



Technische Universität Berlin

Faculty of Electrical Engineering and Computer Science

Institute of Software Engineering and Theoretical
Computer Science

PySimLib - Python Simulation Library

User Guide

Release date: October 18, 2016

This is a work in progress version

Contents

1	Introduction	3
2	Setting up PySimLib	4
2.1	Command line	4
2.1.1	Windows	4
2.2	Unix-like	5
2.3	Installing Python	5
2.3.1	Windows	5
2.3.2	Ubuntu	6
2.4	Installing a Simulator	6
2.5	Installing PySimLib	7
2.6	Configuring PySimLib	7

List of Tables

1	List of simulators supported by PySimLib	6
2	Testing Constellations of Operating Systems and Simulators using PySimLib . .	7

List of Figures

1	Windows command prompt under Windows 7 in Administrator mode	4
2	Python 3.5 Windows setup	5
3	Python 3.5 Windows customized setup	6

Listings

1 Introduction

This user guide is aimed to aid the user in setting up the Python Simulation Library (in the following abbreviated with PySimLib), run it, and explain the libraries features. After reading this document the user should be able to simulate models using the PySimLib and be able to use it in Python in order to create own simulation descriptions.

If you have any questions or issues, please feel free to write an e-mail to *a.mehlhase@tu-berlin.de*. In case you find bugs in our software we would be pleased for a report at our repository <https://gitlab.tubit.tu-berlin.de/a.mehlhase/PySimulationLibrary>.

2 Setting up PySimLib

The following will show you how to set up PySimLib, while mainly focusing on Windows platforms. Installation instructions for systems based on Ubuntu are also given. Due to the variety of Linux systems we can't provide installation instructions for all of them. Unfortunately we currently have no access to a Mac computer and can therefore not provide installation instructions, nor do we know if PySimLib works on Mac at all. We would be pleased to hear feedback on that.

At first we will give a brief introduction on how to open a command line terminal or console, which will be necessary for the setup process described here. The first installation step then will be to install Python, as PySimLib is a Python library. In order to simulate models using the PySimLib you will need a simulator that can simulate your models. Note that PySimLib is not a simulator itself but can communicate with several ones and provides a common interface to do so. The second step of installation will be to install a simulator. After these preparations PySimLib can be installed and finally we will describe how to configure PySimLib correctly.

2.1 Command line

The command line terminal or console is a program by which the user can issue commands in text-form. Interaction is realized in the following way: The program writes text to the screen that the user can read. The user writes text commands that are confirmed by the "Enter" or "Return" keys and are read and executed by the program.

Usually operating systems come with a command line program, in order to control settings, files etc. Specific commands may not be executed by any user but by a system administrator, who is allowed to run commands with advanced privileges, such as installing a program or making any other change to the system.

2.1.1 Windows

In Microsoft Windows, the utility cmd.exe is included, also called. In order to open it, click the "Start"-Button or the Windows-Button. In the search widget, type "cmd" without the quotation marks. The program "Command Prompt", "cmd.exe" or something similar should appear. Click it in order to start it. A window similar to that shown in figure 1 should open up. In order to run the console in administrator mode, repeat the above process but instead of doing a left click, click right and press "Run as administrator".

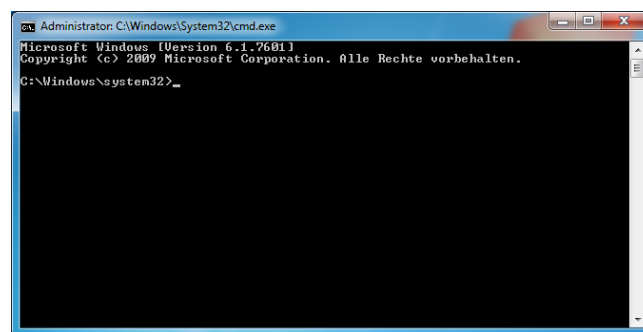


Figure 1: Windows command prompt under Windows 7 in Administrator mode

2.2 Unix-like

Usually a command line program, usually called Terminal in Unix world, is included with the operating system. You should look for it in your program launcher possibly in a category “System tools“ or something close.

You can run a command with advanced privileges by entering the “sudo“ command before it. Say you want to run the command “help“ with advanced privileges enter “sudo help“ without quotation marks. This might ask you for the password of your operating systems’ user account.

2.3 Installing Python

The first step in setting up PySimLib is installing Python. PySimLib was developed for Python 3, in particular version 3.5. The library might also work with previous versions of Python 3 but surely not with Python 2. By us, PySimLib has only been tested in Python 3.5 and we would recommend you to use it. In general it is more likely that we will support a newer version than an older one than Python 3.5.

In this guide we will expect users to install apart from Python also “pip“ (this will be shown later). Pip is Python’s package management utility and allows to easily install extensions to Python. This is not required, as extensions can be also installed manually but this is rather cumbersome and we don’t encourage it. In case you don’t want to use pip, you can download the mentioned packages and install them manually. We will not give support on this process.

2.3.1 Windows

Go to Python’s website, download the installer and run it. You should see something like in figure 2. You don’t need to customize the installation, however we encourage you to check the “Add Python 3.5 to PATH“ box. This allows you to execute Python from any directory, which is not necessarily required but makes working with Python easier. Don’t select this in case you have other versions of Python installed and one of them is added to the PATH environment variable. However, in this guide we will expect that the user added Python 3.5 to the PATH variable. If you did not do that you have to specify the full path to Python whenever you want to use it.

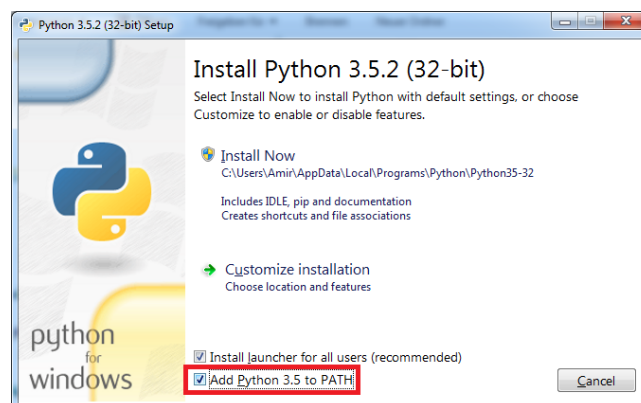


Figure 2: Python 3.5 Windows setup

In case you want to customize the installation, make sure, like in figure 3, that you select at least “pip“ and “tcl/tk and IDLE“, which is required by PySimLib in order to draw and show plots.

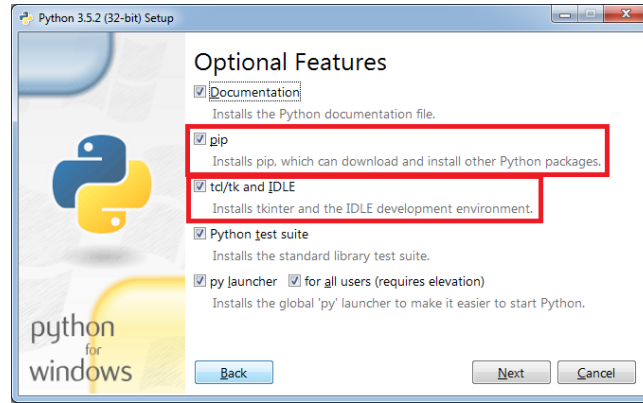


Figure 3: Python 3.5 Windows customized setup

2.3.2 Ubuntu

Python should already be installed on Ubuntu. If not, install it by running the following commands:

```
1 sudo apt-get install python3
2 sudo apt-get install python3-pip
```

If you are unsure whether Python is installed or not, run the commands anyways. In case Python is already installed, apt-get is going to inform you and nothing is going to be done.

2.4 Installing a Simulator

PySimLib currently supports the simulators listed in table 1.

Dymola
OpenModelica
MATLAB/Simulink

Table 1: List of simulators supported by PySimLib

Please follow the manual of the simulator of your choice in order to set it up correctly. Be sure that it is working properly before continuing with installing PySimLib.

It should be noted that these simulators exist in different versions, which differentiate in terms of features etc. We have listed the constellations of operating system and simulator version under which we tested PySimLib in table 2.

	Windows 7 64 bit	Windows 10 64 bit	Ubuntu 14.10	Ubuntu 16.04
Dymola 2013	✓			
OpenModelica 1.9.6			✓	✓
MATLAB/Simulink R2013a				✓

Table 2: Testing Constellations of Operating Systems and Simulators using PySimLib

In case you want to use Dymola under Windows you need to install the package “pywin32” from <https://sourceforge.net/projects/pywin32/>.

You can install (or remove) additional simulators any time but whenever you do that, you have to reconfigure PySimLib. See subsection 2.6 for configuring PySimLib.

2.5 Installing PySimLib

PySimLib can be installed very simply using pip. In order to determine the correct pip version try entering the following commands:

```
1 pip3.5
2 pip3
3 pip
```

Remember the first one that worked and take it for all following commands where "pipxyz" is used.

To install PySimLib execute the following commands (**you might need advanced privileges**):

```
1 pipxyz install zmq
2 pipxyz install PySimLib
```

2.6 Configuring PySimLib

TODO: schreiben

TODO: weiter