

Ecosim - V0.3



USER MANUAL

PREPARED FOR:

Bonneville Power Administration

PREPARED BY:

Paul Kintner

JUNE 2, 2022

TABLE OF CONTENTS

INTRODUCTION	3
INSTALLATION	3
<i>Hardware Requirements</i>	3
<i>Installation</i>	4
RUNNING ECOSIM	4
<i>Command Line</i>	4
<i>Folder Structure</i>	4
INPUT STRUCTURE	4
<i>Climates</i>	5
<i>Cohorts</i>	5
<i>Cases</i>	5
OUTPUT FILES	5
MODELKIT TIPS	6
<i>Skipping Runs</i>	6
<i>Parallel Processing</i>	6
CHPWH SYSTEMS	6
<i>Supported HPWH Models</i>	7
INPUT PARAMETERS	9
<i>Building Level Inputs</i>	9
run_name	9
weather_file	9
num_bedrooms	9
floor_areas	9
num_units	9
weather_mains_model	10
weather_dt_amb_max	10
weather_temp_offset	11
<i>CHPWH System Level Inputs</i>	11
temperature_maintenance_loop_config	11
system_type	11
system_supply_temp	11
<i>Primary Heater Inputs</i>	11
primary_heater_type	11
primary_heater_hpwh_unit	12
primary_heater_volume	12
primary_heater_elec_resistance_capacity_1	12
primary_heater_elec_resistance_capacity_2	12
primary_heater_gas_heating_capacity	12
primary_heater_multiplier	12
primary_heater_location	12
primary_heater_hpwh_source_air	13
primary_heater_setpoint	13
primary_tank_per_heater	13
primary_heater_hpwh_insul_r	13
<i>Loop Heater Inputs</i>	13
loop_heater_config	13
loop_heater_type	14
loop_heater_hpwh_unit	14
loop_heater_volume	14
loop_heater_elec_resistance_capacity_1	14
loop_heater_elec_resistance_capacity_2	14
loop_heater_gas_heating_capacity	15

loop_heater_multiplier	15
loop_heater_location	15
loop_heater_hpwh_source_air	15
loop_heater_setpoint	15
loop_tank_per_heater	15
loop_heater_hpwh_insul_r	15
<i>Draw Inputs</i>	16
kitchen_sink_flow	16
bathroom_faucet_flow	16
shower_flow	16
dishwasher_cycle	17
clotheswasher_cycle	17
shower_day_waste_fraction	17
faucet_day_waste_fraction	17
distribution_system_multiplier	18
draws_shuffle_seed	18
<i>Recirculation Loop Inputs</i>	18
loop_calc_method	18
loop_pump_power	18
loop_flow	19
loop_location	19
loop_loss_rate_per_apartment	19
loop_branch_length	19
loop_branch_number	19
bldg_num_floors	20
bldg_floor_height	20
<i>Buffer Space Inputs</i>	20
buffer_space_underground_config	20
buffer_number_of_floors	21
<i>Autosizing Inputs</i>	21
sizing_duration	21
sizing_storage_efficiency	21
<i>Additional Export Inputs</i>	21
export_minute_day_start	21
export_minute_day_end	21
add_meter_energy_exports	21
add_dhw_meter_flow_exports	22
add_subhourly_exports	22
ADVANCED INPUT STRUCTURE	22
<i>Supporting Functions</i>	23
<i>Specific HPWH data variables</i>	24
OUTPUT PARAMETERS	26
KNOWN ISSUES	28

Introduction

Ecosim is a standalone energy analysis tool focused specifically on multifamily Commercial Heat Pump Water Heater (CHPWH) systems. It uses the same modeling engine for HPWHs in California Title 24 Modeling Software: the California Simulation Engine (CSE)¹. This provides utilities, manufacturers, designers, and regulators a tool to analyze how their specific CHPWH system performs against a baseline, for a wide range of design choices, including:

- Specific heat pump technology
- Temperature maintenance and piping configuration
- Back-up technology
- Climate zone

System designers will be able to investigate the impact of the above design choices on:

- Predicted annual energy use
- Electric load
- System COP

CSE uses the software library HPWHsim² to model HPWHs and electric resistance water heaters. A strong advantage of Ecosim is that HPWHsim models HPWHs with performance maps that are derived from laboratory data or been submitted by the manufacturers to the California Energy Commission for inclusion in the Title 24 modeling software.

There is a brief discussion on CHPWH systems, but this manual is largely designed for those with an understanding of how CHPWH systems are set up and operate. For further reading on CHPWH systems, review the latest version of the NEEA Advanced Water Heater Specification (AWHS)³.

This version of the Ecosim tool was funded by the Bonneville Power Administration. It is not intended for widespread distribution or use by people unfamiliar with the tool. This version was created to develop the basic engine which will be used to create a user-friendly web-accessible tool expected to be released at the end of 2022 with funding from Southern California Edison. This version of the tool will also be used to model system performance for populating the Northwest Energy Efficiency Alliance's Advanced Water Heater Specification v8.

Installation

Ecosim releases can be downloaded here:

<https://bitbucket.org/ecotope-research/ecosim/downloads/?tab=tags>

Hardware Requirements

The hardware requirements are limited by CSE which is a 32-bit Microsoft Windows console application. This should be able to run on Windows Vista, Windows 7, Windows 8, and Windows 10.

¹ <https://github.com/cse-sim/cse>, CSE Copyright (c) 1997-2016 The CSE Authors. All rights reserved.

² <https://github.com/ECotopeResearch/HPWHsim>

³ <https://neea.org/our-work/advanced-water-heating-specification>

Installation

The Ecosim package probably came to you in a zipped file. Unzip and store the Ecosim package in a directory named \Ecosim. Next download Modelkit Catalyst⁴ (hereafter Modelkit) from this link:

<https://share.bigladdersoftware.com/files/modelkit-catalyst-0.7.0.exe>

During Modelkit installation only the Command-Line Tools are needed for running the Ecosim.

Running Ecosim

Command Line

Ecosim is currently run through the templating engine Modelkit, which uses the fully featured Ruby programming language. Modelkit will read and parametrically combine the input files to create individual runs.

To run Ecosim, one must use the command line; on Windows this can be done by searching for “Command Prompt” from the command line; alternatively hold shift and right click in the analysis folder then select “Open PowerShell window here.”

From the command line, navigate to an analysis directory of interest (i.e., /analysis/mf88/) - there should be a rakefile.rb in the folder. To run the models, use the command:

```
> modelkit rake
```

To aggregate annual results use the command:

```
> modelkit rake results
```

This creates a summary file, “results-summary.csv”, with annual results for each run.

Model files and run files can be removed for recreating templates and results files, which can be necessary when debugging issues:

```
> modelkit rake clean
```

Folder Structure

In the root directory of the Ecosim exist important data directories such as /weather, /draws, and /hpwhs. The /weather directory is where Ecosim will look for weather files; more can be added here without interference. The /draws, /hpwhs, and /templates directories contain sensitive data that should not be moved or changed to maintain proper functionality of Ecosim.

Any new suite of runs should be created as a new directory in the /analysis directory and need to contain the rakefile.rb. This could be completed by copying the /analysis/new_project directory in the analysis folder and renaming it; this directory contains all the necessary parts for running Ecosim.

Input Structure

The inputs are contained within two main CSV files (climates.csv and cohorts.csv) and in multiple other possible files in the /analysis/*/cases directory.

⁴ <https://bigladdersoftware.com/projects/modelkit>

In general, every cohort defines a building and is run for each case in the cohort's case csv file in the `/analysis/*/cases` directory. Each combination of cohort and associated cases is run for every climate defined in `climates.csv`.

Every row of the input files has a skip option. That skips the climate, cohort, or case if “#” is entered.

Parameters used directly in templating in the Ecosim are prefixed with “:” (i.e. `:weather_file`).

The zipped Ecosim package comes with an example set of runs in the `/analysis/mf88/` directory to help guide users about how the input structure works. New projects could copy and rename this folder to begin modeling.

Climates

The climate csv contains the paths and shorthand names for each climate zone to run.

The “climate” column is a shorthand name for the climate and is used to create the output directory for each climate run in the `/analysis/*/runs` directory.

The “climate_zone” column is merely a descriptive name for the climate zone.

The “:weather_file” column is the name of the weather file in the `/weather` directory.

Cohorts

The “cohorts.csv” is typically used to define construction parameters, like the building size in apartment units and size of apartment units.

The root column contains the root template file – this has been specifically set for Ecosim.

The cohort column contains the names of the cases to be run. These are the names of the csv files in the `/cases` directory.

Cases

Each csv in the `/cases` directory defines a specific set of runs for each cohort pointing to it. This is where more detailed system level investigations are good to define, such as the system type, configuration, or presence of a temperature maintenance loop.

The `case_name` is used to assign a name to each case or row. This is used to identify the output directory. The next columns are used to assign parameters for specific run.

Output Files

Ecosim automatically aggregates results from each case run in a `results-summary.csv` file in the `/analysis/*/` directory. These results include annual sums of energy use, energy delivered, and volume of hot water delivered, and COP for the various components of the CHPWH system and the system as a whole.

When running the Ecosim detailed output files are created in subfolders deep in the `/analysis/*/runs` directory of the project. The path for each individual model run follows `/analysis/*/runs/climate/cohorts/case_name/`.

Modelkit creates a `instance.cse` file in each of the folders, which is the run file for CSE. Notably all of the parameters used for the run are listed in the beginning of each file, useful for debugging. Modelkit also

writes out instance-inputs.csv, instance-parameters.csv, and instance-rules.csv which provide similar information about the parameters for an individual run.

CSE also writes out numerous output files, most importantly for debugging issues. The *.err file will record errors and warnings message from CSE that could be missed from the Modelkit command line output.

Since Modelkit runs the cases in parallel as a default, the command line output is directed into a file “stdout” for each run, which can be viewed in any plain text editor.

Of most interest to the advanced user is the “EXPORTANNUAL.csv”, “EXPORTSMONTHLY.csv”, and “EXPTHOURLY*.csv” files. If specified by the inputs, the minute timestep output files will also be written out here separately for the primary heater (named: *dhwtr1-t24-CHPWH_hpwh.csv) and the loop heater (named: *dhwlphttr-t24-CHPWH_hpwh.csv). These contain various diagnostics on energy, temperatures, and flows for the modeled CHPWH systems. Model runs are aggregated from these files to annual results in the results-summary.csv file in the /analysis/* directory. For a description of the output parameters in the results-summary.csv see the Output Parameters section.

Modelkit Tips

Skipping Runs

Use # in a skip column to skip that row.

Parallel Processing

In the root directory of Ecosim is a file called “.modelkit-config.” This can be opened in any plain text editor. There is an attribute in the file “max-workers” that can be changed to 1 to prevent parallel processing or increased to allow it.

Note that if the file does not appear there, it may just be hidden. This source explains how to show hidden files: <https://support.microsoft.com/en-us/windows/show-hidden-files-0320fe58-0117-fd59-6851-9b7f9840fdb2>.

CHPWH Systems

Proper design of the full commercial HPWH system, not just the HPWH itself, is essential for reliable and efficient HPWH operation. Heat pump water heating equipment can operate in two distinctive heating cycles: single pass and multi-pass.

- Single pass: Heats water to the storage set point temperature in a single water pass through the refrigeration circuit.
- Multipass: Heats water to the storage set point temperature using multiple water passes through the refrigeration circuit.

Advisable piping configurations depend on the distinct heating cycles. According to V8.0 of the AWHs, these include:

1. Single-pass HPWH
 - a. No hot water circulation, primary heat pump water heating only.
 - b. Hot water circulation returned to the primary storage.

- c. Hot water circulation returned to a temperature maintenance tank in series with electric resistance element, also referred to as a “swing tank”.
 - d. Hot water circulation returned to a temperature maintenance storage tank in parallel with multi-pass HPWH for reheat (Figure 1).
2. Multi-pass HPWH
 - a. Integrated HPWH, no hot water circulation.
 - b. Integrated HPWH, hot water circulation returned to primary storage.
 - c. Split-system, hot water circulation returned to the primary storage.

The implantation in Ecosim gives the option of having two separate heaters, the primary and a “loop” heater, referred to in the AWHs as the temperature maintenance tank. In Figure 1, the primary heater is on the left of the cold-water supply (CW supply) and the temperature maintenance or loop tank is on the right. All of the above systems are possible in the Ecosim by changing the primary and loop heater inputs.

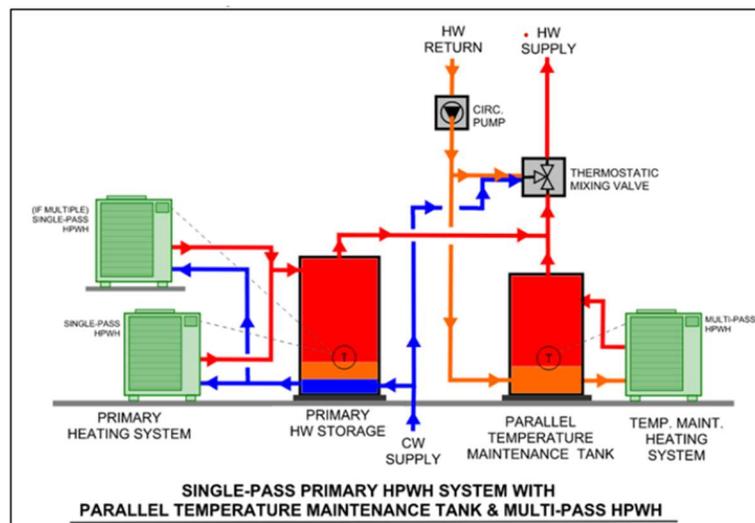


Figure 1. Single-Pass Primary HPWH with Parallel Temperature Maintenance Tank System

Ecosim also has no rules for if the primary or loop heaters are HPWHs or Electric Resistance (ER) tanks, although rules do exist for gas heaters based around their currently arcane modeling methodology. For example, the primary heater type could be ER and the loop heater type could be a HPWH. This system might not make the most sense, but users have freedom to make exploratory decisions here.

Supported HPWH Models

The following is a table of all HPWH model names currently available in Ecosim. These are the HPWH models which have been uploaded to the CEC website at the time of the writing of this manual. As more products are become available and they are expected to be on this list when the data is submitted to the CEC⁵. Models ending in `_SP` or `_MP` are indeed split systems of single pass or multi-pass heaters and

⁵ <https://www.energy.ca.gov/rules-and-regulations/building-energy-efficiency/manufacture-certification-building-equipment-8>

have a configurable number of tanks and heaters, the one exception to the rule is the Sanden40, Sanden80, Sanden120, and SandenGS3⁶, which is a split system single pass heater. Unitary models in general do not have that flag in their name and will have a fixed ratio between number of tanks and heaters, as 1.

AOSmithCAHP120	AOSmithHPTU50	AOSmithHPTU66	AOSmithHPTU80
AOSmithHPTU80DR	AOSmithPHPT60	AOSmithPHPT80	AWHSTier3Generic40
AWHSTier3Generic50	AWHSTier3Generic65	AWHSTier3Generic80	BasicIntegrated
BWC202065	ColmacCxA_10_MP	ColmacCxA_15_MP	ColmacCxA_20_MP
ColmacCxA_25_MP	ColmacCxA_30_MP	ColmacCxA10_SP	ColmacCxA15_SP
ColmacCxA20_SP	ColmacCxA25_SP	ColmacCxA30_SP	ColmacCxV_5_MP
ColmacCxV5_SP	GE2012	GE2014	GE2014_80
GE2014_80DR	GE2014StdMode	GE2014StdMode80	Mitsubishi_QAHV_N13 6TAU_HP_B_SP
NyleC125A_C_MP	NyleC125A_CWP_SP	NyleC125A_MP	NyleC125A_SP
NyleC185A_C_MP	NyleC185A_CWP_SP	NyleC185A_MP	NyleC185A_SP
NyleC250A_C_MP	NyleC250A_CWP_SP	NyleC250A_MP	NyleC250A_SP
NyleC25A_SP	NyleC60A_C_MP	NyleC60A_CWP_SP	NyleC60A_MP
NyleC60A_SP	NyleC90A_C_MP	NyleC90A_CWP_SP	NyleC90A_MP
NyleC90A_SP	ResTank	Rheem_HPHD135HNU _MP	Rheem_HPHD135VNU _MP
Rheem_HPHD60HNU_ MP	Rheem_HPHD60VNU_ MP	Rheem2020Build40	Rheem2020Build50
Rheem2020Build65	Rheem2020Build80	Rheem2020Prem40	Rheem2020Prem50
Rheem2020Prem65	Rheem2020Prem80	RheemHB50	RheemHBDR2250
RheemHBDR2265	RheemHBDR2280	RheemHBDR4550	RheemHBDR4565
RheemHBDR4580	Sanden120	Sanden40	Sanden80
SandenGS3	Scalable_SP	Scalable_MP	Stiebel220E

⁶ Sanden is now known by SanCO2 in the US.

Input Parameters

In general, each model is organized starting at the building level (i.e. number of apartment units), and is broken down to the CHPWH system, which must have a primary water heater, and could have a loop heater, and/or a temperature maintenance loop.

As a reminder each input parameter must be preceded by ":" (i.e. :weather_file) to tell Modelkit that the input needs to be passed to the template.

Building Level Inputs

run_name

type: str

Description: The name of the weather file to be modeled against. This sets wfName in CSE input file.

Required: No

Default: None

weather_file

type: str

Description: The name of the weather file to be modeled against. This sets wfName in CSE input file.

Required: Yes

Default: None

num_bedrooms

type: array

Description: An array with sizes of apartments by the number of bedrooms. For example, num_bedrooms could be [0, 0, 2, 6] which would correspond to a studio, a studio, a two bedroom, and a six bedroom apartment. This is primarily used to set the hot water draws.

Required: Yes

Default: None

Range: > 0

floor_areas

type: array

Description: An array of the condition square footage for the apartment sizes in num_bedrooms. For example floor_areas could be [520, 600, 1200, 3000] for the example in num_bedrooms.

This is used to set the standard distribution loss multiplier, if using the default draws, $SDLM = 1.0032 - 0.0001864 * floor_areas[i] - 0.00000002165 * floor_areas[i]^2$. Where $floor_areas[i]$ is the square footage of an apartment with $num_bedrooms[i]$ bedrooms.

Required: Yes

Default: None

Range: > 0

num_units

type: array

Description: An array of the condition square footage for the apartment sizes in num_bedrooms. For example, num_units could be [10, 10, 12, 1] for the example in num_bedrooms and would create a 33 unit apartment building.

Required: Yes

Default: None

Range: > 0

weather_mains_model

type: string

Description: The mains water temperature model feeding the CHPWH system. Four options are available (“cbecc”, “ground”, “max_ground”, “nrel”):

cbecc – Uses the default mains water temperature in CSE as defined by the 2019 RACM.

ground – Uses the ground temperature as calculated by CSE.

max_ground – Uses the max between the ground temperature and 33 °F to prevent freezing.

nrel – Follows the sinusoidal equation developed by the National Renewable Energy Laboratory (NREL) in Burch and Christensen (2007)⁷, which requires the extra input parameter *weather_dt_amb_max*; note that this variable must be set there is no way to get it out of the weather processing that CSE does. The *offset* variable can also be adjusted, from the parameter *weather_temp_offset*, to ensure freezing does not occur. The equation is described in Hendron et al. (2004)⁸ as:

$$T_{mains} = (T_{amb,avg} + offset) + R * \left(\frac{dT_{amb,max}}{2} \right) \sin(0.986(day\# - 15 - lag) - 90)$$

Where:

$T_{amb,avg}$ is the ambient average temperature of the weather file in °F,

offset is 6 °F,

$R = 0.4 + 0.1 (T_{amb,avg} - 44)$

$dT_{amb,max}$ is the maximum difference between monthly average ambient temperatures in °F, defined in the parameter *weather_dt_amb_max*,

day# is the day of the year, and

$lag = 35 - 0.01 (T_{amb,avg} - 44)$

Range: "cbecc" | "ground" | "max_ground" | "nrel"

Required: Yes

Default: “cbecc”

weather_dt_amb_max

type: float

Description: The maximum difference between monthly average ambient temperatures in °F (e.g., $T_{amb,avg,july} - T_{amb,avg,January}$). Used if *weather_mains_model* is set to “nrel”. This currently must be calculated by the user currently before running the model.

⁷ Burch, J., Christensen, C. (2007). “Towards Development of an Algorithm for Mains Water Temperature.” Proceedings of the 2007 ASES Annual Conference. Accessed:

<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.515.6885>

⁸ Hendron, R., Anderson, R., Christensen, C., Eastment, M., and Reeves, P. 2004. “Development of an Energy Savings Benchmark for All Residential End-Uses”, Proceedings of SimBuild 2004, IBPSA-USA National Conference, Boulder, CO, August 4 - 6, 2004.

Range: ≥ 0

Required: IFF `weather_mains_model` is "nrel"

Default: None

weather_temp_offset

type: float

Description: This is the offset used in the mains water temperature equation when `weather_mains_model` is "nrel". Recommended to only be used in cold climates when mains temperatures can drop below zero.

Range: Any float.

Required: IFF `weather_mains_model` is "nrel"

Default: 6.0

CHPWH System Level Inputs

temperature_maintenance_loop_config

type: boolean

Description: If the system has a temperature maintenance loop (recirculation loop) or not. If true the inputs for the temperature maintenance water heater are read in. If `loop_heater_config` is set to "None" the recirculation loop is then returned to the primary system.

Range: true || false

Required: Yes

Default: true

system_type

type: str

Description: Defines if the system is a central system or if to model as a collect of in-unit systems.

Range: "Central" || "In-Unit"

Required: Yes

Default: "Central"

system_supply_temp

type: float

Description: The supply water temperature or the temperature the mixing valve supplies mixed water to the occupants. 1 to 1 with CSE input `wsTUse`: "Hot water delivery temperature (at output of water heater(s) and at point of use). Delivered water is mixed down to `wsTUse` (with cold water) or heated to `wsTUse` (with extra electric resistance backup, see the CSE manual for [DHWHEATER](#) `whXBUEndUse`)."

Range: 32 – 200 F

Required: Yes

Default: 120

Primary Heater Inputs

primary_heater_type

type: str

Description: Sets the type of WH for the primary part of the system. If set to "HPWH" this follows uses the HPWH inputs below, otherwise ignores them. If set to "Electric Resistance" it will use the input for capacity to set the element size. If "Gas" minimal inputs are needed.

Range: "HPWH" || "Electric Resistance" || "Gas"

Required: Yes
Default: "HPWH"

primary_heater_hpwh_unit

type: str

Description: Sets the model of the HPWH.

Range: See table in Support HPWH Models

Required: If primary_heater_type == "HPWH"

Default: None

primary_heater_volume

type: float

Description: The total storage volume of the primary heater system in gallons.

This can be set to "Autosize" when primary_heater_type == "Electric Resistance" to create a reasonable volume following the ASHRAE 2015 HVAC Applications simple methodology for sizing DHW systems. To control the sizing in more detail, see the Autosizing InputsAutosizing Inputs.

Range: > 0

Required: Yes

Default: None

primary_heater_elec_resistance_capacity_1

primary_heater_elec_resistance_capacity_2

type: float

Description: The heating capacity for the top (1) and bottom (2) elements in an electric resistance tank in W. Both can be set to "Autosize" when primary_heater_type == "Electric Resistance" to create a reasonable capacity following the ASHRAE 2015 HVAC Applications simple methodology for sizing DHW systems. To control the sizing in more detail, see the Autosizing Inputs. Default size set to match a residential in-unit water.

Range: >= 0

Required: If primary_heater_type == "Electric Resistance"

Default: 4500

primary_heater_gas_heating_capacity

type: float

Description: The heating capacity when specifying a gas heater.

Range: > 0

Required: If primary_heater_type == "Gas"

Default: None

primary_heater_multiplier

type: int

Description: the number of identical primary water heaters.

Range: > 0

Required: Yes

Default: 1

primary_heater_location

type: str

Description: The location of the primary water heater tanks, NOT the source air for the HPWH. The Conditioned Space option fixes the temperature to 70 F.

Range: "Outside" | "Buffer Space" | "Conditioned Space"

Required: Yes

Default: "Conditioned Space"

primary_heater_hpwh_source_air

type: str

Description: The location of the primary water heater tanks, NOT the source air for the HPWH.

Range: "Outside" | "Buffer Space"

Required: If primary_heater_type == "HPWH"

Default: "Outside"

primary_heater_setpoint

type: float

Description: The setpoint for the primary heater in degrees Fahrenheit.

Range: 32 – 200 F

Required: Yes

Default: 140, or if primary_heater_type == "HPWH" it is defaulted by the primary_heater_hpwh_unit

primary_tank_per_heater

type: float

Description: The number of water heater tanks per compressor.

Range: > 0

Required: If primary_heater_type == "HPWH" || "Electric Resistance"

Default: 1

primary_heater_hpwh_insul_r

type: float

Description: The water heater tank insulation R-value (interior + exterior insulation), ft²-F/Btuh

Range: > 0

Required: If primary_heater_type == "HPWH" || "Electric Resistance"

Default: 12

Loop Heater Inputs

The loop heater inputs are largely the same as the primary heater inputs, except replacing "primary" with "loop" in the input parameter name. However, some defaults are different, so the inputs are enumerated here.

The default configuration here is not to have a loop heater, but the rest of the inputs are set up to model a swing tank.

loop_heater_config

type: str

Description: The type of auxiliary, loop, or temperature maintenance tank.

If set to "None" then no loop tank is added. If temperature_maintenance_loop_config == true then the recirculation loop is returned to the primary system. If temperature_maintenance_loop_config == false then there is no loop or loop heater, only the primary heater is modeled.

Note that if `loop_heater_config` is not set to "None" and the `temperature_maintenance_loop_config == false` then the loop heater is made but no recirculation loop is. This could be done with a "Swing" tank to add electric resistance back up for example.

"Swing" is used to set up a temperature maintenance tank in series. The series tank has the hot water input from the primary heater and from the recirculation loop into the bottom of the tank.

"Parallel" (Figure 1) is used to set up a parallel loop tank that only sees the load from the temperature maintenance system. The hot water input from the primary heater enters the top of the tank and the recirculation loop enters the bottom of the tank.

"Trim" is typically used with multi-pass HPWHS in the primary. This uses a return to primary configuration but has an extra auxiliary electric resistance tank to ensure the outlet temperature of the water is above the use temperature. Currently there is no way to model a trim tank in CSE.

Range: "None" | "Swing" | "Parallel"

Required: Yes

Default: "None"

loop_heater_type

type: str

Description: Sets the type of WH for the primary part of the system, this is `cse:DHWEATER`. If set to "HPWH" this follows uses the HPWH inputs below, otherwise ignores them. If set to "Electric Resistance" it will use the input for capacity to set the element size. If "Gas" minimal inputs are needed.

Range: "HPWH" || "Electric Resistance" || "Gas"

Required: Yes

Default: "Electric Resistance"

loop_heater_hpwh_unit

type: str

Description: Sets the model of the HPWH.

Range: See table in Support HPWH Models

Required: If `primary_heater_type == "HPWH"`

Default: None

loop_heater_volume

type: float

Description: The total storage volume of the loop heater system in gallons.

Range: > 0

Required: Yes

Default: 900?

loop_heater_elec_resistance_capacity_1

loop_heater_elec_resistance_capacity_2

type: float

Description: The heating capacity for the top (1) and bottom (2) elements in an electric resistance loop tank in W.

Range: > 0
Required: If loop_heater_type == "Electric Resistance"
Default: None

loop_heater_gas_heating_capacity
type: float
Description: The heating capacity when specifying a gas heater.
Range: > 0
Required: If loop_heater_type == "Gas"
Default: None

loop_heater_multiplier
type: int
Description: the number of identical loop water heaters.
Range: > 0
Required: Yes
Default: 1

loop_heater_location
type: str
Description: The location of the loop water heater tanks, NOT the source air for the HPWH. The Conditioned Space option fixes the temperature to 70 F.
Range: "Outside" | "Buffer Space" | "Conditioned Space"
Required: Yes
Default: "Conditioned Space"

loop_heater_hpwh_source_air
type: str
Description: The location of the loop water heater tanks, NOT the source air for a HPWH.
Range: "Outside" | "Buffer Space"
Required: If loop_heater_type == "HPWH"
Default: "Outside"

loop_heater_setpoint
type: float
Description: The setpoint for the loop heater in degrees Fahrenheit.
Range: 32 – 200 F
Required: Yes
Default: 131, or if loop_heater_type == "HPWH" it is defaulted by the loop_heater_hpwh_unit

loop_tank_per_heater
type: float
Description: The number of water heater tanks per heater.
Range: > 0
Required: If loop_heater_type == "HPWH" || "Electric Resistance"
Default: 1

loop_heater_hpwh_insul_r
type: float
Description: The water heater tank insulation R-value (interior + exterior insulation), ft2-F/Btuh

Range: > 0

Required: If loop_heater_type == "HPWH" || "Electric Resistance"

Default: 12

Draw Inputs

The draw related inputs are primarily to define the flow rates of fixtures and appliances, as well as the waste factors associated with the draws.

kitchen_sink_flow

type: float

Description: This sets the expected flow rate for kitchen sinks, which is mixed with the bathroom faucet flow. The standard flow rate here is 2.2 GPM but other values according to different standards are given in the table below.

Fixture	Standard	WaterSense	CAWA	Unit
Kitchen Sink	2.2	2.2	1.8	GPM

Because of limitations with the definition of fixture flows in the draws used by CSE stemming from the difficulty in separating faucet flows, kitchen_sink_flow and the bathroom_faucet_flow are weighted together by the kitchen_sink_weighting_fraction (hardcoded at 0.5) to define a single faucet flow rate.

Range: > 0

Required: No

Default: 2.2

bathroom_faucet_flow

type: float

Description: This sets the expected flow rate for bathroom faucets. The standard flow rate here is 2.2 GPM but other values according to different standards are given in the table below. For more details see kitchen_sink_flow.

Fixture	Standard	WaterSense	CAWA	Unit
Bathroom Faucet	2.2	1.5	1.2	GPM

Range: > 0

Required: No

Default: 2.2

shower_flow

type: float

Description: Sets the expected flow rate for showers. Different standards are given below.

Fixture	CSE Default	Standard	WaterSense	CAWA	Unit
Shower	2	2.5	2	1.8	GPM

Range: > 0

Required: No

Default: 2.0

dishwasher_cycle

type: float

Description: Sets the dishwasher hot water use per cycle. *This is not a flow rate.*

Note that “None” isn’t a valid option given the dataset, where they only selected homes with dishwashers and as such these profiles are not appropriate for homes where dishes are hand-washed. Furthermore, no one knows what the difference in hot water use is, and we should not expect timing of dishwasher events to be the same as handwashing events so the replacement of one with the other would be difficult.

Appliance	CSE Default	Fed	Estar	CA/WA	Unit
Dishwasher	5	5	3.5	5	Gal/Cycle

Range: > 0

Required: No

Default: 5

clotheswasher_cycle

type: float

Description: Sets the clothes washer hot water use per cycle. *This is not a flow rate.*

The standard method follows what is currently run in CBEC-res. The EnergyStar model was derived in a similar manner; the data was pulled from the “United States Department of Energy (DOE) Appliance & Equipment Standards Program Compliance Certification Database.”

This can be set to 0 to remove clothes washers from the simulation.

Appliance	Standard/CSE Default	EnergyStar	Unit
Clothes washer	25.73	17.29	Gal/Cycle

Range: >= 0

Required: No

Default: 25.73

shower_day_waste_fraction

type: float

Description: Along with the faucet_day_waste_fraction describes the ratio of day waste volume that are distributed to the shower draws and the faucet draws. The default ratio between showers and faucets 3 : 1 means shower draws get 3 times the waste volume that faucets do. This is done to emulate people waiting for hot water in their showers, but not when washing hands or filling cooking pots.

Range: >= 1

Required: No

Default: 3

faucet_day_waste_fraction

type: float

Description: See shower_day_waste_fraction.

Range: >= 1

Required: No

Default: 1

distribution_system_multiplier

type: float

Description: The distribution system multiplier, 1 to 1 with the CSE input wsDSM. “See RACM App B Eqn 4. To duplicate CEC 2016 methods, wsDSM should be set to the appropriate value from App B Table B-2. Note the NCF (non-compliance factor) included in App B Eqn 4 is not a CSE input and thus must be applied externally to wsDSM.”

Range: > 0

Required: No

Default: 1

draws_shuffle_seed

type: integer

Description: The seed number for randomizing the order the draws appear in for each unit size. If 0 then there is no randomization in the order of unit draws, the units will simply count to from 1 to the max number of draw profiles available for that unit size. Currently the max draw profiles for studios is 50, 1 bedrooms is 50, 2 bedrooms is 50, 3 bedrooms is 20, 4 bedrooms is 10, and 5 bedrooms is 10.

Range: >= 0

Required: No

Default: 0

Recirculation Loop Inputs*loop_calc_method*

type: str

The method used to calculate the recirculation loop efficiency.

- Simulated will follow the rules defined in the 2019 RACM to calculate the recirculation loop lengths, branches, and associated losses.
- Heat Loss Rate calculates the hourly recirculation pipe heat loss with non-zero water flow in the 2019 RACM to match the input value by estimating the recirculation pipe lengths and the imperfect insulation correction factor.

For all methods chosen the pipe diameters are sized in accordance with the 2019 RACM; the table was built looking at sizes pipe diameters according to the 2009 Uniform Plumbing Code (UPC) and finds the diameters from on a regression curve dependent on the number of multifamily units⁹.

Range: Heat Loss Rate || Simulated

Default: Heat Loss Rate

Required: If temperature_maintenance_loop_config == true

loop_pump_power

type: float

Description: The recirculation pump power in Watts. The heat created by the pump is added to the recirculation loop return water. The default follows the rule used by CBECC-Res: the power is the summed number of units * 0.5 + 4 W.

Range: > 0

Default: “default” = #AssignedDwellingUnits * 0.5 + 4 W

⁹ http://title24stakeholders.com/wp-content/uploads/2017/10/2013_CASE-Report_Water-and-Space-Heating-ACM-Improvement.pdf

Required: If temperature_maintenance_loop_config == true

loop_flow

type: float

Description: The recirculation loop flow rate this is CSE::DHWLOOP.wlFlow. If not specified it defaults to the rule used in CBECC-Res 0.17 * #AssignedDwellingUnits GPM

Range: > 0

Default: "default" = 0.17 * #AssignedDwellingUnits [GPM]

Required: If temperature_maintenance_loop_config == true

loop_location

type: str

Description: The location of the recirculation loop, used to determine the exterior temperature of the recirculation loop segments. Branches are assumed to be in conditioned space.

If the loop_calc_method == "Heat Loss Rate" the loops are assumed to be in conditioned space (70 °F).

Range: Outside || Buffer Space || Conditioned Space

Default: "Conditioned Space"

Required: If temperature_maintenance_loop_config == true AND loop_calc_method == "Simulated"

loop_loss_rate_per_apartment

type: float

Description: The hourly recirculation pipe heat loss with non-zero water flow irrespective of hot water draws and assuming there is constant recirculation flow, described by equation 17 of Appendix B in the 2019 RACM. This is used to calculate the recirculation pipe imperfect insulation factor input in CSE, f_{UA} , as found in the 2019 RACM equation 20. The assumption here is that the ambient temperature is at room temperature.

f_{UA} can be solved for when the total recirculation heat loss, $PLWF$, is specified, all pipe segments are in the same ambient condition, and $U_{insul,n} \geq U_{bare,n}$.

$$f_{UA} = \frac{\rho c_p Flow_R}{\sum_n Len_n U_{insul,n}} * \left(\ln(T_{1,in} - T_{amb}) - \ln\left(\frac{PLWF}{\rho c_p Flow_R} + T_{1,in} - T_{amb}\right) \right)$$

Here T_{amb} is forced to room temperature, 70 °F, $T_{1,in}$ is system_supply_temp, $Flow_R$ is loop_flow, $Len_n U_{insul,n}$ is the length and U value of pipe segment n.

An issue arises when $U_{insul,n} < U_{bare,n}$, which can occur for the small diameter return pipe. The solution is to increase pipe lengths by a constant to force f_{UA} to be a smaller number.

Range: > 0

Default: 80 [W/Apt]

Required: If temperature_maintenance_loop_config == true And loop_calc_method == "Heat Loss Rate"

loop_branch_length

type: float

Description: Sets the length of branches from the recirculation loop to the apartment units.

If simulated the value is set according to the rules of the 2019 RACM Appendix B.

Range: >= 0

Default: 4 [ft]

Required: If temperature_maintenance_loop_config == true And loop_calc_method == "Heat Loss Rate"

loop_branch_number

type: int

Description: Sets the number of branches from the recirculation loop to the apartment units. It is used only when `loop_calc_method == "Heat Loss Rate"`. Use will primarily be to set the number of branches to the number of apartment units.

If simulated the value is set according to the rules of the 2019 RACM Appendix B.

Range: ≥ 0

Default: "default" = total number of units

Required: If `temperature_maintenance_loop_config == true` And `loop_calc_method == "Heat Loss Rate"`

bldg_num_floors

type: float

Description: The number of floors in the building. Used to find pipe segment lengths according to the RACM if `loop_calc_method = Simulated`. This also used to set the number of units per branch ($N_{Unit_b} = bldgFloors/2$), which in turns sets number of branches the length and diameter of the branches. If the simulation method is NOT used for recirc then the number of branches is set to the number of units, the branch length is 4 ft and, and the branch diameter is found the same way.

In the CA model, the number of branches is set to $N_{branchk} = N_{unit_k} / N_{unit_b}$, the diameter of the branches is set by Table B-5 of the 2019 RACM, and the branch lengths are found from $4 + bldg_floor_height * bldg_num_floors / 2$.

Range: ≥ 1

Default: "default", which assumes $\max(1, \text{number of units}/20)$

Required: If `temperature_maintenance_loop_config == true` And `loop_calc_method == "Simulated"`

bldg_floor_height

type: float

Description: The height of the floors in ft.

Range: > 0

Default: 10 ft

Required: If `temperature_maintenance_loop_config == true` And `loop_calc_method == "Simulated"`

Buffer Space Inputs

If the locational or source air inputs for the primary heater, loop heater, or recirculation loop are set to the "Buffer Space" input, the buffer space is constructed as a garage, whose primary control will be if its above or below grade.

The buffer space makes a lot of assumptions to get a reasonable air temperature in the space. For example, there are assumptions about the light power density and occupancy density to provide internal gains, as well as assumptions on wall materials and thickness.

The floor area of the buffer space is assumed to be the average floor area, that is to say the total building area divided by the number of floors in the building.

buffer_space_underground_config

type: boolean

Description: Determines if the buffer space is underground, i.e. below grade, (true) or above grade (false).

Range: true || false

Default: true

Required: If the buffer space is used by another object.

buffer_number_of_floors

type: int

Description: The number of floors in the buffer space, which has a large effect on the volume.

Range: ≥ 1

Default: 1

Required: If the buffer space is used by another object.

Autosizing Inputs

Currently the autosizing inputs are only applicable to ER heaters.

sizing_duration

type: int

Description: The peak period, or the number of hours for which the autosizing algorithm looks for the peak hot water load. This volume and time period is used to follow the ASHRAE methodology on sizing service water heating, following the simple method (see 2015 ASHRAE Handbook – HVAC Applications 50.15).

In the methodology the volume during the peak period defined by *sizing_duration* is the storage volume divided by a storage efficiency number (*sizing_storage_efficiency*) is the volume in the “simple” method. The capacity is the power it takes to heat the volume of hot water used during the peak period to temperature over the length of the peak period.

Range: ≥ 1

Default: 1

Required: No

sizing_storage_efficiency

type: float

Description: The storage efficiency number that is used for the sizing methodology. The recommended value in ASHRAE is 0.8.

Range: $1 \geq x > 0$

Default: 0.8

Required: No

Additional Export Inputs*export_minute_day_start**export_minute_day_end*

type: int

Description: Define *export_minute_day_start* and *export_minute_day_end* to define the starting day of the year and the ending day of the year for which to print minute by minute output data. Integer days start counting at the beginning of the year. If either input is less than 0, they are ignored.

Required: Yes

Default: -1

add_meter_energy_exports

type: boolean

Description: Provides detailed energy outputs using the CSE “meter” construct in the hourly output file for each run.

Required: Yes

Default: false

add_dhwmeter_flow_exports

type: boolean

Description: Provides detailed flow outputs using the CSE “meter” construct for flows in the hourly output file for each run. The outputs give water use at the fixture and just the hot water fraction.

Required: Yes

Default: false

add_subhourly_exports

type: boolean

Description: Provides a subhourly stepped export file for each run. This is also controlled by `add_meter_energy_reports`, and provides little information if `add_meter_energy_reports = false`.

Required: Yes

Default: false

Advanced Input Structure

At times users, may want to define inputs for the primary HPWH and loop HPWH based on the characteristics of the specific HPWH. Since HPWH metadata is already checked against inputs this has been exposed so that it the HPWH metadata can be used for that HPWH in the primary or loop input scopes. This additional feature also will evaluate simple equations such as:

$$(10.-9.)/(2.*5.)*1e3$$

Pieces of metadata can be evaluated by encapsulating their values in square brackets, [x]. Parameters starting with “primary_” reference the metadata associated with the `primary_heater_hpwh_unit`. For example the `primary_heater_setpoint` could be set to

$$[\text{setpoint}]$$

To use the `primary_heater_hpwh_unit` default setpoint in the software. Or the `primary_heater_volume` could be based of the capacity at the minimum operating air temperature as

$$2e6 / [\text{capacityAtMinOperatingTemp}]$$

These equations are only evaluated when, the parameter `primary_heater_type` is “HPWH”, and the metadata scope is defined by the `primary_heater_hpwh_unit` parameter. Equations are only evaluated for the variables that have int or float type, not string, and the electric resistance and gas heating capacities are ignored. The same rules apply to the loop heater, when `loop_heater_type` is “HPWH”, and the metadata is defined by the `loop_heater_hpwh_unit` parameter. The parameters which allow references to the HPWH metadata are given in Table 1.

Table 1. Variables where functions are evaluated and the associated scope for data.

Scope	<code>primary_heater_hpwh_unit</code>	<code>loop_heater_hpwh_unit</code>
Variables	<code>primary_heater_volume</code>	<code>loop_heater_volume</code>

	primary_heater_multiplier	loop_heater_multiplier
	primary_heater_setpoint	loop_heater_setpoint
	primary_tank_per_heater	loop_tank_per_heater
	primary_heater_hpwh_insul_r	loop_heater_hpwh_insul_r

Supporting Functions

Users should be wary of the integer vs float inputs when using this equation structure. Five basic functions following Ruby syntax (see examples for syntax), are allowed to help with dealing with variable types:

[floor](#) : returns the largest number greater than or equal to the variable i.e. (3.1).floor and (3.99).floor will return 3.

[ceil](#) : returns the smallest number greater than or equal to the variable i.e. (3.1).ceil and (3.99).ceil will return 4.

[round](#) : rounds the float to the nearest integer i.e. (3.49).round will return 3 and (3.5).round will return 4.

[to_i](#) : returns the integer value of a string or float, i.e. (3.1).to_i and (3.99).to_i will return 3.

[to_f](#) : returns the float value of an integer, i.e. (3).to_f will return 3.0.

Round, floor, and ceil functions also supported to allow the intentional conversion from float to integer. The syntax follows the input structure of the Ruby scripting language, for example:

```
(8.0 / 3.0).floor    #=> 2
(8.0 / 3.0).ceil    #=> 3
```

Note the importance of declaring floats and integers for mathematical operations. The above example would quickly break down if not:

```
(8 / 3).floor       #=> 2
(8 / 3).ceil        #=> 2
```

Which occurs because $8/3$ returns an integer. Note that Ruby will assume if there is one float in the equation that the result of the division should be float, i.e.

```
(8.0 / 3).floor     #=> 2
(8.0 / 3).ceil      #=> 3
```

And

```
(8 / 3.0).floor     #=> 2
(8 / 3.0).ceil      #=> 3
```

The ternary operator is also supported, the basic of this operator looks like this:

```
condition ? true : false
```

For example this could be used to reduce the warning messages from CSE if the setpoint is fixed and cannot be changed, the primary_heater_setpoint could be set to:

```
[isSetpointFixed] == 1 ? [setpoint] : 140
```

Specific HPWH data variables

This section describes the variables associated with the HPWH presets, with each HPWH have unique their own values.

setpoint

Type: float

Units: Fahrenheit

Description: The default setpoint for the HPWH unit. For integrated units this is based on a calibration to a large dataset¹⁰.

minCompOperatingTemp

Type: float

Units: Fahrenheit

Description: The minimum operating air temperature of the heat pump in the HPWH unit in degrees Fahrenheit. Below this temperature resistance elements turn on for integrated units.

capacityAtMinOperatingTemp

Type: float

Units: BTU/hr

Description: The minimum possible capacity of the HPWH. The capacity is evaluated at the minimum operating air temperature with an inlet water temperature of 32 °F and a setpoint of 140 °F, unless the setpoint is fixed. Note that if the HPWH is a multi-pass unit the information on setpoint is not important to the capacity of the unit at 32 °F inlet water.

Note that the model Scalable_SP and Scalable_MP will only return its default heating capacity, not the version a user defines in CSE.

capacityAt40F

Type: float

Units: BTU/hr

Description: Same as the capacityAtMinOperatingTemp except evaluated at 40F. If 40 F is beneath the minimum operating air temperature of the unit this value is 0.

Note that the model Scalable_SP and Scalable_MP will only return its default heating capacity, not the version a user defines in CSE.

numResistanceElements

Type: int

Units: Count

Description: The number of resistance elements in the HPWH or resistance tank, external tanks typically have 0 resistance elements.

¹⁰ Ecotope (2015). Heat Pump Water Heater Model Validation Study. Report #E15-306. Prepared for Northwest Energy Efficiency Alliance

totalResistanceCapacity

Type: float

Units: BTU/hr

Description: The total resistance capacity of all the resistance elements in the water heater. If numResistanceElements is 0 then this value is 0.

Below are boolean values represented as integer 0 or 1.

isSetpointFixed

Type: int

Units: NA

Description: If the setpoint can be changed or not. If not the setpoint set by the user is ignored and an warning is issued. If the heaters' setpoint is to [setpoint] there is no warning issued.

tankSizeFixed

Type: int

Units: NA

Description: If the tank size is fixed or not. Typically integrated or unitary units have fixed tank sizes and these volumes cannot be changed.

isMultipass

Type: int

Units: NA

Description: If the HPWH is multi-pass or not, if not the unit would be single-pass.

canCompScale

Type: int

Units: NA

Description: If the compressor can be scaled or not, this is only true currently for the Scalable_SP and Scalable_MP models, which are split system models with a low efficiency for the product type. This is useful if users want to scale the capacity of their equipment. However the CSE input for nominal heating capacity (wh_heatingCap) in Btuh, is not currently a parameter in Ecosim, so users will have to define this in the CSE instance files.

isSplitSystem

Type: int

Units: NA

Description: If the system has a split HPWH and storage tank, or not. If not it is unitary (or integrated) HPWH or electric resistance tank, or the unit is a storage tank.

Output Parameters

Model runs are aggregated into the results-summary.csv file for each analysis, the header in the csv contains various output parameter names. Each output here is an annual sum or average derived from the hourly output file and used to summarize the annual performance of the system. Below is a list of the outputs found in the results-summary.csv file.

File Name

Units: NA

Description: The path to the CSE model file, instance.cse. Due to the folder structure this variable encodes the climate zone, cohort, as well as the run name.

COP Primary []

Units: NA

Description: The average coefficient of performance for just the primary water heating object over the year of simulation. This is the energy output from the WH storage tank divide by the total input energy for the primary heater.

COP Temperature Maintenance []

Units: NA

Description: The average coefficient of performance for just the secondary water heating object over the year of simulation

COP System []

Units: NA

Description: The average coefficient of performance for the whole system, which includes the primary and secondary energy inputs, and all the hot water energy for temperature maintenance and hot water delivery measured at the mixing valve outlet temperature over the year of simulation. Note that this does not include the circulation pump.

COP Fixture []

Units: NA

Description: Same as COP System except the hot water energy is measured at the end use fixtures in apartments, should be lower than COP System.

Energy In Total Hourly [kWh]

Units: kWh

Description: The annual sum of energy inputs into the primary, auxiliary water heaters, and the recirculation pump summed from the hourly output.

Energy In Heaters Hourly [kWh]

Units: kWh

Description: The annual sum of all energy inputs into the primary and auxiliary water heaters from the hourly output.

Energy Primary Hourly [kWh]

Units: kWh

Description: Annual sum of energy into the primary water heater including whole HPWH energy input, backup electric resistance energy, and extra backup from this heater (or missed energy).

Energy Auxiliary Hourly [kWh]

Units: kWh

Description: Annual sum of energy into the loop heater, ie the auxiliary or secondary water heater including whole HPWH energy input, backup electric resistance energy, and extra backup from this heater (or missed energy).

Energy Miss Hourly [kWh]

Units: kWh

Description: The annual sum of extra backup energy from the primary and auxiliary water heaters, this is the total energy beneath the loop supply temperature that was not met by the system. This is calculated as if there is an infinitely variable instantaneous electric resistance heater at the outlet of the mixing valve to maintain the supply temperature.

Energy Pump Hourly [kWh]

Units: kWh

Description: The annual sum of the recirculation loop pump energy.

Net Energy Recirc Load Hourly [kWh]

Units: kWh

Description: The net energy loss of the recirculation loop, including the gains from the recirculation pump, over the year of simulation.

Energy Loop Segment Pipes Loss [kWh]

Units: kWh

Description: The annual sum of the recirculation loop energy losses, not including uncirculated branches.

Energy DHW Load Hourly [kWh]

Units: kWh

Description: The annual sum of hot water draw energy measured at the fixtures. This is the heat delivered at the fixtures not at the mixing valve.

Energy Out Primary Hourly [kWh]

Units: kWh

Description: The annual sum of energy out of the primary water heater including the HPWH and a storage tank.

Energy Out System [kWh]

Units: kWh

Description: The annual sum of energy out of the whole central water heating system from hot water draws measured at the outlet of the mixing valve.

Swing Draw Energy Out [kWh]

Units: kWh

Description: The energy change in the secondary series (or swing) tank from the primary tank outlet to the swing tank outlet during hot water draws. When loop_heater_config is set to "Series" the energy difference between hot water leaving and hot water entering the series tank is needed to derive "COP Temperature Maintenance." When the loop_heater_config is not set to "Series" this output is irrelevant.

Under typical operation the energy change will be negative, as the primary setpoint should be higher than the series tank setpoint and the loop supply temperature. The energy is primarily used by the temperature maintenance system, and will offset electric energy use by the series secondary tank.

Volume Draw Primary [gal]

Units: gallon

Description: The total annual volume of hot water drawn from the primary water heater. Under different loop_heater_config settings this will match or not the "Volume Draw Temperature Maintenance" output. If loop_heater_config is set to "None" then expect that this should be less than "Volume Flow Loop" plus "Volume Flow Draw", due to the mixing valve reducing the volume of hot water to reach the system_supply_temp.

Volume Draw Temperature Maintenance [gal]

Units: gallon

Description: The total annual volume of hot water drawn from the auxiliary or secondary water heater. This should be less than "Volume Flow Loop" plus "Volume Flow Draw", due to the mixing valve reducing the volume of hot water to reach the system_supply_temp.

Volume Flow Loop [gal]

Units: gallon

Description: The total flow volume of hot water through the recirculation loop over the entire year.

Volume Flow Draw [gal]

Units: gallon

Description: The total annual volume of hot water draws measured at the mixing valve, equivalent to the hot water portion of the fixture draws.

Known Issues

A CSE probe on the DHWHeater makes finding the volume out of the primary heater when there is a loop heater impossible. This also affects output for the primary COP, and thus the temperature maintenance COP calculation for a swing tank.

Limit input or bounds checking is done on strings so be cautious with inputs and follow the documentation. These kinds of errors will most likely manifest in the CSE *.err file for each individual run.

Do not try to autosize HPWH objects, the current sizing practice in Ecosim does not yet include the Ecosizer methodology, but could be adapted in the future.