City Energy Analyst Documentation

Release 3.31.0

Architecture and Building Systems

Jul 11, 2022

User Documentation:

City Energy Analyst (CEA) is an open-source software for the analysis of energy systems in cities. CEA helps you to analyse the effects of building retrofits, land-use planning, district heating and cooling and renewable energy on the future costs, emissions and energy consumption of neighbourhoods and districts. In Addition CEA helps you to find the optimal location, size and operation of energy generation and distribution technologies for a neighbourhood or a district of your choice.

For the moment, CEA supports neighbourhoods and districts in Temperate (e.g., Switzerland) and Tropical climates (e.g., Singapore).

Learn more..

Visit www.cityenergyanalyst.com for more information on the CEA.

CHAPTER 1

Installation

CEA can be installed in Windows, Mac OS, Ubuntu and on the computer cluster of the ETH Zurich Euler. The latter is only available for students and faculty of the ETH Zurich.

1.1 Installation guide for Windows

Follow these instructions to install the CityEnergyAnalyst (CEA) on a Windows system (tested with Windows 10).

1. Download the latest version of CEA in here.
2. Open the installer and follow the instructions

Note: For installing the development version of CEA, tick the box "development version" during the installation.

Note: For previous releases check here.

Note: To install from the source check installation-on-windows-manual

1.1.1 Interfaces

There are different ways in which you can interact with the code of CEA.

1. The command line interface: This is the command line to all the commands of CEA from your computer terminal
2. The dashboard: This a web-based interface to CEA, open source and developed by the CEA team.
3. The pycharm interface: this interface provides access to all the source code of CEA.

The command line interface and dashboard interface are included during the installation of CEA. Other interfaces require a few additional steps to get them up and running.

Pycharm

In order to access and work on the source code of CEA from pycharm do:

1. Make sure to have installed the development version of CEA (see step 2 of the installation guide).
2. Download and install Github Desktop (64-bit).
3. Download and install Pycharm Community edition (64-bit)
4. Open PyCharm from the start menu and open project CityEnergyAnalyst (default location is C:\Users\<you>\Documents\CityEnergyAnalyst\CityEnergyAnalyst).
5. Open File>Settings>Project:CityEnergyAnalyst>Project Interpreter>Project Interpreter.
6. Click on the settings button (it looks like a wheel) next to the current interpreter path, and click Add.
7. Click System Interpreter from the left hand list and select existing environment.
8. Point to C:\Users\<you>\Documents\CityEnergyAnalyst\Dependencies\Python\python.exe
9. Click apply changes.

Attention: We ended Support of Grashopper on 20.03.20. The legacy code can be found in our github repositry/legacy

Attention: We ended Support of ArcGIS on 15.04.19. The legacy code can be found in our github repository/legacy

1.2 Installation guide for the Euler cluster

Disclaimer: for this to work, you must be an ETH Zurich affiliated person and own a nethz-account.

EULER stands for *Erweiterbarer, Umweltfreundlicher, Leistungsfähiger ETH-Rechner*. It is a high performance cluster available to users affiliated to the ETH Zurich. See more information about the computing cluster on the clusterwiki.

This section describes the steps necessary to get the CEA running on the Euler cluster.

1.2.1 Logging on to the Euler cluster

Estimated time: 1 hr

You can login to the Euler cluster via the SSH protocol. If you use Linux or Mac OS X, then you can directly use SSH from within a shell as it is part of the operating system. If you are on Windows, you will need an ssh client. The CEA Console includes the ssh command, otherwise, install a third-party application in order to use SSH (Putty, Cygwin, Git for Windows).

You can only log in to Euler from within the ETH network or when connected via VPN.

Once in the terminal in Linux or Mac OS X or in a terminal of thrid-party application of your choicse, do:

```
ssh <your nethz-name>@euler.ethz.ch
```

After entering the above command in the shell, you will be asked for a password. Enter your nethz password. You are then greeted with the Euler welcome message.

Detailed steps are described in the Euler wiki of the Scientific Computing Service in ETHZ. For Windows users, it is recomendded to download WinSCP and MobaXterm. Please follow the steps in the wiki carefully, and consult the cluster support when Troubleshooting section (2.9) is not enough to solve your problwm.

1.2.2 Build a CEA Singularity container

Estimated time: 20 mins

You need to build a Singularity container via a cea docker image. The latest docker image of cea is published here. Please login to Euler and conduct the following steps.

- Request a compute node with Singularity

```
$ bsub -n 1 -R singularity -R light -Is bash
```

- Load eth_proxy to connect to the internet from compute nodes

```
$ module load eth_proxy
```

- Go to the scratch folder

```
$ cd $SCRATCH
```

- Build a Singularity container based on the cea docker image

```
$ singularity pull docker://cityenergyanalyst/cea
```

- Check if Singularity has been built

```
$ ls
```

You should find the CEA Singularity container, `cea_latest.sif`, in the list of files. Congratulations! You can start running CEA on Euler!

- If wish to run `cea test` to test the CEA Singularity container:

```
$ SINGULARITY_HOME=/projects singularity shell -B $SCRATCH cea_latest.sif
Singularity> source /venv/bin/activate
(venv) Singularity> cea test
```

1.2.3 Running the CEA

You need to run the CEA scripts with their command line interface (CLI). Be sure to learn how to use the job system on Euler, as the login nodes are not intended for running simulations. See clusterwiki.

- Upload your CEA projects to `/cluster/scratch/nethz-username`.
- Upload a `workflow.yml` to `/cluster/scratch/nethz-username`.
- Open `workflow.yml`, point the project path to `/cluster/scratch/ethz-username/PATH_TO_PROJECT` (be aware of the linux path format).

- In the same `workflow.yml`, specify the steps you wish to simulate. Please refer to this blog post on how to edit `workflow.yml`.
- Submit a batch job following this example command:

```
$ bsub -n 1 -R "span[host=1]" -R singularity -R "rusage[mem=2048,scratch=2048]" -W␣
↪1:00 "SINGULARITY_HOME=/projects singularity run -B \$TMPDIR:/tmp -B $SCRATCH cea_
↪latest.sif cea workflow --workflow /cluster/scratch/nethz-username/workflow.yml"
```

1.2.4 Other Commands

Before building a new singularity container, it is suggested to clean up the folders first.

- To remove a singularity container (e.g., a container named `cea_latest.sif` that is in `$SCRATCH`)

```
$ cd $SCRATCH
$ rm cea_latest.sif
```

- To clean up cache files

```
$ singularity cache clean
```

1.3 Installation guide for Mac OS

Working with the CityEnergyAnalyst (CEA) on a Mac OS system is a little bit messier than on Windows and requires using the Terminal to launch CEA. But don't worry - it works!

There are two installation methods, you can either: *(1) use the CEA source code from GitHub*, or *(2) use the CEA Docker image*. The former option gives you full access to CEA but is significantly more cumbersome to install as it involves running a lot of commands on Terminal. The latter option is much easier to install but working in a virtual container on Docker takes some getting used to. Both options are perfectly suited for users, but only the full installation is suitable for developers.

Choose the one that suits your needs!

1.3.1 (1) Use the CEA source code from GitHub

If you would like to develop CEA, this will be your method. Follow these instructions to install the CityEnergyAnalyst (CEA) on a Mac system (tested with macOS Mojave 10.14.6) from the source

Attention: We advise to follow the guide precisely:

- Be sure to **USE** `conda env create` **NOT** `conda create` familiar to experienced conda users. This command not only creates an environment, but also reads the `environment.yml` file, containing a list of packages (and versions) to install, as well as a definition of the channels to check.
- If you need to create a conda environment for the CEA that has a specific name (the default is `cea`) then use the `name` parameter: `conda env create --name your-env-name-here`
- `CONDA_SUBDIR=osx-64` will be used when creating the CEA conda environment to support both Intel and Apple Silicon Macs (this will be removed once all dependencies have been ported to ARM). Instructions below will indicate where it is being used.

- (Only applicable for Apple Silicon Macs) If at any point during installation or running CEA and it prompts you to install Rosetta, click install.
- This guide **SHOULD** work for both Intel and Apple Silicon Macs. If you encounter any issues when installing on either, help us report it as an issue on our GitHub page.

Prerequisites

- Download and install Homebrew.
- (*Optional*) Download and install Github Desktop (64-bit). Only required if you are unfamiliar with using Terminal and/or git commands.
- (*Optional*) Install Mamba. This will help to speed up the creation of the CEA conda environment.

Installation of the code base

Excluding the above software, CEA installation requires approximately 13 GB of storage (depending on your existing Python library) and 1 hour of your time.

Note: (*Experimental*) We have a script that can automate the process below. Just open a Terminal console and enter `/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/architecture-building-systems/CityEnergyAnalyst/update-mac-installation/install/mac_installation.sh)"` Continue on the next section to find out how to interact with CEA.

1. Open GitHub Desktop from the start menu.
2. **Clone the CEA repository:**
    1. Press `Cmd+Shift+O` (clone repository) and select the URL tab.
    2. Paste the CEA GitHub address: https://github.com/architecture-building-systems/CityEnergyAnalyst
    3. Click Clone, this will take ~ 5-10 minutes (Size 1.65 GB).
3. **Clone the CEA GUI repository:**
    1. Press `Cmd+Shift+O` (clone repository) and select the URL tab.
    2. Paste the CEA GUI GitHub address: https://github.com/architecture-building-systems/CityEnergyAnalyst-GUI
    3. Click Clone, this will take ~ 5 minutes (Size 600MB).
4. **Install CEA:**
    1. Open a Terminal console (you can find it in your Mac's *Applications* folder).
    2. Type `cd Documents/GitHub/CityEnergyAnalyst` and press ENTER.
    3. Type `CONDA_SUBDIR=osx-64 conda env create --name cea` and press ENTER. (If `mamba` is installed, replace `conda` in command to `mamba`. i.e. `CONDA_SUBDIR=osx-64 mamba env create --name cea`.)
    4. Grab a cup of tea and some toast, this will take ~45 minutes.
    5. Type `conda activate cea` and press ENTER.
    6. Type `pip install -e .` and press ENTER *(mind the dot '.' included in this command!)*.

5. **Build the CEA dashboard GUI:**
    1. Type `cd ..` and press ENTER, then type `cd CityEnergyAnalyst-GUI` and press ENTER.
    2. Install Yarn by typing `brew install yarn` and press ENTER.
    3. Type `yarn` and press ENTER.
    4. Type `yarn package` and press ENTER.
    5. You will find the CEA application in the folder `/Users/your_name/Documents/GitHub/CityEnergyAnalyst-GUI/out/CityEnergyAnalyst-GUI-darwin-*`
6. **Running CEA:**
    - You can run CEA a few different ways (see *Mac Interfaces* below).
    - If you are familiar with running CEA on a Windows computer, **please note that there are a few additional steps when running the dashboard on a Mac!**

> **Attention:** In order to run CEA on Mac, you will need to select the correct Daysim binaries:
>
> - If you are running the *Building Solar radiation* tool using the dashboard, make sure the parameter *daysim-bin-directory* (under *Advanced*) points to the correct Daysim binary folder (by default, this should be `/Users/your_name/Documents/GitHub/CityEnergyAnalyst/setup/Dependencies/Daysim/mac`).
> - If you are using the command line interface or Pycharm, you will need to modify the same parameter (i.e., `config.radiation.daysim_bin_directory`) in the config file (usually located in `/Users/your_name/cea.config`, where *your_name* represents your user name on your Mac).

Mac Interfaces

There are different ways in which you can interact with the code of CEA.

1. The command line interface: This is the command line to all the commands of CEA from your computer terminal
2. The dashboard: This a web-based interface to CEA, open source and developed by the CEA team.
3. The Pycharm interface: this interface provides access to all the source code of CEA.

Command line interface

You can run the command line interface by on the Terminal by running the command `conda activate cea`.

That's it! You can run the CEA command interface normally.

Dashboard (GUI)

In order to run the dashboard, you will need to do the following **each time you want to start the dashboard**:

1. Open the Terminal (you can find it in your Mac's *Applications* folder) and run the following command depending on your installation type:
2. Type `conda activate cea` and press ENTER, then type `cea dashboard` and press ENTER.

3. Wait for `start socketio.run` to appear in the Terminal before proceeding to the next step. (This might take a while if you are running this for the first time, around 3-5 min)
4. Run the CEA dashboard application you created in the last step of the installation above (`/Users/your_name/Documents/GitHub/CityEnergyAnalyst-GUI/out/CityEnergyAnalyst-GUI-darwin-*`).

You can now run the CEA dashboard interface normally!

Here you can find a series of blog posts to help you get started!

Pycharm

The Pycharm interface can be helpful if you would like to contribute to CEA, but it requires a few steps to get it up and running. In order to access and work on the source code of CEA from pycharm do:

1. Download and install Pycharm Community edition (64-bit) OR your own favorite editor.
2. Open PyCharm from the start menu and open project CityEnergyAnalyst (stored where you downloaded CityEnergyAnalyst).
3. Open `File>Settings>Project:CityEnergyAnalyst>Project Interpreter>Project Interpreter`.
4. Click on the settings button (it looks like a wheel) next to the current interpreter path, and click Add.
5. Click `Conda Environment` from the left hand list and select existing environment.
6. Point to the location of your conda environment. It should look something like `/Users/your_name/Miniconda2/envs/cea/python.exe` or `/Users/your_name/AppData/Local/conda/conda/envs/cea/python.exe` where *your_name* represents your user name on your Mac.
7. Click apply changes.

1.3.2 (2) Use the CEA docker image

If you would like using docker containers, follow these instructions to run CEA on a Mac OS system (tested with Mac OS Catalina). This method is suitable for users, but not developers. For developers, please refer to the second method below.

1. **Install Docker and run CEA:**
    1. You can find instructions on how to do that here.
    2. If you only plan to run CEA from the command line interface, you're done!
2. **If you would like to use the CEA dashboard, you will need to download and build it manually:**
    1. Download and install Github Desktop (64-bit).
    2. Download and install Miniconda(64-bit) for Python 3.8.
    3. Download and install Homebrew.
    4. **Clone the CEA GUI repository:**
        1. Press `Cmd+Shift+O` (clone repository) and select the URL tab.
        2. Paste the CEA GUI GitHub address: https://github.com/architecture-building-systems/CityEnergyAnalyst-GUI
        3. Click Clone, this will take ~ 5 minutes (Size 600MB).

5. **Build the CEA dashboard GUI:**

   1. Open a Terminal console (you can find it in your Mac's *Applications* folder).
   2. Type `cd Documents/GitHub/CityEnergyAnalyst-GUI` and press ENTER.

6. Install Yarn by typing `brew install yarn` and press ENTER.
7. Type `yarn` and press ENTER.
8. Type `yarn package` and press ENTER.
9. You will find the CEA application in the folder `/Users/your_name/Documents/GitHub/CityEnergyAnalyst-GUI/out/CityEnergyAnalyst-GUI-darwin-*`

3. **Running CEA:**

   - You can run CEA a couple of different ways (see *Docker Interfaces* below).
   - If you are familiar with running CEA on a Windows computer, **please note that there are a few additional steps when running the dashboard on a Mac!**

Docker Interfaces

There are different ways in which you can interact with the code of CEA.

1. The command line interface: This is the command line to all the commands of CEA from your computer terminal
2. The dashboard: This a web-based interface to CEA, open source and developed by the CEA team.
3. The Pycharm interface: this interface provides access to all the source code of CEA.

Command line interface

In order to run the command line interface in Docker, you will need to run the following command instead: `docker run --name cea_container -v path_to_your_cea_projects:/projects dockeruser/cea cea workflow --workflow /projects/workflow.yml`

That's it! You can run the CEA command interface normally.

Dashboard

In order to run the dashboard, you will need to do the following **each time you want to start the dashboard**:

1. Open the Terminal (you can find it in your Mac's *Applications* folder) and run the following command depending on your installation type:
2. Type `docker run -t -p 5050:5050 -v path_to_your_cea_projects:/projects dockeruser/cea`.
3. Run the CEA dashboard application you created in the last step of the installation above.

You can now run the CEA dashboard normally… well, mostly. You will need to pay attention to a few details, described below.

Since you will not be running CEA directly on your computer, you will need to select a project on your Docker container. So if your project is located, for example, in the directory `/Users/username/Documents/CEA_projects/my_project` you will need to select `/projects/my_project` as your project in the CEA Dashboard.

Also, note that your jobs in the dashboard might be listed as “pending” even when they have finished. If you would like to check if your job has finished, you can check the Terminal - it’s still running in the background.

CHAPTER 2

Cite

Please cite us like this in your publications:

The CEA team. The City Energy Analyst (Version 3.31.0). Zenodo. https://doi.org/10.5281/zenodo.598221

You can see a list of our own publications here: https://cityenergyanalyst.com/publications

CHAPTER 3

Tutorials

3.1 Users

General tutorials are constantly updated through blogs or videos.

- Blog posts
- Video tutorials

3.2 Developers

These are a collection of tutorials we consider essential for any new developer of CEA. We suggest to take them in order for a better learning experience.

1. *Contributing to City Energy Analyst (CEA)*
2. *User Stories*
3. how-to-report-bugs
4. how-to-use-github
5. how-to-name-variables
6. how-to-add-a-new-script-to-the-cea
7. how-to-test-the-cea
8. how-to-set-up-nsis
9. how-to-publish-cea
10. how-trace-inputlocator-works
11. how-to-document-cea

3.3 Legacy

Collection of outdated information about past functionality of CEA (use at your own risk)

1. script-data-flow
2. How are databases classified in CEA?
3. What are the input databases of CEA?
4. What are the default databases of CEA?
5. How to edit the input databases of CEA?
6. new-project-guide
7. cea-workflow-guide
8. How to create your own input geometry?
9. how-are-schedules-defined
10. How does the Urban Solar Radiation tool work?
11. How does the Dynamic Demand Forecast feature work?
12. How does the Renewable Energy Assessment tool work?
13. How does the Life Cycle Assessment tool work?
14. How does the 2000-Watt Bench-marking tool work?
15. How to study building retrofits in the CEA?
16. How does the Sensitivity Analysis tool work?
17. how-to-prepare-inputs-for-network-simulation
18. how-to-run-thermal-network-optimization
19. how-to-run-thermal-electrical-grid-planning
20. how-to-run-MPC-building
21. how-to-run-MPC-district
22. how-to-run-CEA-optimization

CHAPTER 4

Known issues

CEA uses Github Issues to document new ideas as well as issues and bugs.

The table below contains a number of the most common issues:

Issue #	Regarding	Description
1577	ArcGIS < 10.6	Internet Explorer Script Error: An error has occurred in the script of this page. Do you want to continue running scripts on this page?
1704	Installation	Error during installation via command line: Import Error: no module named arcpy
1708	Fiona/GDAL	Error after installation via command line ImportError: DLL load failed: The specified module could not be found.
1709	daysim_main.py	Error when running daysim_main.py: .wea weather file cannot be found

If you cannot find your issue here, please check the complete list of known issues.

4.1 Report a new issue

For any problems please post a new issue here.

Please read the how-to-report-bugs guide and review the open issues before posting.

We have a turn-over time of a couple of days.

We appreciate your contribution!

CHAPTER 5

Glossary

This glossary contains all the written input and output variables used by CEA. These variables are stored in databases, themed by the type of information they contain. There are two main types of databases in CEA: input and output. This glossary is organised through the cea's inputlocator method, which is used to retrieve the information within each file.

5.1 Input

Basic input provided by users:

5.1.1 get_site_polygon

path: `inputs/building-geometry/site.shp`

The following file is used by these scripts: `zone_helper`

Variable	Description
`FID`	Shapefile ID
`geometry`	Shapefile POLYGON

5.1.2 get_street_network

path: `inputs/networks/streets.shp`

The following file is used by these scripts: `network_layout`, `optimization`

Variable	Description
`geometry`	Geometry
`Id`	Unique building ID. It must start with a letter.

5.1.3 get_surroundings_geometry

path: `inputs/building-geometry/surroundings.shp`

The following file is used by these scripts: `radiation`, `schedule_maker`

Variable	Description
`floors_ag`	Number of floors above ground (incl. ground floor, Minimum one floor is needed)
`geometry`	Shapefile POLYGON
`height_ag`	Height above ground (incl. ground floor, Minimum one floor is needed)
`Name`	Unique building ID. It must start with a letter.
`REFERENCE`	Reference to data (if any)

5.1.4 get_terrain

path: `inputs/topography/terrain.tif`

The following file is used by these scripts: `radiation`, `schedule_maker`

Variable	Description
`raster_value`	TODO

5.1.5 get_weather

path: `databases/weather/Zug-inducity_1990_2010_TMY.epw`

The following file is used by these scripts: `weather_helper`

Variable	Description
`aerosol_opt_thousandths (index = 29)`	TODO
`Albedo (index = 32)`	Albedo
`atmos_Pa (index = 9)`	Atmospheric pressure
`ceiling_hgt_m (index = 25)`	TODO
`datasource (index = 5)`	Source of data
`day (index = 2)`	TODO
`days_last_snow (index = 31)`	Days since last snow
`dewpoint_C (index = 7)`	TODO
`difhorillum_lux (index = 18)`	TODO
`difhorrad_Whm2 (index = 15)`	TODO
`dirnorillum_lux (index = 17)`	TODO
`dirnorrad_Whm2 (index = 14)`	TODO
`drybulb_C (index = 6)`	TODO
`extdirrad_Whm2 (index = 11)`	TODO
`exthorrad_Whm2 (index = 10)`	TODO
`glohorillum_lux (index = 16)`	TODO
`glohorrad_Whm2 (index = 13)`	TODO
`horirsky_Whm2 (index = 12)`	TODO
`hour (index = 3)`	TODO
`liq_precip_depth_mm (index = 33)`	TODO
`liq_precip_rate_Hour (index = 34)`	TODO

Continued on next page

Table 1 – continued from previous page

Variable	Description
minute (index = 4)	TODO
month (index = 1)	TODO
opaqskycvr_tenths (index = 23)	TODO
precip_wtr_mm (index = 28)	TODO
presweathcodes (index = 27)	TODO
presweathobs (index = 26)	TODO
relhum_percent (index = 8)	TODO
snowdepth_cm (index = 30)	TODO
totskycvr_tenths (index = 22)	TODO
visibility_km (index = 24)	TODO
winddir_deg (index = 20)	TODO
windspd_ms (index = 21)	TODO
year (index = 0)	TODO
zenlum_lux (index = 19)	TODO

5.1.6 get_zone_geometry

path: `inputs/building-geometry/zone.shp`

The following file is used by these scripts: `archetypes_mapper`, `decentralized`, `demand`, `emissions`, `network_layout`, `optimization`, `photovoltaic`, `photovoltaic_thermal`, `radiation`, `schedule_maker`, `sewage_potential`, `shallow_geothermal_potential`, `solar_collector`, `thermal_network`

Variable	Description
floors_ag	Number of floors above ground (incl. ground floor, minimum one floor is needed)
floors_bg	Number of floors below ground (basement)
geometry	Shapefile POLYGON
height_ag	Height above ground (incl. ground floor, minimum one floor is needed)
height_bg	Height below ground (basement)
Name	Unique building ID. It must start with a letter.
REFERENCE	Reference to data (if any)

5.2 Intermediate Input

Intermediate input generated by archetypes-mapper and data-initializer:

5.2.1 get_building_air_conditioning

path: `inputs/building-properties/air_conditioning_systems.dbf`

The following file is used by these scripts: `demand`

Variable	Description
cool_ends	End of the cooling season - use 00|00 when there is none
cool_starts	Start of the cooling season - use 00|00 when there is none
heat_ends	End of the heating season - use 00|00 when there is none
heat_starts	Start of the heating season - use 00|00 when there is none
Name	Unique building ID. It must start with a letter.
type_cs	Type of cooling HVAC assembly (relates to "code" in HVAC assemblies)
type_ctrl	Type of heating and cooling control HVAC assembly (relates to "code" in HVAC assemblies)
type_dhw	Type of hot water HVAC assembly (relates to "code" in HVAC assemblies)
type_hs	Type of heating HVAC assembly (relates to "code" in HVAC assemblies)
type_vent	Type of ventilation HVAC assembly (relates to "code" in HVAC assemblies)

5.2.2 get_building_architecture

path: `inputs/building-properties/architecture.dbf`

The following file is used by these scripts: `demand`, `emissions`, `radiation`, `schedule_maker`

Variable	Description
Es	Fraction of gross floor area with electrical demands.
Hs_ag	Fraction of above ground gross floor area air-conditioned.
Hs_bg	Fraction of below ground gross floor area air-conditioned.
Name	Unique building ID. It must start with a letter.
Ns	Fraction of net gross floor area.
type_base	Basement floor construction assembly (relates to "code" in ENVELOPE assemblies)
type_cons	Type of construction assembly (relates to "code" in ENVELOPE assemblies)
type_floor	Internal floor construction assembly (relates to "code" in ENVELOPE assemblies)
type_leak	Tightness level assembly (relates to "code" in ENVELOPE assemblies)
type_part	Internal partitions construction assembly (relates to "code" in ENVELOPE assemblies)
type_roof	Roof construction assembly (relates to "code" in ENVELOPE assemblies)
type_shade	Shading system assembly (relates to "code" in ENVELOPE assemblies)
type_wall	External wall construction assembly (relates to "code" in ENVELOPE assemblies)
type_win	Window assembly (relates to "code" in ENVELOPE assemblies)
void_deck	Number of floors (from the ground up) with an open envelope (default = 0, should be lower than floors_ag.)
wwr_east	Window to wall ratio in in facades facing east
wwr_north	Window to wall ratio in in facades facing north
wwr_south	Window to wall ratio in in facades facing south
wwr_west	Window to wall ratio in in facades facing west

5.2.3 get_building_comfort

path: `inputs/building-properties/indoor_comfort.dbf`

The following file is used by these scripts: `demand`, `schedule_maker`

Variable	Description
Name	Unique building ID. It must start with a letter.
RH_max_pc	Upper bound of relative humidity
RH_min_pc	Lower_bound of relative humidity
Tcs_set_C	Setpoint temperature for cooling system
Tcs_setb_C	Setback point of temperature for cooling system
Ths_set_C	Setpoint temperature for heating system
Ths_setb_C	Setback point of temperature for heating system
Ve_lsp	Minimum outdoor air ventilation rate per person for Air Quality

5.2.4 get_building_internal

path: `inputs/building-properties/internal_loads.dbf`

The following file is used by these scripts: `demand`, `schedule_maker`

Variable	Description
Ea_Wm2	Peak specific electrical load due to computers and devices
Ed_Wm2	Peak specific electrical load due to servers/data centres
El_Wm2	Peak specific electrical load due to artificial lighting
Epro_Wm2	Peak specific electrical load due to industrial processes
Ev_kWveh	Peak capacity of electric battery per vehicle
Name	Unique building ID. It must start with a letter.
Occ_m2p	Occupancy density
Qcpro_Wm2	Peak specific process cooling load
Qcre_Wm2	Peak specific cooling load due to refrigeration (cooling rooms)
Qhpro_Wm2	Peak specific process heating load
Qs_Wp	Peak sensible heat load of people
Vw_ldp	Peak specific fresh water consumption (includes cold and hot water)
Vww_ldp	Peak specific daily hot water consumption
X_ghp	Moisture released by occupancy at peak conditions

5.2.5 get_building_supply

path: `inputs/building-properties/supply_systems.dbf`

The following file is used by these scripts: `decentralized`, `demand`, `emissions`, `system_costs`

Variable	Description
Name	Unique building ID. It must start with a letter.
type_cs	Type of cooling supply assembly (refers to "code" in SUPPLY assemblies)
type_dhw	Type of hot water supply assembly (refers to "code" in SUPPLY assemblies)
type_el	Type of electrical supply assembly (refers to "code" in SUPPLY assemblies)
type_hs	Type of heating supply assembly (refers to "code" in SUPPLY assemblies)

5.2.6 get_building_weekly_schedules

path: `inputs/building-properties/schedules/B001.csv`

The following file is used by these scripts: `demand`, `schedule_maker`

Variable	Description
METADATA	TODO
MONTHLY_MULTIPLIER	Monthly probabilities of occupancy throughout the year

5.2.7 get_database_air_conditioning_systems

path: inputs/technology/assemblies/HVAC.xlsx

The following file is used by these scripts: demand

Table 2: Worksheet: CONTROLLER

Variable	Description
code	Unique ID of the controller
Description	Describes the type of controller
dT_Qcs	correction temperature of emission losses due to control system of cooling
dT_Qhs	correction temperature of emission losses due to control system of heating

Table 3: Worksheet: COOLING

Variable	Description
class_cs	Type or class of the cooling system
code	Unique ID of the heating system
convection_cs	Convective part of the power of the heating system in relation to the total power
Description	Describes the type of cooling system
dTcs0_ahu_C	Nominal temperature increase on the water side of the air-handling units
dTcs0_aru_C	Nominal temperature increase on the water side of the air-recirculation units
dTcs0_scu_C	Nominal temperature increase on the water side of the sensible cooling units
dTcs_C	Set-point correction for space emission systems
Qcsmax_Wm2	Maximum heat flow permitted by cooling system per m2 gross floor area
Tc_sup_air_ahu_C	Supply air temperature of the air-handling units
Tc_sup_air_aru_C	Supply air temperature of the air-recirculation units
Tscs0_ahu_C	Nominal supply temperature of the water side of the air-handling units
Tscs0_aru_C	Nominal supply temperature of the water side of the air-recirculation units
Tscs0_scu_C	Nominal supply temperature of the water side of the sensible cooling units

Table 4: Worksheet: HEATING

Variable	Description
class_hs	Type or class of the heating system
code	Unique ID of the heating system
convection_hs	Convective part of the power of the heating system in relation to the total power
Description	Description
dThs0_ahu_C	Nominal temperature increase on the water side of the air-handling units
dThs0_aru_C	Nominal temperature increase on the water side of the air-recirculation units
dThs0_shu_C	Nominal temperature increase on the water side of the sensible heating units
dThs_C	correction temperature of emission losses due to type of heating system
Qhsmax_Wm2	Maximum heat flow permitted by heating system per m2 gross floor area
Th_sup_air_ahu_C	Supply air temperature of the air-recirculation units
Th_sup_air_aru_C	Supply air temperature of the air-handling units
Tshs0_ahu_C	Nominal supply temperature of the water side of the air-handling units
Tshs0_aru_C	Nominal supply temperature of the water side of the air-recirculation units
Tshs0_shu_C	Nominal supply temperature of the water side of the sensible heating units

Table 5: Worksheet: HOT_WATER

Variable	Description
code	Unique ID of the hot water supply system
Description	Describes the Type of hot water supply system
Qwwmax_Wm2	Maximum heat flow permitted by hot water system per m2 gross floor area
Tsww0_C	Typical supply water temperature.

Table 6: Worksheet: VENTILATION

Variable	Description	
code	Unique ID of the type of ventilation	
Description	Describes the Type of ventilation	
ECONOMIZER	Boolean	economizer on
HEAT_REC	Boolean	heat recovery on
MECH_VENT	Boolean	mechanical ventilation on
NIGHT_FLSH	Boolean	night flush on
WIN_VENT	Boolean	window ventilation on

5.2.8 get_database_construction_standards

path: inputs/technology/archetypes/CONSTRUCTION_STANDARDS.xlsx

The following file is used by these scripts: archetypes_mapper

Table 7: Worksheet: ENVELOPE_ASSEMBLIES

Variable	Description
Es	Fraction of gross floor area with electrical demands.
Hs_ag	Fraction of above ground gross floor area air-conditioned.
Hs_bg	Fraction of below ground gross floor area air-conditioned
Ns	Fraction of net gross floor area.
STANDARD	Unique ID of Construction Standard
type_base	Basement floor construction assembly (relates to "code" in ENVELOPE assemblies)
type_cons	Type of construction assembly (relates to "code" in ENVELOPE assemblies)
type_floor	Internal floor construction assembly (relates to "code" in ENVELOPE assemblies)
type_leak	Tightness level assembly (relates to "code" in ENVELOPE assemblies)
type_part	Internal partitions construction assembly (relates to "code" in ENVELOPE assemblies)
type_roof	Roof construction assembly (relates to "code" in ENVELOPE assemblies)
type_shade	Shading system assembly (relates to "code" in ENVELOPE assemblies)
type_wall	External wall construction assembly (relates to "code" in ENVELOPE assemblies)
type_win	Window assembly (relates to "code" in ENVELOPE assemblies)
void_deck	Number of floors (from the ground up) with an open envelope (default = 0)
wwr_east	Window to wall ratio in in facades facing east
wwr_north	Window to wall ratio in in facades facing north
wwr_south	Window to wall ratio in in facades facing south
wwr_west	Window to wall ratio in in facades facing west

Table 8: Worksheet: HVAC_ASSEMBLIES

Variable	Description
cool_ends	End of the cooling season - use 00|00 when there is none
cool_starts	Start of the cooling season - use 00|00 when there is none
heat_ends	End of the heating season - use 00|00 when there is none
heat_starts	Start of the heating season - use 00|00 when there is none
STANDARD	Unique ID of Construction Standard
type_cs	Type of cooling HVAC assembly (relates to "code" in HVAC assemblies)
type_ctrl	Type of heating and cooling control HVAC assembly (relates to "code" in HVAC assemblies)
type_dhw	Type of hot water HVAC assembly (relates to "code" in HVAC assemblies)
type_hs	Type of heating HVAC assembly (relates to "code" in HVAC assemblies)
type_vent	Type of ventilation HVAC assembly (relates to "code" in HVAC assemblies)

Table 9: Worksheet: STANDARD_DEFINITION

Variable	Description
Description	Description of the construction standard
STANDARD	Unique ID of Construction Standard
YEAR_END	Upper limit of year interval where the building properties apply
YEAR_START	Lower limit of year interval where the building properties apply

Table 10: Worksheet: SUPPLY_ASSEMBLIES

Variable	Description
STANDARD	Unique ID of Construction Standard
type_cs	Type of cooling supply assembly (refers to "code" in SUPPLY assemblies)
type_dhw	Type of hot water supply assembly (refers to "code" in SUPPLY assemblies)
type_el	Type of electrical supply assembly (refers to "code" in SUPPLY assemblies)
type_hs	Type of heating supply assembly (refers to "code" in SUPPLY assemblies)

5.2.9 get_database_conversion_systems

path: inputs/technology/components/CONVERSION.xlsx

The following file is used by these scripts: decentralized, optimization, photovoltaic, photovoltaic_thermal, solar_collector

Table 11: Worksheet: Absorption_chiller

Variable	Description		
a	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
a_e	parameter in the characteristic equations to calculate the evaporator side		
a_g	parameter in the characteristic equations to calculate the generator side		
assumption	items made by assumptions in this technology		
b	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
c	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
cap_max	maximum capacity		
cap_min	minimum capacity		
code	identifier of each unique equipment		
currency	currency-year information of the investment cost function	should be unified to USD	
d	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
Description	Describes the Type of Absorption Chiller		
e	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
e_e	parameter in the characteristic equations to calculate the evaporator side		
e_g	parameter in the characteristic equations to calculate the generator side		
IR_%	interest rate charged on the loan for the capital cost		
LT_yr	lifetime of this technology		
m_cw	external flow rate of cooling water at the condenser and absorber		
m_hw	external flow rate of hot water at the generator		
O&M_%	operation and maintenance cost factor (fraction of the investment cost)		
r_e	parameter in the characteristic equations to calculate the evaporator side		
r_g	parameter in the characteristic equations to calculate the generator side		
s_e	parameter in the characteristic equations to calculate the evaporator side		
s_g	parameter in the characteristic equations to calculate the generator side		
type	type of absorption chiller		
unit	unit of the min/max capacity		

Table 12: Worksheet: BH

Variable	Description	
a	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
assumption	items made by assumptions in this technology	
b	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
c	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
cap_max	maximum capacity	
cap_min	minimum capacity	
code	identifier of each unique equipment	
currency	currency-year information of the investment cost function	
d	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
Description	Describes the type of borehole heat exchanger	
e	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
IR_%	interest rate charged on the loan for the capital cost	
LT_yr	lifetime of this technology	
O&M_%	operation and maintenance cost factor (fraction of the investment cost)	
unit	unit of the min/max capacity	

Table 13: Worksheet: Boiler

Variable	Description	
a	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
assumption	items made by assumptions in this technology	
b	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
c	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
cap_max	maximum capacity	
cap_min	minimum capacity	
code	identifier of each unique equipment	
currency	currency-year information of the investment cost function	
d	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
Description	Describes the type of boiler	
e	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
IR_%	interest rate charged on the loan for the capital cost	
LT_yr	lifetime of this technology	
O&M_%	operation and maintenance cost factor (fraction of the investment cost)	
unit	unit of the min/max capacity	

Table 14: Worksheet: CCGT

Variable	Description	
a	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
assumption	items made by assumptions in this technology	
b	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
c	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
cap_max	maximum capacity	
cap_min	minimum capacity	
code	identifier of each unique equipment	
currency	currency-year information of the investment cost function	should be unified to USD
d	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
Description	Describes the type of combined-cycle gas turbine	
e	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
IR_%	interest rate charged on the loan for the capital cost	
LT_yr	lifetime of this technology	
O&M_%	operation and maintenance cost factor (fraction of the investment cost)	
unit	unit of the min/max capacity	

Table 15: Worksheet: Chiller

Variable	Description	
a	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
assumption	items made by assumptions in this technology	
b	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
c	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
cap_max	maximum capacity	
cap_min	minimum capacity	
code	identifier of each unique equipment	
currency	currency-year information of the investment cost function	should be unified to USD
d	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
Description	Describes the source of the benchmark standards.	
e	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
IR_%	interest rate charged on the loan for the capital cost	
LT_yr	lifetime of this technology	
O&M_%	operation and maintenance cost factor (fraction of the investment cost)	
unit	unit of the min/max capacity	

Table 16: Worksheet: CT

Variable	Description	
a	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
assumption	items made by assumptions in this technology	
b	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
c	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
cap_max	maximum capacity	
cap_min	minimum capacity	
code	identifier of each unique equipment	
currency	currency-year information of the investment cost function	should be unified to USD
d	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
Description	Describes the type of cooling tower	
e	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
IR_%	interest rate charged on the loan for the capital cost	
LT_yr	lifetime of this technology	
O&M_%	operation and maintenance cost factor (fraction of the investment cost)	
unit	unit of the min/max capacity	

Table 17: Worksheet: FC

Variable	Description	
a	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
assumption	items made by assumptions in this technology	
b	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
c	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
cap_max	maximum capacity	
cap_min	minimum capacity	
code	identifier of each unique equipment	
currency	currency-year information of the investment cost function	should be unified to USD
d	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
Description	Describes the type of fuel cell	
e	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
IR_%	interest rate charged on the loan for the capital cost	
LT_yr	lifetime of this technology	
O&M_%	operation and maintenance cost factor (fraction of the investment cost)	
unit	unit of the min/max capacity	

Table 18: Worksheet: Furnace

Variable	Description	
a	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
assumption	items made by assumptions in this technology	
b	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
c	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
cap_max	maximum capacity	
cap_min	minimum capacity	
code	identifier of each unique equipment	
currency	currency-year information of the investment cost function	should be unified to USD
d	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
Description	Describes the type of furnace	
e	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)
IR_%	interest rate charged on the loan for the capital cost	
LT_yr	lifetime of this technology	
O&M_%	operation and maintenance cost factor (fraction of the investment cost)	
unit	unit of the min/max capacity	

Table 19: Worksheet: HEX

Variable	Description		
a	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	
a_p	parameter in the pressure loss function	f(x) = a_p + b_p*x^c_p + d_p*ln(x) + e_p*x*ln*(x)	where x is the capacity mass flow rate [W/K]
assumption	items made by assumptions in this technology		
b	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	
b_p	parameter in the pressure loss function	f(x) = a_p + b_p*x^c_p + d_p*ln(x) + e_p*x*ln*(x)	where x is the capacity mass flow rate [W/K]
c	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	
c_p	parameter in the pressure loss function	f(x) = a_p + b_p*x^c_p + d_p*ln(x) + e_p*x*ln*(x)	where x is the capacity mass flow rate [W/K]
cap_max	maximum capacity		
cap_min	minimum capacity		
code	identifier of each unique equipment		
currency	currency-year information of the investment cost function	should be unified to USD	
d	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	
d_p	parameter in the pressure loss function	f(x) = a_p + b_p*x^c_p + d_p*ln(x) + e_p*x*ln*(x)	where x is the capacity mass flow rate [W/K]
Description	Describes the type of heat exchanger		
e	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	
e_p	parameter in the pressure loss function	f(x) = a_p + b_p*x^c_p + d_p*ln(x) + e_p*x*ln*(x)	where x is the capacity mass flow rate [W/K]
IR_%	interest rate charged on the loan for the capital cost		
LT_yr	lifetime of this technology		
O&M_%	operation and maintenance cost factor (fraction of the investment cost)		
unit	unit of the min/max capacity		

Table 20: Worksheet: HP

Variable	Description		
a	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
assumption	items made by assumptions in this technology		
b	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
c	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
cap_max	maximum capacity		
cap_min	minimum capacity		
code	identifier of each unique equipment		
currency	currency-year information of the investment cost function	should be unified to USD	
d	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
Description	Describes the source of the heat pump		
e	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
IR_%	interest rate charged on the loan for the capital cost		
LT_yr	lifetime of this technology		
O&M_%	operation and maintenance cost factor (fraction of the investment cost)		
unit	unit of the min/max capacity		

Table 21: Worksheet: Pump

Variable	Description		
a	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
assumption	items made by assumptions in this technology		
b	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
c	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
cap_max	maximum capacity		
cap_min	minimum capacity		
code	identifier of each unique equipment		
currency	currency-year information of the investment cost function	should be unified to USD	
d	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
Description	Describes the source of the benchmark standards.		
e	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
IR_%	interest rate charged on the loan for the capital cost		
LT_yr	lifetime of this technology		
O&M_%	operation and maintenance cost factor (fraction of the investment cost)		
unit	unit of the min/max capacity		

Table 22: Worksheet: PV

Variable	Description		
a	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
assumption	items made by assumptions in this technology		
b	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
c	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
cap_max	maximum capacity		
cap_min	minimum capacity		
code	identifier of each unique equipment		
currency	currency-year information of the investment cost function	should be unified to USD	
d	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
Description	Describes the source of the benchmark standards.		
e	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
IR_%	interest rate charged on the loan for the capital cost		
LT_yr	lifetime of this technology		
misc_losses	losses from cabling	resistances etc…	
module_length	length of the PV module		
O&M_%	operation and maintenance cost factor (fraction of the investment cost)		
PV_a0	parameters for air mass modifier	f(x) = a0 + a1*x + a2*x**2 + a3*x**3 + a4*x**4	where x is the relative air mass
PV_a1	parameters for air mass modifier	f(x) = a0 + a1*x + a2*x**2 + a3*x**3 + a4*x**4	where x is the relative air mass
PV_a2	parameters for air mass modifier	f(x) = a0 + a1*x + a2*x**2 + a3*x**3 + a4*x**4	where x is the relative air mass
PV_a3	parameters for air mass modifier	f(x) = a0 + a1*x + a2*x**2 + a3*x**3 + a4*x**4	where x is the relative air mass
PV_a4	parameters for air mass modifier	f(x) = a0 + a1*x + a2*x**2 + a3*x**3 + a4*x**4	where x is the relative air mass
PV_Bref	cell maximum power temperature coefficient		
PV_n	nominal efficiency		
PV_noct	nominal operating cell temperature		
PV_th	glazing thickness		
type	redundant		
unit	unit of the min/max capacity		

Table 23: Worksheet: PVT

Variable	Description		
a	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
assumption	items made by assumptions in this technology		
b	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
c	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
cap_max	maximum capacity		
cap_min	minimum capacity		
code	identifier of each unique equipment		
currency	currency-year information of the investment cost function	should be unified to USD	
d	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
Description	Describes the type of photovoltaic thermal technology		
e	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
IR_%	interest rate charged on the loan for the capital cost		
LT_yr	lifetime of this technology		
O&M_%	operation and maintenance cost factor (fraction of the investment cost)		
unit	unit of the min/max capacity		

Table 24: Worksheet: SC

Variable	Description	
a	parameter in the investment cost function	f(x) = a + b*x^c + d*ln
aperture_area_ratio	ratio of aperture area to panel area	
assumption	items made by assumptions in this technology	
b	parameter in the investment cost function	f(x) = a + b*x^c + d*ln
c	parameter in the investment cost function	f(x) = a + b*x^c + d*ln
c1	collector heat loss coefficient at zero temperature difference and wind speed	
c2	ctemperature difference dependency of the heat loss coefficient	
C_eff	thermal capacity of module	
cap_max	maximum capacity	
cap_min	minimum capacity	
code	identifier of each unique equipment	
Cp_fluid	heat capacity of the heat transfer fluid	
currency	currency-year information of the investment cost function	should be unified to US
d	parameter in the investment cost function	f(x) = a + b*x^c + d*ln
Description	Describes the type of solar collector	
dP1	pressure drop at zero flow rate	
dP2	pressure drop at nominal flow rate (mB0)	
dP3	pressure drop at maximum flow rate (mB_max)	
dP4	pressure drop at minimum flow rate (mB_min)	
e	parameter in the investment cost function	f(x) = a + b*x^c + d*ln
IAM_d	incident angle modifier for diffuse radiation	

Table 24 – continued from previous page

Variable	Description	
IR_%	interest rate charged on the loan for the capital cost	
LT_yr	lifetime of this technology	
mB0_r	nominal flow rate per aperture area	
mB_max_r	maximum flow rate per aperture area	
mB_min_r	minimum flow rate per aperture area	
module_area_m2	module area of a solar collector	
module_length_m	lengh of a solar collector module	
n0	zero loss efficiency at normal incidence	
O&M_%	operation and maintenance cost factor (fraction of the investment cost)	
t_max	maximum operating temperature	
type	type of the solar collector (FP: flate-plate or ET: evacuated-tube)	
unit	unit of the min/max capacity	

Table 25: Worksheet: TES

Variable	Description		
a	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
assumption	items made by assumptions in this storage technology		
b	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
c	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
C_mat_%	Working fluid replacement cost factor (fraction of the investment cost)		
cap_max	maximum capacity		
cap_min	minimum capacity		
code	Unique code that identifies the thermal energy storage technology		
Cp_kJkgK	heat capacity of working fluid		
currency	currency-year information of the investment cost function	should be unified to USD	
d	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
Description	Describes the thermal energy storage technology		
e	parameter in the investment cost function	f(x) = a + b*x^c + d*ln(x) + e*x*ln*(x)	where x is the capacity
HL_kJkg	Lantent heat of working fluid at phase change temperature		
IR_%	interest rate charged on the loan for the capital cost		
LT_mat_yr	lifetime of the working fluid of this storage technology		
LT_yr	lifetime of this storage technology		
O&M_%	operation and maintnance cost factor (fraction of the investment cost)		
Rho_T_PHCH_kgm3	Density of working fluid at phase change temperature		
T_max_C	Maximum temperature of working fluid at full discharge		
T_min_C	Minimum temperature of working fluid at full charge		
T_PHCH_C	Phase change temperature of working fluid		
type	code that identifies whether the storage is used for heating or cooling (different properties of the transport media)		
unit	unit which describes the minimum and maximum capacity		

5.2.10 get_database_distribution_systems

path: inputs/technology/components/DISTRIBUTION.xlsx

The following file is used by these scripts: optimization, thermal_network

Table 26: Worksheet: THERMAL_GRID

Variable	Description
Code	pipe ID from the manufacterer
D_ext_m	external pipe diameter tolerance for the nominal diameter (DN)
D_ins_m	maximum pipe diameter tolerance for the nominal diameter (DN)
D_int_m	internal pipe diameter tolerance for the nominal diameter (DN)
Inv_USD2015perm	Typical cost of investment for a given pipe diameter.
Pipe_DN	Nominal pipe diameter
Vdot_max_m3s	maximum volumetric flow rate for the nominal diameter (DN)
Vdot_min_m3s	minimum volumetric flow rate for the nominal diameter (DN)

5.2.11 get_database_envelope_systems

path: inputs/technology/assemblies/ENVELOPE.xlsx

The following file is used by these scripts: demand, radiation, schedule_maker

Table 27: Worksheet: CONSTRUCTION

Variable	Description
Cm_Af	Internal heat capacity per unit of air conditioned area. Defined according to ISO 13790.
code	Type of construction
Description	Describes the Type of construction

Table 28: Worksheet: FLOOR

Variable	Description
code	Type of roof
Description	Describes the Type of roof
GHG_FLOOR_kgCO2m2	Embodied emissions per m2 of roof.(entire building life cycle)
U_base	Thermal transmittance of floor including linear losses (+10%). Defined according to ISO 13790.

Table 29: Worksheet: ROOF

Variable	Description
a_roof	Solar absorption coefficient. Defined according to ISO 13790.
code	Type of roof
Description	Describes the Type of roof
e_roof	Emissivity of external surface. Defined according to ISO 13790.
GHG_ROOF_kgCO2m2	Embodied emissions per m2 of roof.(entire building life cycle)
r_roof	Reflectance in the Red spectrum. Defined according Radiance. (long-wave)
U_roof	Thermal transmittance of windows including linear losses (+10%). Defined according to ISO 13790.

Table 30: Worksheet: SHADING

Variable	Description
code	Type of shading
Description	Describes the source of the benchmark standards.
rf_sh	Shading coefficient when shading device is active. Defined according to ISO 13790.

Table 31: Worksheet: TIGHTNESS

Variable	Description
code	Type of tightness
Description	Describes the Type of tightness
n50	Air exchanges per hour at a pressure of 50 Pa.

Table 32: Worksheet: WALL

Variable	Description
a_wall	Solar absorption coefficient. Defined according to ISO 13790.
code	Type of wall
Description	Describes the Type of wall
e_wall	Emissivity of external surface. Defined according to ISO 13790.
GHG_WALL_kgCO2m2	Embodied emissions per m2 of walls (entire building life cycle)
r_wall	Reflectance in the Red spectrum. Defined according Radiance. (long-wave)
U_wall	Thermal transmittance of windows including linear losses (+10%). Defined according to ISO 13790.

Table 33: Worksheet: WINDOW

Variable	Description
code	Window type code to relate to other databases
Description	Describes the source of the benchmark standards.
e_win	Emissivity of external surface. Defined according to ISO 13790.
F_F	Window frame fraction coefficient. Defined according to ISO 13790.
G_win	Solar heat gain coefficient. Defined according to ISO 13790.
GHG_WIN_kgCO2m2	Embodied emissions per m2 of windows.(entire building life cycle)
U_win	Thermal transmittance of windows including linear losses (+10%). Defined according to ISO 13790.

5.2.12 get_database_feedstocks

path: `inputs/technology/components/FEEDSTOCKS.xlsx`

The following file is used by these scripts: `decentralized`, `emissions`, `system_costs`, `optimization`

Table 34: Worksheet: BIOGAS

Variable	Description
GHG_kgCO2MJ	Non-renewable Green House Gas Emissions factor
hour	hour of a 24 hour day
Opex_var_buy_USD2015kWh	buying price
Opex_var_sell_USD2015kWh	selling price
reference	reference

Table 35: Worksheet: COAL

Variable	Description
GHG_kgCO2MJ	Non-renewable Green House Gas Emissions factor
hour	hour of a 24 hour day
Opex_var_buy_USD2015kWh	buying price
Opex_var_sell_USD2015kWh	selling price
reference	reference

Table 36: Worksheet: DRYBIOMASS

Variable	Description
GHG_kgCO2MJ	Non-renewable Green House Gas Emissions factor
hour	hour of a 24 hour day
Opex_var_buy_USD2015kWh	buying price
Opex_var_sell_USD2015kWh	selling price
reference	reference

Table 37: Worksheet: GRID

Variable	Description
GHG_kgCO2MJ	Non-renewable Green House Gas Emissions factor
hour	hour of a 24 hour day
Opex_var_buy_USD2015kWh	buying price
Opex_var_sell_USD2015kWh	selling price
reference	reference

Table 38: Worksheet: NATURALGAS

Variable	Description
GHG_kgCO2MJ	Non-renewable Green House Gas Emissions factor
hour	hour of a 24 hour day
Opex_var_buy_USD2015kWh	buying price
Opex_var_sell_USD2015kWh	selling price
reference	reference

Table 39: Worksheet: OIL

Variable	Description
GHG_kgCO2MJ	Non-renewable Green House Gas Emissions factor
hour	hour of a 24 hour day
Opex_var_buy_USD2015kWh	buying price
Opex_var_sell_USD2015kWh	selling price
reference	reference

Table 40: Worksheet: SOLAR

Variable	Description
GHG_kgCO2MJ	Non-renewable Green House Gas Emissions factor
hour	hour of a 24 hour day
Opex_var_buy_USD2015kWh	buying price
Opex_var_sell_USD2015kWh	selling price
reference	reference

Table 41: Worksheet: WETBIOMASS

Variable	Description
GHG_kgCO2MJ	Non-renewable Green House Gas Emissions factor
hour	hour of a 24 hour day
Opex_var_buy_USD2015kWh	buying price
Opex_var_sell_USD2015kWh	selling price
reference	reference

Table 42: Worksheet: WOOD

Variable	Description
GHG_kgCO2MJ	Non-renewable Green House Gas Emissions factor
hour	hour of a 24 hour day
Opex_var_buy_USD2015kWh	buying price
Opex_var_sell_USD2015kWh	selling price
reference	reference

5.2.13 get_database_standard_schedules_use

path: inputs/technology/archetypes/schedules/RESTAURANT.csv

The following file is used by these scripts: archetypes_mapper

Table 43: Worksheet: APPLIANCES

Variable	Description		
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
DAY	Day of the week (weekday	saturday	or sunday)

Table 44: Worksheet: COOLING

Variable	Description		
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
DAY	Day of the week (weekday	saturday	or sunday)

Table 45: Worksheet: ELECTROMOBILITY

Variable	Description		
1	Average number of electric vehicles in this hour		
2	Average number of electric vehicles in this hour		
3	Average number of electric vehicles in this hour		
4	Average number of electric vehicles in this hour		
5	Average number of electric vehicles in this hour		
6	Average number of electric vehicles in this hour		
7	Average number of electric vehicles in this hour		
8	Average number of electric vehicles in this hour		
9	Average number of electric vehicles in this hour		
10	Average number of electric vehicles in this hour		
11	Average number of electric vehicles in this hour		
12	Average number of electric vehicles in this hour		
13	Average number of electric vehicles in this hour		
14	Average number of electric vehicles in this hour		
15	Average number of electric vehicles in this hour		
16	Average number of electric vehicles in this hour		
17	Average number of electric vehicles in this hour		
18	Average number of electric vehicles in this hour		
19	Average number of electric vehicles in this hour		
20	Average number of electric vehicles in this hour		
21	Average number of electric vehicles in this hour		
22	Average number of electric vehicles in this hour		
23	Average number of electric vehicles in this hour		
24	Average number of electric vehicles in this hour		
DAY	Day of the week (weekday	saturday	or sunday)

Table 46: Worksheet: HEATING

Variable	Description		
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
DAY	Day of the week (weekday	saturday	or sunday)

Table 47: Worksheet: LIGHTING

Variable	Description		
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
DAY	Day of the week (weekday	saturday	or sunday)

Table 48: Worksheet: METADATA

Variable	Description
metadata	

Table 49: Worksheet: MONTHLY_MULTIPLIER

Variable	Description
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	

Table 50: Worksheet: OCCUPANCY

Variable	Description		
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
DAY	Day of the week (weekday	saturday	or sunday)

Table 51: Worksheet: PROCESSES

Variable	Description		
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
DAY	Day of the week (weekday	saturday	or sunday)

Table 52: Worksheet: SERVERS

Variable	Description		
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
DAY	Day of the week (weekday	saturday	or sunday)

Table 53: Worksheet: WATER

Variable	Description		
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
DAY	Day of the week (weekday	saturday	or sunday)

5.2.14 get_database_supply_assemblies

path: inputs/technology/assemblies/SUPPLY.xlsx

The following file is used by these scripts: demand, emissions, system_costs

Table 54: Worksheet: COOLING

Variable	Description
CAPEX_USD2015kW	Capital costs per kW
code	Code of cooling supply assembly
Description	description
efficiency	efficiency of the all in one system
feedstock	feedstock used by the the all in one system (refers to the FEEDSTOCK database)
IR_%	interest rate charged on the loan for the capital cost
LT_yr	lifetime of assembly
O&M_%	operation and maintenance cost factor (fraction of the investment cost)
reference	reference
scale	whether the all in one system is used at the building or the district scale

Table 55: Worksheet: ELECTRICITY

Variable	Description
CAPEX_USD2015kW	Capital costs per kW
code	Type of all in one system
Description	Description of Type of all in one system
efficiency	efficiency of the all in one system
feedstock	feedstock used by the the all in one system (refers to the FEEDSTOCK database)
IR_%	interest rate charged on the loan for the capital cost
LT_yr	lifetime of assembly
O&M_%	operation and maintenance cost factor (fraction of the investment cost)
reference	Reference of the data
scale	whether the all in one system is used at the building or the district scale

Table 56: Worksheet: HEATING

Variable	Description
CAPEX_USD2015kW	Capital costs per kW
code	Type of all in one system
Description	Description of Type of all in one system
efficiency	efficiency of the all in one system
feedstock	feedstock used by the the all in one system (refers to the FEEDSTOCK database)
IR_%	interest rate charged on the loan for the capital cost
LT_yr	lifetime of assembly
O&M_%	operation and maintenance cost factor (fraction of the investment cost)
reference	Reference of the data
scale	whether the all in one system is used at the building or the district scale

Table 57: Worksheet: HOT_WATER

Variable	Description
CAPEX_USD2015kW	Capital costs per kW
code	Type of all in one system
Description	Description of Type of all in one system
efficiency	efficiency of the all in one system
feedstock	feedstock used by the the all in one system (refers to the FEEDSTOCK database)
IR_%	interest rate charged on the loan for the capital cost
LT_yr	lifetime of assembly
O&M_%	operation and maintenance cost factor (fraction of the investment cost)
reference	Reference of the data
scale	whether the all in one system is used at the building or the district scale

5.2.15 get_database_use_types_properties

path: `inputs/technology/archetypes/use_types/USE_TYPE_PROPERTIES.xlsx`

The following file is used by these scripts: `archetypes_mapper`

Table 58: Worksheet: INDOOR_COMFORT

Variable	Description
code	use type code (refers to building use type)
RH_max_pc	Upper bound of relative humidity
RH_min_pc	Lower_bound of relative humidity
Tcs_set_C	Setpoint temperature for cooling system
Tcs_setb_C	Setback point of temperature for cooling system
Ths_set_C	Setpoint temperature for heating system
Ths_setb_C	Setback point of temperature for heating system
Ve_lsp	Indoor quality requirements of indoor ventilation per person

Table 59: Worksheet: INTERNAL_LOADS

Variable	Description
code	use type code (refers to building use type)
Ea_Wm2	Peak specific electrical load due to computers and devices
Ed_Wm2	Peak specific electrical load due to servers/data centres
El_Wm2	Peak specific electrical load due to artificial lighting
Epro_Wm2	Peak specific electrical load due to industrial processes
Ev_kWveh	Peak capacity of electrical battery per vehicle
Occ_m2p	Occupancy density
Qcpro_Wm2	Peak specific process cooling load
Qcre_Wm2	Peak specific cooling load due to refrigeration (cooling rooms)
Qhpro_Wm2	Peak specific process heating load
Qs_Wp	Peak sensible heat load of people
Vw_ldp	Peak specific fresh water consumption (includes cold and hot water)
Vww_ldp	Peak specific daily hot water consumption
X_ghp	Moisture released by occupancy at peak conditions

5.3 Output

Output generated from simulations:

5.3.1 get_building_typology

path: `inputs/building-properties/typology.dbf`

The following file is used by these scripts: `archetypes_mapper`, `demand`, `emissions`

Variable	Description
1ST_USE	First (Main) Use type of the building
1ST_USE_R	Fraction of gross floor area for first Use Type
2ND_USE	Second Use type of the building
2ND_USE_R	Fraction of gross floor area for second Use Type
3RD_USE	Third Use type of the building
3RD_USE_R	Fraction of gross floor area for third Use Type
Name	Unique building ID. It must start with a letter.
REFERENCE	Reference to data (if any)
STANDARD	Construction Standard (relates to "code" in Supply Assemblies)
YEAR	Construction year

5.3.2 get_costs_operation_file

path: `outputs/data/costs/operation_costs.csv`

The following file is used by these scripts:

Variable	Description
Aocc_m2	Occupied floor area (heated/cooled)
Capex_a_sys_building_scale_USD	Annualized capital expenditures of building-scale systems
Capex_a_sys_district_scale_USD	Annualized capital expenditures of district-scale systems
COAL_hs_cost_m2yr	Operation costs of coal due to space heating per unit conditioned floor area
COAL_hs_cost_yr	Operation costs of coal due to space heating
COAL_ww_cost_m2yr	Operation costs of coal due to hotwater per unit conditioned floor area
COAL_ww_cost_yr	Operation costs of coal due to hotwater
DC_cdata_cost_m2yr	Operation costs of district cooling due to data center cooling per unit conditioned floo
DC_cdata_cost_yr	Operation costs of district cooling due to data center cooling
DC_cre_cost_m2yr	Operation costs of district cooling due to cool room refrigeration per unit conditioned
DC_cre_cost_yr	Operation costs of district cooling due to cool room refrigeration
DC_cs_cost_m2yr	Operation costs of district cooling due to space cooling per unit conditioned floor area
DC_cs_cost_yr	Operation costs of district cooling due to space cooling
DH_hs_cost_m2yr	Operation costs of district heating due to space heating per unit conditioned floor area
DH_hs_cost_yr	Operation costs of district heating due to space heating
DH_ww_cost_m2yr	Operation costs of district heating due to domestic hot water per unit conditioned floor
DH_ww_cost_yr	Operation costs of district heating due to domestic hot water
GRID_cost_m2yr	Operation costs due to electricity supply from the grid per unit conditioned floor area
GRID_cost_yr	Operation costs due to electricity supply from the grid
Name	Unique building ID. It must start with a letter.
NG_hs_cost_m2yr	Operation costs of natural gas due to space heating per unit conditioned floor area
NG_hs_cost_yr	Operation costs of natural gas due to space heating
NG_ww_cost_m2yr	Operation costs of natural gas due to domestic hot water per unit conditioned floor are
NG_ww_cost_yr	Operation costs of natural gas due to domestic hot water
OIL_hs_cost_m2yr	Operation costs of oil due to space heating per unit conditioned floor area
OIL_hs_cost_yr	Operation costs of oil due to space heating
OIL_ww_cost_m2yr	Operation costs of oil due to domestic hot water per unit conditioned floor area
OIL_ww_cost_yr	Operation costs of oil due to domestic hot water
Opex_a_sys_building_scale_USD	Annual operational expenditures of building-scale systems
Opex_a_sys_district_scale_USD	Annual operational expenditures of district-scale systems
PV_cost_m2yr	Operation costs due to electricity supply from PV per unit conditioned floor area

Continued on r

Table 60 – continued from previous page

Variable	Description
PV_cost_yr	Operation costs due to electricity supply from PV
SOLAR_hs_cost_m2yr	Operation costs due to solar collectors for space heating per unit conditioned floor are
SOLAR_hs_cost_yr	Operation costs due to solar collectors for space heating
SOLAR_ww_cost_m2yr	Operation costs due to solar collectors for domestic hot water per unit conditioned flo
SOLAR_ww_cost_yr	Operation costs due to solar collectors for domestic hot water
WOOD_hs_cost_m2yr	Operation costs of wood due to space heating per unit conditioned floor area
WOOD_hs_cost_yr	Operation costs of wood due to space heating
WOOD_ww_cost_m2yr	Operation costs of wood due to domestic hot water per unit conditioned floor area
WOOD_ww_cost_yr	Operation costs of wood due to domestic hot water

5.3.3 get_demand_results_file

path: `outputs/data/demand/B001.csv`

The following file is used by these scripts: `decentralized`, `optimization`, `sewage_potential`, `thermal_network`

Variable	Description
COAL_hs_kWh	Coal requirement for space heating supply
COAL_ww_kWh	Coal requirement for hotwater supply
DATE	Time stamp for each day of the year ascending in hour intervals.
DC_cdata_kWh	District cooling for data center cooling demand
DC_cre_kWh	District cooling for refrigeration demand
DC_cs_kWh	Energy consumption of space cooling system (if supplied by District Cooling), DC_cs = Qcs_sys / ef
DH_hs_kWh	Energy requirement by district heating (space heating supply)
DH_ww_kWh	Energy requirement by district heating (hotwater supply)
E_cdata_kWh	Data centre cooling specific electricity consumption.
E_cre_kWh	Refrigeration system electricity consumption.
E_cs_kWh	Energy consumption of cooling system (if supplied by electricity grid), E_cs = Qcs_sys / eff_cs
E_hs_kWh	Heating system electricity consumption.
E_sys_kWh	End-use total electricity consumption = Ea + El + Edata + Epro + Eaux + Ev + Eve
E_ww_kWh	Hot water system electricity consumption.
Ea_kWh	End-use electricity for appliances
Eal_kWh	End-use electricity consumption of appliances and lighting, Eal = El_W + Ea_W
Eaux_kWh	End-use auxiliary electricity consumption, Eaux = Eaux_fw + Eaux_ww + Eaux_cs + Eaux_hs + Ehs
Edata_kWh	End-use data centre electricity consumption.
El_kWh	End-use electricity for lights
Epro_kWh	End-use electricity consumption for industrial processes.
Ev_kWh	End-use electricity for electric vehicles
Eve_kWh	End-use electricity for ventilation
GRID_a_kWh	Grid electricity consumption for appliances
GRID_aux_kWh	Grid electricity consumption for auxiliary loads
GRID_cdata_kWh	Grid electricity consumption for servers cooling
GRID_cre_kWh	Grid electricity consumption for refrigeration
GRID_cs_kWh	Grid electricity consumption for space cooling
GRID_data_kWh	Grid electricity consumption for servers
GRID_hs_kWh	Grid electricity consumption for space heating
GRID_kWh	Grid total electricity consumption, GRID_a + GRID_l + GRID_v + GRID_ve + GRID_data + GRID
GRID_l_kWh	Grid electricity consumption for lighting

Table 61 – continued from previous page

Variable	Description
GRID_pro_kWh	Grid electricity consumption for industrial processes
GRID_ve_kWh	Grid electricity consumption for ventilation
GRID_ww_kWh	Grid electricity consumption for hot water supply
I_rad_kWh	Radiative heat loss
I_sol_and_I_rad_kWh	Net radiative heat gain
I_sol_kWh	Solar heat gain
mcpcdata_sys_kWperC	Capacity flow rate (mass flow* specific heat capacity) of the chilled water delivered to data centre.
mcpcre_sys_kWperC	Capacity flow rate (mass flow* specific heat Capacity) of the chilled water delivered to refrigeration.
mcpcs_sys_ahu_kWperC	Capacity flow rate (mass flow* specific heat Capacity) of the chilled water delivered to air handling u
mcpcs_sys_aru_kWperC	Capacity flow rate (mass flow* specific heat Capacity) of the chilled water delivered to air recirculati
mcpcs_sys_kWperC	Capacity flow rate (mass flow* specific heat Capacity) of the chilled water delivered to space cooling
mcpcs_sys_scu_kWperC	Capacity flow rate (mass flow* specific heat Capacity) of the chilled water delivered to sensible cooli
mcphs_sys_ahu_kWperC	Capacity flow rate (mass flow* specific heat Capacity) of the warm water delivered to air handling ur
mcphs_sys_aru_kWperC	Capacity flow rate (mass flow* specific heat Capacity) of the warm water delivered to air recirculatio
mcphs_sys_kWperC	Capacity flow rate (mass flow* specific heat Capacity) of the warm water delivered to space heating.
mcphs_sys_shu_kWperC	Capacity flow rate (mass flow* specific heat Capacity) of the warm water delivered to sensible heatin
mcptw_kWperC	Capacity flow rate (mass flow* specific heat capacity) of the fresh water
mcpww_sys_kWperC	Capacity flow rate (mass flow* specific heat capacity) of domestic hot water
Name	Unique building ID. It must start with a letter.
NG_hs_kWh	NG requirement for space heating supply
NG_ww_kWh	NG requirement for hotwater supply
OIL_hs_kWh	OIL requirement for space heating supply
OIL_ww_kWh	OIL requirement for hotwater supply
people	Predicted occupancy: number of people in building
PV_kWh	PV electricity consumption
Q_gain_lat_peop_kWh	Latent heat gain from people
Q_gain_sen_app_kWh	Sensible heat gain from appliances
Q_gain_sen_base_kWh	Sensible heat gain from transmission through the base
Q_gain_sen_data_kWh	Sensible heat gain from data centres
Q_gain_sen_light_kWh	Sensible heat gain from lighting
Q_gain_sen_peop_kWh	Sensible heat gain from people
Q_gain_sen_pro_kWh	Sensible heat gain from industrial processes.
Q_gain_sen_roof_kWh	Sensible heat gain from transmission through the roof
Q_gain_sen_vent_kWh	Sensible heat gain from ventilation and infiltration
Q_gain_sen_wall_kWh	Sensible heat gain from transmission through the walls
Q_gain_sen_wind_kWh	Sensible heat gain from transmission through the windows
Q_loss_sen_ref_kWh	Sensible heat loss from refrigeration systems
QC_sys_kWh	Total energy demand for cooling, QC_sys = Qcs_sys + Qcdata_sys + Qcre_sys + Qcpro_sys
Qcdata_kWh	Data centre space cooling demand
Qcdata_sys_kWh	End-use data center cooling demand
Qcpro_sys_kWh	Process cooling demand
Qcre_kWh	Refrigeration space cooling demand
Qcre_sys_kWh	End-use refrigeration demand
Qcs_dis_ls_kWh	Cooling system distribution losses
Qcs_em_ls_kWh	Cooling system emission losses
Qcs_kWh	Specific cooling demand
Qcs_lat_ahu_kWh	AHU latent cooling demand
Qcs_lat_aru_kWh	ARU latent cooling demand
Qcs_lat_sys_kWh	Total latent cooling demand for all systems

Table 61 – continued from previous page

Variable	Description
Qcs_sen_ahu_kWh	AHU sensible cooling demand
Qcs_sen_aru_kWh	ARU sensible cooling demand
Qcs_sen_scu_kWh	SHU sensible cooling demand
Qcs_sen_sys_kWh	Total sensible cooling demand for all systems
Qcs_sys_ahu_kWh	AHU system cooling demand
Qcs_sys_aru_kWh	ARU system cooling demand
Qcs_sys_kWh	End-use space cooling demand, Qcs_sys = Qcs_sen_sys + Qcs_lat_sys + Qcs_em_ls + Qcs_dis_ls
Qcs_sys_scu_kWh	SCU system cooling demand
QH_sys_kWh	Total energy demand for heating, QH_sys = Qww_sys + Qhs_sys + Qhpro_sys
Qhpro_sys_kWh	Process heating demand
Qhs_dis_ls_kWh	Heating system distribution losses
Qhs_em_ls_kWh	Heating system emission losses
Qhs_kWh	Sensible heating system demand
Qhs_lat_ahu_kWh	AHU latent heating demand
Qhs_lat_aru_kWh	ARU latent heating demand
Qhs_lat_sys_kWh	Total latent heating demand for all systems
Qhs_sen_ahu_kWh	AHU sensible heating demand
Qhs_sen_aru_kWh	ARU sensible heating demand
Qhs_sen_shu_kWh	SHU sensible heating demand
Qhs_sen_sys_kWh	Total sensible heating demand
Qhs_sys_ahu_kWh	Space heating demand in AHU
Qhs_sys_aru_kWh	Space heating demand in ARU
Qhs_sys_kWh	End-use space heating demand, Qhs_sys = Qhs_sen_sys + Qhs_em_ls + Qhs_dis_ls
Qhs_sys_shu_kWh	SHU system heating demand
Qww_kWh	DHW specific heating demand
Qww_sys_kWh	End-use hotwater demand
SOLAR_hs_kWh	Solar thermal energy requirement for space heating supply
SOLAR_ww_kWh	Solar thermal energy requirement for hotwater supply
T_ext_C	Outdoor temperature
T_int_C	Indoor temperature
Tcdata_sys_re_C	Cooling supply temperature of the data centre
Tcdata_sys_sup_C	Cooling return temperature of the data centre
Tcre_sys_re_C	Cooling return temperature of the refrigeration system.
Tcre_sys_sup_C	Cooling supply temperature of the refrigeration system.
Tcs_sys_re_ahu_C	Return temperature cooling system
Tcs_sys_re_aru_C	Return temperature cooling system
Tcs_sys_re_C	System cooling return temperature.
Tcs_sys_re_scu_C	Return temperature cooling system
Tcs_sys_sup_ahu_C	Supply temperature cooling system
Tcs_sys_sup_aru_C	Supply temperature cooling system
Tcs_sys_sup_C	System cooling supply temperature.
Tcs_sys_sup_scu_C	Supply temperature cooling system
theta_o_C	Operative temperature in building (RC-model) used for comfort plotting
Ths_sys_re_ahu_C	Return temperature heating system
Ths_sys_re_aru_C	Return temperature heating system
Ths_sys_re_C	Heating system return temperature.
Ths_sys_re_shu_C	Return temperature heating system
Ths_sys_sup_ahu_C	Supply temperature heating system
Ths_sys_sup_aru_C	Supply temperature heating system

Table 61 – continued from previous page

Variable	Description
Ths_sys_sup_C	Heating system supply temperature.
Ths_sys_sup_shu_C	Supply temperature heating system
Tww_sys_re_C	Return temperature hotwater system
Tww_sys_sup_C	Supply temperature hotwater system
WOOD_hs_kWh	WOOD requirement for space heating supply
WOOD_ww_kWh	WOOD requirement for hotwater supply
x_int	Internal mass fraction of humidity (water/dry air)

5.3.4 get_geothermal_potential

path: `outputs/data/potentials/Shallow_geothermal_potential.csv`

The following file is used by these scripts: `optimization`

Variable	Description
Area_avail_m2	areas available to install ground source heat pumps
QGHP_kW	geothermal heat potential
Ts_C	ground temperature

5.3.5 get_lca_embodied

path: `outputs/data/emissions/Total_LCA_embodied.csv`

The following file is used by these scripts:

Variable	Description
GFA_m2	Gross floor area
GHG_sys_embodied_kgCO2m2	Embodied emissions per conditioned floor area due to building construction and decomissioning
GHG_sys_embodied_tonCO2	Embodied emissions due to building construction and decomissioning
Name	Unique building ID. It must start with a letter.

5.3.6 get_lca_mobility

path: `outputs/data/emissions/Total_LCA_mobility.csv`

The following file is used by these scripts:

Variable	Description
GFA_m2	Gross floor area
GHG_sys_mobility_kgCO2m2	Operational emissions per unit of conditioned floor area due to mobility
GHG_sys_mobility_tonCO2	Operational emissions due to mobility
Name	Unique building ID. It must start with a letter.

5.3.7 get_lca_operation

path: `outputs/data/emissions/Total_LCA_operation.csv`

The following file is used by these scripts:

Variable	Description
COAL_hs_ghg_kgm2	Operational emissions per unit of conditioned floor area of the coal powererd heating system
COAL_hs_ghg_ton	Operational emissions of the coal powered heating system
COAL_hs_nre_pen_GJ	Operational primary energy demand (non-renewable) for coal powered heating system
COAL_hs_nre_pen_MJm2	Operational primary energy demand per unit of conditioned floor area (non-renewable) of the coal p
COAL_ww_ghg_kgm2	Operational emissions per unit of conditioned floor area of the coal powered domestic hot water sys
COAL_ww_ghg_ton	Operational emissions of the coal powered domestic hot water system
COAL_ww_nre_pen_GJ	Operational primary energy demand (non-renewable) for coal powered domestic hot water system
COAL_ww_nre_pen_MJm2	Operational primary energy demand per unit of conditioned floor area (non-renewable) of the coal p
DC_cdata_ghg_kgm2	Operational emissions per unit of conditioned floor area of the district cooling for the data center
DC_cdata_ghg_ton	Operational emissions of the district cooling for the data center
DC_cdata_nre_pen_GJ	Operational primary energy demand (non-renewable) for district cooling system for cool room refri
DC_cdata_nre_pen_MJm2	Operational primary energy demand per unit of conditioned floor area (non-renewable) for district c
DC_cre_ghg_kgm2	Operational emissions per unit of conditioned floor area for district cooling system for cool room re
DC_cre_ghg_ton	Operational emissions for district cooling system for cool room refrigeration
DC_cre_nre_pen_GJ	Operational primary energy demand (non-renewable) for district cooling system for cool room refri
DC_cre_nre_pen_MJm2	Operational primary energy demand per unit of conditioned floor area (non-renewable) for cool roo
DC_cs_ghg_kgm2	Operational emissions per unit of conditioned floor area of the district cooling
DC_cs_ghg_ton	Operational emissions of the district cooling
DC_cs_nre_pen_GJ	Operational primary energy demand (non-renewable) for district cooling system
DC_cs_nre_pen_MJm2	Operational primary energy demand per unit of conditioned floor area (non-renewable) of the distric
DH_hs_ghg_kgm2	Operational emissions per unit of conditioned floor area of the district heating system
DH_hs_ghg_ton	Operational emissions of the district heating system
DH_hs_nre_pen_GJ	Operational primary energy demand (non-renewable) for district heating system
DH_hs_nre_pen_MJm2	Operational primary energy demand per unit of conditioned floor area (non-renewable) of the distric
DH_ww_ghg_kgm2	Operational emissions per unit of conditioned floor area of the district heating domestic hot water sy
DH_ww_ghg_ton	Operational emissions of the district heating powered domestic hot water system
DH_ww_nre_pen_GJ	Operational primary energy demand (non-renewable) for district heating powered domestic hot wat
DH_ww_nre_pen_MJm2	Operational primary energy demand per unit of conditioned floor area (non-renewable) of the distric
GFA_m2	Gross floor area
GHG_sys_kgCO2m2	Total operational emissions per unit of conditioned floor area
GHG_sys_tonCO2	Total operational emissions
GRID_ghg_kgm2	Operational emissions per unit of conditioned floor area from grid electricity
GRID_ghg_ton	Operational emissions of the electrictiy from the grid
GRID_nre_pen_GJ	Operational primary energy demand (non-renewable) from the grid
GRID_nre_pen_MJm2	Operational primary energy demand per unit of conditioned floor area (non-renewable) from grid el
Name	Unique building ID. It must start with a letter.
NG_hs_ghg_kgm2	Operational emissions per unit of conditioned floor area of the natural gas powered heating system
NG_hs_ghg_ton	Operational emissions of the natural gas powered heating system
NG_hs_nre_pen_GJ	Operational primary energy demand (non-renewable) for natural gas powered heating system
NG_hs_nre_pen_MJm2	Operational primary energy demand per unit of conditioned floor area (non-renewable) of the natura
NG_ww_ghg_kgm2	Operational emissions per unit of conditioned floor area of the gas powered domestic hot water syst
NG_ww_ghg_ton	Operational emissions of the solar powered domestic hot water system
NG_ww_nre_pen_GJ	Operational primary energy demand (non-renewable) for natural gas powered domestic hot water sy
NG_ww_nre_pen_MJm2	Operational primary energy demand per unit of conditioned floor area (non-renewable) of the natura
OIL_hs_ghg_kgm2	Operational emissions per unit of conditioned floor area of the oil powered heating system
OIL_hs_ghg_ton	Operational emissions of the oil powered heating system
OIL_hs_nre_pen_GJ	Operational primary energy demand (non-renewable) for oil powered heating system
OIL_hs_nre_pen_MJm2	Operational primary energy demand per unit of conditioned floor area (non-renewable) of the oil po

Table 62 – continued from previous page

Variable	Description
OIL_ww_ghg_kgm2	Operational emissions per unit of conditioned floor area of the oil powered domestic hot water syste
OIL_ww_ghg_ton	Operational emissions of the oil powered domestic hot water system
OIL_ww_nre_pen_GJ	Operational primary energy demand (non-renewable) for oil powered domestic hot water system
OIL_ww_nre_pen_MJm2	Operational primary energy demand per unit of conditioned floor area (non-renewable) of the oil po
PV_ghg_kgm2	Operational emissions per unit of conditioned floor area for PV-System
PV_ghg_ton	Operational emissions of the PV-System
PV_nre_pen_GJ	Operational primary energy demand (non-renewable) for PV-System
PV_nre_pen_MJm2	Operational primary energy demand per unit of conditioned floor area (non-renewable) for PV Syst
SOLAR_hs_ghg_kgm2	Operational emissions per unit of conditioned floor area of the solar powered heating system
SOLAR_hs_ghg_ton	Operational emissions of the solar powered heating system
SOLAR_hs_nre_pen_GJ	Operational primary energy demand (non-renewable) of the solar powered heating system
SOLAR_hs_nre_pen_MJm2	Operational primary energy demand per unit of conditioned floor area (non-renewable) of the solar
SOLAR_ww_ghg_kgm2	Operational emissions per unit of conditioned floor area of the solar powered domestic hot water sy
SOLAR_ww_ghg_ton	Operational emissions of the solar powered domestic hot water system
SOLAR_ww_nre_pen_GJ	Operational primary energy demand (non-renewable) for solar powered domestic hot water system
SOLAR_ww_nre_pen_MJm2	Operational primary energy demand per unit of conditioned floor area (non-renewable) of the solar
WOOD_hs_ghg_kgm2	Operational emissions per unit of conditioned floor area of the wood powered heating system
WOOD_hs_ghg_ton	Operational emissions of the wood powered heating system
WOOD_hs_nre_pen_GJ	Operational primary energy demand (non-renewable) for wood powered heating system
WOOD_hs_nre_pen_MJm2	Operational primary energy demand per unit of conditioned floor area (non-renewable) of the wood
WOOD_ww_ghg_kgm2	Operational emissions per unit of conditioned floor area of the wood powered domestic hot water sy
WOOD_ww_ghg_ton	Operational emissions of the wood powered domestic hot water system
WOOD_ww_nre_pen_GJ	Operational primary energy demand (non-renewable) for wood powered domestic hot water system
WOOD_ww_nre_pen_MJm2	Operational primary energy demand per unit of conditioned floor area (non-renewable) of the wood

5.3.8 get_multi_criteria_analysis

path: `outputs/data/multicriteria/gen_2_multi_criteria_analysis.csv`

The following file is used by these scripts:

Variable	Description
Capex_a_sys_building_scale_USD	Annualized Capital costs of building-scale systems
Capex_a_sys_district_scale_USD	Capital costs of district-scale systems
Capex_a_sys_USD	Capital costs of all systems
Capex_total_sys_building_scale_USD	Capital costs of building-scale systems
Capex_total_sys_district_scale_USD	Capital costs of district-scale systems
Capex_total_sys_USD	Capital costs of all systems
generation	Generation or iteration
GHG_rank	Rank for emissions
GHG_sys_building_scale_tonCO2	Green house gas emissions of building-scale systems
GHG_sys_district_scale_tonCO2	Green house gas emissions of building-scale systems
GHG_sys_tonCO2	Green house gas emissions of all systems
individual	system number
individual_name	Name of system
normalized_Capex_total	normalization of CAPEX
normalized_emissions	normalization of GHG
normalized_Opex	Normalization of OPEX
normalized_TAC	normalization of TAC
Opex_a_sys_building_scale_USD	Operational costs of building-scale systems
Opex_a_sys_district_scale_USD	Operational costs of district-scale systems
Opex_a_sys_USD	Operational costs of all systems
TAC_rank	Rank of TAC
TAC_sys_building_scale_USD	Equivalent annual costs of building-scale systems
TAC_sys_district_scale_USD	Equivalent annual of district-scale systems
TAC_sys_USD	Equivalent annual costs of all systems
user_MCDA	Best system accoridng to user mult-criteria weights
user_MCDA_rank	Rank of Best system according to user mult-criteria weights

5.3.9 get_network_energy_pumping_requirements_file

path: `outputs/data/thermal-network/DH__plant_pumping_load_kW.csv`

The following file is used by these scripts:

Variable	Description
pressure_loss_return_kW	pumping electricity required to overcome pressure losses in the return network
pressure_loss_substations_kW	pumping electricity required to overcome pressure losses in the substations
pressure_loss_supply_kW	pumping electricity required to overcome pressure losses in the supply network
pressure_loss_total_kW	pumping electricity required to overcome pressure losses in the entire network

5.3.10 get_network_layout_edges_shapefile

path: `outputs/data/thermal-network/DH/edges.shp`

The following file is used by these scripts: `thermal_network`

Variable	Description
geometry	Geometry
length_m	lenth of this edge
Name	Unique network pipe ID.
Pipe_DN	Classifies nominal pipe diameters (DN) into typical bins.
Type_mat	Material type

5.3.11 get_network_layout_nodes_shapefile

path: `outputs/data/thermal-network/DH/nodes.shp`

The following file is used by these scripts: `thermal_network`

Variable	Description
Building	Unique building ID. e.g. "B01"
geometry	Geometry
Name	Unique node ID. e.g. "NODE1"
Type	Type of node.

5.3.12 get_network_linear_pressure_drop_edges

path: `outputs/data/thermal-network/DH__linear_pressure_drop_edges_Paperm.csv`

The following file is used by these scripts:

Variable	Description
PIPE0	linear pressure drop in this pipe section

5.3.13 get_network_linear_thermal_loss_edges_file

path: `outputs/data/thermal-network/DH__linear_thermal_loss_edges_Wperm.csv`

The following file is used by these scripts:

Variable	Description
PIPE0	linear thermal losses in thie pipe section

5.3.14 get_network_pressure_at_nodes

path: `outputs/data/thermal-network/DH__pressure_at_nodes_Pa.csv`

The following file is used by these scripts:

Variable	Description
NODE0	pressure at this node

5.3.15 get_network_temperature_plant

path: `outputs/data/thermal-network/DH__temperature_plant_K.csv`

The following file is used by these scripts:

Variable	Description
`temperature_return_K`	Plant return temperature at each time step
`temperature_supply_K`	Plant supply temperature at each time step

5.3.16 get_network_temperature_return_nodes_file

path: `outputs/data/thermal-network/DH__temperature_return_nodes_K.csv`

The following file is used by these scripts:

Variable	Description
`NODE0`	Return temperature at node NODE0 (outlet temperature of NODE0) at each time step

5.3.17 get_network_temperature_supply_nodes_file

path: `outputs/data/thermal-network/DH__temperature_supply_nodes_K.csv`

The following file is used by these scripts:

Variable	Description
`NODE0`	Supply temperature at node NODE0 (inlet temperature of NODE0) at each time step

5.3.18 get_network_thermal_loss_edges_file

path: `outputs/data/thermal-network/DH__thermal_loss_edges_kW.csv`

The following file is used by these scripts:

Variable	Description
`PIPE0`	Thermal losses along pipe PIPE0 at each time step

5.3.19 get_network_total_pressure_drop_file

path: `outputs/data/thermal-network/DH__plant_pumping_pressure_loss_Pa.csv`

The following file is used by these scripts: `optimization`

Variable	Description
`pressure_loss_return_Pa`	Pressure losses in the return network at each time step
`pressure_loss_substations_Pa`	Pressure losses in all substations at each time step
`pressure_loss_supply_Pa`	Pressure losses in the supply network at each time step
`pressure_loss_total_Pa`	Total pressure losses in the entire thermal network at each time step

5.3.20 get_network_total_thermal_loss_file

path: `outputs/data/thermal-network/DH__total_thermal_loss_edges_kW.csv`

The following file is used by these scripts: `optimization`

Variable	Description
`thermal_loss_return_kW`	Thermal losses in the supply network at each time step
`thermal_loss_supply_kW`	Thermal losses in the return network at each time step
`thermal_loss_total_kW`	Thermal losses in the entire thermal network at each time step

5.3.21 get_nominal_edge_mass_flow_csv_file

path: `outputs/data/thermal-network/Nominal_EdgeMassFlow_at_design_{network_type}__kgpers.csv`

The following file is used by these scripts: `thermal_network`

Variable	Description
`PIPE0`	Mass flow rate in pipe PIPE0 at design operating conditions

5.3.22 get_nominal_node_mass_flow_csv_file

path: `outputs/data/thermal-network/Nominal_NodeMassFlow_at_design_{network_type}__kgpers.csv`

The following file is used by these scripts: `thermal_network`

Variable	Description
`NODE0`	Mass flow rate in node NODE0 at design operating conditions

5.3.23 get_optimization_building_scale_cooling_capacity

path: `outputs/data/optimization/slave/gen_1/ind_0_building_scale_cooling_capacity.csv`

The following file is used by these scripts:

Variable	Description
Capacity_ACH_SC_FP_cool_building_scale_W	Thermal Capacity of Absorption Chiller and Solar Collector (Flat Plate) for Decentralized Building
Capacity_ACHHT_FP_cool_building_scale_W	Thermal Capacity of High-Temperature Absorption Chiller and Solar Collector (Flat Plate) for Decentralized Building
Capacity_BaseVCC_AS_cool_building_scale_W	Thermal Capacity of Base load Vapor Compression Chiller for Decentralized Building
Capacity_DX_AS_cool_building_scale_W	Thermal Capacity of Direct Expansion Air-Source for Decentralized Building
Capacity_VCCHT_AS_cool_building_scale_W	Thermal Capacity of High-Temperature Vapor Compression Chiller (Air-Source) for Decentralized Building
Capaticy_ACH_SC_ET_cool_building_scale_W	Thermal Capacity of Absorption Chiller and Solar Collector (Evacuated Tube)for Decentralized Building
Name	Unique building ID. It must start with a letter

5.3.24 get_optimization_building_scale_heating_capacity

path: `outputs/data/optimization/slave/gen_0/ind_1_building_scale_heating_capacity.csv`

The following file is used by these scripts:

Variable	Description
Capacity_BaseBoiler_NG_heat_building_scale_W	Thermal capacity of Base load boiler for Decentralized building
Capacity_FC_NG_heat_building_scale_W	Thermal capacity of Fuel Cell for Decentralized building
Capacity_GS_HP_heat_building_scale_W	Thermal capacity of ground-source heat pump for Decentralized building
Name	Unique building ID. It must start with a letter

5.3.25 get_optimization_checkpoint

path: `outputs/data/optimization/master/CheckPoint_1`

The following file is used by these scripts:

Variable	Description
difference_generational_distances	TODO
generation	TODO
generational_distances	TODO
selected_population	TODO
tested_population	TODO

5.3.26 get_optimization_decentralized_folder_building_result_cooling

path: `outputs/data/optimization/decentralized/{building}_{configuration}_cooling.csv`

The following file is used by these scripts: `optimization`

Variable	Description
Best configuration	Index of best configuration simulated
Capacity_ACH_SC_FP_W	Thermal Capacity of Absorption Chiller connected to Flat-plate Solar Collector
Capacity_ACHHT_FP_W	Thermal Capacity of High Temperature Absorption Chiller connected to Solar Collector (flat Plate)
Capacity_BaseVCC_AS_W	Thermal Capacity of Base Vapor compression chiller (air-source)
Capacity_DX_AS_W	Thermal Capacity of Direct-Expansion Unit Air-source
Capacity_VCCHT_AS_W	Thermal Capacity of High Temperature Vapor compression chiller (air-source)
Capaticy_ACH_SC_ET_W	Thermal Capacity of Absorption Chiller connected to Evacuated-Tube Solar Collector
Capex_a_USD	Annualized Capital Costs
Capex_total_USD	Total Capital Costs
GHG_tonCO2	Annual Green House Gas Emissions
Nominal heating load	Nominal heat load
Opex_fixed_USD	Fixed Annual Operational Costs
Opex_var_USD	Variable Annual Operational Costs
TAC_USD	Total Annualized Costs

5.3.27 get_optimization_decentralized_folder_building_result_cooling_activation

path: `outputs/data/optimization/decentralized/{building}_{configuration}_cooling_activation.csv`

The following file is used by these scripts: `optimization`

Variable	Description
E_ACH_req_W	Electricity requirements of Absorption Chillers
E_BaseVCC_AS_req_W	Electricity requirements of Vapor Compression Chillers and refrigeration
E_cs_cre_cdata_req_W	Electricity requirements due to space cooling, servers cooling and refrigeration
E_CT_req_W	Electricity requirements of Cooling Towers
E_DX_AS_req_W	Electricity requirements of Air-Source direct expansion chillers
E_SC_ET_req_W	Electricity requirements of Solar Collectors (evacuated-tubes)
E_SC_FP_req_W	Electricity requirements of Solar Collectors (flat-plate)
NG_Boiler_req	Requirements of Natural Gas for Boilers
NG_Burner_req	Requirements of Natural Gas for Burners
Q_ACH_gen_directload_W	Thermal energy generated by Absorption chillers
Q_BaseVCC_AS_gen_directload_W	Thermal energy generated by Air-Source Vapor-compression chillers
Q_Boiler_NG_ACH_W	Thermal energy generated by Natural gas Boilers to Absorption chillers
Q_Burner_NG_ACH_W	Thermal energy generated by Natural gas Burners to Absorption chillers
Q_DX_AS_gen_directload_W	Thermal energy generated by Air-Source direct expansion chillers
Q_SC_ET_ACH_W	Thermal energy generated by Solar Collectors (evacuated-tubes) to Absorption chillers
Q_SC_FP_ACH_W	Thermal energy generated by Solar Collectors (flat-plate) to Absorption chillers

5.3.28 get_optimization_decentralized_folder_building_result_heating

path: `outputs/data/optimization/decentralized/DiscOp_B001_result_heating.csv`

The following file is used by these scripts: `optimization`

Variable	Description
`Best configuration`	Index of best configuration simulated
`Capacity_BaseBoiler_NG_W`	Thermal capacity of Baseload Boiler NG
`Capacity_FC_NG_W`	Thermal Capacity of Fuel Cell NG
`Capacity_GS_HP_W`	Thermal Capacity of Ground Source Heat Pump
`Capex_a_USD`	Annualized Capital Costs
`Capex_total_USD`	Total Capital Costs
`GHG_tonCO2`	Annual Green House Gas Emissions
`Nominal heating load`	Nominal heat load
`Opex_fixed_USD`	Fixed Annual Operational Costs
`Opex_var_USD`	Variable Annual Operational Costs
`TAC_USD`	Total Annualized Costs

5.3.29 get_optimization_decentralized_folder_building_result_heating_activation

path: `outputs/data/optimization/decentralized/DiscOp_B001_result_heating_activation.csv`

The following file is used by these scripts: `optimization`

Variable	Description
`BackupBoiler_Status`	Status of the BackupBoiler (1=on, 0 =off)
`BG_Boiler_req_W`	Requirements of Bio-gas for Base load Boiler
`Boiler_Status`	Status of the Base load Boiler (1=on, 0 =off)
`E_Fuelcell_gen_export_W`	Electricity generation of fuel cell exported to the grid
`E_hs_ww_req_W`	Electricity Requirements for heat pump compressor and auxiliary uses (if required)
`Fuelcell_Status`	Status of the fuel cell (1=on, 0 =off)
`GHP_Status`	Status of the ground-source heat pump (1=on, 0 =off)
`NG_BackupBoiler_req_W`	Requirements of Natural Gas for Back-up Boiler
`NG_Boiler_req_W`	Requirements of Natural Gas for Base load Boiler
`NG_FuelCell_req_W`	Requirements of Natural Gas for fuel cell
`Q_BackupBoiler_gen_directload_W`	Thermal generation of Back-up Boiler to direct load
`Q_Boiler_gen_directload_W`	Thermal generation of Base load Boiler to direct load
`Q_Fuelcell_gen_directload_W`	Thermal generation of fuel cell to direct load
`Q_GHP_gen_directload_W`	Thermal generation of ground-source heat pump to direct load

5.3.30 get_optimization_district_scale_cooling_capacity

path: `outputs/data/optimization/slave/gen_1/ind_1_district_scale_cooling_capacity.csv`

The following file is used by these scripts:

Variable	Description
Capacity_ACH_SC_FP_cool_building_scale_W	Thermal Capacity of Absorption Chiller and Solar Collector (Flat Plate) for Decentralized Building
Capacity_ACHHT_FP_cool_building_scale_W	Thermal Capacity of High-Temperature Absorption Chiller and Solar Collector (Flat Plate) for Decentralized Building
Capacity_BaseVCC_AS_cool_building_scale_W	Thermal Capacity of Base load Vapor Compression Chiller for Decentralized Building
Capacity_DX_AS_cool_building_scale_W	Thermal Capacity of Direct Expansion Air-Source for Decentralized Building
Capacity_VCCHT_AS_cool_building_scale_W	Thermal Capacity of High-Temperature Vapor Compression Chiller (Air-Source) for Decentralized Building
Capaticy_ACH_SC_ET_cool_building_scale_W	Thermal Capacity of Absorption Chiller and Solar Collector (Evacuated Tube)for Decentralized Building
Name	Unique building ID. It must start with a letter

5.3.31 get_optimization_district_scale_electricity_capacity

path: outputs/data/optimization/slave/gen_2/ind_0_district_scale_electrical_capacity.csv

The following file is used by these scripts:

Variable	Description
Capacity_GRID_el_district_scale_W	Electrical Capacity Required from the local Grid
Capacity_PV_el_district_scale_m2	Area Coverage of PV in central Plant
Capacity_PV_el_district_scale_W	Electrical Capacity of PV in central Plant

5.3.32 get_optimization_district_scale_heating_capacity

path: outputs/data/optimization/slave/gen_0/ind_2_district_scale_heating_capacity.csv

The following file is used by these scripts:

Variable	Description
Capacity_BackupBoiler_NG_heat_district_scale_W	ThermalCapacity of Back-up Boiler - Natural Gas in central plant
Capacity_BaseBoiler_NG_heat_district_scale_W	Thermal Capacity of Base Load Boiler - Natural Gas in central plant
Capacity_CHP_DB_el_district_scale_W	Electrical Capacity of CHP Dry-Biomass in central plant
Capacity_CHP_DB_heat_district_scale_W	ThermalCapacity of CHP Dry-Biomass in central plant
Capacity_CHP_NG_el_district_scale_W	Electrical Capacity of CHP Natural-Gas in central plant
Capacity_CHP_NG_heat_district_scale_W	Thermal Capacity of CHP Natural-Gas in central plant
Capacity_CHP_WB_el_district_scale_W	Electrical Capacity of CHP Wet-Biomass in central plant
Capacity_CHP_WB_heat_district_scale_W	Thermal Capacity of CHP Wet-Biomass in central plant
Capacity_HP_DS_heat_district_scale_W	Thermal Capacity of Heat Pump Server-Source in central plant
Capacity_HP_GS_heat_district_scale_W	Thermal Capacity of Heat Pump Ground-Source in central plant
Capacity_HP_SS_heat_district_scale_W	Thermal Capacity of Heat Pump Sewage-Source in central plant
Capacity_HP_WS_heat_district_scale_W	Thermal Capacity of Heat Pump Water-Source in central plant
Capacity_PeakBoiler_NG_heat_district_scale_W	Thermal Capacity of Peak Boiler - Natural Gas in central plant
Capacity_PVT_el_district_scale_W	Electrical Capacity of PVT Field in central plant
Capacity_PVT_heat_district_scale_W	Thermal Capacity of PVT panels in central plant
Capacity_SC_ET_heat_district_scale_W	Thermal Capacity of Solar Collectors (Evacuated-tube) in central plant
Capacity_SC_FP_heat_district_scale_W	Thermal Capacity of Solar Collectors (Flat-plate) in central plant
Capacity_SeasonalStorage_WS_heat_district_scale_W	Thermal Capacity of Seasonal Thermal Storage in central plant

5.3.33 get_optimization_generation_building_scale_performance

path: `outputs/data/optimization/slave/gen_2/gen_2_building_scale_performance.csv`

The following file is used by these scripts:

Variable	Description
Capex_a_cooling_building_scale_USD	Annualized Capital Costs of building-scale systems due to cooling
Capex_a_heating_building_scale_USD	Annualized Capital Costs of building-scale systems due to heating
Capex_total_cooling_building_scale_USD	Total Capital Costs of building-scale systems due to cooling
Capex_total_heating_building_scale_USD	Total Capital Costs of building-scale systems due to heating
generation	No. Generation or Iteration (genetic Algorithm)
GHG_cooling_building_scale_tonCO2	Green House Gas Emissions of building-scale systems due to Cooling
GHG_heating_building_scale_tonCO2	Green House Gas Emissions of building-scale systems due to Heating
individual	No. Individual unique ID
individual_name	Name of Individual unique ID
Opex_fixed_cooling_building_scale_USD	Fixed Operational Costs of building-scale systems due to cooling
Opex_fixed_heating_building_scale_USD	Fixed Operational Costs of building-scale systems due to heating
Opex_var_cooling_building_scale_USD	Variable Operational Costs of building-scale systems due to cooling
Opex_var_heating_building_scale_USD	Variable Operational Costs of building-scale systems due to heating

5.3.34 get_optimization_generation_district_scale_performance

path: `outputs/data/optimization/slave/gen_1/gen_1_district_scale_performance.csv`

The following file is used by these scripts:

Variable	Description
Capex_a_BackupBoiler_NG_district_scale_USD	Annualized Capital Costs of Back-up Boiler Natural Gas
Capex_a_BaseBoiler_NG_district_scale_USD	Annualized Capital Costs of Base Load Boiler Boiler Nat
Capex_a_CHP_NG_district_scale_USD	Annualized Capital Costs of CHP Natural Gas in Central
Capex_a_DHN_district_scale_USD	Annualized Capital Costs of District Heating Network
Capex_a_Furnace_dry_district_scale_USD	Annualized Capital Costs of CHP Dry-Biomass in Centra
Capex_a_Furnace_wet_district_scale_USD	Annualized Capital Costs of CHP Wet-Biomass in Centra
Capex_a_GHP_district_scale_USD	Annualized Capital Costs of Ground-Source Heat Pump i
Capex_a_GRID_district_scale_USD	Annualized Capital Costs of connection to local electrical
Capex_a_HP_Lake_district_scale_USD	Annualized Capital Costs of Lake-Source Heat Pump in C
Capex_a_HP_Server_district_scale_USD	Annualized Capital Costs of Server-Source Heat Pump in
Capex_a_HP_Sewage_district_scale_USD	Annualized Capital Costs of Sewage-Source Heat Pump i
Capex_a_PeakBoiler_NG_district_scale_USD	Annualized Capital Costs of Peak Load Boiler Boiler Nat
Capex_a_PV_district_scale_USD	Annualized Capital Costs of Photovoltaic Panels in Centr
Capex_a_PVT_district_scale_USD	Annualized Capital Costs of PVT Panels in Central Plant
Capex_a_SC_ET_district_scale_USD	Annualized Capital Costs of Solar Collectors (evacuated-
Capex_a_SC_FP_district_scale_USD	Annualized Capital Costs of Solar Collectors (Flat-Plate)
Capex_a_SeasonalStorage_WS_district_scale_USD	Annualized Capital Costs of Seasonal Thermal Storage in
Capex_a_SubstationsHeating_district_scale_USD	Annualized Capital Costs of Thermal Substations
Capex_total_BackupBoiler_NG_district_scale_USD	Total Capital Costs of Back-up Boiler Natural Gas in Cen
Capex_total_BaseBoiler_NG_district_scale_USD	Total Capital Costs of Base Load Boiler Boiler Natural G

C

Table 63 – continued from previous page

Variable	Description
Capex_total_CHP_NG_district_scale_USD	Total Capital Costs of CHP Natural Gas in Central Plant
Capex_total_DHN_district_scale_USD	Total Capital Costs of District Heating Network
Capex_total_Furnace_dry_district_scale_USD	Total Capital Costs of CHP Dry-Biomass in Central Plant
Capex_total_Furnace_wet_district_scale_USD	Total Capital Costs of CHP Wet-Biomass in Central Plant
Capex_total_GHP_district_scale_USD	Total Capital Costs of Ground-Source Heat Pump in Cent
Capex_total_GRID_district_scale_USD	Total Capital Costs of connection to local electrical grid
Capex_total_HP_Lake_district_scale_USD	Total Capital Costs of Lake-Source Heat Pump in Central
Capex_total_HP_Server_district_scale_USD	Total Capital Costs of Server-Source Heat Pump in Centr
Capex_total_HP_Sewage_district_scale_USD	Total Capital Costs of Sewage-Source Heat Pump in Cent
Capex_total_PeakBoiler_NG_district_scale_USD	Total Capital Costs of Peak Load Boiler Boiler Natural G
Capex_total_PV_district_scale_USD	Total Capital Costs of Photovoltaic Panels in Central Plan
Capex_total_PVT_district_scale_USD	Total Capital Costs of PVT Panels in Central Plant
Capex_total_SC_ET_district_scale_USD	Total Capital Costs of Solar Collectors (evacuated-Tube)
Capex_total_SC_FP_district_scale_USD	Total Capital Costs of Solar Collectors (Flat-Plate) in Cen
Capex_total_SeasonalStorage_WS_district_scale_USD	Total Capital Costs of Seasonal Thermal Storage in Centr
Capex_total_SubstationsHeating_district_scale_USD	Total Capital Costs of Thermal Substations
generation	Number of the Generation or Iteration (Genetic algorithm
GHG_DB_district_scale_tonCO2yr	Green House Gas Emissions of Dry-Biomass of district-s
GHG_GRID_exports_district_scale_tonCO2yr	Green House Gas Emissions of Exports of Electricity
GHG_GRID_imports_district_scale_tonCO2yr	Green House Gas Emissions of Import of Electricity
GHG_NG_district_scale_tonCO2yr	Green House Gas Emissions of Natural Gas of district-sca
GHG_WB_district_scale_tonCO2yr	Green House Gas Emissions of Wet-Biomass of district-s
individual	Unique numerical ID of individual
individual_name	Unique alphanumerical ID of individual
Opex_fixed_BackupBoiler_NG_district_scale_USD	Fixed Operation Costs of Back-up Boiler Natural Gas in C
Opex_fixed_BaseBoiler_NG_district_scale_USD	Fixed Operation Costs of Base Load Boiler Boiler Natura
Opex_fixed_CHP_NG_district_scale_USD	Fixed Operation Costs of CHP Natural Gas in Central Pla
Opex_fixed_DHN_district_scale_USD	Fixed Operation Costs of District Heating Network
Opex_fixed_Furnace_dry_district_scale_USD	Fixed Operation Costs of CHP Dry-Biomass in Central P
Opex_fixed_Furnace_wet_district_scale_USD	Fixed Operation Costs of CHP Wet-Biomass in Central P
Opex_fixed_GHP_district_scale_USD	Fixed Operation Costs of Ground-Source Heat Pump in C
Opex_fixed_GRID_district_scale_USD	Fixed Operation Costs of Electricity in Buildings Connec
Opex_fixed_HP_Lake_district_scale_USD	Fixed Operation Costs of Lake-Source Heat Pump in Cen
Opex_fixed_HP_Server_district_scale_USD	Fixed Operation Costs of Server-Source Heat Pump in Ce
Opex_fixed_HP_Sewage_district_scale_USD	Fixed Operation Costs of Sewage-Source Heat Pump in C
Opex_fixed_PeakBoiler_NG_district_scale_USD	Fixed Operation Costs of Peak Load Boiler Boiler Natura
Opex_fixed_PV_district_scale_USD	Fixed Operation Costs of Photovoltaic Panels in Central P
Opex_fixed_PVT_district_scale_USD	Fixed Operation Costs of PVT Panels in Central Plant
Opex_fixed_SC_ET_district_scale_USD	Fixed Operation Costs of Solar Collectors (evacuated-Tub
Opex_fixed_SC_FP_district_scale_USD	Fixed Operation Costs of Solar Collectors (Flat-Plate) in C
Opex_fixed_SeasonalStorage_WS_district_scale_USD	Fixed Operation Costs of Seasonal Thermal Storage in Ce
Opex_fixed_SubstationsHeating_district_scale_USD	Fixed Operation Costs of Thermal Substations
Opex_var_DB_district_scale_USD	Variable Operation Costs due to consumption of Dry-Bio
Opex_var_GRID_exports_district_scale_USD	Variable Operation Costs due to electricity exported
Opex_var_GRID_imports_district_scale_USD	Variable Operation Costs due to electricity imported
Opex_var_NG_district_scale_USD	Variable Operation Costs due to consumption of Natural C
Opex_var_WB_district_scale_USD	Variable Operation Costs due to consumption of Wet-Bio

5.3.35 get_optimization_generation_total_performance

path: `outputs/data/optimization/slave/gen_2/gen_2_total_performance.csv`

The following file is used by these scripts:

Variable	Description
`Capex_a_sys_building_scale_USD`	Annualized Capital Costs of building-scale systems
`Capex_a_sys_district_scale_USD`	Annualized Capital Costs of district-scale systems
`Capex_a_sys_USD`	Annualized Capital Costs of all systems
`Capex_total_sys_building_scale_USD`	Total Capital Costs of building-scale systems
`Capex_total_sys_district_scale_USD`	Total Capital Costs of district-scale systems
`Capex_total_sys_USD`	Total Capital Costs of district-scale systems and Decentralized Buildings
`generation`	No. Generation or Iteration (genetic Algorithm)
`GHG_sys_building_scale_tonCO2`	Green House Gas Emissions of building-scale systems
`GHG_sys_district_scale_tonCO2`	Green House Gas Emissions Central Plant
`GHG_sys_tonCO2`	Green House Gas Emissions of all systems
`individual`	No. Individual unique ID
`individual_name`	Name of Individual unique ID
`Opex_a_sys_building_scale_USD`	Operation Costs of building-scale systems
`Opex_a_sys_district_scale_USD`	Operation Costs of district-scale systems
`Opex_a_sys_USD`	Operation Costs of all systems
`TAC_sys_building_scale_USD`	Total Anualized Costs of building-scale systems
`TAC_sys_district_scale_USD`	Total Anualized Costs of district-scale systems
`TAC_sys_USD`	Total Anualized Costs of all systems

5.3.36 get_optimization_generation_total_performance_pareto

path: `outputs/data/optimization/slave/gen_2/gen_2_total_performance_pareto.csv`

The following file is used by these scripts: `multi_criteria_analysis`

Variable	Description
Capex_a_sys_building_scale_USD	Annualized Capital Costs of building-scale systems
Capex_a_sys_district_scale_USD	Annualized Capital Costs of district-scale systems
Capex_a_sys_USD	Annualized Capital Costs of all systems
Capex_total_sys_building_scale_USD	Total Capital Costs of building-scale systems
Capex_total_sys_district_scale_USD	Total Capital Costs of district-scale systems
Capex_total_sys_USD	Total Capital Costs of district-scale systems and Decentralized Buildings
generation	No. Generation or Iteration (genetic Algorithm)
GHG_sys_building_scale_tonCO2	Green House Gas Emissions of building-scale systems
GHG_sys_district_scale_tonCO2	Green House Gas Emissions Central Plant
GHG_sys_tonCO2	Green House Gas Emissions of all systems
individual	No. Individual unique ID
individual_name	Name of Individual unique ID
Opex_a_sys_building_scale_USD	Operation Costs of building-scale systems
Opex_a_sys_district_scale_USD	Operation Costs of district-scale systems
Opex_a_sys_USD	Operation Costs of all systems
TAC_sys_building_scale_USD	Total Anualized Costs of building-scale systems
TAC_sys_district_scale_USD	Total Anualized Costs of district-scale systems
TAC_sys_USD	Total Anualized Costs of all systems

5.3.37 get_optimization_individuals_in_generation

path: `outputs/data/optimization/slave/gen_2/generation_2_individuals.csv`

The following file is used by these scripts:

Variable	Description
B01_DH	TODO
B02_DH	TODO
B03_DH	TODO
B04_DH	TODO
B05_DH	TODO
B06_DH	TODO
B07_DH	TODO
B08_DH	TODO
B09_DH	TODO
DB_Cogen	TODO
DS_HP	TODO
generation	TODO
GS_HP	TODO
individual	TODO
NG_BaseBoiler	TODO
NG_Cogen	TODO
NG_PeakBoiler	TODO
PV	TODO
PVT	TODO
SC_ET	TODO
SC_FP	TODO
SS_HP	TODO
WB_Cogen	TODO
WS_HP	TODO

5.3.38 get_optimization_network_results_summary

path: `outputs/data/optimization/network/DH_Network_summary_result_0x1be.csv`

The following file is used by these scripts: `optimization`

Variable	Description
DATE	Time stamp (hourly) for one year
mcpdata_netw_total_kWperC	Capacity mass flow reate for server cooling of this network
mdot_DH_netw_total_kgpers	Total mass flow rate in this district heating network
Q_DH_losses_W	Thermal losses of this district heating network
Q_DHNf_W	Total thermal demand of district heating network
Qcdata_netw_total_kWh	Thermal Demand for server cooling in this network
T_DHNf_re_K	Average Temperature of return of this district heating network
T_DHNf_sup_K	Average Temperature of supply of this district heating network

5.3.39 get_optimization_slave_building_connectivity

path: `outputs/data/optimization/slave/gen_2/ind_1_building_connectivity.csv`

The following file is used by these scripts:

Variable	Description
DC_connectivity	Flag to know if building is connected to District Heating or not
DH_connectivity	Flag to know if building is connected to District Cooling or not
Name	Unique building ID. It must start with a letter.)

5.3.40 get_optimization_slave_building_scale_performance

path: outputs/data/optimization/slave/gen_2/ind_0_buildings_building_scale_performance.csv

The following file is used by these scripts:

Variable	Description
Capex_a_cooling_building_scale_USD	Annualized Capital Costs of building-scale systems due to cooling
Capex_a_heating_building_scale_USD	Annualized Capital Costs of building-scale systems due to heating
Capex_total_cooling_building_scale_USD	Total Capital Costs of building-scale systems due to cooling
Capex_total_heating_building_scale_USD	Total Capital Costs of building-scale systems due to heating
GHG_cooling_building_scale_tonCO2	Green House Gas Emissions of building-scale systems due to Cooling
GHG_heating_building_scale_tonCO2	Green House Gas Emissions of building-scale systems due to Heating
Opex_fixed_cooling_building_scale_USD	Fixed Operational Costs of building-scale systems due to cooling
Opex_fixed_heating_building_scale_USD	Fixed Operational Costs of building-scale systems due to heating
Opex_var_cooling_building_scale_USD	Variable Operational Costs of building-scale systems due to cooling
Opex_var_heating_building_scale_USD	Variable Operational Costs of building-scale systems due to heating

5.3.41 get_optimization_slave_cooling_activation_pattern

path: outputs/data/optimization/slave/gen_1/ind_2_Cooling_Activation_Pattern.csv

The following file is used by these scripts: optimization

Variable	Description
Capacity_DailyStorage_WS_cool_district_scale_W	Installed capacity of the short-term thermal storage
Capex_a_DailyStorage_WS_cool_district_scale_USD	Annualized capital costs of the short-term thermal storage
Capex_total_DailyStorage_WS_cool_district_scale_USD	Total capital costs of the short-term thermal storage
Opex_fixed_DailyStorage_WS_cool_district_scale_USD	Fixed operational costs of the short-term thermal storage
Q_DailyStorage_content_W	Thermal energy content of the short-term thermal storage
Q_DailyStorage_gen_directload_W	Thermal energy supplied from the short-term thermal storage

5.3.42 get_optimization_slave_district_scale_performance

path: `outputs/data/optimization/slave/gen_1/ind_2_buildings_district_scale_performance.csv`

The following file is used by these scripts:

Variable	Description
`Capex_a_BackupBoiler_NG_district_scale_USD`	Annualized Capital Costs of Back-up Boiler Natural Gas
`Capex_a_BaseBoiler_NG_district_scale_USD`	Annualized Capital Costs of Base load Boiler Natural Ga
`Capex_a_CHP_NG_district_scale_USD`	Annualized Capital Costs of CHP Natural Gas in Central
`Capex_a_DHN_district_scale_USD`	Annualized Capital Costs of District Heating Network
`Capex_a_Furnace_dry_district_scale_USD`	Annualized Capital Costs of CHP Dry-Biomass in Centra
`Capex_a_Furnace_wet_district_scale_USD`	Annualized Capital Costs of CHP Wet-Biomass in Centra
`Capex_a_GHP_district_scale_USD`	Annualized Capital Costs of Ground-Source Heat-Pump i
`Capex_a_GRID_district_scale_USD`	Annualized Capital Costs of connection to local grid
`Capex_a_HP_Lake_district_scale_USD`	Annualized Capital Costs of Lake-Source Heat Pump in C
`Capex_a_HP_Server_district_scale_USD`	Annualized Capital Costs of Server-Source Heat Pump in
`Capex_a_HP_Sewage_district_scale_USD`	Annualized Capital Costs of Sewage-Source Heat Pump i
`Capex_a_PeakBoiler_NG_district_scale_USD`	Annualized Capital Costs of Peak Boiler in Central Plant
`Capex_a_PV_district_scale_USD`	Annualized Capital Costs of PV panels
`Capex_a_PVT_district_scale_USD`	Annualized Capital Costs of PVT panels
`Capex_a_SC_ET_district_scale_USD`	Annualized Capital Costs of Solar collectors (evacuated T
`Capex_a_SC_FP_district_scale_USD`	Annualized Capital Costs of Solar collectors (Flat-Plate)
`Capex_a_SeasonalStorage_WS_district_scale_USD`	Annualized Capital Costs of Seasonal Thermal Storage in
`Capex_a_SubstationsHeating_district_scale_USD`	Annualized Capital Costs of Heating Substations
`Capex_total_BackupBoiler_NG_district_scale_USD`	Total Capital Costs of Back-up Boiler Natural Gas in Cen
`Capex_total_BaseBoiler_NG_district_scale_USD`	Total Capital Costs of Base load Boiler Natural Gas in Ce
`Capex_total_CHP_NG_district_scale_USD`	Total Capital Costs of CHP Natural Gas in Central Plant
`Capex_total_DHN_district_scale_USD`	Total Capital Costs of District Heating Network
`Capex_total_Furnace_dry_district_scale_USD`	Total Capital Costs of CHP Dry-Biomass in Central Plant
`Capex_total_Furnace_wet_district_scale_USD`	Total Capital Costs of CHP Wet-Biomass in Central Plant
`Capex_total_GHP_district_scale_USD`	Total Capital Costs of Ground-Source Heat-Pump in Cen
`Capex_total_GRID_district_scale_USD`	Total Capital Costs of connection to local grid
`Capex_total_HP_Lake_district_scale_USD`	Total Capital Costs of Lake-Source Heat Pump in Central
`Capex_total_HP_Server_district_scale_USD`	Total Capital Costs of Server-Source Heat Pump in Centr
`Capex_total_HP_Sewage_district_scale_USD`	Total Capital Costs of Sewage-Source Heat Pump in Cent
`Capex_total_PeakBoiler_NG_district_scale_USD`	Total Capital Costs of Peak Boiler in Central Plant
`Capex_total_PV_district_scale_USD`	Total Capital Costs of PV panels
`Capex_total_PVT_district_scale_USD`	Total Capital Costs of PVT panels
`Capex_total_SC_ET_district_scale_USD`	Total Capital Costs of Solar collectors (evacuated Tubes)
`Capex_total_SC_FP_district_scale_USD`	Total Capital Costs of Solar collectors (Flat-Plate)
`Capex_total_SeasonalStorage_WS_district_scale_USD`	Total Capital Costs of Seasonal Thermal Storage
`Capex_total_SubstationsHeating_district_scale_USD`	Total Capital Costs of Heating Substations
`GHG_DB_district_scale_tonCO2yr`	Green House Gas Emissions of Dry-Biomass in Central p
`GHG_GRID_exports_district_scale_tonCO2yr`	Green House Gas Emissions of Electricity Exports in Cen
`GHG_GRID_imports_district_scale_tonCO2yr`	Green House Gas Emissions of Electricity Import in Cent
`GHG_NG_district_scale_tonCO2yr`	Green House Gas Emissions of Natural Gas in Central pla
`GHG_WB_district_scale_tonCO2yr`	Green House Gas Emissions of Wet-Biomass in Central p
`Opex_fixed_BackupBoiler_NG_district_scale_USD`	Fixed Operation Costs of Back-up Boiler Natural Gas in C
`Opex_fixed_BaseBoiler_NG_district_scale_USD`	Fixed Operation Costs of Base load Boiler Natural Gas in
`Opex_fixed_CHP_NG_district_scale_USD`	Fixed Operation Costs of CHP Natural Gas in Central Pla

Continue

Table 64 – continued from previous page

Variable	Description
Opex_fixed_DHN_district_scale_USD	Fixed Operation Costs of District Heating Network
Opex_fixed_Furnace_dry_district_scale_USD	Fixed Operation Costs of CHP Dry-Biomass in Central P
Opex_fixed_Furnace_wet_district_scale_USD	Fixed Operation Costs of CHP Wet-Biomass in Central P
Opex_fixed_GHP_district_scale_USD	Fixed Operation Costs of Ground-Source Heat-Pump in C
Opex_fixed_GRID_district_scale_USD	Fixed Operation Costs of connection to local grid
Opex_fixed_HP_Lake_district_scale_USD	Fixed Operation Costs of Lake-Source Heat Pump in Cen
Opex_fixed_HP_Server_district_scale_USD	Fixed Operation Costs of Server-Source Heat Pump in Ce
Opex_fixed_HP_Sewage_district_scale_USD	Fixed Operation Costs of Sewage-Source Heat Pump in C
Opex_fixed_PeakBoiler_NG_district_scale_USD	Fixed Operation Costs of Peak Boiler in Central Plant
Opex_fixed_PV_district_scale_USD	Fixed Operation Costs of PV panels
Opex_fixed_PVT_district_scale_USD	Fixed Operation Costs of PVT panels
Opex_fixed_SC_ET_district_scale_USD	Fixed Operation Costs of Solar collectors (evacuated Tub
Opex_fixed_SC_FP_district_scale_USD	Fixed Operation Costs of Solar collectors (Flat-Plate)
Opex_fixed_SeasonalStorage_WS_district_scale_USD	Fixed Operation Costs of Seasonal Thermal Storage
Opex_fixed_SubstationsHeating_district_scale_USD	Fixed Operation Costs of Heating Substations
Opex_var_DB_district_scale_USD	Variable Operation Costs
Opex_var_GRID_exports_district_scale_USD	Variable Operation Costs
Opex_var_GRID_imports_district_scale_USD	Variable Operation Costs
Opex_var_NG_district_scale_USD	Variable Operation Costs
Opex_var_WB_district_scale_USD	Variable Operation Costs

5.3.43 get_optimization_slave_electricity_activation_pattern

path: `outputs/data/optimization/slave/gen_1/ind_1_Electricity_Activation_Pattern.csv`

The following file is used by these scripts:

Variable	Description
DATE	Time stamp (hourly) for one year
E_CHP_gen_directload_W	Electricity Generated to direct load by CHP Natural Gas
E_CHP_gen_export_W	Electricity Exported by CHP Natural Gas
E_Furnace_dry_gen_directload_W	Electricity Generated to direct load by CHP Dry Biomass
E_Furnace_dry_gen_export_W	Electricity Exported by CHP Dry Biomass
E_Furnace_wet_gen_directload_W	Electricity Generated to direct load by CHP Wet Biomass
E_Furnace_wet_gen_export_W	Electricity Exported by CHP Wet Biomass
E_GRID_directload_W	Electricity Imported from the local grid
E_PV_gen_directload_W	Electricity Generated to direct load by PV panels
E_PV_gen_export_W	Electricity Exported by PV panels
E_PVT_gen_directload_W	Electricity Generated to direct load by PVT panels
E_PVT_gen_export_W	Electricity Exported by PVT panels
E_Trigen_gen_directload_W	Electricity Generated to direct load by Trigen CHP Natural Gas
E_Trigen_gen_export_W	Electricity Exported by Trigen CHP Natural Gas

5.3.44 get_optimization_slave_electricity_requirements_data

path: `outputs/data/optimization/slave/gen_1/ind_1_Electricity_Requirements_Pattern.csv`

The following file is used by these scripts:

Variable	Description
DATE	Time stamp (hourly) for one year
E_BackupBoiler_req_W	Electricity (auxiliary) Required by Back-up Boiler
E_BackupVCC_AS_req_W	Electricity Required by Back-up Vapor Compression Chiller (Air-Source)
E_BaseBoiler_req_W	Electricity (auxiliary) Required by Base Load Boiler
E_BaseVCC_AS_req_W	Electricity Required by Base Load Vapor Compression Chiller (Air-Source)
E_BaseVCC_WS_req_W	Electricity Required by Base Load Vapor Compression Chiller (Water-Source)
E_cs_cre_cdata_req_building_scale_W	Electricity Required for space cooling, server cooling and refrigeration of building-scale systems
E_cs_cre_cdata_req_district_scale_W	Electricity Required for space cooling, server cooling and refrigeration of Buildings Connected to Network
E_DCN_req_W	Electricity Required for Chilled water Pumping in District Cooling Network
E_DHN_req_W	Electricity Required for Chilled water Pumping in District Heating Network
E_electricalnetwork_sys_req_W	Total Electricity Requirements
E_GHP_req_W	Electricity Required by Ground-Source Heat Pumps
E_HP_Lake_req_W	Electricity Required by Lake-Source Heat Pumps
E_HP_PVT_req_W	Electricity Required by Auxiliary Heat Pumps of PVT panels
E_HP_SC_ET_req_W	Electricity Required by Auxiliary Heat Pumps of Solar collectors (Evacuated tubes)
E_HP_SC_FP_req_W	Electricity Required by Auxiliary Heat Pumps of Solar collectors (Evacuated Flat Plate)
E_HP_Server_req_W	Electricity Required by Server-Source Heat Pumps
E_HP_Sew_req_W	Electricity Required by Sewage-Source Heat Pumps
E_hs_ww_req_building_scale_W	Electricity Required for space heating and hotwater of building-scale systems
E_hs_ww_req_district_scale_W	Electricity Required for space heating and hotwater of Buildings Connected to Network
E_PeakBoiler_req_W	Electricity (auxiliary) Required by Peak-Boiler
E_PeakVCC_AS_req_W	Electricity Required by Peak Vapor Compression Chiller (Air-Source)
E_PeakVCC_WS_req_W	Electricity Required by Peak Vapor Compression Chiller (Water-Source)
E_Storage_charging_req_W	Electricity Required by Auxiliary Heatpumps for charging Seasonal Thermal Storage
E_Storage_discharging_req_W	Electricity Required by Auxiliary Heatpumps for discharging Seasonal Thermal Storage
Eal_req_W	Electricity Required for Appliances and Lighting in all Buildings
Eaux_req_W	Electricity Required for Fans and others in all Buildings
Edata_req_W	Electricity Required for Servers in all Buildings
Epro_req_W	Electricity Required for Industrial Processes in all Buildings

5.3.45 get_optimization_slave_heating_activation_pattern

path: `outputs/data/optimization/slave/gen_2/ind_0_Heating_Activation_Pattern.csv`

The following file is used by these scripts:

Variable	Description
DATE	Time stamp (hourly) for one year
E_CHP_gen_W	Electricity Generation by CHP Natural Gas
E_Furnace_dry_gen_W	Electricity Generation by CHP Dry-Biomass
E_Furnace_wet_gen_W	Electricity Generation by CHP Wet-Biomass
E_PVT_gen_W	Electricity Generation by PVT
Q_BackupBoiler_gen_directload_W	Thermal generation of Back-up Boiler to direct load
Q_BaseBoiler_gen_directload_W	Thermal generation of Base load Boiler to direct load
Q_CHP_gen_directload_W	Thermal generation of CHP Natural Gas to direct load
Q_districtheating_sys_req_W	Thermal requirements of District Heating Network
Q_Furnace_dry_gen_directload_W	Thermal generation of CHP Dry-Biomass to direct load
Q_Furnace_wet_gen_directload_W	Thermal generation of CHP Wet-Biomass to direct load
Q_GHP_gen_directload_W	Thermal generation of ground-source heat pump to direct load
Q_HP_Lake_gen_directload_W	Thermal generation of Lake-Source Heatpump to direct load
Q_HP_Server_gen_directload_W	Thermal generation of Server-Source Heatpump to direct load
Q_HP_Server_storage_W	Thermal generation of Server-Source Heatpump to Seasonal Thermal Storage
Q_HP_Sew_gen_directload_W	Thermal generation of Sewage-Source Heatpump to direct load
Q_PeakBoiler_gen_directload_W	Thermal generation of Peak Boiler to direct load
Q_PVT_gen_directload_W	Thermal generation of PVT to direct load
Q_PVT_gen_storage_W	Thermal generation of PVT to Seasonal Thermal Storage
Q_SC_ET_gen_directload_W	Thermal generation of Solar Collectors (Evacuated Tubes) to direct load
Q_SC_ET_gen_storage_W	Thermal generation of Solar Collectors (Evacuated Tubes) to Seasonal Thermal Storage
Q_SC_FP_gen_directload_W	Thermal generation of Solar Collectors (Flat Plate) to direct load
Q_SC_FP_gen_storage_W	Thermal generation of Solar Collectors (Flat Plate) to Seasonal Thermal Storage
Q_Storage_gen_directload_W	Discharge from Storage to Direct Load

5.3.46 get_optimization_slave_total_performance

path: `outputs/data/optimization/slave/gen_0/ind_2_total_performance.csv`

The following file is used by these scripts:

Variable	Description
Capex_a_sys_building_scale_USD	Annualized Capital Costs of building-scale systems
Capex_a_sys_district_scale_USD	Annualized Capital Costs of district-scale systems
Capex_a_sys_USD	Annualized Capital Costs of all systems
Capex_total_sys_building_scale_USD	Total Capital Costs of building-scale systems
Capex_total_sys_district_scale_USD	Total Capital Costs of district-scale systems
Capex_total_sys_USD	Total Capital Costs of all systems
GHG_sys_building_scale_tonCO2	Green House Gas Emissions of building-scale systems
GHG_sys_district_scale_tonCO2	Green House Gas Emissions Central Plant
GHG_sys_tonCO2	Green House Gas Emissions of all systems
Opex_a_sys_building_scale_USD	Operation Costs of building-scale systems
Opex_a_sys_district_scale_USD	Operation Costs of district-scale systems
Opex_a_sys_USD	Operation Costs of all systems
TAC_sys_building_scale_USD	Total Anualized Costs of building-scale systems
TAC_sys_district_scale_USD	Total Anualized Costs of district-scale systems
TAC_sys_USD	Total Anualized Costs of all systems

5.3.47 get_optimization_substations_results_file

path: outputs/data/optimization/substations/110011011DH_B001_result.csv

The following file is used by these scripts: optimization

Variable	Description
A_hex_dhw_design_m2	Substation heat exchanger area to supply domestic hot water
A_hex_heating_design_m2	Substation heat exchanger area to supply space heating
mdot_DH_result_kgpers	Substation flow rate on the DH side.
Q_dhw_W	Substation heat requirement to supply domestic hot water
Q_heating_W	Substation heat requirement to supply space heating
T_return_DH_result_K	Substation return temperature of the district heating network
T_supply_DH_result_K	Substation supply temperature of the district heating network.

5.3.48 get_optimization_substations_total_file

path: outputs/data/optimization/substations/Total_DH_111111111.csv

The following file is used by these scripts:

Variable	Description
Af_m2	Conditioned floor area (heated/cooled)
Aocc_m2	Occupied floor area (heated/cooled)
Aroof_m2	Roof area
COAL_hs0_kW	Nominal Coal requirement for space heating supply
COAL_hs_MWhyr	Coal requirement for space heating supply
COAL_ww0_kW	Nominal Coal requirement for hotwater supply
COAL_ww_MWhyr	Coal requirement for hotwater supply
DC_cdata0_kW	Nominal district cooling for final data center cooling demand
DC_cdata_MWhyr	District cooling for data center cooling demand
DC_cre0_kW	Nominal district cooling for refrigeration demand
DC_cre_MWhyr	District cooling for refrigeration demand
DC_cs0_kW	Nominal district cooling for space cooling demand
DC_cs_MWhyr	Energy consumption of space cooling system (if supplied by District Cooling), DC_cs = Qcs_sys / eff_cs
DH_hs0_kW	Nominal energy requirement by district heating (space heating supply)
DH_hs_MWhyr	Energy requirement by district heating (space heating supply)
DH_ww0_kW	Nominal Energy requirement by district heating (hotwater supply)
DH_ww_MWhyr	Energy requirement by district heating (hotwater supply)
E_cdata0_kW	Nominal Data centre cooling specific electricity consumption.
E_cdata_MWhyr	Electricity consumption due to data center cooling
E_cre0_kW	Nominal Refrigeration system electricity consumption.
E_cre_MWhyr	Electricity consumption due to refrigeration
E_cs0_kW	Nominal Cooling system electricity consumption.

Continued on next page

Table 65 – continued from previous page

Variable	Description
E_cs_MWhyr	Energy consumption of cooling system (if supplied by electricity grid), E_cs = Qcs_sys / eff_cs
E_hs0_kW	Nominal Heating system electricity consumption.
E_hs_MWhyr	Electricity consumption due to space heating
E_sys0_kW	Nominal end-use electricity demand
E_sys_MWhyr	End-use total electricity consumption, E_sys = Eve + Ea + El + Edata + Epro + Eaux + Ev
E_ww0_kW	Nominal Domestic hot water electricity consumption.
E_ww_MWhyr	Electricity consumption due to hot water
Ea0_kW	Nominal end-use electricity for appliances
Ea_MWhyr	End-use electricity for appliances
Eal0_kW	Nominal Total net electricity for all sources and sinks.
Eal_MWhyr	End-use electricity consumption of appliances and lighting, Eal = El_W + Ea_W
Eaux0_kW	Nominal Auxiliary electricity consumption.
Eaux_MWhyr	End-use auxiliary electricity consumption, Eaux = Eaux_fw + Eaux_ww + Eaux_cs + Eaux_hs + Ehs_lat_aux
Edata0_kW	Nominal Data centre electricity consumption.
Edata_MWhyr	Electricity consumption for data centers
El0_kW	Nominal end-use electricity for lights
El_MWhyr	End-use electricity for lights
Epro0_kW	Nominal Industrial processes electricity consumption.
Epro_MWhyr	Electricity supplied to industrial processes
Ev0_kW	Nominal end-use electricity for electric vehicles
Ev_MWhyr	End-use electricity for electric vehicles
Eve0_kW	Nominal end-use electricity for ventilation
Eve_MWhyr	End-use electricity for ventilation
GFA_m2	Gross floor area
GRID0_kW	Nominal Grid electricity consumption
GRID_a0_kW	Nominal grid electricity requirements for appliances
GRID_a_MWhyr	Grid electricity requirements for appliances
GRID_aux0_kW	Nominal grid electricity requirements for auxiliary loads
GRID_aux_MWhyr	Grid electricity requirements for auxiliary loads
GRID_cdata0_kW	Nominal grid electricity requirements for servers cooling
GRID_cdata_MWhyr	Grid electricity requirements for servers cooling
GRID_cre0_kW	Nominal grid electricity requirements for refrigeration
GRID_cre_MWhyr	Grid electricity requirements for refrigeration
GRID_cs0_kW	Nominal grid electricity requirements for space cooling
GRID_cs_MWhyr	Grid electricity requirements for space cooling
GRID_data0_kW	Nominal grid electricity requirements for servers
GRID_data_MWhyr	Grid electricity requirements for servers
GRID_hs0_kW	Nominal grid electricity requirements for space heating
GRID_hs_MWhyr	Grid electricity requirements for space heating
GRID_l0_kW	Nominal grid electricity consumption for lights
GRID_l_MWhyr	Grid electricity requirements for lights

Continued on next page

Table 65 – continued from previous page

Variable	Description
GRID_MWhyr	Grid electricity consumption, GRID = GRID_a + GRID_l + GRID_v + GRID_ve + GRID_data + GRID_pro + GRID_aux + GRID_ww + GRID_cs + GRID_hs + GRID_cdata + GRID_cre
GRID_pro0_kW	Nominal grid electricity requirements for industrial processes
GRID_pro_MWhyr	Grid electricity requirements for industrial processes
GRID_v0_kW	Nominal grid electricity consumption for electric vehicles
GRID_v_MWhyr	Grid electricity requirements for electric vehicles
GRID_ve0_kW	Nominal grid electricity consumption for ventilatioon
GRID_ve_MWhyr	Grid electricity requirements for ventilatioon
GRID_ww0_kW	Nominal grid electricity requirements for hot water supply
GRID_ww_MWhyr	Grid electricity requirements for hot water supply
Name	Unique building ID. It must start with a letter.
NG_hs0_kW	Nominal NG requirement for space heating supply
NG_hs_MWhyr	NG requirement for space heating supply
NG_ww0_kW	Nominal NG requirement for hotwater supply
NG_ww_MWhyr	NG requirement for hotwater supply
OIL_hs0_kW	Nominal OIL requirement for space heating supply
OIL_hs_MWhyr	OIL requirement for space heating supply
OIL_ww0_kW	Nominal OIL requirement for hotwater supply
OIL_ww_MWhyr	OIL requirement for hotwater supply
people0	Nominal occupancy
PV0_kW	Nominal PV electricity consumption
PV_MWhyr	PV electricity consumption
QC_sys0_kW	Nominal Total system cooling demand.
QC_sys_MWhyr	Total system cooling demand, QC_sys = Qcs_sys + Qcdata_sys + Qcre_sys + Qcpro_sys
Qcdata0_kW	Nominal Data centre cooling demand.
Qcdata_MWhyr	Data centre cooling demand
Qcdata_sys0_kW	Nominal end-use data center cooling demand
Qcdata_sys_MWhyr	End-use data center cooling demand
Qcpro_sys0_kW	Nominal process cooling demand.
Qcpro_sys_MWhyr	Yearly processes cooling demand.
Qcre0_kW	Nominal Refrigeration cooling demand.
Qcre_MWhyr	Refrigeration cooling demand for the system
Qcre_sys0_kW	Nominal refrigeration cooling demand
Qcre_sys_MWhyr	End-use refrigeration demand
Qcs0_kW	Nominal Total cooling demand.
Qcs_dis_ls0_kW	Nominal Cooling distribution losses.
Qcs_dis_ls_MWhyr	Cooling distribution losses
Qcs_em_ls0_kW	Nominal Cooling emission losses.
Qcs_em_ls_MWhyr	Cooling emission losses
Qcs_lat_ahu0_kW	Nominal AHU latent cooling demand.
Qcs_lat_ahu_MWhyr	AHU latent cooling demand
Qcs_lat_aru0_kW	Nominal ARU latent cooling demand.

Continued on next page

Table 65 – continued from previous page

Variable	Description
Qcs_lat_aru_MWhyr	ARU latent cooling demand
Qcs_lat_sys0_kW	Nominal System latent cooling demand.
Qcs_lat_sys_MWhyr	System latent cooling demand
Qcs_MWhyr	Total cooling demand
Qcs_sen_ahu0_kW	Nominal AHU system cooling demand.
Qcs_sen_ahu_MWhyr	Sensible cooling demand in AHU
Qcs_sen_aru0_kW	Nominal ARU system cooling demand.
Qcs_sen_aru_MWhyr	ARU system cooling demand
Qcs_sen_scu0_kW	Nominal SCU system cooling demand.
Qcs_sen_scu_MWhyr	SCU system cooling demand
Qcs_sen_sys0_kW	Nominal Sensible system cooling demand.
Qcs_sen_sys_MWhyr	Total sensible cooling demand
Qcs_sys0_kW	Nominal end-use space cooling demand
Qcs_sys_ahu0_kW	Nominal AHU system cooling demand.
Qcs_sys_ahu_MWhyr	AHU system cooling demand
Qcs_sys_aru0_kW	Nominal ARU system cooling demand.
Qcs_sys_aru_MWhyr	ARU system cooling demand
Qcs_sys_MWhyr	End-use space cooling demand, Qcs_sys = Qcs_sen_sys + Qcs_lat_sys + Qcs_em_ls + Qcs_dis_ls
Qcs_sys_scu0_kW	Nominal SCU system cooling demand.
Qcs_sys_scu_MWhyr	SCU system cooling demand
QH_sys0_kW	Nominal total building heating demand.
QH_sys_MWhyr	Total building heating demand
Qhpro_sys0_kW	Nominal process heating demand.
Qhpro_sys_MWhyr	Yearly processes heating demand.
Qhs0_kW	Nominal space heating demand.
Qhs_dis_ls0_kW	Nominal Heating system distribution losses.
Qhs_dis_ls_MWhyr	Heating system distribution losses
Qhs_em_ls0_kW	Nominal Heating emission losses.
Qhs_em_ls_MWhyr	Heating system emission losses
Qhs_lat_ahu0_kW	Nominal AHU latent heating demand.
Qhs_lat_ahu_MWhyr	AHU latent heating demand
Qhs_lat_aru0_kW	Nominal ARU latent heating demand.
Qhs_lat_aru_MWhyr	ARU latent heating demand
Qhs_lat_sys0_kW	Nominal System latent heating demand.
Qhs_lat_sys_MWhyr	System latent heating demand
Qhs_MWhyr	Total space heating demand.
Qhs_sen_ahu0_kW	Nominal AHU sensible heating demand.
Qhs_sen_ahu_MWhyr	AHU sensible heating demand
Qhs_sen_aru0_kW	ARU sensible heating demand
Qhs_sen_aru_MWhyr	ARU sensible heating demand
Qhs_sen_shu0_kW	Nominal SHU sensible heating demand.
Qhs_sen_shu_MWhyr	SHU sensible heating demand
Qhs_sen_sys0_kW	Nominal HVAC systems sensible heating demand.
Qhs_sen_sys_MWhyr	SHU sensible heating demand
Qhs_sys0_kW	Nominal end-use space heating demand
Qhs_sys_ahu0_kW	Nominal AHU sensible heating demand.
Qhs_sys_ahu_MWhyr	AHU system heating demand
Qhs_sys_aru0_kW	Nominal ARU sensible heating demand.

Continued on next page

Table 65 – continued from previous page

Variable	Description
Qhs_sys_aru_MWhyr	ARU sensible heating demand
Qhs_sys_MWhyr	End-use space heating demand, Qhs_sys = Qhs_sen_sys + Qhs_em_ls + Qhs_dis_ls
Qhs_sys_shu0_kW	Nominal SHU sensible heating demand.
Qhs_sys_shu_MWhyr	SHU sensible heating demand
Qww0_kW	Nominal DHW heating demand.
Qww_MWhyr	DHW heating demand
Qww_sys0_kW	Nominal end-use hotwater demand
Qww_sys_MWhyr	End-use hotwater demand
SOLAR_hs0_kW	Nominal solar thermal energy requirement for space heating supply
SOLAR_hs_MWhyr	Solar thermal energy requirement for space heating supply
SOLAR_ww0_kW	Nominal solar thermal energy requirement for hotwater supply
SOLAR_ww_MWhyr	Solar thermal energy requirement for hotwater supply
WOOD_hs0_kW	Nominal WOOD requirement for space heating supply
WOOD_hs_MWhyr	WOOD requirement for space heating supply
WOOD_ww0_kW	Nominal WOOD requirement for hotwater supply
WOOD_ww_MWhyr	WOOD requirement for hotwater supply

5.3.49 get_radiation_building

path: `outputs/data/solar-radiation/{building}_radiation.csv`

The following file is used by these scripts: `demand`, `photovoltaic`, `photovoltaic_thermal`, `solar_collector`

Variable	Description
Date	Date and time in hourly steps
roofs_top_kW	solar incident on the roof tops
roofs_top_m2	roof top area
walls_east_kW	solar incident on the east facing facades excluding windows
walls_east_m2	area of the east facing facades excluding windows
walls_north_kW	solar incident on the north facing facades excluding windows
walls_north_m2	area of the north facing facades excluding windows
walls_south_kW	solar incident on the south facing facades excluding windows
walls_south_m2	area of the south facing facades excluding windows
walls_west_kW	solar incident on the west facing facades excluding windows
walls_west_m2	area of the south facing facades excluding windows
windows_east_kW	solar incident on windows on the south facing facades
windows_east_m2	window area on the east facing facades
windows_north_kW	solar incident on windows on the south facing facades
windows_north_m2	window area on the north facing facades
windows_south_kW	solar incident on windows on the south facing facades
windows_south_m2	window area on the south facing facades
windows_west_kW	solar incident on windows on the west facing facades
windows_west_m2	window area on the west facing facades

5.3.50 get_radiation_building_sensors

path: `outputs/data/solar-radiation/B001_insolation_Whm2.json`

The following file is used by these scripts: `demand`, `photovoltaic`, `photovoltaic_thermal`, `solar_collector`

Variable	Description
`srf0`	TODO

5.3.51 get_radiation_materials

path: `outputs/data/solar-radiation/buidling_materials.csv`

The following file is used by these scripts:

Variable	Description
`G_win`	Solar heat gain coefficient. Defined according to ISO 13790.
`Name`	Unique building ID. It must start with a letter.
`r_roof`	Reflectance in the Red spectrum. Defined according Radiance. (long-wave)
`r_wall`	Reflectance in the Red spectrum. Defined according Radiance. (long-wave)
`type_base`	Basement floor construction assembly (relates to “code” in ENVELOPE assemblies)
`type_floor`	Internal floor construction assembly (relates to “code” in ENVELOPE assemblies)
`type_roof`	Roof construction assembly (relates to “code” in ENVELOPE assemblies)
`type_wall`	External wall construction assembly (relates to “code” in ENVELOPE assemblies)
`type_win`	Window assembly (relates to “code” in ENVELOPE assemblies)

5.3.52 get_radiation_metadata

path: `outputs/data/solar-radiation/B001_geometry.csv`

The following file is used by these scripts: `demand`, `photovoltaic`, `photovoltaic_thermal`, `solar_collector`

Variable	Description
`AREA_m2`	Surface area.
`BUILDING`	Unique building ID. It must start with a letter.
`intersection`	flag to indicate whether this surface is intersecting with another surface (0: no intersection, 1: intersected)
`orientation`	Orientation of the surface (north/east/south/west/top)
`SURFACE`	Unique surface ID for each building exterior surface.
`TYPE`	Surface typology.
`Xcoor`	Describes the position of the x vector.
`Xdir`	Directional scalar of the x vector.
`Ycoor`	Describes the position of the y vector.
`Ydir`	Directional scalar of the y vector.
`Zcoor`	Describes the position of the z vector.
`Zdir`	Directional scalar of the z vector.

5.3.53 get_schedule_model_file

path: `outputs/data/occupancy/B001.csv`

The following file is used by these scripts: `demand`

Variable	Description
DATE	Time stamp for each day of the year ascending in hourly intervals
Ea_W	Electrical load due to processes
Ed_W	Electrical load due to servers/data centers
El_W	Electrical load due to lighting
Epro_W	Electrical load due to processes
people_p	Number of people in the building
Qcpro_W	Process cooling load
Qcre_W	Cooling load due to cool room refrigeration
Qhpro_W	Process heat load
Qs_W	Sensible heat load of people
Tcs_set_C	Set point temperature of space cooling system
Ths_set_C	Set point temperature of space heating system
Ve_lps	Ventilation rate
Vw_lph	Fresh water consumption (includes cold and hot water)
Vww_lph	Domestic hot water consumption
X_gh	Moisture released by occupants

5.3.54 get_sewage_heat_potential

path: `outputs/data/potentials/Sewage_heat_potential.csv`

The following file is used by these scripts: `optimization`

Variable	Description
mww_zone_kWperC	heat capacity of total sewage in a zone
Qsw_kW	heat extracted from sewage flows
T_in_HP_C	Inlet temperature of the sweage heapump
T_in_sw_C	Inlet temperature of sewage flows
T_out_HP_C	Outlet temperature of the sewage heatpump
T_out_sw_C	Outlet temperature of sewage flows
Ts_C	Average temperature of sewage flows

5.3.55 get_thermal_demand_csv_file

path: `outputs/data/thermal-network/DH__thermal_demand_per_building_W.csv`

The following file is used by these scripts:

Variable	Description
B01	Thermal demand for building B01 at each simulation time step

5.3.56 get_thermal_network_edge_list_file

path: `outputs/data/thermal-network/DH__metadata_edges.csv`

The following file is used by these scripts: `optimization`

Variable	Description
`D_int_m`	Internal pipe diameter for the nominal diameter
`length_m`	Length of each pipe in the network
`Name`	Unique network pipe ID.
`Pipe_DN`	Nominal pipe diameter (e.g. DN100 refers to pipes of approx. 100 mm in diameter)
`Type_mat`	Material of the pipes

5.3.57 get_thermal_network_edge_node_matrix_file

path: `outputs/data/thermal-network/{network_type}__EdgeNode.csv`

The following file is used by these scripts: `thermal_network`

Variable	Description
`NODE`	Names of the nodes in the network
`PIPE0`	Indicates the direction of flow of PIPE0 with respect to each node NODEn: if equal to PIPE0 and NODEn are not connected / if equal to 1 PIPE0 enters NODEn / if equal to -1 PIPE0 leaves NODEn

5.3.58 get_thermal_network_layout_massflow_edges_file

path: `outputs/data/thermal-network/DH__massflow_edges_kgs.csv`

The following file is used by these scripts:

Variable	Description
`PIPE0`	Mass flow rate in pipe PIPE0 at each time step

5.3.59 get_thermal_network_layout_massflow_nodes_file

path: `outputs/data/thermal-network/DH__massflow_nodes_kgs.csv`

The following file is used by these scripts:

Variable	Description
`NODE0`	Mass flow rate in node NODE0 at each time step

5.3.60 get_thermal_network_node_types_csv_file

path: `outputs/data/thermal-network/DH__metadata_nodes.csv`

The following file is used by these scripts:

Variable	Description
`Building`	Unique building ID. It must start with a letter.
`Name`	Unique network node ID.
`Type`	Type of node: "PLANT" / "CONSUMER" / "NONE" (if it is neither)

5.3.61 get_thermal_network_plant_heat_requirement_file

path: `outputs/data/thermal-network/DH__plant_thermal_load_kW.csv`

The following file is used by these scripts:

Variable	Description
`thermal_load_kW`	Thermal load supplied by the plant at each time step

5.3.62 get_thermal_network_pressure_losses_edges_file

path: `outputs/data/thermal-network/DH__pressure_losses_edges_kW.csv`

The following file is used by these scripts:

Variable	Description
`PIPE0`	Pressure losses at pipe PIPE0 at each time step

5.3.63 get_thermal_network_substation_ploss_file

path: `outputs/data/thermal-network/DH__pumping_load_due_to_substations_kW.csv`

The following file is used by these scripts:

Variable	Description
`B01`	Pumping load at building substation B01 for each timestep

5.3.64 get_thermal_network_velocity_edges_file

path: `outputs/data/thermal-network/DH__velocity_edges_mpers.csv`

The following file is used by these scripts:

Variable	Description
`PIPE0`	Flow velocity of heating/cooling medium in pipe PIPE0

5.3.65 get_total_demand

path: `outputs/data/demand/Total_demand.csv`

The following file is used by these scripts: `decentralized`, `emissions`, `network_layout`, `system_costs`, `optimization`, `sewage_potential`, `thermal_network`

Variable	Description
`Af_m2`	Conditioned floor area (heated/cooled)
`Aocc_m2`	Occupied floor area (heated/cooled)
`Aroof_m2`	Roof area
`COAL_hs0_kW`	Nominal Coal requirement for space heating supply
`COAL_hs_MWhyr`	Coal requirement for space heating supply

Continued on next page

Table 66 – continued from previous page

Variable	Description
COAL_ww0_kW	Nominal Coal requirement for hotwater supply
COAL_ww_MWhyr	Coal requirement for hotwater supply
DC_cdata0_kW	Nominal district cooling for final data center cooling demand
DC_cdata_MWhyr	District cooling for data center cooling demand
DC_cre0_kW	Nominal district cooling for refrigeration demand
DC_cre_MWhyr	District cooling for refrigeration demand
DC_cs0_kW	Nominal district cooling for space cooling demand
DC_cs_MWhyr	Energy consumption of space cooling system (if supplied by District Cooling), DC_cs = Qcs_sys / eff_cs
DH_hs0_kW	Nominal energy requirement by district heating (space heating supply)
DH_hs_MWhyr	Energy requirement by district heating (space heating supply)
DH_ww0_kW	Nominal Energy requirement by district heating (hotwater supply)
DH_ww_MWhyr	Energy requirement by district heating (hotwater supply)
E_cdata0_kW	Nominal Data centre cooling specific electricity consumption.
E_cdata_MWhyr	Electricity consumption due to data center cooling
E_cre0_kW	Nominal Refrigeration system electricity consumption.
E_cre_MWhyr	Electricity consumption due to refrigeration
E_cs0_kW	Nominal Cooling system electricity consumption.
E_cs_MWhyr	Energy consumption of cooling system (if supplied by electricity grid), E_cs = Qcs_sys / eff_cs
E_hs0_kW	Nominal Heating system electricity consumption.
E_hs_MWhyr	Electricity consumption due to space heating
E_sys0_kW	Nominal end-use electricity demand
E_sys_MWhyr	End-use total electricity consumption, E_sys = Eve + Ea + El + Edata + Epro + Eaux + Ev
E_ww0_kW	Nominal Domestic hot water electricity consumption.
E_ww_MWhyr	Electricity consumption due to hot water
Ea0_kW	Nominal end-use electricity for appliances
Ea_MWhyr	End-use electricity for appliances
Eal0_kW	Nominal Total net electricity for all sources and sinks.
Eal_MWhyr	End-use electricity consumption of appliances and lighting, Eal = El_W + Ea_W
Eaux0_kW	Nominal Auxiliary electricity consumption.
Eaux_MWhyr	End-use auxiliary electricity consumption, Eaux = Eaux_fw + Eaux_ww + Eaux_cs + Eaux_hs + Ehs_lat_aux
Edata0_kW	Nominal Data centre electricity consumption.
Edata_MWhyr	Electricity consumption for data centers
El0_kW	Nominal end-use electricity for lights
El_MWhyr	End-use electricity for lights
Epro0_kW	Nominal Industrial processes electricity consumption.
Epro_MWhyr	Electricity supplied to industrial processes
Ev0_kW	Nominal end-use electricity for electric vehicles
Ev_MWhyr	End-use electricity for electric vehicles

Continued on next page

Table 66 – continued from previous page

Variable	Description
Eve0_kW	Nominal end-use electricity for ventilation
Eve_MWhyr	End-use electricity for ventilation
GFA_m2	Gross floor area
GRID0_kW	Nominal Grid electricity consumption
GRID_a0_kW	Nominal grid electricity requirements for appliances
GRID_a_MWhyr	Grid electricity requirements for appliances
GRID_aux0_kW	Nominal grid electricity requirements for auxiliary loads
GRID_aux_MWhyr	Grid electricity requirements for auxiliary loads
GRID_cdata0_kW	Nominal grid electricity requirements for servers cooling
GRID_cdata_MWhyr	Grid electricity requirements for servers cooling
GRID_cre0_kW	Nominal grid electricity requirements for refrigeration
GRID_cre_MWhyr	Grid electricity requirements for refrigeration
GRID_cs0_kW	Nominal grid electricity requirements for space cooling
GRID_cs_MWhyr	Grid electricity requirements for space cooling
GRID_data0_kW	Nominal grid electricity requirements for servers
GRID_data_MWhyr	Grid electricity requirements for servers
GRID_hs0_kW	Nominal grid electricity requirements for space heating
GRID_hs_MWhyr	Grid electricity requirements for space heating
GRID_l0_kW	Nominal grid electricity consumption for lights
GRID_l_MWhyr	Grid electricity requirements for lights
GRID_MWhyr	Grid electricity consumption, GRID_a + GRID_l + GRID_v + GRID_ve + GRID_data + GRID_pro + GRID_aux + GRID_ww + GRID_cs + GRID_hs + GRID_cdata + GRID_cre
GRID_pro0_kW	Nominal grid electricity requirements for industrial processes
GRID_pro_MWhyr	Grid electricity requirements for industrial processes
GRID_v0_kW	Nominal grid electricity consumption for electric vehicles
GRID_v_MWhyr	Grid electricity requirements for electric vehicles
GRID_ve0_kW	Nominal grid electricity consumption for ventilation
GRID_ve_MWhyr	Grid electricity requirements for ventilation
GRID_ww0_kW	Nominal grid electricity requirements for hot water supply
GRID_ww_MWhyr	Grid electricity requirements for hot water supply
Name	Unique building ID. It must start with a letter.
NG_hs0_kW	Nominal NG requirement for space heating supply
NG_hs_MWhyr	NG requirement for space heating supply
NG_ww0_kW	Nominal NG requirement for hotwater supply
NG_ww_MWhyr	NG requirement for hotwater supply
OIL_hs0_kW	Nominal OIL requirement for space heating supply
OIL_hs_MWhyr	OIL requirement for space heating supply
OIL_ww0_kW	Nominal OIL requirement for hotwater supply
OIL_ww_MWhyr	OIL requirement for hotwater supply
people0	Nominal occupancy
PV0_kW	Nominal PV electricity consumption
PV_MWhyr	PV electricity consumption
QC_sys0_kW	Nominal Total system cooling demand.

Continued on next page

Table 66 – continued from previous page

Variable	Description
QC_sys_MWhyr	Total energy demand for cooling, QC_sys = Qcs_sys + Qcdata_sys + Qcre_sys + Qcpro_sys
Qcdata0_kW	Nominal Data centre cooling demand.
Qcdata_MWhyr	Data centre cooling demand
Qcdata_sys0_kW	Nominal end-use data center cooling demand
Qcdata_sys_MWhyr	End-use data center cooling demand
Qcpro_sys0_kW	Nominal process cooling demand.
Qcpro_sys_MWhyr	Yearly processes cooling demand.
Qcre0_kW	Nominal Refrigeration cooling demand.
Qcre_MWhyr	Refrigeration cooling demand for the system
Qcre_sys0_kW	Nominal refrigeration cooling demand
Qcre_sys_MWhyr	End-use refrigeration demand
Qcs0_kW	Nominal Total cooling demand.
Qcs_dis_ls0_kW	Nominal Cooling distribution losses.
Qcs_dis_ls_MWhyr	Cooling distribution losses
Qcs_em_ls0_kW	Nominal Cooling emission losses.
Qcs_em_ls_MWhyr	Cooling emission losses
Qcs_lat_ahu0_kW	Nominal AHU latent cooling demand.
Qcs_lat_ahu_MWhyr	AHU latent cooling demand
Qcs_lat_aru0_kW	Nominal ARU latent cooling demand.
Qcs_lat_aru_MWhyr	ARU latent cooling demand
Qcs_lat_sys0_kW	Nominal System latent cooling demand.
Qcs_lat_sys_MWhyr	Latent cooling demand
Qcs_MWhyr	Total cooling demand
Qcs_sen_ahu0_kW	Nominal AHU system cooling demand.
Qcs_sen_ahu_MWhyr	AHU system cooling demand
Qcs_sen_aru0_kW	Nominal ARU system cooling demand.
Qcs_sen_aru_MWhyr	ARU system cooling demand
Qcs_sen_scu0_kW	Nominal SCU system cooling demand.
Qcs_sen_scu_MWhyr	SCU system cooling demand
Qcs_sen_sys0_kW	Nominal Sensible system cooling demand.
Qcs_sen_sys_MWhyr	Sensible system cooling demand
Qcs_sys0_kW	Nominal end-use space cooling demand
Qcs_sys_ahu0_kW	Nominal AHU system cooling demand.
Qcs_sys_ahu_MWhyr	AHU system cooling demand
Qcs_sys_aru0_kW	Nominal ARU system cooling demand.
Qcs_sys_aru_MWhyr	ARU system cooling demand
Qcs_sys_MWhyr	End-use space cooling demand, Qcs_sys = Qcs_sen_sys + Qcs_lat_sys + Qcs_em_ls + Qcs_dis_ls
Qcs_sys_scu0_kW	Nominal SCU system cooling demand.
Qcs_sys_scu_MWhyr	SCU system cooling demand
QH_sys0_kW	Nominal total building heating demand.
QH_sys_MWhyr	Total energy demand for heating
Qhpro_sys0_kW	Nominal process heating demand.
Qhpro_sys_MWhyr	Yearly processes heating demand.
Qhs0_kW	Nominal Total heating demand.
Qhs_dis_ls0_kW	Nominal Heating system distribution losses.

Continued on next page

Table 66 – continued from previous page

Variable	Description
Qhs_dis_ls_MWhyr	Heating system distribution losses
Qhs_em_ls0_kW	Nominal Heating emission losses.
Qhs_em_ls_MWhyr	Heating system emission losses
Qhs_lat_ahu0_kW	Nominal AHU latent heating demand.
Qhs_lat_ahu_MWhyr	AHU latent heating demand
Qhs_lat_aru0_kW	Nominal ARU latent heating demand.
Qhs_lat_aru_MWhyr	ARU latent heating demand
Qhs_lat_sys0_kW	Nominal System latent heating demand.
Qhs_lat_sys_MWhyr	System latent heating demand
Qhs_MWhyr	Total heating demand
Qhs_sen_ahu0_kW	Nominal AHU sensible heating demand.
Qhs_sen_ahu_MWhyr	Sensible heating demand in AHU
Qhs_sen_aru0_kW	ARU sensible heating demand
Qhs_sen_aru_MWhyr	ARU sensible heating demand
Qhs_sen_shu0_kW	Nominal SHU sensible heating demand.
Qhs_sen_shu_MWhyr	SHU sensible heating demand
Qhs_sen_sys0_kW	Nominal HVAC systems sensible heating demand.
Qhs_sen_sys_MWhyr	Sensible heating demand
Qhs_sys0_kW	Nominal end-use space heating demand
Qhs_sys_ahu0_kW	Nominal AHU sensible heating demand.
Qhs_sys_ahu_MWhyr	AHU system heating demand
Qhs_sys_aru0_kW	Nominal ARU sensible heating demand.
Qhs_sys_aru_MWhyr	ARU sensible heating demand
Qhs_sys_MWhyr	End-use space heating demand, Qhs_sys = Qhs_sen_sys + Qhs_em_ls + Qhs_dis_ls
Qhs_sys_shu0_kW	Nominal SHU sensible heating demand.
Qhs_sys_shu_MWhyr	SHU sensible heating demand
Qww0_kW	Nominal DHW heating demand.
Qww_MWhyr	DHW heating demand
Qww_sys0_kW	Nominal end-use hotwater demand
Qww_sys_MWhyr	End-use hotwater demand
SOLAR_hs0_kW	Nominal solar thermal energy requirement for space heating supply
SOLAR_hs_MWhyr	Solar thermal energy requirement for space heating supply
SOLAR_ww0_kW	Nominal solar thermal energy requirement for hotwater supply
SOLAR_ww_MWhyr	Solar thermal energy requirement for hotwater supply
WOOD_hs0_kW	Nominal WOOD requirement for space heating supply
WOOD_hs_MWhyr	WOOD requirement for space heating supply
WOOD_ww0_kW	Nominal WOOD requirement for hotwater supply
WOOD_ww_MWhyr	WOOD requirement for hotwater supply

5.3.66 get_water_body_potential

path: `outputs/data/potentials/Water_body_potential.csv`

The following file is used by these scripts: `optimization`

Variable	Description
QLake_kW	thermal potential from water body
Ts_C	average temperature of the water body

5.3.67 get_weather_file

path: inputs/weather/weather.epw

The following file is used by these scripts: decentralized, demand, optimization, photovoltaic, photovoltaic_thermal, radiation, schedule_maker, shallow_geothermal_potential, solar_collector, thermal_network

Variable	Description
aerosol_opt_thousandths (index = 29)	TODO
Albedo (index = 32)	TODO
atmos_Pa (index = 9)	TODO
ceiling_hgt_m (index = 25)	TODO
datasource (index = 5)	TODO
day (index = 2)	TODO
days_last_snow (index = 31)	TODO
dewpoint_C (index = 7)	TODO
difhorillum_lux (index = 18)	TODO
difhorrad_Whm2 (index = 15)	TODO
dirnorillum_lux (index = 17)	TODO
dirnorrad_Whm2 (index = 14)	TODO
drybulb_C (index = 6)	TODO
extdirrad_Whm2 (index = 11)	TODO
exthorrad_Whm2 (index = 10)	TODO
glohorillum_lux (index = 16)	TODO
glohorrad_Whm2 (index = 13)	TODO
horirsky_Whm2 (index = 12)	TODO
hour (index = 3)	TODO
liq_precip_depth_mm (index = 33)	TODO
liq_precip_rate_Hour (index = 34)	TODO
minute (index = 4)	TODO
month (index = 1)	TODO
opaqskycvr_tenths (index = 23)	TODO
precip_wtr_mm (index = 28)	TODO
presweathcodes (index = 27)	TODO
presweathobs (index = 26)	TODO
relhum_percent (index = 8)	TODO
snowdepth_cm (index = 30)	TODO
totskycvr_tenths (index = 22)	TODO
visibility_km (index = 24)	TODO
winddir_deg (index = 20)	TODO
windspd_ms (index = 21)	TODO
year (index = 0)	TODO
zenlum_lux (index = 19)	TODO

5.3.68 PV_metadata_results

path: `outputs/data/potentials/solar/B001_PV_sensors.csv`

The following file is used by these scripts:

Variable	Description
`area_installed_module_m2`	The area of the building suface covered by one solar panel
`AREA_m2`	Surface area.
`array_spacing_m`	Spacing between solar arrays.
`B_deg`	Tilt angle of the installed solar panels
`BUILDING`	Unique building ID. It must start with a letter.
`CATB`	Category according to the tilt angle of the panel
`CATGB`	Category according to the annual radiation on the panel surface
`CATteta_z`	Category according to the surface azimuth of the panel
`intersection`	flag to indicate whether this surface is intersecting with another surface (0: no intersection, 1: intersected)
`orientation`	Orientation of the surface (north/east/south/west/top)
`SURFACE`	Unique surface ID for each building exterior surface.
`surface`	Unique surface ID for each building exterior surface.
`surface_azimuth_deg`	Azimuth angle of the panel surface e.g. south facing = 180 deg
`tilt_deg`	Tilt angle of roof or walls
`total_rad_Whm2`	Total radiatiative potential of a given surfaces area.
`TYPE`	Surface typology.
`type_orientation`	Concatenated surface type and orientation.
`Xcoor`	Describes the position of the x vector.
`Xdir`	Directional scalar of the x vector.
`Ycoor`	Describes the position of the y vector.
`Ydir`	Directional scalar of the y vector.
`Zcoor`	Describes the position of the z vector.
`Zdir`	Directional scalar of the z vector.

5.3.69 PV_results

path: `outputs/data/potentials/solar/B001_PV.csv`

The following file is used by these scripts:

Variable	Description
Area_PV_m2	Total area of investigated collector.
Date	Date and time in hourly steps.
E_PV_gen_kWh	Total electricity generated by the collector.
PV_roofs_top_E_kWh	Electricity production from photovoltaic panels on roof tops
PV_roofs_top_m2	Collector surface area on roof tops.
PV_walls_east_E_kWh	Electricity production from photovoltaic panels on east facades
PV_walls_east_m2	Collector surface area on east facades.
PV_walls_north_E_kWh	Electricity production from photovoltaic panels on north facades
PV_walls_north_m2	Collector surface area on north facades.
PV_walls_south_E_kWh	Electricity production from photovoltaic panels on south facades
PV_walls_south_m2	Collector surface area on south facades.
PV_walls_west_E_kWh	Electricity production from photovoltaic panels on west facades
PV_walls_west_m2	West facing wall collector surface area.
radiation_kWh	Total radiatiative potential.

5.3.70 PV_total_buildings

path: `outputs/data/potentials/solar/PV_total_buildings.csv`

The following file is used by these scripts:

Variable	Description
Area_PV_m2	Total area of investigated collector.
E_PV_gen_kWh	Total electricity generated by the collector.
Name	Unique building ID. It must start with a letter.
PV_roofs_top_E_kWh	Electricity production from photovoltaic panels on roof tops
PV_roofs_top_m2	Collector surface area on roof tops.
PV_walls_east_E_kWh	Electricity production from photovoltaic panels on east facades
PV_walls_east_m2	Collector surface area on east facades.
PV_walls_north_E_kWh	Electricity production from photovoltaic panels on north facades
PV_walls_north_m2	Collector surface area on north facades.
PV_walls_south_E_kWh	Electricity production from photovoltaic panels on south facades
PV_walls_south_m2	Collector surface area on south facades.
PV_walls_west_E_kWh	Electricity production from photovoltaic panels on west facades
PV_walls_west_m2	West facing wall collector surface area.
radiation_kWh	Total radiatiative potential.

5.3.71 PV_totals

path: `outputs/data/potentials/solar/PV_total.csv`

The following file is used by these scripts: `optimization`

Variable	Description
Area_PV_m2	Total area of investigated collector.
Date	Date and time in hourly steps.
E_PV_gen_kWh	Total electricity generated by the collector.
PV_roofs_top_E_kWh	Electricity production from photovoltaic panels on roof tops
PV_roofs_top_m2	Collector surface area on roof tops.
PV_walls_east_E_kWh	Electricity production from photovoltaic panels on east facades
PV_walls_east_m2	Collector surface area on east facades.
PV_walls_north_E_kWh	Electricity production from photovoltaic panels on north facades
PV_walls_north_m2	Collector surface area on north facades.
PV_walls_south_E_kWh	Electricity production from photovoltaic panels on south facades
PV_walls_south_m2	Collector surface area on south facades.
PV_walls_west_E_kWh	Electricity production from photovoltaic panels on west facades
PV_walls_west_m2	West facing wall collector surface area.
radiation_kWh	Total radiatiative potential.

5.3.72 PVT_metadata_results

path: `outputs/data/potentials/solar/B001_PVT_sensors.csv`

The following file is used by these scripts:

Variable	Description
area_installed_module_m2	The area of the building suface covered by one solar panel
AREA_m2	Surface area.
array_spacing_m	Spacing between solar arrays.
B_deg	Tilt angle of the installed solar panels
BUILDING	Unique building ID. It must start with a letter.
CATB	Category according to the tilt angle of the panel
CATGB	Category according to the annual radiation on the panel surface
CATteta_z	Category according to the surface azimuth of the panel
intersection	flag to indicate whether this surface is intersecting with another surface (0: no intersection, 1: intersected)
orientation	Orientation of the surface (north/east/south/west/top)
SURFACE	Unique surface ID for each building exterior surface.
surface	Unique surface ID for each building exterior surface.
surface_azimuth_deg	Azimuth angle of the panel surface e.g. south facing = 180 deg
tilt_deg	Tilt angle of roof or walls
total_rad_Whm2	Total radiatiative potential of a given surfaces area.
TYPE	Surface typology.
type_orientation	Concatenated surface type and orientation.
Xcoor	Describes the position of the x vector.
Xdir	Directional scalar of the x vector.
Ycoor	Describes the position of the y vector.
Ydir	Directional scalar of the y vector.
Zcoor	Describes the position of the z vector.
Zdir	Directional scalar of the z vector.

5.3.73 PVT_results

path: `outputs/data/potentials/solar/B001_PVT.csv`

The following file is used by these scripts:

Variable	Description
Area_PVT_m2	Total area of investigated collector.
Date	Date and time in hourly steps.
E_PVT_gen_kWh	Total electricity generated by the collector.
Eaux_PVT_kWh	Auxiliary electricity consumed by the collector.
mcp_PVT_kWperC	Capacity flow rate (mass flow* specific heat capacity) of the hot water delivered by the collector.
PVT_roofs_top_E_kWh	Electricity production from photovoltaic-thermal panels on roof tops
PVT_roofs_top_m2	Collector surface area on roof tops.
PVT_roofs_top_Q_kWh	Heat production from photovoltaic-thermal panels on roof tops
PVT_walls_east_E_kWh	Electricity production from photovoltaic-thermal panels on east facades
PVT_walls_east_m2	Collector surface area on east facades.
PVT_walls_east_Q_kWh	Heat production from photovoltaic-thermal panels on east facades
PVT_walls_north_E_kWh	Electricity production from photovoltaic-thermal panels on north facades
PVT_walls_north_m2	Collector surface area on north facades.
PVT_walls_north_Q_kWh	Heat production from photovoltaic-thermal panels on north facades
PVT_walls_south_E_kWh	Electricity production from photovoltaic-thermal panels on south facades
PVT_walls_south_m2	Collector surface area on south facades.
PVT_walls_south_Q_kWh	Heat production from photovoltaic-thermal panels on south facades
PVT_walls_west_E_kWh	Electricity production from photovoltaic-thermal panels on west facades
PVT_walls_west_m2	West facing wall collector surface area.
PVT_walls_west_Q_kWh	Heat production from photovoltaic-thermal panels on west facades
Q_PVT_gen_kWh	Total heat generated by the collector.
Q_PVT_l_kWh	Collector heat loss.
radiation_kWh	Total radiatiative potential.
T_PVT_re_C	Collector hot water return temperature.
T_PVT_sup_C	Collector heating supply temperature.

5.3.74 PVT_total_buildings

path: `outputs/data/potentials/solar/PVT_total_buildings.csv`

The following file is used by these scripts:

Variable	Description
Area_PVT_m2	Total area of investigated collector.
E_PVT_gen_kWh	Total electricity generated by the collector.
Eaux_PVT_kWh	Auxiliary electricity consumed by the collector.
Name	Unique building ID.
PVT_roofs_top_E_kWh	Electricity production from photovoltaic-thermal panels on roof tops
PVT_roofs_top_m2	Collector surface area on roof tops.
PVT_roofs_top_Q_kWh	Heat production from photovoltaic-thermal panels on roof tops
PVT_walls_east_E_kWh	Electricity production from photovoltaic-thermal panels on east facades
PVT_walls_east_m2	Collector surface area on east facades.
PVT_walls_east_Q_kWh	Heat production from photovoltaic-thermal panels on east facades
PVT_walls_north_E_kWh	Electricity production from photovoltaic-thermal panels on north facades
PVT_walls_north_m2	Collector surface area on north facades.
PVT_walls_north_Q_kWh	Heat production from photovoltaic-thermal panels on north facades
PVT_walls_south_E_kWh	Electricity production from photovoltaic-thermal panels on south facades
PVT_walls_south_m2	Collector surface area on south facades.
PVT_walls_south_Q_kWh	Heat production from photovoltaic-thermal panels on south facades
PVT_walls_west_E_kWh	Electricity production from photovoltaic-thermal panels on west facades
PVT_walls_west_m2	West facing wall collector surface area.
PVT_walls_west_Q_kWh	Heat production from photovoltaic-thermal panels on west facades
Q_PVT_gen_kWh	Total heat generated by the collector.
Q_PVT_l_kWh	Collector heat loss.
radiation_kWh	Total radiatiative potential.

5.3.75 PVT_totals

path: outputs/data/potentials/solar/PVT_total.csv

The following file is used by these scripts: optimization

Variable	Description
Area_PVT_m2	Total area of investigated collector.
Date	Date and time in hourly steps.
E_PVT_gen_kWh	Total electricity generated by the collector.
Eaux_PVT_kWh	Auxiliary electricity consumed by the collector.
mcp_PVT_kWperC	Capacity flow rate (mass flow* specific heat capacity) of the hot water delivered by the collector.
PVT_roofs_top_E_kWh	Electricity production from photovoltaic-thermal panels on roof tops
PVT_roofs_top_m2	Collector surface area on roof tops.
PVT_roofs_top_Q_kWh	Heat production from photovoltaic-thermal panels on roof tops
PVT_walls_east_E_kWh	Electricity production from photovoltaic-thermal panels on east facades
PVT_walls_east_m2	Collector surface area on east facades.
PVT_walls_east_Q_kWh	Heat production from photovoltaic-thermal panels on east facades
PVT_walls_north_E_kWh	Electricity production from photovoltaic-thermal panels on north facades
PVT_walls_north_m2	Collector surface area on north facades.
PVT_walls_north_Q_kWh	Heat production from photovoltaic-thermal panels on north facades
PVT_walls_south_E_kWh	Electricity production from photovoltaic-thermal panels on south facades
PVT_walls_south_m2	Collector surface area on south facades.
PVT_walls_south_Q_kWh	Heat production from photovoltaic-thermal panels on south facades
PVT_walls_west_E_kWh	Electricity production from photovoltaic-thermal panels on west facades
PVT_walls_west_m2	West facing wall collector surface area.
PVT_walls_west_Q_kWh	Heat production from photovoltaic-thermal panels on west facades
Q_PVT_gen_kWh	Total heat generated by the collector.
Q_PVT_l_kWh	Collector heat loss.
radiation_kWh	Total radiatiative potential.
T_PVT_re_C	Collector heating supply temperature.
T_PVT_sup_C	Collector heating supply temperature.

5.3.76 SC_metadata_results

path: `outputs/data/potentials/solar/B001_SC_ET_sensors.csv`

The following file is used by these scripts:

Variable	Description
area_installed_module_m2	The area of the building suface covered by one solar panel
AREA_m2	Surface area.
array_spacing_m	Spacing between solar arrays.
B_deg	Tilt angle of the installed solar panels
BUILDING	Unique building ID. It must start with a letter.
CATB	Category according to the tilt angle of the panel
CATGB	Category according to the annual radiation on the panel surface
CATteta_z	Category according to the surface azimuth of the panel
intersection	flag to indicate whether this surface is intersecting with another surface (0: no intersection, 1: intersected)
orientation	Orientation of the surface (north/east/south/west/top)
SURFACE	Unique surface ID for each building exterior surface.
surface	Unique surface ID for each building exterior surface.
surface_azimuth_deg	Azimuth angle of the panel surface e.g. south facing = 180 deg
tilt_deg	Tilt angle of roof or walls
total_rad_Whm2	Total radiatiative potential of a given surfaces area.
TYPE	Surface typology.
type_orientation	Concatenated surface type and orientation.
Xcoor	Describes the position of the x vector.
Xdir	Directional scalar of the x vector.
Ycoor	Describes the position of the y vector.
Ydir	Directional scalar of the y vector.
Zcoor	Describes the position of the z vector.
Zdir	Directional scalar of the z vector.

5.3.77 SC_results

path: `outputs/data/potentials/solar/B001_SC_ET.csv`

The following file is used by these scripts: `decentralized`

Variable	Description
Area_SC_m2	Total area of investigated collector.
Date	Date and time in hourly steps.
Eaux_SC_kWh	Auxiliary electricity consumed by the collector.
mcp_SC_kWperC	Capacity flow rate (mass flow* specific heat capacity) of the hot water delivered by the collector.
Q_SC_gen_kWh	Total heat generated by the collector.
Q_SC_l_kWh	Collector heat loss.
radiation_kWh	Total radiatiative potential.
SC_ET_roofs_top_m2	Collector surface area on roof tops.
SC_ET_roofs_top_Q_kWh	Heat production from solar collectors on roof tops
SC_ET_walls_east_m2	Collector surface area on east facades.
SC_ET_walls_east_Q_kWh	Heat production from solar collectors on east facades
SC_ET_walls_north_m2	Collector surface area on north facades.
SC_ET_walls_north_Q_kWh	Heat production from solar collectors on north facades
SC_ET_walls_south_m2	Collector surface area on south facades.
SC_ET_walls_south_Q_kWh	Heat production from solar collectors on south facades
SC_ET_walls_west_m2	Collector surface area on west facades.
SC_ET_walls_west_Q_kWh	Heat production from solar collectors on west facades
T_SC_re_C	Collector hot water return temperature.
T_SC_sup_C	Collector hot water supply temperature.

5.3.78 SC_total_buildings

path: `outputs/data/potentials/solar/SC_ET_total_buildings.csv`

The following file is used by these scripts:

Variable	Description
Area_SC_m2	Total area of investigated collector.
Eaux_SC_kWh	Auxiliary electricity consumed by the collector.
Name	Unique building ID.
Q_SC_gen_kWh	Total heat generated by the collector.
Q_SC_l_kWh	Collector heat loss.
radiation_kWh	Total radiatiative potential.
SC_ET_roofs_top_m2	Roof top collector surface area.
SC_ET_roofs_top_Q_kWh	Heat production from solar collectors on roof tops
SC_ET_walls_east_m2	East facing wall collector surface area.
SC_ET_walls_east_Q_kWh	Heat production from solar collectors on east facades
SC_ET_walls_north_m2	North facing wall collector surface area.
SC_ET_walls_north_Q_kWh	Heat production from solar collectors on west facades
SC_ET_walls_south_m2	South facing wall collector surface area.
SC_ET_walls_south_Q_kWh	Heat production from solar collectors on south facades
SC_ET_walls_west_m2	West facing wall collector surface area.
SC_ET_walls_west_Q_kWh	Heat production from solar collectors on west facades

5.3.79 SC_totals

path: `outputs/data/potentials/solar/SC_FP_total.csv`

The following file is used by these scripts: `optimization`

Variable	Description
Area_SC_m2	Collector surface area on south facades.
Date	Date and time in hourly steps.
Eaux_SC_kWh	Auxiliary electricity consumed by the collector.
mcp_SC_kWperC	Capacity flow rate (mass flow* specific heat capacity) of the hot water delivered by the collector.
Q_SC_gen_kWh	Total heat generated by the collector.
Q_SC_l_kWh	Collector heat loss.
radiation_kWh	Total radiatiative potential.
SC_FP_roofs_top_m2	Collector surface area on roof tops.
SC_FP_roofs_top_Q_kWh	Heat production from solar collectors on roof tops
SC_FP_walls_east_m2	Collector surface area on east facades.
SC_FP_walls_east_Q_kWh	Heat production from solar collectors on east facades
SC_FP_walls_north_m2	Collector surface area on north facades.
SC_FP_walls_north_Q_kWh	Heat production from solar collectors on north facades
SC_FP_walls_south_m2	Collector surface area on south facades.
SC_FP_walls_south_Q_kWh	Heat production from solar collectors on south facades
SC_FP_walls_west_m2	Collector surface area on west facades.
SC_FP_walls_west_Q_kWh	Heat production from solar collectors on west facades
T_SC_re_C	Collector hot water return temperature.
T_SC_sup_C	Collector hot water supply temperature.

CHAPTER 6

Legal

6.1 License

The `core` of the City Energy Analyst is registered under The MIT License (MIT).

6.1.1 for V0.3c

6.1.2 for V0.1

6.2 Disclaimer

The City Energy Analyst is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall the authors or copyright holders be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with the software or the use or other dealings in the software.

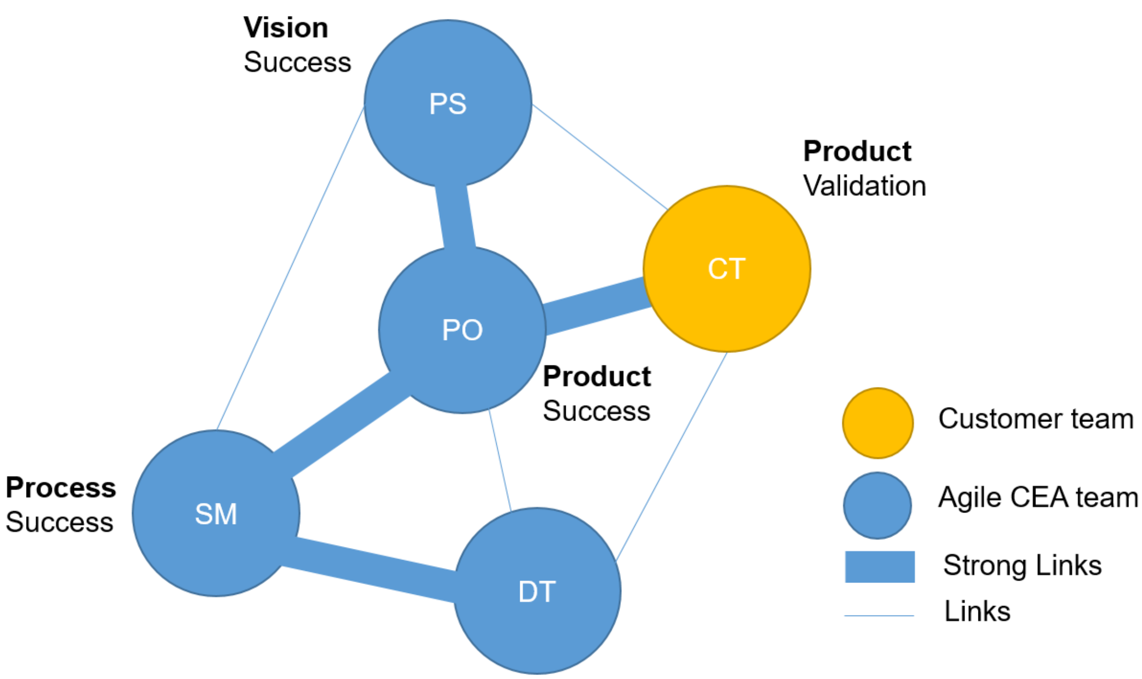
CHAPTER 7

Being Agile with CEA

7.1 Roles and Responsibilities

by Dr. Jimeno A. Fonseca, 2020, updated by Shanshan Hsieh, March 2020

This is a guide of roles and responsibilities in CEA. We use this to more formally coordinate our work. To check who is currently holding each position, we invite you to check here for more information.



7.1.1 Product Sponsor (PS)

The Project Sponsor is the person that has overall responsibility and accountability.

Responsibilities

1. Guarantee that our vision be successful by finding and securing the budget and ensuring that high-level organizational risks are mitigated.
2. Champion the project based on whether the work fits our research needs and overall strategy.
3. Empower the Product Owner to act for him/her on a more tactical basis.

Engagement

1. Participate in the Roadmap planning and review event (see *Activities*).

7.1.2 Product Owner (PO)

The Product Owner is the voice of the customer and performs all tactical operations.

PO Responsibilities

1. Guarantee our product success by acting as a conduit/facilitator for communication between the team and the customers.

The means to communication to the customers include:

- Conducting Usability Tests
- Organizing Teaching Activities
- Maintaining Documentation
- Maintaining CEA website
- Maintaining cea@arch.ethz.ch
- Distributing Newsletters

2. Coordinate the Roadmap planning and review event (see *Activities*).
3. Break down the overall vision into feasible and tactical pieces of work for the development team to understand and create.
4. Prioritizing and selecting user stories. This entails grooming the **product backlogs** with the help of the Srcum Master.

The **product backlogs** contains issues that is labelled as Must-have and Should-have. Here are the details of the labels used to categorize new issues in the CEA:

Label	Description
Must-have	• Features defined in the roadmap. • Bugs.
Should-have	• Issues that can't be categorized for the moment.
Won't have	• Features that are not in the roadmap. • Features that are hard to maintain. These features can become external modules of the CEA.

PO Engagement

1. Participate in all *Activities*.
2. One day per week on communication to the customer team and event coordination.

7.1.3 Scrum Master (SM)

The Scrum Master is the servant leader and maintainer of agile practices.

SM Responsibilities

1. Coordinate all *Activities* that form the agile practices of the CEA.
2. Coordinate the training of new developers.
   1. Introduce the functionalities in the CEA.
   2. Introduce the CEA-ecosystem
3. Coordinate the training of the team
   1. Pair-programming sessions
   2. Refresher course on programming
4. Support the Product Owner on setting priorities and grooming the product backlog.
5. Maintaining the internal communication channel (slack).

7.1.4 Development Team (DT)

The Development team is the group of individuals who build CEA. His/her main responsibilities are:

1. Create the best working software possible.
2. Design, analyze, develop, test and document new features in CEA.
3. Be self-organized and self-directed.
4. Participate in all agile practices led by the Scrum Master (*Activities*).

7.1.5 Customer Team (CT)

The Customer team is the group of individuals who use CEA. They are not part of the CEA team, but their function is extremely important for the succes of the project.

1. Represent one or more User Personas in CEA.
2. Validate CEA and give feedback to the Product Owner
3. Communicate issues and new features to the Product Owner.

7.2 User Personas

This is a guide of User Personas in CEA. These are potential end-users of CEA. We describe the goals and priorities of these user personas hereafter. We use this information to build User Stories / Report bugs / Develop new features. It is of utmost importance for every developer to know this by heart.

7.2.1 Researcher

The priorities of this User Persona are:

1. To find an answer to a unique research question/hypothesis
2. To build upon the state-of-the-art, saving time in coding newly established algorithms.
3. To conduct experiments quickly.
4. to generate replicable and transparent results.

7.2.2 Student

The priorities of this User Persona are:

1. To understand the trade-offs of planning cities for energy efficiency.
2. To build intuition about the mechanics of different analysis and modeling techniques.
3. To generate replicable and transparent results.

7.2.3 Practitioner

The priorities of this User Persona are:

1. To analyze real case studies quickly.
2. To generate great visualizations which aid communication to stakeholders.
3. To generate standard, replicable and transparent results.

7.3 User Stories

User stories define WHAT (need), WHY (reason) and for WHOM (User persona) we aim to solve a new bug or implement a new feature. This helps our Project Owner to better prioritize issues around CEA.

We have a standard yet simple way to define these user stories. This guide walks you through defining your first user story.

7.3.1 Step 1. define a user persona

At CEA we differentiate four (4) potential USERs for creating an user story. These are key groups of users we want CEA to serve and are defined as user personas. The first step in creating a new story is to select a User Persona from the following list:

1. Researcher: A member of the CEA research team or network of contributors.
2. Student or Practitioner: An active user of CEA.

For more information about the goals and priorities of the different user personas check: *User Personas*

7.3.2 Step 2. define its needs

Think about the NEED this user persona has. Is it about a problem/bug or rather about a new feature you want to be implemented? It is important that you think about the NEED from the point of view of the user persona.

7.3.3 Step 3. define a reason

Now think about why the user persona needs that.

7.3.4 Step 4. put the story together

Now put it all together using the following template:

As a USER PERSONA **I want to** NEED **so I can** REASON

Here is an example:

USER PERSONA Researcher

NEED know how to define user stories

REASON add new bugs and features to the pipeline of CEA

The result will be the title of your user story:

As a Researcher **I want to** know how to define user stories **so I** can add new bugs and features to the pipeline of CEA.

One more example:

USER PERSONA Student

NEED understand how the dynamic tool works

REASON use CEA more effectively

The result will be the title of your user story:

As a Student **I want to** understand how the dynamic tool works **so I** can use CEA more effectively.

7.3.5 Step 5. submit a new user story

Now it is time to submit a new user story in CEA. For this:

1. Go to the CEA repository in Github.
2. Click *New Issue*
3. in Title, add the name of the user story
4. Finally, in description, give a more detailed description of the problem.
5. In the description you can directly connect to other user stories using # or connect to other people using @ in the text.

7.4 Activities

This is a guide of Activities carried out during the development of CEA. This activities are part of our concept of operation under responsibilities of the Scrum Master.

As the project evolves, so do the activities: We try to keep them as lean as possible and expect to adapt this document as needed.

One important aspect of the team is how it is distributed between (mainly) 2 timezones (Singapore, Switzerland) and a large part of the team has other responsibilities (e.g. writing their PhD thesis) that needs to be accounted for and impacts the amount of "ceremony" we can sustain.

7.4.1 The Kanban board

The written record of these activities is reflected in the GitHub issues and are maintained using ZenHub to produce a Kanban view of our process. The columns on our Kanban board are:

- New Issues
- Project Backlog
- Sprint Backlog
- In Progress
- In Review
- Closed

The different events focus on different parts of the process and their respective Kanban columns are mentioned below.

7.4.2 Pre-Planning event

- Duration: 1 - 2 hours.
- Frequency: Before each Planning event (bi-weekly)
- Scope: Prepare the Kanban board to streamline the Planning event.
- Kanban:
  - Move User stories from "New Issues" to either "Product Backlog" (if in scope) or close them with the label "wont-fix" or "known issue".

  - Make sure each User story in Product backlog has a (preliminary) time estimation (measured in story points, each story point is roughly 1 day of work)
  - Make sure each User story in Product backlog has a priority label: "Must have" or "Should have"
  - Order the Product backlog with most important issues at the top of the column

NOTE: Currently, the Pre-Planning event is held every second Wednesday at 9am Zurich time on Zoom, the day before the Planning event.

Attendees: Product Owner, Scrum Master

7.4.3 Planning event

- Duration: 1 - 2 hours.
- Frequency: After every block of work (sprint, bi-weekly)
- Scope: Define what User stories should go in the next block of work (sprint). These stories are attached to a new time-bounded milestone.
- Kanban: User stories are moved from the Product Backlog and to the Sprint Backlog and assigned to a member of the Development team. Each User story requires a time estimate and the developer the issue is assigned to is responsible for updating this estimate based on his/her experience.

Currently, the Planning event is held every second Thursday at 9am Zurich time on Zoom. Subject to change.

Attendees: Scrum Master, Product Owner, Product Sponsor, and Development Team.

7.4.4 Daily stand-up

- Duration: 15 min.
- Frequency: daily
- Scope: Provide early support and mentoring
- Kanban: Main focus is on User stories in the "In Progress" and "In Review" columns. This is also time to point out important "New Issues" and discuss problems blocking "Sprint Backlog" issues from being moved to "In Progress".

NOTE: Currently this is being held every morning at 9am (Switzerland time) on Zoom with the SM and a reduced set of the DT.

Attendees: Scrum Master and Development Team.

7.4.5 Roadmap planning and review event

- Duration: 1 - 2 hours.
- Frequency: quarterly
- **Scope:**
  1. Demonstration of the newest development in the CEA.
  2. Bottlenecks that are faced by the development team.
  3. Planning for the next quarter.

Attendees: Scrum Master, Product Owner, Product Sponsor, and Development Team.

NOTE: While this event is mainly an update to the product sponsors, it is also open to the entire CEA team and the customer team.

7.4.6 Retrospective

- Duration: 1 hour.
- Frequency: After every review event
- Scope: Discuss what was wrong and set next steps to follow.

NOTE: This activity is currently not part of our repertoire. We are trying to figure out the right parameters for this.

Attendees: Scrum Master and Development Team.

7.5 Communication channels

This is a guide of Communication channels in CEA. These are maintained by our Scrum Master.

7.5.1 CEA website

- Host: squarespace.com
- Credentials: Ask the Product Owner
- Administrator: Product Owner
- Access granted to: Scrum Master, Product Owner, Development Team

7.5.2 CEA e-mail

- Host: arch.ethz.ch
- Credentials: Ask the Product Owner
- Administrator: Product Owner
- Access granted to: Scrum Master, Product Owner

7.5.3 CEA messenger

- Host: ceadev.slack.com
- Credential: Ask the Scrum Master
- Administrator: Scrum Master
- Access granted to: Scrum Master, Product Owner, Development team

7.5.4 CEA newsletter

- Host: mailchimp.com
- Credentials: Ask the Product Owner
- Administrator: Product Owner
- Access granted to: Scrum Master, Product Owner

7.5.5 CEA documents

- Host: gmail.com
- Credentials: Ask the Scrum Master Daren Thomas
- Administrator: Scrum Master
- Access granted to: Scrum Master, Product Owner, Development Team

CHAPTER 8

Contributing to City Energy Analyst (CEA)

First off, **thank you** for taking the time to contribute!

The following is a set of guidelines for contributing to CEA, which is hosted on GitHub. These are mostly guidelines, not rules. Use your best judgment, and feel free to propose changes to this document in a pull request.

8.1 Step 1. Let us know about it

Whether you would like to implement a new feature or fix a bug, let the CEA team know. This will help us to coordinate efforts across network of developers. It will also help us to better support your work.

You can let us know by reporting a new issue https://github.com/architecture-building-systems/CityEnergyAnalyst/issues/new/choose

8.2 Step 2. Install our development version

If you have not done it yet, please install the development version of CEA. Make sure you have the CEA repository (included in the development version) connected via github and your favorite python editor (ours is PyCharm). You can download the development version with the same link as the one used www.cityenergyanalyst.com/tryit.

Here we tell you how to do it https://cityenergyanalyst.com/blog/2019/6/3/how-to-install-the-cea-on-windows-part-2

8.3 Step 3. Branch out and code

Branch out from out main 'Master branch" of our github repository and start coding. This can be done with the CEA development version. For this make sure to use one of our template scripts and follow the documentation guide. This could help to maintain an homogenous structure, and help us to acknowledge you.

Check these guides for more details: how-to-use-github, how-to-add-a-new-script-to-the-cea.

8.4 Step 4. Check style

If you have not done it yet, take some time to get acquainted with variable names in CEA. This would make easier for you to understand and develop consistent code.

Here we tell some basic hints how-to-name-variables.

8.5 Step 5. Run some local tests

Now test if your creation does not brake CEA's functionality.

The next guide explains how to run local unittests in CEA how-to-test-the-cea.

8.6 Step 6. Create a Pull request

Now it is time to ask other developer of CEA to review your code so we get to make it part of the CEA main core. We do this by creating a Pull Request in github.

Check this guide for more details on how to do it: how-to-use-github.

8.7 Step 7. Claim your CEA T-shirt!

What happens after that? We will check the code, and if all is correct we will proceed to make it part of CEA's main source code. If your work has been merged, give yourself an applause. You have just made part of the growing network of developers of CEA.

Your are entitled to claim a CEA T-shirt after this to cea@arch.ethz.ch

CHAPTER 9

Developer walkthrough

9.1 The Configuration File

The City Energy Analyst uses a configuration file for storing user preferences. User preferences are inputs to simulation runs like what weather file to use, what scenario to use and script-specific inputs.

When you first run the `cea` tool (e.g. with `cea install-toolbox` during the installation process), the default configuration file is copied to your home folder.

On Windows systems, the home folder is usually something like `C:\Users\YourUserName`, so the configuration file would be stored in `C:\Users\michelle\cea.config`, assuming that your username is `michelle`.

9.1.1 Setting values in the configuration file

The most important values to set when working with the CEA are probably those under the `[general]` section, specifically `scenario`, and `weather`

Open the `cea.config` file with a text editor (`notebook.exe` will do just fine) and update the values.

Note: We expect to implement an editor for the configuration file soon.

9.1.2 The configuration file and the command line interface

When you run the CEA from the command line (with the `cea` command), then the values to use as inputs to the scripts are taken from the configuration file. You can override each value by adding it as a parameter to the `cea` command, using the syntax `--` + `parameter-name` + ‘‘ ‘‘ + `value`. Example:

```
$ cea demand --scenario C:\scenario\baseline --weather Brussels
```

9.1.3 The configuration file and the ArcGIS interface

The values in the configuration file are used as the default values when you open up a cea tool in the ArcGIS interface.

9.2 Configuration File Details

Let's explore the details of how the configuration file works!

The configuration file edited by the user (`~/cea.config`) is only the tip of the iceberg, resting on a foundation of the default configuraiton file `default.config` file and the class `cea.config.Configuration`, which reads in the default configuration file as well as the user configuration file and makes those parameters available to the scripts. Each script is provided with an instance of `cea.config.Configuration` called `config`.

Each parameter is defined in a section. Each parameter has a type, which specifies the range of values allowed for that parameter and also how to read and write them to the configuration file.

Access to parameters through the `config` variable happens by section. Since all section names and parameter names in the configuration file follow the kebab-case naming convention, and these are not valid python identifiers, a translation is made to the snake_case naming convention: All hyphens (`-`) are replaced by underscores (`_`).

The syntax is simple:

```
"config." + [section] + "." + parameter
```

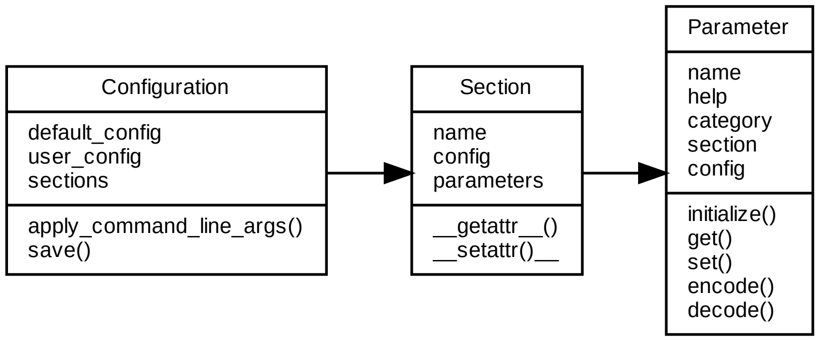
The section name is optional for the section `general`, so `config.general.scenario` refers to the same parameter as `config.scenario`. Note that these parameters can also be set:

```
config.scenario = r'C:\hoenggerberg\baseline'
```

If you want to persist these changes to disk, you need to explicitly save them with `cea.config.Configuration.save()`.

Note: It is a bad idea to have multiple instances of `cea.config.Configuration`, as if one part of a script changes a parameter, this will not be reflected in the other instances. Each CEA script accepts a `config` argument to it's `main` function and should only use that.

9.2.1 Overview



9.2.2 Initialization of the config object

An instance of `cea.config.Configuration` is created with an optional `config_file` parameter that specifies the configuration file to load as the user configuration file. This defaults to `~/cea.config`. This file is parsed as a `ConfigParser.SafeConfigParser`, using the default configuration as a backup for the values and stored in the attribute `user_config`. Another `ConfigParser.SafeConfigParser` is created for the default configuration and stored in the attribute `default_config`.

Next, the `default_config` is used to create a dictionary of :py:class‘cea.config.Section‘ objects and each section is populated with a dictionary of `cea.config.Parameter` instances. The default configuration file lists not only each parameter, but additional keys for each parameter as well. Example:

```
[general]
scenario = C:\reference-case-open\baseline
scenario.type = PathParameter
scenario.help = Path to the scenario to run
```

Using this information, the parameter `general:scenario` is assigned a default value of `C:\reference-case-open\baseline`, is represented by a subtype of ``cea.config.Parameter` `` called `cea.config.PathParameter` and has a help text “Path to the scenario to run” - which is stored in the `help` attribute of the parameter object.

Some subclasses of `cea.config.Parameter` have additional configuration, like the *cea.config.ChoiceParameter*:

```
[data-helper]
region = CH
region.type = ChoiceParameter
region.choices = CH SIN custom
region.help = The region to use for the databases (either CH or SIN) - set to "custom
↪" if you want to edit them
```

When the `config` instance is creating the parameters, each parameter object is given a chance to initialize itself with a call to `cea.config.Parameter.initialize(parser)()` with `parser` set to the `default_config`. Subclasses of `Parameter` can override this method to read this additional configuration.

9.2.3 How a value is read from the config file

When a script does something like `config.general.weather`, the `config.sections` dictionary is checked for the section named `general` and the `parameters` dictionary in that section is checked for a parameter named `weather`. The `cea.config.Parameter.get()` method is called on that parameter and the result of this call is returned.

Based on the default configuration file, this is defined as:

```
[general]
weather = Zug-inducity_1990_2010_TMY
weather.type = WeatherPathParameter
weather.help = either a full path to a weather file or the name of one of the weather␣
↪files shipped with the CEA
```

So the parameter is of type `cea.config.WeatherPathParameter`.

Inside the `cea.config.Parameter.get()` method, a call is made to `cea.config.Parameter.decode()`, passing in the value read from the user configuration file. Subclasses of `Parameter` specify how to encode and decode values to the configuration file. The semantics are:

- `decode` takes a string from a configuration file (or from the command line) and returns a typed value (e.g. a `bool` if the parameter type is `cea.config.BooleanParameter`).
- `encode` takes a typed value (e.g. a boolean value) and encodes it to a string that can be stored in the configuration file.

In the case of `cea.config.WeatherPathParameter`, `decode` will ensure that the path to the weather file exists and, if just the name of a weather file in the CEA weather file database is returned, resolves that to the full path to that file. Hence, on my system, the value of `config.weather` is `C:\Users\darthoma\Documents\GitHub\CityEnergyAnalyst\cea\databases\weather\Zurich.epw`.

9.2.4 How a value is saved to the config file

The mechanism for saving a value to the config file works similarly: `cea.config.Parameter.set()` is called, which in turn calls `cea.config.Parameter.encode()` - subclasses can override this to provide type specific behaviour.

9.2.5 How to create new parameter types

Steps:

1. subclass `cea.config.Parameter`
2. optional: override `initialize` to settings
3. optional: override `encode` to format the parameter value as a string
4. optional: override `decode` to read the parameter value from a string

Check the existing parameter types for ideas!

9.3 User Interfaces

The CEA code exposes multiple interfaces as an API:

- CLI (Command Line Interface) - each module in the CEA implements a CLI for calling it from the command line.
- euler - a set of scripts for running the CEA sensitivity analysis on the ETH Euler cluster is provided in the folder `euler` and can be used as a starting point for running the analysis on similar clusters and / or linux machines.

9.3.1 The Command Line Interface

The most portable way to interact with the CEA is via the CLI. Type the following command in your shell to see the list of commands available:

```
> cea --help
usage: cea SCRIPT [OPTIONS]
       to run a specific script
usage: cea --help SCRIPT
       to get additional help specific to a script

SCRIPT can be one of: benchmark-graphs, compile, data-helper,
    dbf-to-excel, demand, demand-graphs, embodied-energy, emissions,
    excel-to-dbf, extract-reference-case, install-toolbox,
    latitude, list-demand-graphs-fields, locate, longitude, mobility,
    operation-costs, photovoltaic, photovoltaic-thermal, read-config, read-config-
↪section,
    retrofit-potential, scenario-plots, sensitivity-demand-analyze,
    sensitivity-demand-samples, sensitivity-demand-simulate,
    solar-collector, test, weather-files, weather-path, write-config
```

All scripts use the configuration file as the default source of parameters. See the *Configuration File Details* for information on the configuration file.

The parameters in the configuration file relevant to a script can be overridden. To see which parameters are used by a certain script, use the syntax `cea --help SCRIPT`:

```
> cea --help data-helper

building properties algorithm

OPTIONS for data-helper:
--scenario: C:/reference-case-open/baseline
    Path to the scenario to run
--archetypes: ['comfort', 'architecture', 'HVAC', 'internal-loads']
    List of archetypes to process
```

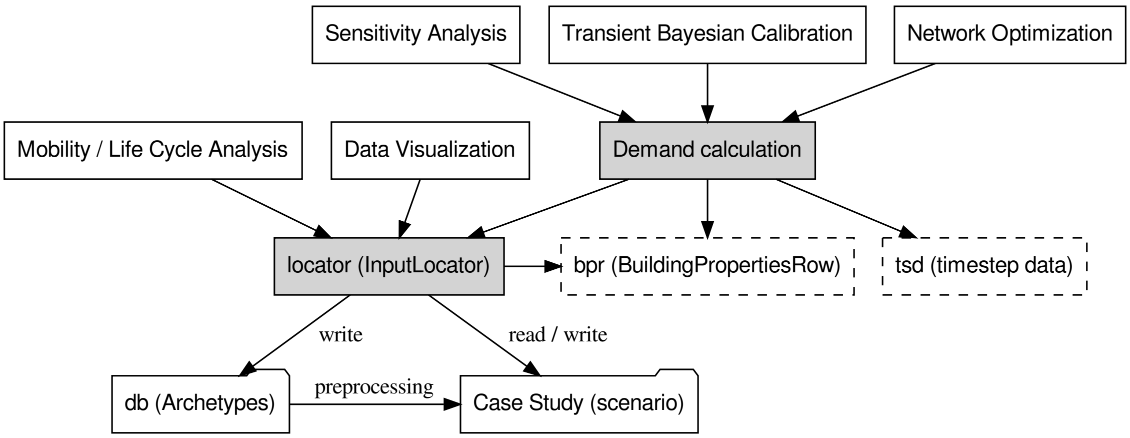
This displays some documentation on the script as well as a list of parameters, their default values and a description of the parameter. Using this information, the `data-helper` script can be run like this:

```
> cea data-helper --scenario C:/reference-case-open/scenario1 --archetypes HVAC
```

Note: All options are optional and have default values as defined in the configuration file!

9.4 Architecture

The architecture of the CEA is still a bit in flux, but some main components have already been developed and will be explained in this chapter. The following figure shows a high-level view of the main components of the CEA:



9.4.1 Demand calculation

At the core of the CEA is the demand calculation. The demand calculation retrieves inputs from the scenario folder and stores outputs back to the scenario folder. A preprocessing step can be used to add archetype data to a scenario as a first guess of certain parameters.

The demand calculation uses a special variable called `tsd` to store information about the timestep data during the calculation of thermal loads for each building. The data structure used is a simple python dictionary of NumPy arrays. Each of these arrays has the length 8760, to total number of hours of the year. The keys of the `tsd` dictionary are the names of the state variables of the simulation.

The demand calculation also uses a variable `bpr` to store building properties of a single building.

9.4.2 InputLocator

The `cea.inputlocator.InputLocator` class encapsulates the code for creating paths for input and output to the archetypes and the contents of the scenario (input and output files). An instance of this class is found in most of the code and is always named `locator`, unless multiple `InputLocator` objects are used, e.g. for comparing scenarios.

Each method of the `locator` starts with `get_*` and returns a string containing the full path to the resource requested. These `get_*` methods should be the only way to obtain file- and folder names in the CEA - files and folders should especially not be concatenated with strings and backslashes (\). Instead, new paths should be introduced as methods of the `InputLocator` class.

One of the main benefits of doing this is that it makes documentation of what files are read / written by what module of the CEA easier. The `funcionlogger` module can be used to trace these calls for generating documentation.

The private method `_ensure_folder(*paths)` is used to join path components and at the same time ensure that the folders are present in the scenario folder, creating them with `os.makedirs` if necessary.

NOTE: The list of `get_*` methods is getting very long. We might end up creating a namespace hierarchy for grouping related paths together.

9.4.3 Analysis and Visualization

Separate modules exist for analyzing different aspects of a scenario. Some of the analysis modules operate only on the input data (LCA for embedded emissions, mobility) and others operate on the output of the demand module (LCA for emissions due to operation). These modules are grouped in the folder `cea/analysis`.

The folder `cea/plots` contains modules for plotting outputs of the calculations.

9.4.4 "Higher order" modules

Some of the modules in the CEA use the demand calculation to calculate variants of a scenario. This includes the sensitivity analysis, the calibration and the network optimization code. All these modules call the demand calculation as part of their process.

9.5 How to review a pull request

Code review could be time-consuming, but it is extremely important. All pull requests (PR) to the CEA should be reviewed by at least one contributor with maintenance right. The reviewer needs to ensure the changes in the code are aligned with the authors' description and do not compromise the existing functionalities in the CEA.

9.5.1 1. Read the PR description and follow the test

The author of the PR should provide an instruction on how to test the implementation of the new changes. As the reviewer, you should be able to follow the instruction provided by the PR's author, or provide feedback if the instruction is unclear. Once the test provided by the author has passed, the reviewer may proceed to the next step.

9.5.2 2. Go through the file changes

It is always a good idea to go through all the changes at least once. Please follow this guide to review the file changes on GitHub.

During this process, the reviewer should check for:

- Conflicts with master. Make sure the branch is updated with the latest master, and all conflicts are resolved.
- Sufficient documentation. Check if the documentation is sufficient for the next person to understand the code.
- Hard-coded values. All hard-coded values should be avoided if possible.
- Unit tests to implement. The reviewer should decide whether a unit test should be implemented, and request the PR author to implement one accordingly.
- Changes that might affect other existing functions. In this case, the reviewer should come up with a test to ensure the existing functions are still working as intended.

Once all the points are checked out, the reviewer may proceed to the next step.

9.5.3 3. Run tests

All PRs are automatically sent to test by Jenkins, it executes `cea test --workflow quick` on a remote computer. The test result is directly shown in the PR page on GitHub.

Additionally, it is always a good idea to run a complete test (`cea test --workflow slow`) on your local computer. If Jenkins encounters any errors, you can also reproduce those errors by running `cea test --workflow quick` locally. See here for more information on `cea test`.

Once `cea test` is passed, the reviewer may proceed to the last step!

9.5.4 4. Merge the Pull Request

Now you have made sure the PR is going to improve the CEA, thank you for your time! You may go ahead and merge the PR. If the new changes would affect many users, you might want to consider publishing it on the #_critical_updates channel on Slack.

9.6 How to add a heating/cooling system in CEA

9.6.1 Step 0: Make an issue and create a branch

As this procedure requires adding scripts in CEA master, please make a branch before performing the changes.

9.6.2 Step 1: Add the new system to the database

1. Open `cea/databases/systems/emission_systems.xls`
2. In the tab `heating` or `cooling`, add a row for the new system.
3. Specify the operating conditions of the new system, for cooling systems:

- `code`: add a new code `Tx` that has not been used.
- `Qcsmax_Wm2`: maximum cooling capacity of the system.
- `dTcs_C`:

For Air Handling Units (ahu), if applicable:

- `Tscs0_ahu_C`: coolant (water) supply temperature at the primary side
- `dTcs0_ahu_C`: temperature change of the coolant at the primary side
- `Tc_sup_air_ahu_C`: air supply temperature from ahu to the room

For Air Recirculation Units (aru), if applicable:

- `Tscs0_aru_C`: coolant (water) supply temperature at the primary side
- `dTcs0_aru_C`: temperature change of the coolant at the primary side
- `Tc_sup_air_aru_C`: air supply temperature from ahu to the room

For Sensible Cooling Units (scu), if applicable:

- `Tscs0_scu_C`: coolant (water) supply temperature at the primary side
- `dTcs0_scu_C`: temperature change of the coolant at the primary side

9.6.3 Step 2: Add the new system to the options

1. Go to script `cea/demand/control_heating_cooling_systems.py`
2. Add the code of the new systems (Tx) to function `has_cooling_systems` or `has_heating_systems`
3. Add a new function that check the type of the system, similar to `has_3for2_cooling_systems`

9.6.4 Step 3: Add a new function to model new technologies

Currently, there are models for AHU, ARU, SCU running wiht heating/cooling coil. If the new systems is utilizing different technologies, the models should be added to *airconditioning_model.py*.

9.6.5 Step 3: Add a new function to calculate cooling/heating loads

1. Go to script `cea/demand/hourly_procedure_heating_cooling_system_load.py`
2. Add a new function to set up the calculation procedure for cooling/heating loads, similar to `calc_cool_loads_3for2`

9.6.6 Step 4: Add distribution losses

1. Go to *cea/demand/sensible_loads.py*
2. Update *calc_Qhs_Qcs_loss*

9.6.7 Step 5: Calculate temperature and mass flow primary supply systems

1. Go to *cea/demand/sensible_loads.py*
2. Update *calc_temperatures_emission_systems*

9.6.8 Step 6: Calculate auxiliary electricity

calc_Eauxf_cs_dis calc_Eauxf_hs_dis

9.7 How to create a new release?

This section describes the steps necessary to create a new release of the City Energy Analyst (CEA).

9.7.1 Versioning

Each release of the CEA needs a version number. Version numbers need to increase for PyPI. The relevant documentation for python version numbers is documented in PEP440. The CEA uses the following versioning scheme in compliance with PEP440:

> major.minor[.revision][pre-release]

Major and minor version segments in this scheme refer to the milestone (sprint) the release was developed for. The major version segment works on roughly a yearly time scale while the minor version segment tracks sprints inside the major release. Each such pair (major.minor) refers to a "milestone" in the GitHub issues milestones list.

If a release needs to be updated after publishing, an optional revision can be used, starting at 1 and incrementing.

During the sprint, the pre-release section is used to represent the current state of the master branch. At the beginning of the sprint, alpha versions are used. Examples: 2.2a0, 2.2a1, 2.2a2, etc. In this phase the issues belonging to the milestone are being added.

Once the code base settles down, beta versions can be used. Examples: 2.2b0, 2.2b1, 2.2b2, etc. In this phase, new features should not be added anymore and instead testing / bug fixing activities should dominate.

Before releasing a milestone, the release candidates can be used. Examples: 2.2rc0, 2.2rc1, 2.2rc2, etc. In this phase the software is just being tested with show-stopping bugs being fixed if possible.

Where to find the current version number

The current version number can be found in the module `cea` (actually, since `cea` is a package, you need to look into the file `__init__.py`) in the variable `__version__`.

All code requiring knowledge of the current version number should read the version from here.

In python modules this can be achieved by:

```
import cea
version_number = cea.__version__
```

The NSIS installer (see section *Creating the installer*) uses the helper tool `setup/get_version.exe` to extract the version and write it to the file `setup/cea_version.txt` - if importing `cea` is not an option, you could explore this avenue too...

Responsibility for version numbers

The repository admin merging a pull request to master is responsible for updating the version number.

9.7.2 Create a Release Branch

- Create a branch `Release-x.x.x` from master.

9.7.3 Update the CREDITS.md file

For each minor release (2.2, 2.3, ...) the `CREDITS.md` file needs to be updated to include all the authors that worked on that release. Update the "How to Cite" section with the Zenodo link to the correct version and doi.

9.7.4 Update CHANGELOG

- Run `create-changelog.py` in `CityEnergyAnalyst\bin`.
- Update `CHANGELOG.md` with the latest changes from the outputs.

9.7.5 Updating the CEA GUI interface

You'll need yarn and Node.js installed.

For the installer to be able to pick up the newest version of the CEA GUI interface, make sure you

- Pull the newest version of the `CityEnergyAnalyst-GUI` repository
- Open CEA Console, navigate to the GitHub repo of the `CityEnergyAnalyst-GUI` repository
- Type `yarn`, wait for the command to complete (this will update packages if necessary)

9.7.6 Creating the installer

- First, make sure you have the Nullsoft Scriptable Installation System (NSIS) installed. See **:docs:'how-to-set-up-nsis'**
- Next, make sure the command *cea-dev build* is configured properly. The configuration should look something like this:

```
(CEA) λ cea-config build
City Energy Analyst version 3.11.0
Configuring `cea build` with the following parameters:
- development:nsis = C:\Program Files (x86)\NSIS\Bin\makensis.exe
  (default: )
- development:conda = C:\Users\darthoma\miniconda3\condabin\conda.bat
  (default: )
- development:gui = c:\Users\darthoma\Documents\GitHub\CityEnergyAnalyst-GUI
  (default: )
- development:yarn = C:\Users\darthoma\AppData\Roaming\npm\yarn.cmd
  (default: )
```

You can either edit the `cea.config` file directly or use `cea-config build --nsis C:\...\makensis.exe --conda ....`

Note: The paths will be different on your system. Use the `conda.bat` in `condabin` of your Anaconda/Miniconda installation. The path to `gui` should be set to the repository folder of the CityEnergyAnalyst-GUI repository.

- Install `conda-pack` by typing `conda install conda-pack`.
- Creating the installer is then as easy as `cea-dev build`. This will run quite some time as it will create a new conda environment for the version, conda-pack it, and do a lot of compressing.
- Locate the installer in the CityEnergyAnalyst repository under `setup/Output`.

9.7.7 Create a Release Draft on GitHub

- Tag the release with the correct version number.

9.7.8 Testing in a virtual machine

In order to test the release, it is a good idea to run the installation guide / installer on a clean virtual machine, e.g. with VirtualBox.

This test should go as far as running `cea test --workflow slow` just to be sure everything is still working. This test goes a bit further than the regular test in that it makes sure the installation instructions still work on a new installation. This is important because it can find missing packages in the dependency lists etc.

It's a good idea to use a different username on the VM as the one you used to create the installer - some `pip` bugs can be found that way.

9.7.9 Merge the Release Branch

- Update the "How to Cite" section inside CREDITS.md with the Zenodo link to the correct version and doi.
- Merge the branch `Release-x.x.x` into master.

9.7.10 Publish the Release on GitHub

- The release should be published so that it could be found on the CityEnergyAnalyst repository on GitHub. Add the

installer you created in the previous step. - It is recommended to also publish a release on the CityEnergyAnalyst-GUI repository that corresponds to the version on the CityEnergyAnalyst repository.

9.7.11 Building the documentation

Well documented code is an essential part of the release, allowing your code's legacy to only grow in glory and admiration.

The documentation will be rendered via the readthedocs site, allowing future developers, practitioners, researchers and students to understand and build upon your work. CEA uses sphinx to document all module code, and GraphViz to render flow charts (please install Graphviz to view graphs).

First, launch the CEA Console created by the installer and call (please address any errors (red text) which appears during the sphinx build):

```
cea-doc html
```

This tool will:

- Remove any outdated module rst files
- Rebuild all module rst files
- Render all rst files to html
- Open any documentation html's for files identified by a Gitdiff.

Finally, any changes to the conda environment need to be reflected in the `CityEnergyAnalyst/environment.yml` file and if your code writes any new output variables or files, the `CityEnergyAnalyst/cea/schemas.yml` should be updated accordingly.

For more information, check out the how-to-document-cea.

9.7.12 Updating Link in www.cityenergyanalyst.com/try-cea

- Go to http://www.cityenergyanalyst.com
- Press Esc and try logging into squarespace
- Go to Pages/Try CEA (it is the last page in the list)
- Go to edit 'Page content'
- Go to edit 'Form'

- Change 'Form Name' to the name of the new version of CEA you just released
- Go to the tab 'Advanced'
- Change 'POST-SUBMIT REDIRECT' to the link where the .exe of CEA can be downloaded from
- Change 'POST-SUBMIT MESSAGE'/here, to the link where the .exe of CEA can be downloaded from
- Click 'Apply'
- Click 'Save'

9.7.13 Uploading to PyPI

Note: This step is not necessary anymore for installation.

- Check long-description with this commandline:

```
python setup.py --long-description | for /f %i in ('where rst2html.py') do python
↪%i > %temp%\ld.html && start %temp%\ld.html
```

  - make sure the output is valid / no errors, as this will be the text of the CEA on PyPI
- Delete any old distributions from dist folder (you can just delete the whole `dist` folder if you like)
- Do `python setup.py sdist bdist_wheel`
  - this will recreate the `dist` folder with two files that look similar to these:
    * cityenergyanalyst-2.2-py2-none-any.whl
    * cityenergyanalyst-2.2.tar.gz
- Use twine to upload to PyPI (`twine upload dist/*`)
  - you can get twine with `pip install twine` (it should be pre-installed in the CEA Console)
  - the command above assumes you have set the `TWINE_PASSWORD` and `TWINE_USERNAME` environment variables if not, use the `--username` and `--password` positional arguments
  - ask the repository admins for username and password

9.8 How to set up the Jenkins server on a new PC

Note: you only need to do this when the current Jenkins server dies

Note: this guide assumes you are installing on a Windows 10 Professional system. Adjust accordingly for other systems, but keep in mind that some functionality of the CEA is dependant on Windows.

There are a few steps to take to setting up a Jenkins server:

- installation of some prerequisites
- installation of Jenkins
- installation of a tunnel to the Jenkins server

- global configuration of Jenkins
- configuration of the Jenkins items
  - cea test for new pull requests
  - cea test for merges to master

9.8.1 Installation of some prerequisites

You will need to install these softwares:

- CityEnergyAnalyst (install with the `Setup_CityEnergyAnalyst_<VERSION>.exe` installer)
  - we'll be using the Python environment shipped with the CEA to test the CEA
  - we'll also be using the *git.exe* shipped with the CEA

9.8.2 Installation of Jenkins

- Download & install jenkins from https://jenkins.io
  - LTS version Jenkins for Windows (last time this document was used, it was version 2.204.4)
  - just double click the installer, next, next, next (all default values)
  - set jenkins service to use local user
    - Open up the Services Manager (search for "Services" in the Windows menu)
    - locate and open the "Jenkins" service
    - make sure the Startup type is set to "Automatic" so the Jenkins starts up again after reboots
    - on the tab "Log On", select "This account" instead of "Local System account" and enter in your credentials
      - this will allow the Jenkins to have access to your user profile. You can create an account just for this service and use that for the rest of this guide.
- open browser to http://localhost:8080 (NOTE: the installer did this automatically last time tried)
  - follow instructions to enter initial admin password
    - click "install suggested plugins"
    - create first admin user
      - Username: *cea*
      - Password: (same as *cityea* user in outlook, ask Jimeno or Daren for the password)
      - Full name: *City Energy Analyst*
      - E-mail address: *cea@arch.ethz.ch*
    - Click "Manage Jenkins"
      - click "Configure System" (following this guide here: https://wiki.jenkins.io/display/JENKINS/Github+Plugin#GitHubPlugin-GitHubhooktriggerforGITScmpolling)
      - set "# of executors" to 1 (let's just make it dead simple, no concurrency, less headache)

9.8.3 Installation of a tunnel to the Jenkins server

This guide assumes you're running the Jenkins on a Windows PC inside a corporate network. We use the ngrok service to tunnel webhooks triggered by GitHub back to the Jenkins server.

- download ngrok for Windows (https://ngrok.com/download)
- extract `ngrok.exe` to `%PROGRAMDATA%\ceajenkins\ngrok.exe`
  - (you might need to create the folder `ceajenkins` first)
- create a file `ngrok.yml` in the folder `%PROGRAMDATA%\ceajenkins` with the following contents:

```
authtoken: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
tunnels:
  ceajenkins:
    proto: http
    addr: 8080
    subdomain: ceajenkins
```

  - (replace the authtoken variable with the authtoken obtained from ngrok
- test it with this command: `%PROGRAMDATA%\ceajenkins\ngrok.exe start --config %PROGRAMDATA%\ceajenkins\ngrok.yml ceajenkins`
  - you should now be able to access your Jenkins installation by going to https://ceajenkins.ngrok.io from any computer with access to the internet
  - press CTRL+C to shutdown the tunnel
- copy the `CityEnergyAnalyst\bin\ceajenkins.py` file to `%PROGRAMDATA%\ceajenkins`
  - if you haven't checked out the CEA, download it from the CEA GitHub repository
- copy the CEA Dependencies folder (after installing CEA, it should be in `%USERPROFILE%\Documents\CityEnergyAnalysts\Dependencies`) twice
  - once to `C:\ProgramData\ceajenkins\ceatest`
  - once to `C:\ProgramData\ceajenkins\ceatestall`
  - (actually rename the folder `Dependencies` to `ceatest` and `ceatestall` respectively)
- in order for the service to find required DLL's, ensure the PATH includes the following folders (use the windows search function to find the control panel item "Edit System Environment Variables"):
  - `C:\ProgramData\ceajenkins\ceatestall\Python\`
  - `C:\ProgramData\ceajenkins\ceatestall\Python\lib\site-packages\win32`
  - `C:\ProgramData\ceajenkins\ceatestall\Python\lib\site-packages\pywin32_system32`
  - make sure you edit the System Variables, not the User Environment Variables
- open `cmd.exe` with admin rights (right click, then "Run as Administrator")
- run `python %PROGRAMDATA%\ceajenkins\ceajenkins.py install`
- open the windows services panel (just search for "Services" in the windows menu)
  - locate "CEA Jenkins keepalive", right click, "Properties"
  - set Startup type to "Automatic"
  - set the account in the "Log On" tab to your user account (the one that you used to install all of the above stuff)

  - start the service!
  - you should now be able to access your Jenkins installation by going to https://ceajenkins.ngrok.io from any computer with access to the internet (test this)

9.8.4 Global configuration of Jenkins

Now that we have a tunnel set up, we can start configuring the Jenkins server, mainly following this guide:

- open browser to http://ceajenkins.ngrok.io and log in
- click "Manage Jenkins" and then "Configure System"
  - set "# of executors" to 1 (let's just make it dead simple, no concurrency, less headache)
  - in the "Jenkins Location" section set Jenkins URL to "https://ceajenkins.ngrok.io"
    * (Jenkins might be smart enough to figure this out and has filled it in for you already)
  - scroll to "GitHub" section
  - click "Advanced"
  - dropdown "Manage additional GitHub actions", click "Convert login and password to token"
  - choose "From login and password", enter GitHub user and password, click "Create token credentials"
  - Click "Add GitHub Server"
    * Name: (leave blank)
    * Credentials: (choose the GitHub credentials auto-generated for your username)
    * click "Test connection" - expect this message: "Credentials verified for user <username>"
    * check "Override Hook URL"
    * enter hook url https://ceajenkins.ngrok.io
  - click "Save"

Next, we make sure all the required Jenkins plugins are installed

- open browser to http://ceajenkins.ngrok.io and log in
- click "Manage Jenkins" and then "Manage Plugins"
  - install the following plugin:
    * GitHub Pull Request Builder Plugin (https://github.com/jenkinsci/ghprb-plugin)

Next, we configure the GitHub Pull Request Builder plugin, following the instructions here: https://github.com/jenkinsci/ghprb-plugin

- open browser to http://ceajenkins.ngrok.io and log in
- click "Manage Jenkins" and then "Configure System"
- scroll down to the "GitHub Pull Request Builder" section
  - leave the GitHub Server API URL: `https://api.github.com`
  - set the Jenkins URL overrride: `https://ceajenkins.ngrok.io`
  - leave the Shared secret: (bunch of *'s… idk…)
  - select the credentials (This should be the GitHub auto generated token credentials you created above)
  - select Auto-manage webhooks

    - set the Admin list to the two lines `daren-thomas` and `JIMENOFONSECA`

- click Save

Finally, make sure Jenkins knows where to find `git.exe` - if it's not in `%PATH%`:

- open browser to https://ceajenkins.ngrok.io and log in
- click "Manage Jenkins" and then "Global Tool Configuration"
- set "Path to Git executable" to `C:\ProgramData\ceajenkins\ceatestall\cmder\vendor\git-for-windows\` `exe`

9.8.5 Configuration of the Jenkins items

First, we configure a Jenkins item for pull requests:

- open browser to https://ceajenkins.ngrok.io and log in
- click "New Item"
- Enter an item name: `run cea test for pull requests`
    - Choose "Freestyle project"
    - Project name: "run cea test for pull requests"
    - Description: "Check out the CityEnergyAnalyst, and run binceatest.bat"
    - check "Discard old builds"
        * Strategy: "Log Rotation"
        * Max # of builds to keep: 10
    - check "GitHub project"
    - Project url: "https://github.com/architecture-building-systems/CityEnergyAnalyst"
    - section "Source Code Management":
        * select "Git"
        * Repository URL: `https://github.com/architecture-building-systems/CityEnergyAnalyst.git`
        * Credentials: (add a new username/password credential)
        * Branches to build: `${ghprbActualCommit}`
    - section "Build Triggers":
        * check "GitHub Pull Request Builder"
        * GitHub API credentials: choose your credentials from the list
        * check "Use github hooks for build triggering"
        * click "Advanced"
        * List of organizations. Their members will be whitelisted: `architecture-building-systems`
    - section "Build"
        * Execute Windows batch command: `bin\ceatest.bat`
    - section "Build Environment"
        * select "Delete workspace before build starts"

Next, we configure a Jenkins item for merging to master:

- open browser to https://ceajenkins.ngrok.io and log in
- click "New Item"
- Enter an item name: `run cea test on merge to master`
  - Choose "Freestyle project"
  - Project name: "run cea test on merge to master"
  - Description: "Check out the CityEnergyAnalyst, and run binceatestall.bat"
  - check "Discard old builds"
    - Strategy: "Log Rotation"
    - Max # of builds to keep: 10
  - check "GitHub project"
  - Project url: "https://github.com/architecture-building-systems/CityEnergyAnalyst"
  - section "Source Code Management":
    - select "Git"
    - Repository URL: `https://github.com/architecture-building-systems/CityEnergyAnalyst.git`
    - Credentials: (use the ones created above)
    - Branches to build: `refs/heads/master`
  - section "Build Triggers":
    - check "GitHub hook trigger for GITScm polling"
    - check "Poll SCM"
  - section "Build"
    - Execute Windows batch command: `bin\ceatestall.bat`
  - section "Build Environment"
    - select "Delete workspace before build starts"
- open GitHub Webhooks
  - (NOTE: This should already be set up for the CEA Repository, but here's how to configure it just in case)
  - dropdown "Add webhook"
    - Payload URL: `http://ceajenkins.ngrok.io/git/notifyCommit?url=https://github.com/architecture-building-systems/CityEnergyAnalyst`
    - under "Which events would you like to trigger this webhook?" select "Let me select individual events."
    - select "Just the push event"

9.9 Running the CEA in Docker

Docker is an open-source project for automating the deployment of applications as portable, self-sufficient containers that can run on the cloud or on-premises. (Source)

The CEA can be run in docker. The main steps are:

- install Docker on your computer (out of scope of this document)
- build the image
- run the image

Note, that "CEA" in this context refers to the backend (server, cli) part of the CEA and not the GUI.

9.9.1 Building the image

Set server host to `0.0.0.0`. You can do it by editing `default.config` in the `cea` folder. In the `[server]` section, change the default address from `127.0.0.1` to `0.0.0.0`.

To build the docker image, navigate to CityEnergyAnalyst repository where the `Dockerfile` is located. Execute the following command:

```
> docker build -t dockeruser/cea:latest .
```

Notice the `.` at the end of the command - be sure to include it, as it tells `docker` where to find the `Dockerfile`.

The docker image should show up in your local computer:

```
> docker images
REPOSITORY                        TAG       IMAGE ID       CREATED            SIZE
dockeruser/cea                    latest    9963cd876a48   19 minutes ago    3.06GB
```

To share the docker image, push the image to Dockerhub:

```
> docker login
    username: dockeruser
    password:
> docker push dockeruser/cea:latest
```

9.9.2 Pull docker image

If you wish to use the latest cea image without building it on your own, you can pull it from our dockerhub.

To pull a docker image from Dockerhub:

```
> docker pull dockeruser/cea
```

9.9.3 Running the image in a new container

1. To test the docker image:

   ```
   > docker run --rm dockeruser/cea cea test
   ```

   - The `--rm` flag removes the container after it finishes running. This is useful when running the `cea test`

command so that the container does not persist after it exits after running the tests.

2. To run the docker container via shell (as the CEA Console):

```
> docker run -it -v /home/cea_projects:/projects dockeruser/cea /bin/bash
root@df9d4b16e5c0:> source /venv/bin/activate
(venv) root@df9d4b16e5c0:> cea --help
```

- The `-it` flag sets up interactive and tty so you can actually _do_ something there. Note, in order to use any of the CEA functionality, you'll need to type `source /venv/bin/activate`.
- The *-v /home/cea_projects:/projects* binds the folder `/home/cea_projects` in host to the folder `/projects` inside the container. Files saved in `/home/cea_projects` will be shared with the container.

3. To run `cea workflow`. First make sure the project folder and `workflow.yml` are in the correct path, in the example is `/home/cea_projects`.:

```
> docker run --name cea_container -v /home/cea_projects:/projects dockeruser/cea␣
↪cea workflow --workflow /projects/workflow.yml
```

4. To connect the GUI, CEA Dashboard, to a container

```
> docker run -t -p 5050:5050 dockeruser/cea
```

This command will start the CEA server and display it's output. You should see something like this:

```
City Energy Analyst version 3.24.0
Running `cea dashboard` with the following parameters:
- general:debug = False
  (default: False)
start socketio.run
```

There's quite a lot going on here and if this is seems daunting, I suggest reading up on some Docker tutorials - I don't understand it well enough myself to feel confident enough to explain. But here are some observations:

- The `-t` flag connects the container to your terminal, so you can see the output. You can drop this argument, but then you'll not be able to see any error messages etc. of the backend.
- The `-p 5050:5050` flag connects the port 5050 on the host machine (your computer) to the port 5050 in the container (an instance of the cea-server docker image).
- If you browse to http://localhost:5050/api/ you will see a description of the api you can use. This is the same api used by the CityEnergyAnalyst-GUI project, so you can essentially do anything that can be done in the GUI programmatically using this api.

9.10 How to clean up CEA git repository

This section describes the steps necessary to clean up the City Energy Analyst (CEA) git repository.

9.10.1 CEA Git Repository

Over the years, the CEA git repository has accumulated a lot of files in it's git history, many of which we do not use anymore. Because of this, the size of the repository to be around 1.7GB (as of December 2021) even though the size of the main source code is only around 200MB. This makes it really slow to clone the repository from GitHub.

9.10.2 Cleaning up the Repository

With the help of the git-filter-repo tool, we will remove large and/or old files, which are not relevant to the current version of CEA anymore, "permanently" from the git history.

Note: These steps are considered "permanent" only if the steps here are followed correctly. It is still possible for the changes to be reverted if the old git history is reintroduced (merged) into the new (clean) history by some way. In that case, we can redo these steps again to remove it.

Pre-flight Checks

1. **Make sure that you have backed up the existing CEA repository somewhere safe.**
   - This could be done by cloning the latest CEA repository to your local machine and copying folder somewhere else.
2. **Inform internal developers to complete and merge any branches that they are currently working on into the main branch.**
   - This is to ensure that local git branches (local machines) are not going to reintroduce any old history into the remote git repository (GitHub).
   - If for some reason that is not possible, one way to solve it is to do a rebase instead of a merge commit when merging the branch using GitHub Pull Requests. Read here if you want to know more about the difference.

Prerequisites (software)

- Have git installed and accessible through terminal.
  - For Windows, use the Git Bash Terminal that is installed with the Git for Windows installation.
- Install git-filter-repo tool.

Using the git-filter-repo tool

After all the necessary checks are done, we can run the git-filter-repo tool using 2 ways, manually or using a script.

Manually

1. Open a Terminal / Command Prompt.
2. Enter `cd PATH_TO_CEA_REPO`, replacing `PATH_TO_CEA_REPO` with the path of your local CEA git repository.
3. Enter `git filter-repo --invert-paths --paths-from-file ./bin/files_to_remove.txt`

Using Script *(Experimental)*

1. Open a Terminal
   - For Windows, use the Git Bash Terminal that is installed with the Git for Windows installation.

2. Enter `PATH_TO_CEA_REPO/bin/repo_cleanup.sh`, replacing `PATH_TO_CEA_REPO` with the path of your local CEA git repository.

We can then proceed to update these changes to GitHub

Updating git history of CEA repo on GitHub

To update the git history on GitHub, follow these steps from step 7.

9.10.3 Adding additional files to clean from history

If you want to remove other files from the history, other than the ones found in `bin/files_to_remove.txt`, add the path of the new lines to the file and re-run the tool as per above. Read this document for more information on how to use the git-filter-repo tool.

CHAPTER 10

API reference

10.1 cea package

exception cea.**ConfigError**

Bases: exceptions.Exception

Raised when the configuration of a tool contains some invalid values.

rc = 100

exception cea.**CustomDatabaseNotFound**

Bases: exceptions.Exception

Raised when the InputLocator can't find a user-provided database (region=='custom')

rc = 101

exception cea.**InvalidOccupancyNameException**

Bases: exceptions.Exception

Raised when the occupancy.dbf has an invalid / unknown occupancy column

rc = 104

exception cea.**MissingInputDataException**

Bases: exceptions.Exception

Raised when a script can't run because some information is missing

rc = 103

exception cea.**ScriptNotFoundException**

Bases: exceptions.Exception

Raised when an invalid script name is used.

rc = 102

cea.**suppress_3rd_party_debug_loggers**()

set logging level to WARN for fiona and shapely and others

10.1.1 Subpackages

cea.analysis package

Subpackages

cea.analysis.costs package

Submodules

cea.analysis.costs.equations module

cea.analysis.costs.system_costs module

cea.analysis.lca package

Submodules

cea.analysis.lca.embodied module

cea.analysis.lca.main module

cea.analysis.lca.operation module

cea.analysis.multicriteria package

Submodules

cea.analysis.multicriteria.main module

cea.datamanagement package

Submodules

cea.datamanagement.archetypes_mapper module

cea.datamanagement.constants module

This file contains the constants used in the data management tools

cea.datamanagement.create_new_scenario module

cea.datamanagement.data_initializer module

cea.datamanagement.data_migrator module

cea.datamanagement.databases_verification module

cea.datamanagement.schedule_helper module

cea.datamanagement.streets_helper module

cea.datamanagement.surroundings_helper module

cea.datamanagement.terrain_helper module

cea.datamanagement.weather_helper module

cea.datamanagement.zone_helper module

cea.demand package

Subpackages

cea.demand.schedule_maker package

Submodules

cea.demand.schedule_maker.schedule_maker module

Submodules

cea.demand.airconditioning_model module

cea.demand.building_properties module

cea.demand.calc_tm module

cea.demand.constants module

This file contains the constants used in the building energy demand calculations

cea.demand.control_heating_cooling_systems module

cea.demand.control_ventilation_systems module

cea.demand.datacenter_loads module

cea.demand.demand_main module

cea.demand.demand_writers module

cea.demand.electrical_loads module

cea.demand.hotwater_loads module

cea.demand.hourly_procedure_heating_cooling_system_load module

cea.demand.latent_loads module

cea.demand.rc_model_SIA module

cea.demand.rc_model_sia_cc module

cea.demand.refrigeration_loads module

cea.demand.sensible_loads module

cea.demand.space_emission_systems module

cea.demand.thermal_loads module

cea.demand.ventilation_air_flows_detailed module

cea.demand.ventilation_air_flows_simple module

cea.dev package

Submodules

cea.dev.build module

cea.examples package

Submodules

cea.examples.extract_reference_case module

cea.examples.template module

cea.interfaces package

Subpackages

cea.interfaces.cli package

Submodules

cea.interfaces.cli.cea_config module

cea.interfaces.cli.cea_dev module

cea.interfaces.cli.cea_doc module

cea.interfaces.cli.cli module

cea.interfaces.cli.dbf_to_excel module

cea.interfaces.cli.excel_to_dbf module

cea.interfaces.cli.excel_to_shapefile module

cea.interfaces.cli.list_demand_graphs_fields module

cea.interfaces.cli.shapefile_to_excel module

cea.interfaces.dashboard package

Subpackages

cea.interfaces.dashboard.api package

Submodules

cea.interfaces.dashboard.api.dashboard module

cea.interfaces.dashboard.api.databases module

cea.interfaces.dashboard.api.glossary module

cea.interfaces.dashboard.api.inputs module

cea.interfaces.dashboard.api.project module

cea.interfaces.dashboard.api.tools module

cea.interfaces.dashboard.api.utils module

cea.interfaces.dashboard.plots package

Submodules

cea.interfaces.dashboard.plots.routes module

cea.interfaces.dashboard.server package

Submodules

cea.interfaces.dashboard.server.jobs module

cea.interfaces.dashboard.server.streams module

Submodules

cea.interfaces.dashboard.dashboard module

cea.optimization package

Subpackages

cea.optimization.distribution package

Submodules

cea.optimization.distribution.network_optimization_features module

cea.optimization.master package

Submodules

cea.optimization.master.cost_model module

cea.optimization.master.crossover module

Crossover routines

class `cea.optimization.master.crossover.`**`CrossOverMethodsContinuous`**(*crossover_method*)

Bases: `object`

mutation methods for integers

`__init__`(*crossover_method*)

x.__init__(. . .) initializes x; see help(type(x)) for signature

`crossover`(*individual_1*, *individual_2*, *probability*)

class `cea.optimization.master.crossover.`**`CrossOverMethodsInteger`**(*crossover_method*)

Bases: `object`

mutation methods for integers

`__init__`(*crossover_method*)

x.__init__(. . .) initializes x; see help(type(x)) for signature

`crossover`(*individual_1*, *individual_2*, *probability*) cea.optimization.master.crossover.**crossover_main**(*ind1*, *ind2*, *indpb*, *column_names*, *heating_unit_names_share*, *cooling_unit_names_share*, *column_names_buildings_heating*, *column_names_buildings_cooling*, *district_heating_network*, *district_cooling_network*, *technologies_heating_allowed*, *technologies_cooling_allowed*, *crossover_method_integer*, *crossover_method_continuous*)

cea.optimization.master.data_saver module

cea.optimization.master.emissions_model module

cea.optimization.master.emissions_model.**calc_emissions_Whyr_to_tonCO2yr**(*E_Wh_yr*, *factor_kgCO2_to_MJ*)

cea.optimization.master.emissions_model.**calc_pen_Whyr_to_MJoilyr**(*E_Wh_yr*, *factor_MJ_to_MJ*)

cea.optimization.master.evaluation module

cea.optimization.master.generation module

Create individuals

cea.optimization.master.generation.**calc_building_connectivity_dict**(*building_names_all*, *building_names_heating*, *building_names_cooling*, *DHN_barcode*, *DCN_barcode*)

cea.optimization.master.generation.**generate_main**(*individual_with_names_dict*, *column_names*, *column_names_buildings_heating*, *column_names_buildings_cooling*, *district_heating_network*, *district_cooling_network*, *technologies_heating_allowed*, *technologies_cooling_allowed*)

Creates an individual configuration for the evolutionary algorithm. The individual is divided into four parts namely Heating technologies, Cooling Technologies, Heating Network and Cooling Network Heating Technologies: This block consists of heating technologies associated with % of the peak capacity each technology is going to supply, i.e. 10.1520.2030, which translates into technology 1 corresponding to 15% of peak capacity, technology 2 corresponding to 20% and technology 3 corresponding to 0%. 0% can also be just done by replacing 3 with 0. The technologies block is then followed by supply temperature of the DHN and the number of units it is supplied to among AHU, ARU, SHU. So if it is 6 degrees C supplied by DHN to AHU and ARU, it is represented as 6.02. The temperature is represented with 1 decimal point. Cooling Technologies: This follows the same syntax as heating technologies, but will be represented with cooling technologies. The block length of heating and cooling can be different. Heating Network: Network of buildings connected to centralized heating Cooling Network: Network of buildings connected to centralized cooling. Both these networks can be different, and will always have a fixed length corresponding to the total number of buildings in the neighborhood :param nBuildings: number of buildings :type nBuildings: int :return: individual: representation of values taken by the individual :rtype: list

cea.optimization.master.generation.**individual_to_barcode**(*individual*, *building_names_all*, *building_names_heating*, *building_names_cooling*, *column_names*, *column_names_buildings_heating*, *column_names_buildings_cooling*)

Reads the 0-1 combination of connected/disconnected buildings and creates a list of strings type barcode i.e. ("12311111123012") :param individual: list containing the combination of connected/disconnected buildings :type individual: list :return: indCombi: list of strings :rtype: list

cea.optimization.master.generation.**populate_individual**(*empty_individual_with_names_dict*, *name_share_conversion_technologies*, *technologies_allowed*, *columns_buildings_name*)

cea.optimization.master.master_main module

cea.optimization.master.master_to_slave module

cea.optimization.master.mutations module

Mutation routines

class cea.optimization.master.mutations.**MutationMethodContinuos**(*mutation_method*)

Bases: object

mutation methods for continuos variables

__init__(*mutation_method*)

x.__init__(...) initializes x; see help(type(x)) for signature

mutate(*individual*, *probability*)

class cea.optimization.master.mutations.**MutationMethodInteger**(*mutation_method*)

Bases: object

mutation methods for integers

__init__(*mutation_method*)

x.__init__(...) initializes x; see help(type(x)) for signature

mutate(*individual*, *probability*)

cea.optimization.master.mutations.**mutation_main**(*individual, indpb, column_names, heating_unit_names_share, cooling_unit_names_share, column_names_buildings_heating, column_names_buildings_cooling, district_heating_network, district_cooling_network, technologies_heating_allowed, technologies_cooling_allowed, mutation_method_integer, mutation_method_continuous*)

cea.optimization.master.normalization module

cea.optimization.master.normalization.**minmax_scaler**(*value, min_value, max_value*)

cea.optimization.master.normalization.**normalize_fitnesses**(*scaler_dict, fitnesses_population*)

cea.optimization.master.normalization.**scaler_for_normalization**(*number_of_objectives, fitnesses*)

cea.optimization.master.performance_aggregation module

cea.optimization.master.summarize_network module

cea.optimization.master.validation module

Validation

cea.optimization.master.validation.**validation_main**(*individual_with_name_dict, column_names_buildings_heating, column_names_buildings_cooling, district_heating_network, district_cooling_network, technologies_heating_allowed, technologies_cooling_allowed*)

cea.optimization.preprocessing package

Submodules

cea.optimization.preprocessing.decentralized_building_main module

cea.optimization.preprocessing.decentralized_buildings_cooling module

cea.optimization.preprocessing.decentralized_buildings_heating module

cea.optimization.preprocessing.preprocessing_main module

cea.optimization.preprocessing.processheat module

cea.optimization.slave package

Subpackages

cea.optimization.slave.seasonal_storage package

Submodules

cea.optimization.slave.seasonal_storage.Import_Network_Data_functions module

cea.optimization.slave.seasonal_storage.SolarPowerHandler_incl_Losses module

cea.optimization.slave.seasonal_storage.design_operation module

cea.optimization.slave.seasonal_storage.storage_main module

Submodules

cea.optimization.slave.cooling_main module

cea.optimization.slave.cooling_resource_activation module

cea.optimization.slave.electricity_main module

cea.optimization.slave.heating_main module

cea.optimization.slave.heating_resource_activation module

cea.optimization.slave.natural_gas_main module

cea.optimization.slave.test module

`cea.optimization.slave.test.main()`

Submodules

cea.optimization.constants module

This file contains the constants used in objective function calculation in optimization

cea.optimization.lca_calculations module

cea.optimization.optimization_main module

cea.optimization.prices module

cea.optimization.slave_data module

cea.plots package

Subpackages

cea.plots.comparisons package

Submodules

cea.plots.comparisons.Annual_costs module

cea.plots.comparisons.Annual_emissions module

cea.plots.demand package

Submodules

cea.plots.demand.comfort_chart module

cea.plots.demand.energy_balance module

cea.plots.demand.energy_end_use module

cea.plots.demand.energy_end_use_intensity module

cea.plots.demand.energy_final_use module

cea.plots.demand.energy_use_intensity module

cea.plots.demand.heating_reset_schedule module

cea.plots.demand.load_curve module

cea.plots.demand.load_curve_supply module

cea.plots.demand.load_duration_curve module

cea.plots.demand.load_duration_curve_supply module

cea.plots.demand.peak_load module

cea.plots.demand.peak_load_supply module

cea.plots.optimization package

Submodules

cea.plots.optimization.a_pareto_curve module

cea.plots.optimization.b_parallel_coordinates module

cea.plots.optimization.c_annual_costs module

cea.plots.optimization.d_annual_emissions module

cea.plots.optimization.e_Investment_costs module

cea.plots.optimization.f_paretocurve_convergence module

cea.plots.solar_potential package

Submodules

cea.plots.solar_potential.a_solar_radiation module

cea.plots.supply_system package

Submodules

cea.plots.supply_system.a_supply_system_map module

cea.plots.supply_system.b_installed_capacities module

cea.plots.supply_system.c_requirements_curve_electricity module

cea.plots.supply_system.d_dispatch_curve_electricity module

cea.plots.supply_system.e_dispatch_curve_heating_plant module

cea.plots.supply_system.f_dispatch_curve_cooling_plant module

cea.plots.supply_system.g_grid_ramping_capacity module

cea.plots.technology_potentials package

Submodules

cea.plots.technology_potentials.a_photovoltaic_potential module

cea.plots.technology_potentials.b_photovoltaic_thermal_potential module

cea.plots.technology_potentials.c_solar_collector_ET_potential module

cea.plots.thermal_networks package

Submodules

cea.plots.thermal_networks.a_network_design module

cea.plots.thermal_networks.b_demand_curve module

cea.plots.thermal_networks.c_annual_energy_consumption module

cea.plots.thermal_networks.d_energy_loss_bar module

cea.plots.thermal_networks.e_heating_reset_curve module

cea.plots.thermal_networks.f_pump_duration_curve module

Submodules

cea.plots.base module

cea.plots.cache module

cea.plots.categories module

cea.plots.colors module

cea.plots.plot_cli module

cea.plots.variable_naming module

cea.resources package

Subpackages

cea.resources.radiation_daysim package

Submodules

cea.resources.radiation_daysim.daysim_main module

cea.resources.radiation_daysim.geometry_generator module

cea.resources.radiation_daysim.radiance module

cea.resources.radiation_daysim.radiation_main module

cea.resources.radiation_daysim.visualization module

Submodules

cea.resources.geothermal module

cea.resources.natural_gas module

natural gas

`cea.resources.natural_gas.`**`calc_Cinv_gas`**(*PnomGas*)
Calculate investment cost of natural gas connections.

Parameters **`PnomGas`** (*`float`*) – peak natural gas supply in [W]

Returns InvCa

Rtype InvCa

cea.resources.sewage_heat_exchanger module

cea.resources.water_body_potential module

cea.technologies package

Subpackages

cea.technologies.network_layout package

Submodules

cea.technologies.network_layout.connectivity_potential module

cea.technologies.network_layout.main module

cea.technologies.network_layout.minimum_spanning_tree module

cea.technologies.network_layout.steiner_spanning_tree module

cea.technologies.network_layout.substations_location module

cea.technologies.network_layout.utility module

Shapefile

Generates a networkx.DiGraph from point and line shapefiles.

"The Esri Shapefile or simply a shapefile is a popular geospatial vector data format for geographic information systems software. It is developed and regulated by Esri as a (mostly) open specification for data interoperability among Esri and other software products." See https://en.wikipedia.org/wiki/Shapefile for additional information.

`cea.technologies.network_layout.utility.`**`read_shp`**(*path*, *simplify=True*, *geom_attrs=True*, *strict=True*)

Generates a networkx.DiGraph from shapefiles. Point geometries are translated into nodes, lines into edges. Coordinate tuples are used as keys. Attributes are preserved, line geometries are simplified into start and end coordinates. Accepts a single shapefile or directory of many shapefiles.

"The Esri Shapefile or simply a shapefile is a popular geospatial vector data format for geographic information systems software[1]."

Parameters

- **path** (*str*) – File, directory, or filename to read.
- **simplify** (*bool*) – If True, simplify line geometries to start and end coordinates. If False, and line feature geometry has multiple segments, the non-geometric attributes for that feature will be repeated for each edge comprising that feature.
- **geom_attrs** (*bool*) – If True, include the Wkb, Wkt and Json geometry attributes with each edge. NOTE: if these attributes are available, write_shp will use them to write the geometry. If nodes store the underlying coordinates for the edge geometry as well (as they do when they are read via this method) and they change, your geomety will be out of sync.
- **strict** (*bool*) – If True, raise NetworkXError when feature geometry is missing or GeometryType is not supported. If False, silently ignore missing or unsupported geometry in features.

Returns the NetworkX graph

Raises

- **ImportError** – If ogr module is not available.
- **RuntimeError** – If file cannot be open or read.
- **NetworkXError** – If strict=True and feature is missing geometry or GeometryType is not supported.

`cea.technologies.network_layout.utility.`**`write_shp`**(*G*, *outdir*)

Writes a networkx.DiGraph to two shapefiles, edges and nodes. Nodes and edges are expected to have a Well Known Binary (Wkb) or Well Known Text (Wkt) key in order to generate geometries. Also acceptable are nodes with a numeric tuple key (x,y).

"The Esri Shapefile or simply a shapefile is a popular geospatial vector data format for geographic information systems software[2]."

:param str outdir : directory path, Output directory for the two shapefiles. :rtype: None

Examples:

```
nx.write_shp(digraph, '/shapefiles') # doctest +SKIP
```

[1] https://en.wikipedia.org/wiki/Shapefile
[2] https://en.wikipedia.org/wiki/Shapefile

cea.technologies.solar package

Submodules

cea.technologies.solar.constants module

Parameters used for solar technologies

cea.technologies.solar.photovoltaic module

cea.technologies.solar.photovoltaic_thermal module

cea.technologies.solar.solar_collector module

cea.technologies.thermal_network package

Submodules

cea.technologies.thermal_network.simplified_thermal_network module

cea.technologies.thermal_network.substation_matrix module

cea.technologies.thermal_network.thermal_network module

cea.technologies.thermal_network.thermal_network_costs module

cea.technologies.thermal_network.thermal_network_loss module

Hydraulic - thermal network

cea.technologies.thermal_network.thermal_network_loss.**calc_temperature_out_per_pipe**(*t_in*, *m*, *k*, *t_ground*)

Parameters

- **t_in** – in Kelvin
- **m** – in kg/s
- **k** – in kW/K
- **t_ground** – in Kelvin

Returns

cea.technologies.thermal_network.thermal_network_optimization module

Submodules

cea.technologies.blinds module

blinds

`cea.technologies.blinds.calc_blinds_activation(radiation, g_gl, Rf_sh)`

This function calculates the blind operation according to ISO 13790.

Parameters

- **radiation** – radiation in [W/m2]
- **g_gl** – window g value
- **Rf_sh** – shading factor

cea.technologies.boiler module

cea.technologies.burner module

cea.technologies.chiller_absorption module

cea.technologies.chiller_vapor_compression module

cea.technologies.cogeneration module

cea.technologies.constants module

Constants used throughout the cea.technologies package.

History lesson: This is a first step at removing the *cea.globalvars.GlobalVariables* object.

cea.technologies.cooling_tower module

cea.technologies.direct_expansion_units module

cea.technologies.furnace module

cea.technologies.heat_exchangers module

cea.technologies.heating_coils module

cea.technologies.heatpumps module

cea.technologies.pumps module

cea.technologies.radiators module

cea.technologies.storage_tank module

cea.technologies.storage_tank_pcm module

cea.technologies.storagetank_cc module

cea.technologies.substation module

cea.technologies.supply_systems_database module

cea.technologies.tabs module

cea.technologies.thermal_storage module

cea.tests package

Subpackages

cea.tests.datamanagement package

Submodules

cea.tests.datamanagement.test_zone_helper module

Submodules

cea.tests.create_unittest_data module

cea.tests.run_all_plots module

cea.tests.run_unit_tests module

cea.tests.test_calc_thermal_loads module

cea.tests.test_check_for_radiation_input_in_demand_script module

cea.tests.test_chiller_vapor_compression module

cea.tests.test_config module

cea.tests.test_dbf module

cea.tests.test_inputlocator module

cea.tests.test_inputs_setup_workflow module

cea.tests.test_plots module

cea.tests.test_schedules module

cea.tests.test_schemas module

cea.tests.test_technologies module

cea.tests.trace_inputlocator module

cea.utilities package

class cea.utilities.**devnull**

Bases: object

Suppress sys.stdout so that it goes to devnull for duration of the with block

__enter__()

__exit__(*exc_type*, *exc_val*, *exc_tb*)

__init__()

x.__init__(...) initializes x; see help(type(x)) for signature

write(_)

cea.utilities.**identifier**(*s*, *sep='-'*)

First, all characters are lowercased, then, any character that is not in ascii_lowercase is replaced with sep.

Parameters

- **s** (*str*) – the string to create an identifier of
- **use_underscores** (*str*) – if set to true, underscores ("_") will be used instead of dashes ("-")

Return type str

class cea.utilities.**pushd**(*path*)

Bases: object

Manage an os.chdir so that at the end of a with block, the path is set back to what it was

__enter__()

__exit__(*exc_type*, *exc_val*, *exc_tb*)

__init__(*path*)

x.__init__(...) initializes x; see help(type(x)) for signature

cea.utilities.**remap**(*x*, *in_min*, *in_max*, *out_min*, *out_max*)

Scale x from range [in_min, in_max] to [out_min, out_max] Based on this StackOverflow answer: https://stackoverflow.com/a/43567380/2260

cea.utilities.**simple_memoize**(*obj*)

cea.utilities.**unique**(*sequence*)

Return only the unique elements in sequence, preserving order.

Parameters **sequence** (*Sequence[T]*) – the sequence to unique-ify

Return type List[T]

Submodules

cea.utilities.color_fader module

cea.utilities.compile_pyd_files module

Compile the .pyd files using Numba pycc to speed up the calculation of certain modules. Currently used for:

- calc_tm.pyd (used in demand/sensible_loads.py)
- calc_radiator.pyd (used in technologies/radiators.py)

In order to run this script, you will need to install Numba. Try: *conda install numba*

`cea.utilities.compile_pyd_files.`**`compile_radiators`**`()`

`cea.utilities.compile_pyd_files.`**`compile_rc_model_sia`**`()`

`cea.utilities.compile_pyd_files.`**`compile_storagetank`**`()`

`cea.utilities.compile_pyd_files.`**`copy_pyd`**`(`*source*, *destination*`)`

`cea.utilities.compile_pyd_files.`**`delete_pyd`**`(`**pathspec*`)`
Delete the file with the pathspec. *pathspec* is an array of path segments.

`cea.utilities.compile_pyd_files.`**`main`**`()`

cea.utilities.create_mixed_use_type module

cea.utilities.create_polygon module

cea.utilities.date module

cea.utilities.dbf module

cea.utilities.doc_glossary module

cea.utilities.doc_graphviz module

cea.utilities.doc_html module

cea.utilities.doc_schemas module

cea.utilities.epwreader module

cea.utilities.latin_hypercube module

cea.utilities.parallel module

Standardizes multiprocessing use. In the CEA, some functions are run using the standard `multiprocessing` library. They are run by `map``ing the function to a list of arguments (see ``multiprocessing.Pool.map_async`) and waiting for the processes to finish, while at the same time piping STDOUT, STDERR through `cea.utilities.workerstream.QueueWorkerStream` - this ensures that the dashboard interface can read the output from the sub-processes.

The way this was done in CEA < v2.23 included boiler plate code that needed to be repeated every time multiprocessing was used. Issue [#2344](https://github.com/architecture-building-systems/CityEnergyAnalyst/issues/2344) was a result of not applying this technique to the demand script.

This module exports the function *map* which is intended to replace both `map_async` and the builtin `map` function (which was used when `config.multiprocessing == False`). This simplifies multiprocessing.

`cea.utilities.parallel.`**`__apply_func_with_worker_stream`**(*args*)
Call func, using `queue` to redirect stdout and stderr, with a tuple of args because multiprocessing.Pool.map only accepts one argument for the function.

This function is called _inside_ a separate process.

`cea.utilities.parallel.`**`__multiprocess_wrapper`**(*func*, *processes*, *on_complete*)
Create a worker pool to map the function, taking care to set up STDOUT and STDERR

`cea.utilities.parallel.`**`single_process_wrapper`**(*func*, *on_complete*)
The simplest form of vectorization: Just loop

`cea.utilities.parallel.`**`test`**(*a*, *b*)

`cea.utilities.parallel.`**`vectorize`**(*func*, *processes=1*, *on_complete=None*)
Similar to `numpy.vectorize`, this function wraps `func` so that it operates on sequences (of same length) of inputs and outputs a sequence of results, similar to `map(func, *args)`.

The main point of using `vectorize` is to unify single-processing with multi-processing - if processes > 1, then multiprocessing is used and the function will be run on a pool of processes. STDOUT and STDERR of these processes are fed through a `cea.workerstream.QueueWorkerStream` so it can be shown in the dashboard job output.

The parameter `on_complete` is an optional callable that is called for each completed call of `func`. It takes 4 arguments:

- i: the 0-based order in which this call was completed
- n: the total number of function calls to be made
- args: the arguments passed to this call to `func`
- result: the return value of this call to `func`

Parameters

- **`func`** – The function to vectorize
- **`processes`** (*`int`*) – The number of processes to use (use `config.get_number_of_processes()`)
- **`on_complete`** – An optional function to call for each completed call to `func`.

cea.utilities.physics module

Physical functions

`cea.utilities.physics.`**`calc_rho_air`**(*temp_air*)
Calculation of density of air according to 6.4.2.1 in [1]

temp_air : air temperature in (°C)

rho_air : air density in (kg/m3)

cea.utilities.physics.**kelvin_to_fahrenheit**(*T_Kelvin*)

cea.utilities.rename_building module

cea.utilities.reporting module

cea.utilities.schedule_reader module

cea.utilities.solar_equations module

cea.utilities.standardize_coordinates module

cea.utilities.workerstream module

This file implements `WorkerStream` for capturing stdout and stderr.

class cea.utilities.workerstream.**HttpWorkerStream**(*name*, *jobid*, *url*)

Bases: `object`

__init__(*name*, *jobid*, *url*)

x.__init__(. . .) initializes x; see help(type(x)) for signature

class cea.utilities.workerstream.**QueueWorkerStream**(*name*, *q*)

Bases: `object`

File-like object for wrapping the output of the scripts with queues - to be created in child process

__init__(*name*, *q*)

x.__init__(. . .) initializes x; see help(type(x)) for signature

__repr__() <==> *repr(x)*

close()

flush()

isatty()

write(*str*)

class cea.utilities.workerstream.**WorkerStream**(*name*, *connection*)

Bases: `object`

File-like object for wrapping the output of the scripts into connection messages

__init__(*name*, *connection*)

x.__init__(. . .) initializes x; see help(type(x)) for signature

__repr__() <==> *repr(x)*

close()

flush()

isatty()

write(*str*)

cea.utilities.workerstream.**stream_from_queue**(*q*)

Stream the contents from the queue to STDOUT / STDERR - to be called from parent process

cea.workflows package

Submodules

cea.workflows.workflow module

10.1.2 Submodules

10.1.3 cea.api module

10.1.4 cea.config module

10.1.5 cea.constants module

This file contains the constants used in many folders in CEA. IF few constants are only used in a subfolder, it is highly recommended to keep those constants in a separate file in the subfolder. This is to make sure we declare the constants closest to the point of usage.

10.1.6 cea.glossary module

10.1.7 cea.inputlocator module

10.1.8 cea.plugin module

10.1.9 cea.schemas module

10.1.10 cea.scripts module

10.1.11 cea.worker module

Python Module Index

Index

Symbols

C

D

F

G

H

I

K

M