Getting Started Guide: E-hub Modeling Tool (v5.4)

This guide is intended to help new users get started with the e-hub modeling tool.

# Requirements

- Python 3.6+

- Python libraries: pandas, numpy, Pyomo

- Solver supported by Pyomo (e.g., glpk, gurobi); note: current model tested/configured with gurobi only

- Spreadsheet editor (e.g., Microsoft Excel, OpenOffice)

# Installation Quick Start

**Python:**

* An easy was to get started with Python is by installing the Anaconda package from: <https://www.anaconda.com/download/>
* Download and install the appropriate Python version 3.6+ for your machine

**Libraries:**

* Launch Anaconda Prompt
* Enter the following commands to install pandas, numpy, and Pyomo libraries, respectively (may require administrator access):
  1. “conda install pandas”
  2. “conda install numpy”
  3. “pip install pyomo”

**Solver:**

* To install the glpk solver, enter the command: “conda install -c conda-forge glpk” (using Anaconda Prompt)

**OpenOffice:**

* Download and install the appropriate OpenOffice software package for your machine from: <https://www.openoffice.org/download/>

# Defining a Model

The energy hub model is defined using a spreadsheet. The input spreadsheet template is located under the “cases” GitHub folder. Several worksheets are used to define the model. These are briefly described below. Further information regarding the input fields and their specification is provided in the comments on individual cells within the worksheets.

Note that the complete optimization problem is defined and documented in the “E-hub optimization problem.docx” file under the GitHub “docs” folder.

**Worksheet Descriptions:**

*General:* Overarching optimization problem and modeling parameters

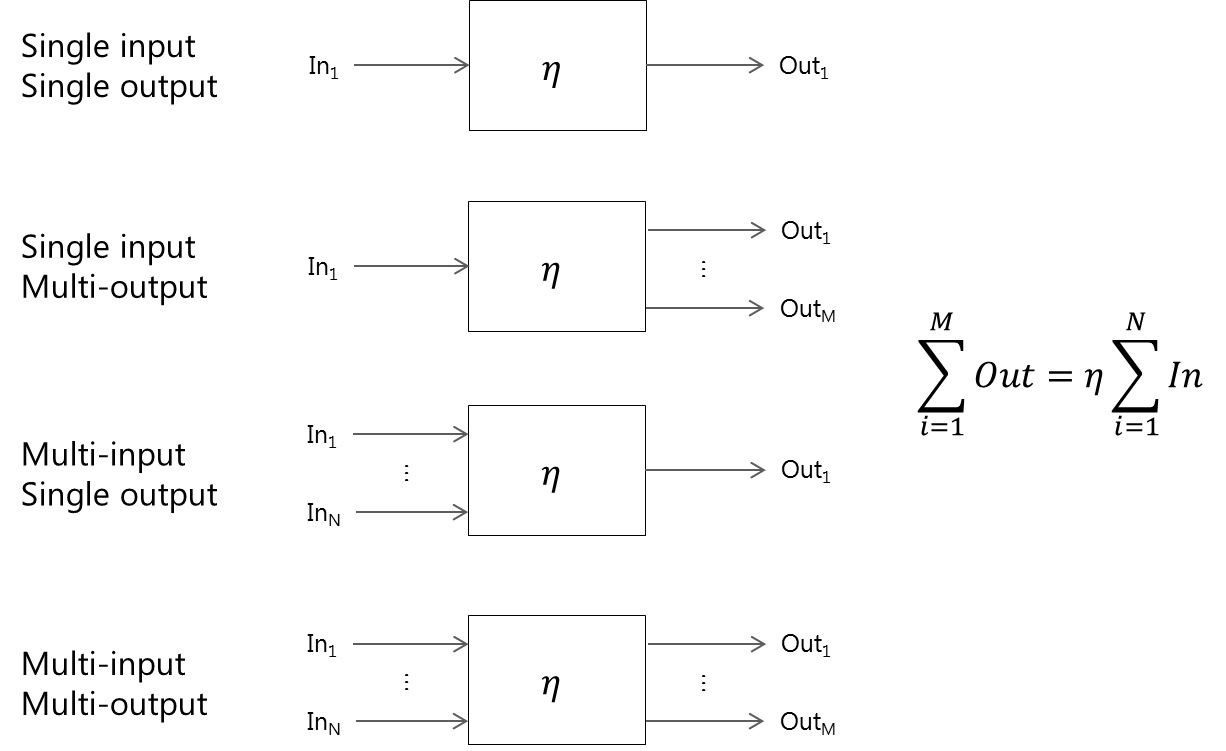
*Energy Carriers:* Complete list of energy carriers referenced in the model

*Imports:* Imported energy carriers; note that every model requires at least one import in order to satisfy energy balance equations.

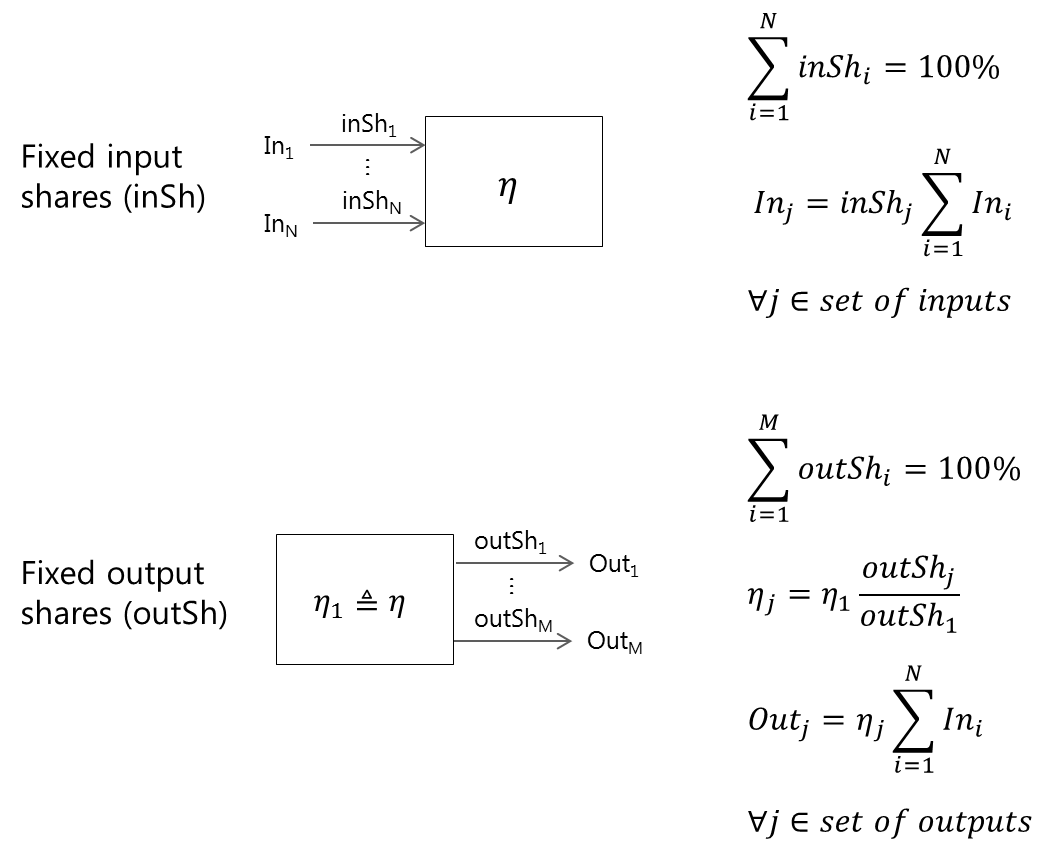
*Exports:* Exported energy carriers (if any)

*Demand:* Energy carrier demand by hub

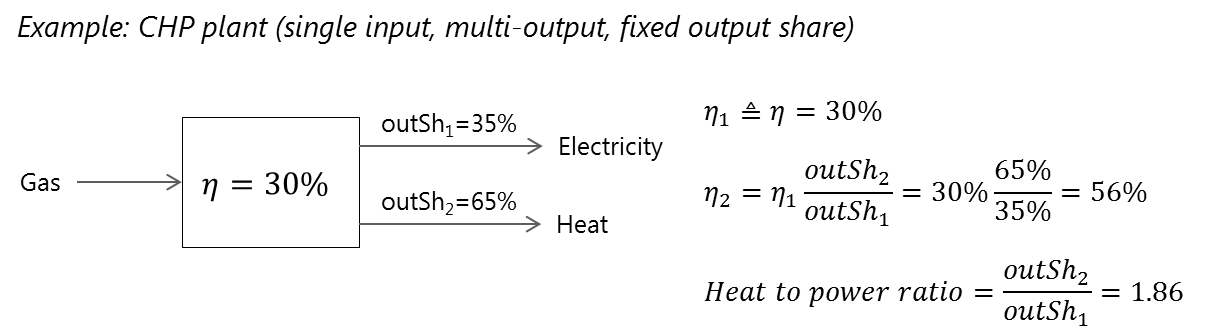
*Energy Converters:* Energy conversion technologies; at least one must be specified. Note that the following general types of technologies can be specified:



Here, η represents the efficiency of the conversion process. The energy balance equation for technologies is also described above. Fixed input and output shares may also be specified for multi input/output technologies according to the following relationships:



For example, a CHP plant can be specified as a single input, multi-output, fixed output share technology as follows:



*Storage:* Storage technologies (if any)

*Network:* Network connections (if any)

*Solar Profile:* Solar irradiation profile and energy carrier (if any)

Note that custom modeling equations can be defined by a user in Python within the “mod\_custom.py” file under the “code” GitHub folder.

# Running a Model

* Download the latest e-hub tool version from GitHub (<https://github.com/hues-platform/python-ehub/tree/NextGen>)
* The entire e-hub model is stored within the “code” folder
* The main run file is “main.py”
* The model can be run from a Python IDE (integrated development environment), such as Spyder (included in the Anaconda package)
* Note: modify the “run\_solve.py” file in order to use a solver other than gurobi. (This file is currently configured for gurobi only.)

1. Launch Spyder
2. Open the file “main.py”
3. Edit the input spreadsheet .xls file path (input\_path)
4. Edit the save .txt file paths (result\_file and param\_file)
5. Save and run “main.py”
6. If executed successfully, results and modeling parameters are saved to the specified .txt files

# Viewing Results

* Results are saved to the specified text file
* Note that the e-hub tool numbers technologies, storages, network links and energy carriers from 1:N in the order that they appear in the input file

A macro-enabled spreadsheet utilizing pivot tables has been developed to ease viewing and navigation of the results. This spreadsheet is located under the “results” GitHub folder. Note that this file is under development and has not been tested on large import files, but if you would like to use it:

* Navigate to the “Setup” worksheet
* Click on the “Get File” button. This button executes a macro which will import your results. You will be prompted to provide two files:
  1. Select the .xls input file associated with your run as the first file. This file is used to relabel the numeric indices of the result variables as the data is imported. (Numeric indices are overwritten with the original naming conventions provided in the input spreadsheet (e.g., for technologies, energy carriers, storage, etc.).)
  2. Select the .txt file associated with the results of your run as the second file. The data will then be imported (with relabeled indices according to file #1). The raw data for each variable under “Variable Name” will be imported to the specified worksheet under “Import Worksheet”.
* Pivot tables have been predefined under the remaining worksheets (e.g., “Ein”, “Eout”, “CapTech”, etc.). These tables should be automatically refreshed via the macro. You may view the data via these pivot tables and manipulate them (or create new ones) as needed.