Modeller normally.

20. In Modeller, open the file Python_test.mdl from your temporary folder.

21. In Modeller, run the script Test_Python_from_Modeller.lpy[2]. If the script ran successfully, the text output should look like the one shown below.



If you see these results, installation has been successful, and you are finished.

If you get any other result, inform your supervisor.

3. Further reading - COM and pywin32

Modeller can only be driven by scripting languages that are compatible with COM. As explained in the Installation section, it is necessary to install the Python package pywin32 to use COM from Python. This is a third-party package that is not maintained by the Python Software Foundation.

[1] If you cannot find this folder, follow these instructions to locate it:

1. Within the Windows Command Prompt window (ran as administrator), type python to start the Python interpreter.

2. To find the Python installation folder, type the following command which prints the absolute path of the executable binary for the Python interpreter (this file is located in the installation folder).
import sys; print(sys.executable)

5. Append Lib\site-packages\win32comext\axscript\client to the folder reported

[2] V19.1 will not show lpy scripts by default in the file open dialog, but you can nonetheless type its name, and it should run successfully. V20.0 will show the file in the list.

It is important to understand the concept of early and late binding in COM. Binding is a process of matching function calls written by the programmer to the actual code that implements the function. In our case we want to match the calls to LUSAS Programmable Interface (LPI) functions to the actual code. The COM support in pywin32 will either generate a class on the fly which represents the COM object we are interested in (late binding) or will make use of a generated Python module which contains the class definition (early binding). It is important to use late binding when calling LPI functions. To ensure this, we use the pywin32 function *win32com.client.dynamic.Dispatch*. Using other pywin32 dispatch functions might result in using early binding.

The following websites contain additional information about the package pywin32:

- github.com/mhammond/pywin32
- pypi.org/project/pywin32

4. References

F1 Quality Procedures Manual, Issue 3 (1993)
Finite Element Analysis Ltd., Kingston-upon-Thames, UK.