Dynamic ISO Building Simulator (DIBS) – Installation Guide

Institut Wohnen und Umwelt

Julian Bischof, Simon Knoll

DIBS – Dynamic ISO Building Simulator has been developed in context of the ENOB:DataNWG Forschungsdatenbank Nichtwohngebäude (www.datanwg.de) project.

Funding code: Fkz.: 03ET1315

Project duration: 01.12.2015 until 31.05.2021

Funded by:

Bundesministerium für Wirtschaft und Energie

in accordance with the parliamentary resolution of the German Parliament

IWU

Table of Content

This installation guide describes all steps required to get DIBS up and running. The process should be operation system independent, but it has only been tested under Windows 10.

Dependencies of the DIBS

- Python 3.7 or 3.8
- Anaconda
- Python module “namedlist”
- Python module “geopy”

Installation steps

The following steps prepares the environment which DIBS need to work. For unexperienced users we are setting of on the starting point of the operational system.

Installation of ANACONDA

Further due to certain dependencies we recommend the ANACONDA distribution. Therefore visit https://www.anaconda.com/products/individual and download the distribution on the bottom of this page. After downloading please install with standard settings (The according Python version is automatically installed in the process).

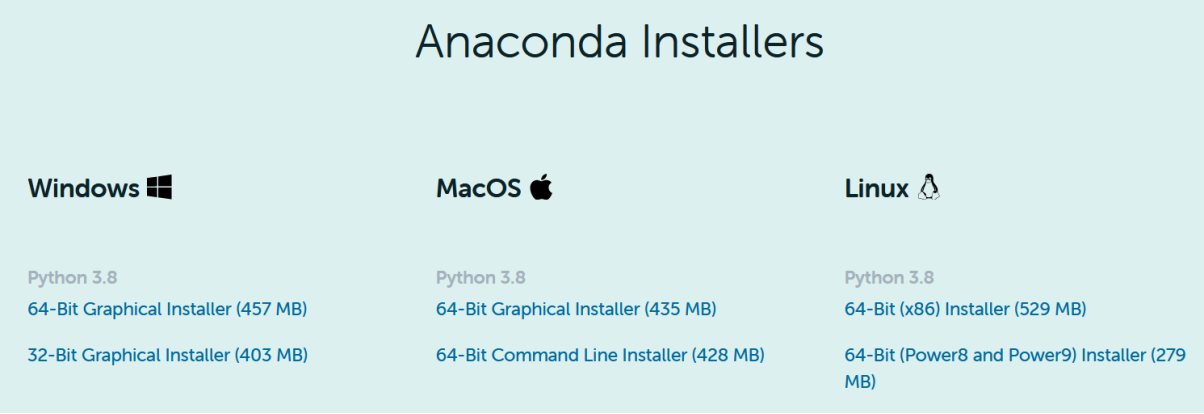

Figure 1: ANACONDA Individual download options (https://www.anaconda.com/products/individual)

Installation of missing Python modules

With the ANACONDA distribution comes the Scientific Python Development Environment *Spyder.* Open Spyder and install the missing Python modules via the console (see **Figure 2**).

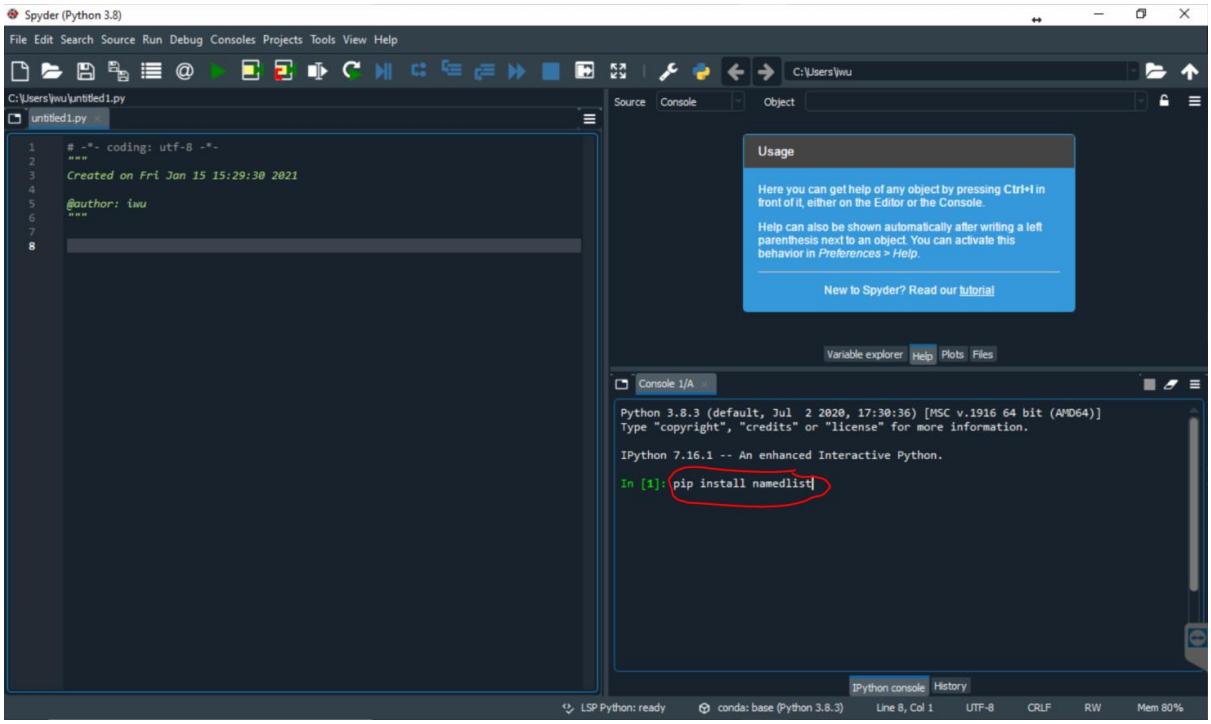

Figure 2: Installation of Python modules via the Spyder console

namedlist

For installing “namedlist” (https://pypi.org/project/namedlist/#files) insert the command “pip install namedlist” into the console and run it by hitting ENTER. Wait for the installation to finish.

geopy

For installing “geopy” (https://pypi.org/project/geopy/) insert the command “pip install geopy” into the console and run it by hitting ENTER. Wait for the installation to finish.

Now the DIBS environment is created. For getting your first building stock simulation on the way, please follow the steps described on the next pages.

Starting your first simulation with DIBS

After setting up the environment DIBS requires to run, we are good to go. The next steps are to download all DIBS files, extract them and to start your first simulation.

Download DIBS

There are two options of getting you hands on DIBS. Either clone the GitHub repository (LINK) or download the DIBS-package via .zip file and extract it on the location of your choice. For the download, please go to the GitHub repository (LINK) and select under the green download menu the "Download ZIP" option (see **Figure 3**).

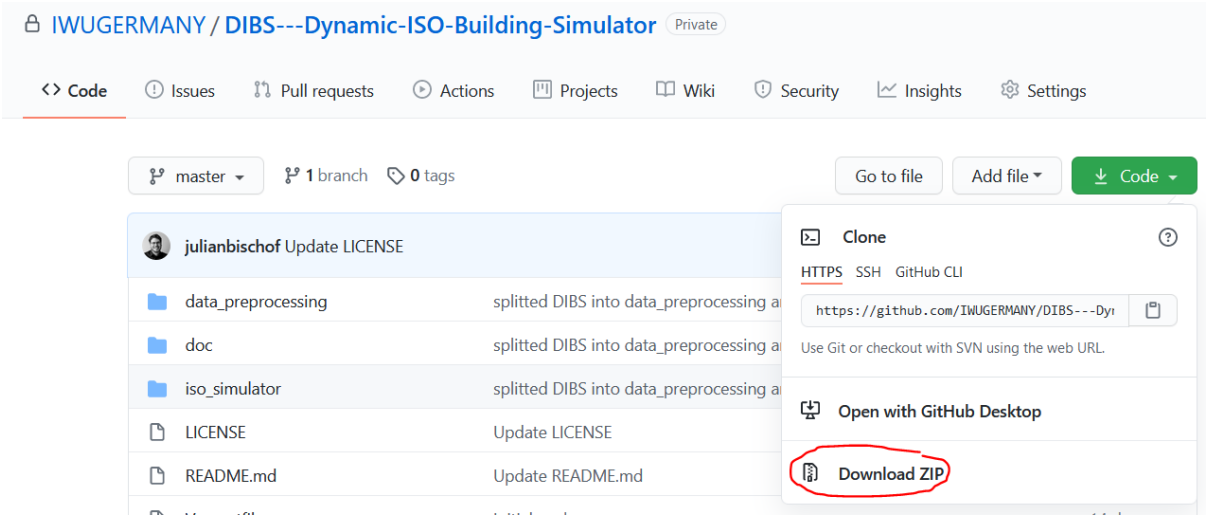

Figure 3: Downloading DIBS as zip-file

Save the zip file to your directory of choice and extract it.

Open Spyder and start simulation

Next open the Spyder-Editor (i.e. by pressing the Windows-key, then type "Spyder" and open with Enter). In Spyder select the right working directory of DIBS. This must be the path to the extracted DIBS-Package. In this working directory now follow the path

DIBS---Dynamic-ISO-Building-Simulator-master\iso_simulator\annualSimulation

and open annualSimulation.py. (see **Figure 4**)

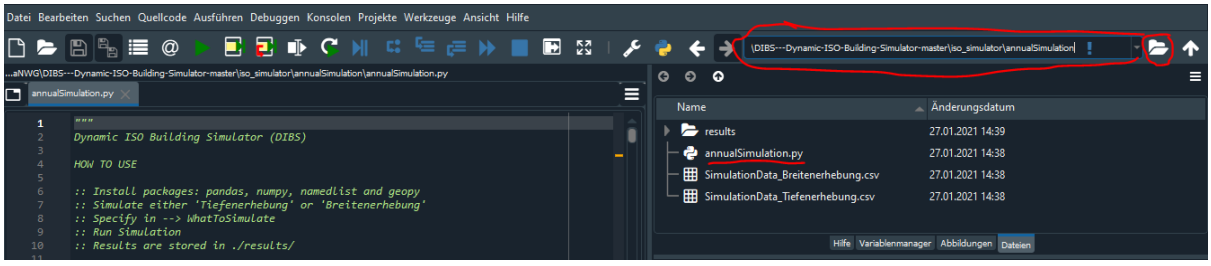

Figure 4: Setting DIBS working directory and selecting the annualSimulation.py file

The final step for running the first dummy simulation is to press the “run” button on top of Spyder (see Figure 5).

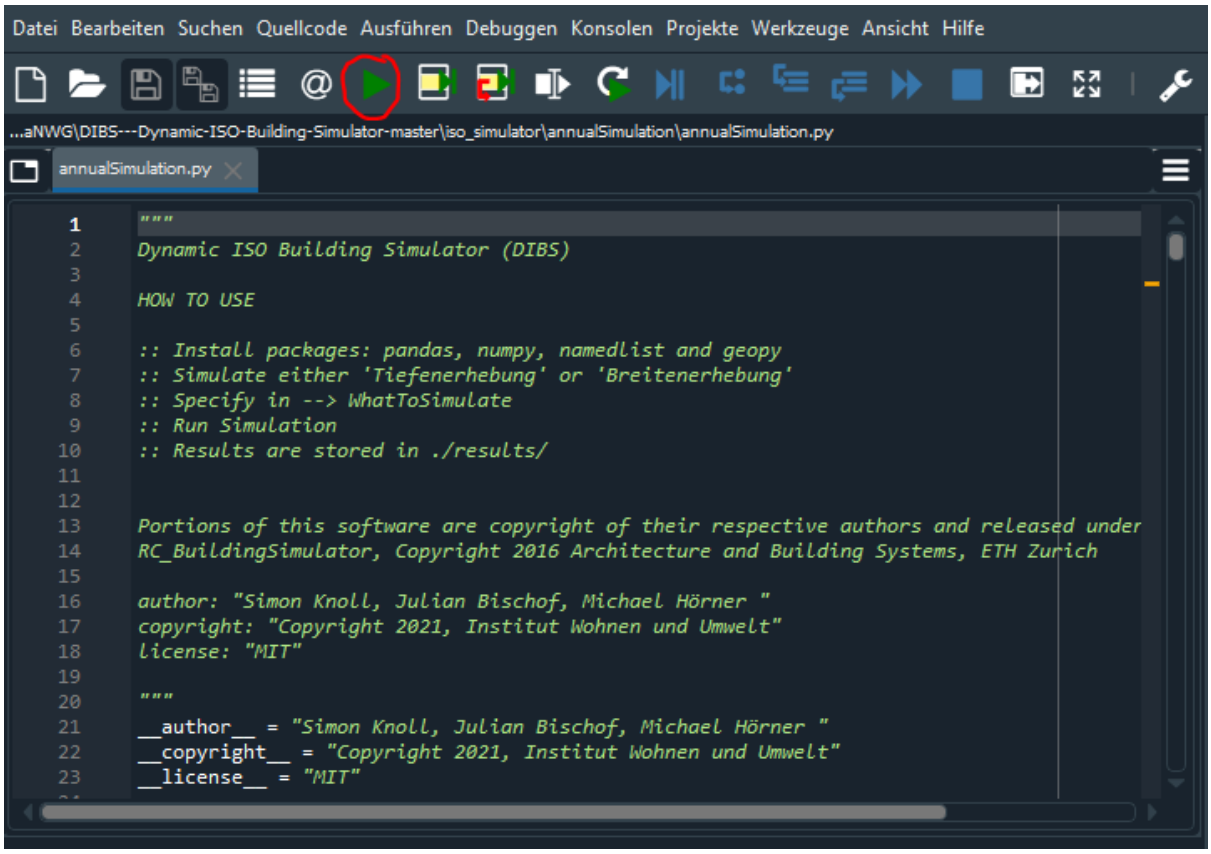

Figure 5: Running DIBS simulation

The results of the dummy-simulation and any other simulation run are placed in the results folder: DIBS---Dynamic-ISO-Building-Simulator-master\iso_simulator\annualSimulation\results

Here a detailed results file for each simulated building and a summary results file are stored.

DIBS data-structure

The script annualSimulation.py organizes the building (stock) simulation and calls other sub-scripts (see **Figure 6**) which are described here briefly.

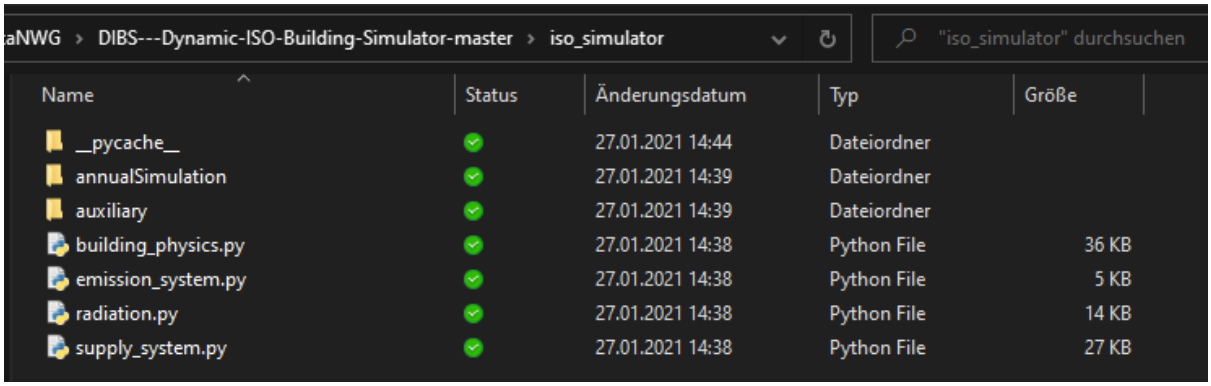

Figure 6: Scripts in DIBS---Dynamic-ISO-Building-Simulator-master\iso_simulator

- building_physics.py: In this script you will find all the equations necessary to calculate the space heating, cooling and electricity demand (including the ISO 13790 calculations and the approach to calculate the lighting demand).
- radiation.py: This script includes methods which depend on the building location e. g. calculation of the sun position, solar gains and illuminance. Furthermore, the script is used to determine the nearest weather station of a building.
- supply_system.py: This script manages different supply systems for both heating and cooling. New supply systems can be introduced by adding new classes.
- emission_system.py: Depending on the emission system of the building the calculated heat/cooling will be supplied to one temperature node in the resistance-capacity network. Information about the supply and return temperatures is only decisive for heat pumps at this stage.

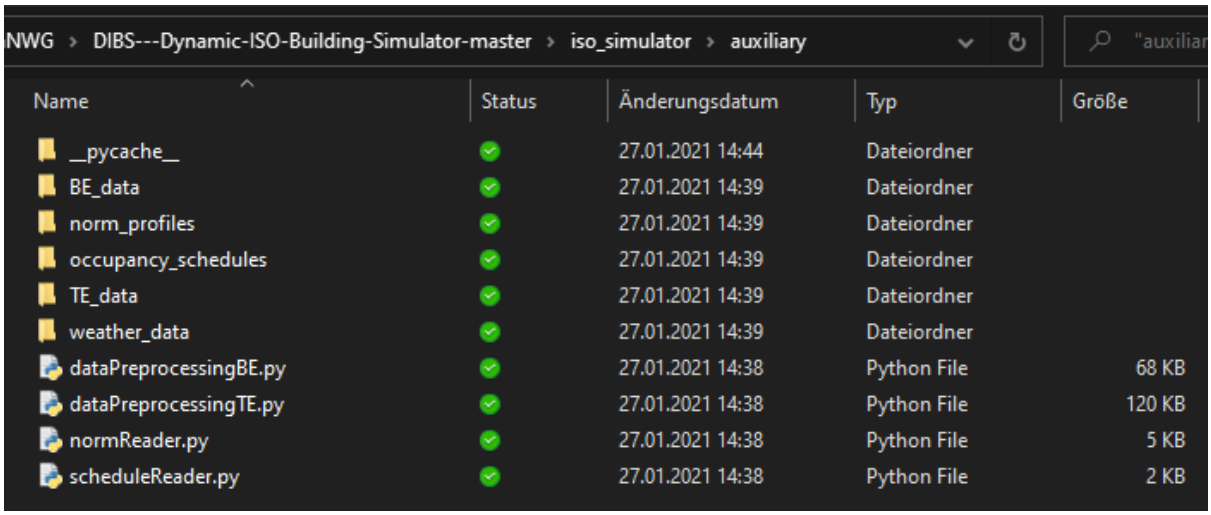

Figure 7: Scripts and data files, supporting the simulation of DIBS

Scripts and data files that support the simulation with additional information than the building input dataset can be found in DIBS---Dynamic-ISO-Building-Simulator-master\iso_simulator\auxiliary (See Figure 7).

- normReader.py: This script returns data from DIN EN ISO 18599-10 (gain per person and appliance gains) as well as the buildings usage time in accordance with SIA 2024.
- scheduleReader.py: This script returns the corresponding occupancy schedule depending on the building subcategory (uk_geb). Occupancy schedules are taken from SIA 2024. These schedules are stored in DIBS---Dynamic-ISO-Building-Simulator-master\iso_simulator\auxiliary\occupancy_schedules (see)

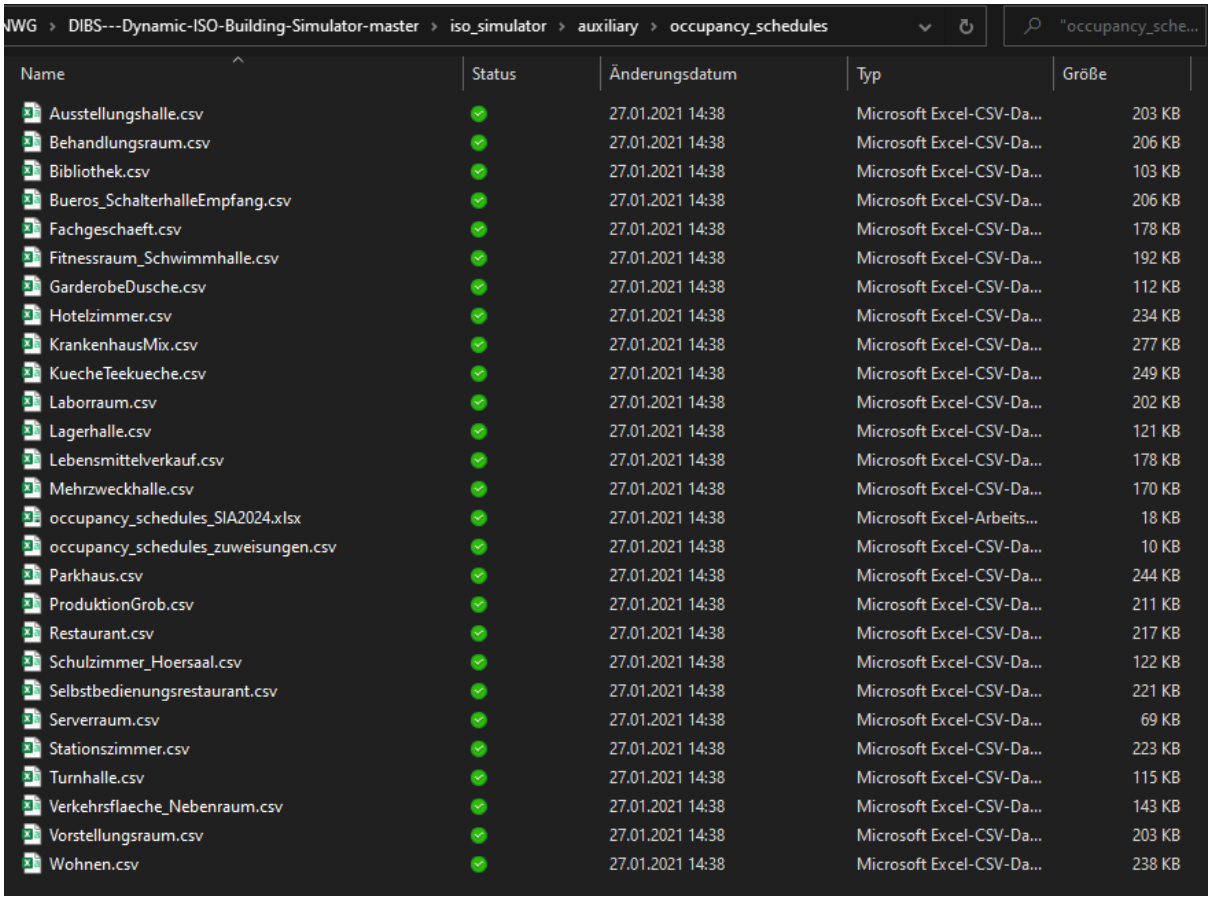

Figure 8: Occupancy schedules used by DIBS

All weather datasets are stored in DIBS---Dynamic-ISO-Building-Simulator-master\iso_simulator\auxiliary\weather_data