HPWHsim Tips and Tricks

***Units***

Most variables used in HPWHsim have the units they are specified in appended with an underscore. For example, the tank temperatures use the variable named tankTemps\_C, which implies they should be supplied in degrees Celsius. There exist functions for converting between units, and many of the output or input functions have versions which take a UNITS input. The functions with this input will take the values specified and, if necessary, convert them to the appropriate units.

***Interior space temperature depression, for single zone models***

The predecessor of HPWHsim was originally written for SEEM, Ecotope's single-zone building energy use simulation engine. Because HPWH's remove energy from the air, there is the possibility that they will depress the temperature of their local environment, thus decreasing their performance. This effect was measured in a field study, with the average temperature depression being approximately 4.5 F with a half life of 9.4 minutes. It works by exponentially decreasing the local (ambient and evaporator) temperature of the HPWH to a fixed level below the actual input ambient temperature when a heat source which has the “depressesTemperature” boolean set to true is running. When no qualifying heat source is running, the local temperature exponentially returns to ambient using the same time constant.

***The verbosity and callback function***

The HPWHsim provides an error/debugging message facility, along with the ability to specify where the output should be sent, via a callback function. There are four levels of verbosity for the messages:

VRB\_silent = 0, VRB\_reluctant = 1, VRB\_typical = 2, and VRB\_emetic = 3

They are ordered by the amount of output they will write: “silent” prints no messages, “reluctant” prints messages only for fatal errors, “typical” prints basic debugging information, and “emetic” prints copious amounts of information. Each level prints all the messages from that below it, i.e. “typical” will print all of the “reluctant” messages and all of its messages. When initialized, the HPWH object defaults to silent, but can be set using the setVerbosity function.

The default behavior is for output to be written to stdout, using cout. If it is desired for the output to be sent to a different place, this can be done by providing a function pointer and a context pointer. These variables are stored as part of the HPWH object and set using the setMessageCallback function:

void setMessageCallback( void (\*callbackFunc)(const std::string message, void\* pContext), void\* pContext);

As can be seen, the callback function should return nothing, and take a string and a pointer as arguments.

***Hysteresis***

One of the parameters for each HeatSource is a term labeled “hysteresis\_dC”. This represents the different behavior that has been observed for heat sources which are running. The units, dC, are differential Celsius degrees, i.e. the conversion between C and F should not use the constant offset term (32 F). It is applied to:

OFFLOGIC\_lowT - when the heat source is engaged, the hysteresis is subtracted from the decisionPoint. If the heat source is not engaged, the decisionPoint is used as is.

OFFLOGIC\_lowTreheat – when the heat source is engaged, the hysteresis is added to the decisionPoint. The lowTreheat logic does nothing when the heat source is not engaged, as it is not intended to prevent a heat source from engaging.

***resetTankToSetpoint***

The resetTankToSetpoint function will set all nodes to the setpoint. The starting temperature of a HPWH is set to the setpoint when initialized. Any change in the setpoint will not alter the current temperature of the tank however, so if a non-default setpoint is desired, the resetTankToSetpoint function can be used.

**Failure**

***Fatal Errors***

There are a number of reasons that the HPWH simulation can fail:

* The draw size in one step is more than the volume of the tank.
* A HeatSource which has defined backupHeatSource or companionHeatSource pointers is copied or assigned.
* HPWHinit\_file or HPWHinit\_presets fails to properly initialize the HPWH.
* A value other than 1 for minutesPerStep is specified when attempting to run with tempDepression.

This kind of fatal error will write out an informative error message if the verbosity is not set to “silent” and will return HPWH\_ABORT.

The simulation is unrecoverable from this kind of error, and must be reinitialized. Care should be taken not to invoke one of these failure conditions again.

***Non-Fatal Errors***

Certain kinds of mistakes will cause errors but not prevent the simulation from continuing to run. In these cases, the function which has failed will return a failure code specified in the header as HPWH\_ABORT. Examples of these failures are the functions which set or get from a particular node or heat source. Specifying a number, N, outside the range of nodes or heatsources (e.g. 13 for a HPWH with 12 nodes) will return HPWH\_ABORT. Typically a failure of this kind will output a useful error message, provided the verbosity is set to at least “reluctant”.