Introduction to HTAP

Batch run and optimization capabilities for HOT2000 V11.3 – Version of Jan 15, 2018.

Required software

HTAP depends on the following third-party components.

- Ruby: http://rubyinstaller.org/
- Git: https://git-scm.com/downloads

In addition to these, you may find the following tools useful:

- A text file editor, such as notepad++
- A data analysis program, such as Matlab, tableau or excel.

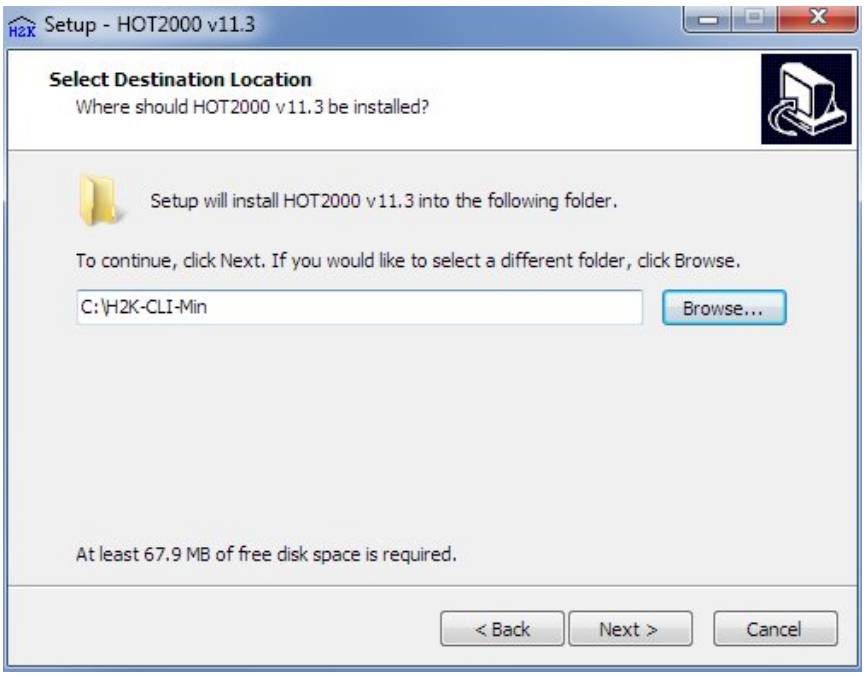
HTAP installation and configuration

HTAP consists of two parts:

a) HOT2000 v11.3 (including command-line client)
b) HTAP scripts and configuration files.

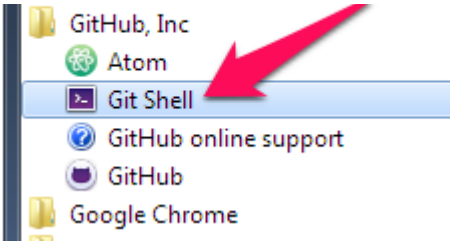
a) To download HOT2000 v11.3, visit https://drive.google.com/drive/folders/1DY1Oae9-d9U8U-jxvRBB54EUM8Df8y-u, and install these files in the following order:

1. vc_redist.x86.exe (install this one first)
2. HOT2000 v11.3b90 Setup.exe (if you don't already have it)
3. HOT2000-CLI v11.3b90 Setup.exe - **When prompted, set the destination location to C:\H2K-CLI-Min (as below)**



b) To Install the HTAP scripts and configuration files, checkout the files from GitHub:

1. Open up the git shell from the start menu



2. Type in the following **command** from location C:\>

```
PS C:\> git clone https://github.com/NRCan-IETS-CE-O-HBC/HTAP.git
Cloning into 'HTAP'...
remote: Counting objects: 996, done.
remote: Total 996 (delta 0), reused 0 (delta 0), pack-reused 996
Receiving objects: 100% (996/996), 12.02 MiB | 2.46 MiB/s, done.
Resolving deltas: 100% (590/590), done.
PS C:\>
```

Git will create a new folder on your computer at location `C:\HTAP`. This folder will contain the HTAP scripts, some archetypes, and configuration files.

3. Before you can run HTAP simulations, you must first copy the archetype files to the `C:\H2K-CLI-Min\User\` directory. HTAP includes a ruby script to do this for you:

```
PS C:\> cd .\HTAP\Archetypes
PS C:\HTAP\Archetypes> ruby .\CopyToH2K.rb

>> Copying ./BC-Step-LargeSFD.h2k to C:\H2K-CLI-Min\User\... done.
>> Copying ./BC-Step-MediumSFD.h2k to C:\H2K-CLI-Min\User\... done.
>> Copying ./BC-Step-MURB10.h2k to C:\H2K-CLI-Min\User\... done.
>> Copying ./BC-Step-MURB20.h2k to C:\H2K-CLI-Min\User\... done.
>> Copying ./BC-Step-Quad-BCH.h2k to C:\H2K-CLI-Min\User\... done.
>> Copying ./BC-Step-Quad-mkt.h2k to C:\H2K-CLI-Min\User\... done.
>> Copying ./BC-Step-rev-LargeSFD.h2k to C:\H2K-CLI-Min\User\... done.
>> Copying ./BC-Step-rev-MediumSFD.h2k to C:\H2K-CLI-Min\User\... done.
>> Copying ./BC-Step-rev-Murb1.h2k to C:\H2K-CLI-Min\User\... done.
>> Copying ./BC-Step-rev-Quad.h2k to C:\H2K-CLI-Min\User\... done.
>> Copying ./BC-Step-rev-Row.h2k to C:\H2K-CLI-Min\User\... done.
>> Copying ./BC-Step-rev-SmallSFD.h2k to C:\H2K-CLI-Min\User\... done.
>> Copying ./BC-Step-Row-11uBCH.h2k to C:\H2K-CLI-Min\User\... done.
>> Copying ./BC-Step-Row-mkt.h2k to C:\H2K-CLI-Min\User\... done.
>> Copying ./BC-Step-SmallSFD.h2k to C:\H2K-CLI-Min\User\... done.
>> Copying ./KelownaHouse.h2k to C:\H2K-CLI-Min\User\... done.
>> Copying ./PrinceGeorgeHouse.h2k to C:\H2K-CLI-Min\User\... done.
>> Copying ./codeLib.cod to C:\H2K-CLI-Min\StdLibs\... done.
>> Copying ./FuelLib16.flc to C:\H2K-CLI-Min\StdLibs\... done.

PS C:\HTAP\Archetypes>
```

Note that the `CopyToH2k.rb` script will copy all HOT200 files and libraries from the `C:\HTAP\Archetypes` folder to `C:\H2K-CLI-Min\User`. To add additional archetypes to the HTAP platform, you merely need to copy them into `C:\HTAP\Archetypes`, and re-run the `CopyToH2k.rb` script

Verifying the installation

1) Test HOT2000 and the substiture-h2k.rb script by running the following command from the `C:\HTAP` directory – abridged output appears below.

```
PS C:\HTAP> .\substitute-h2k.rb -vv -o .\HOT2000.options -c .\HOT2000.choices -b C:\H2K-
CLI-Min\User\KelownaHouse.h2k

 > substitute-h2k.rb
         path: C:\HTAP
         ChoiceFile: .\HOT2000.choices
         OptionFile: .\HOT2000.options
         Base model: C:\H2K-CLI-Min\User\KelownaHouse.h2k
         HOT2000 source folder: C:\H2K-CLI-Min
         HOT2000 run folder: C:\HTAP\H2K


 Reading available options (.\HOT2000.options)...  done.

   ----------------------------------------
                  ^
                  |

        A lot of output appears here

                  |
                  V
   ----------------------------------------

---------------------- SIMULATION RESULTS --------------------------------

 Peak Heating Load (W): 20793.0
 Peak Cooling Load (W): 19403.6

 Energy Consumption:

  55.1 ( Space Heating, GJ )
  5.8 ( Hot Water, GJ )
  1.4 ( Ventilator Electrical, GJ )
  3.9 ( Space Cooling, GJ )
  24.3 ( Appliances + Lights + Plugs + outdoor, GJ )
 ------------------------------------------------------
  90.5 ( H2K Gross energy use GJ )

 Total processing time: 3.75 seconds (H2K run: 1.57 seconds)

PS C:\HTAP>
```

Application — running a HOT2000 single simulation

HTAP provides a means to run HOT2000 from the command line using the substitute-h2k.rb script. The syntax is:

```
PS C:\HTAP> .\substitute-h2k.rb [-v] -o OPTIONS-FILE
                                     -c CHOICE-FILE
```

- The **substitute-h2k.rb** script is the pre- and post-processor that automates HOT2000. It can manipulate HOT2000 input files, start HOT2000 simulations, and parse HOT2000 outputs.
- The **OPTIONS-FILE** defines all of the attributes within a HOT2000 model that can be manipulated via HTAP, and the valid values that they can be set to. It also contains cost data for upgrade specifications.
- The **CHOICE-FILE** defines the values that each HOT2000 parameter should be set to in the current simulation.

The HOT2000.options file

Most of HTAP's data are stored in the .options file. The option file contains a list of attributes that HTAP can edit within HOT2000 input (.h2k) file. An excerpt from the **HOT2000.options** file follows:

```
!-----------------------------------------------------------------------
! Photovoltaics
! Use internal HOT2000 PV Generation model
! Choice file used to specify Internal PV OR External PV but not both!
!-----------------------------------------------------------------------

*attribute:start
*attribute:name  = Opt-H2K-PV
*attribute:tag:1 = Opt-H2K-Area           ! m2
*attribute:tag:2 = Opt-H2K-Slope          ! degrees from horizontal
*attribute:tag:3 = Opt-H2K-Azimuth        ! degrees from S
*attribute:tag:4 = Opt-H2K-PVModuleType   ! 1:Mono-Si, 2:Poly-Si, 3:a-Si, 4:CdTe, 5:CIS,
                                          ! 6:UsrSpec
*attribute:tag:5 = Opt-H2K-GridAbsRate    ! %
*attribute:tag:6 = Opt-H2K-InvEff         ! %
*attribute:default = NA

*option:NA:value:1 = NA
*option:NA:value:2 = NA
*option:NA:value:3 = NA
*option:NA:value:4 = NA
*option:NA:value:5 = NA
*option:NA:value:6 = NA
*option:NA:cost:total  = 0

*option:MonoSi-5kW:value:1 = 53             !53m2 is required area for 5 kW for Mono-Si
*option:MonoSi-5kW:value:2 = 18.4           !22.6 for 5-12 roof in Prince George and 18.4
for 4-12 slope in Kelowna
*option:MonoSi-5kW:value:3 = 0
*option:MonoSi-5kW:value:4 = 1
*option:MonoSi-5kW:value:5 = 90
*option:MonoSi-5kW:value:6 = 90
*option:MonoSi-5kW:cost:total = 21500      !$21500 assumed cost for 5 kW PV system

*option:MonoSi-10kW:value:1 = 107          !107m2 is required area for 10 kW for Mono-Si
*option:MonoSi-10kW:value:2 = 18.4         !22.6 for 5-12 roof in Prince George
*option:MonoSi-10kW:value:3 = 0
*option:MonoSi-10kW:value:4 = 1
*option:MonoSi-10kW:value:5 = 90
*option:MonoSi-10kW:value:6 = 90
*option:MonoSi-10kW:cost:total = 33395     !$33395 assumed cost for 10 kW PV system for !
                                           !Prince George & Kelowna LEEP

*attribute:end
```

This section defines data for the `Opt-H2K-PV` attribute. Three options are available: `NA`, `MonoSi-5kW`, and `MonoSi-10kW`. Throughout the Subsitute-h2k.rb interprets the `NA` specification as instructions to leave the existing .h2k file unaltered – that is, the values for those inputs that were provided when the file was saved in HOT2000 will be preserved when the file is run in HTAP.

The remainder of the data for each attribute describe tags, values, and costs. Each `tag` identifies a key word that substitute-h2k.rb associates with part of the HOT2000 data model. For instance, `Opt-H2K-InvEff` refers to the inverter efficiency of PV modules. Each value provides the alphanumeric input that must be substituted within the .h2k file. For example, the inverter efficiency will be set to 90% for the MonoSI-10kW case in the snippet above.

TODO: Table defining all HTAP options to be inserted here.

TODO: Add note on costing .

The HOT2000.choice file

The .choice file contains a token-value list that defines the option that HTAP should use for each attribute. The syntax for each is `TOKEN : VALUE`, and comments are denoted with a exclamation mark (`!`). Entries in the choice file must obey the following rules:

- Each token must match one of the attributes in the .options file
- Each value must match on of the options given for that attribute in the .options file
- NA values instruct the substiture-h2k.rb script to leave the associated data in the .h2k file alone – that is, whatever inputs were provided when the file was created in HOT2000 will be used in the HTAP simulation.

An example .choice file follows. In this example, the .choice file instructs HTAP to replace the heating system with a cold-climate air source heat pump, the DHW system with a heat pump water heater, and to add a drain-water heat recovery device. All other inputs are left unchanged.

```
!-----------------------------------------------------------------------
! Choice file for use in exercising HOT2000
!
! The H2K model file used is a valid model and nothing needs to be
! changed for it to run!  Using "NA" on any of the options below
! leaves the model unchanged for that option.
!-----------------------------------------------------------------------


! HOT2000 code library file to be used - MUST ALWAYS BE SPECIFIED HERE
Opt-DBFiles  : H2KCodeLibFile

! Weather location
Opt-Location : NA

! Archetype file:
Opt-Archetype: NZEH-Arch-1

! Fuel costs
Opt-FuelCost : rates2016

! Air tightness
Opt-ACH : NA


! Ceiling R-value
```

```
Opt-Ceilings : NA

! Main wall definitions
Opt-GenericWall_1Layer_definitions : NA

! Exposed floor
Opt-ExposedFloor : NA

! Optical and thermal characteristics of casement windows (all)
Opt-CasementWindows  : NA

! Foundation definitions
Opt-H2KFoundation : NA

! Hot water system.
Opt-DHWSystem :  HPHotWater


! Drain-water heat recovery
Opt-DWHRSystem :  DWHR-eff-30

! HVAC system
Opt-HVACSystem  : CCASHP

! HRV spec
Opt-HRVspec : NA

Opt-RoofPitch : NA   !6-12

! External (Opt-StandoffPV) and Internal model (Opt-H2K-PV) PV options.
! Substitute-h2k.rb will select external if both are specified!
Opt-StandoffPV : NoPV      !SizedPV|8kW
Opt-H2K-PV : NA !  MonoSi-200m2   !MonoSi-50m2

!-------------------------------------------------------------------------
! The following options don't do anything for HOT2000
!-------------------------------------------------------------------------
! Set the orientation for the model (N,S,E,W, or AVG to run all four directions
! and compute an average result).
! ***NOTE: As of Dec 2015 this attribute only determines numbers of runs (1 or 4)
!          for HOT2000 -- it doesn't rotate the model yet.
GOconfig_rotate      : S

Opt-Cooling-Spec     : 2TonStdCooling
```

Substitute-h2k.rb output

When executed, Substiture-h2k.rb will perform the requested changes on the .h2k file, invoke HOT2000 and collect the results. Summary results are reported to screen, as shown in the following excerpt:

```
PS C:\HTAP> .\substitute-h2k.rb -v -o .\HOT2000.options -c .\HOT2000.choices

 > substitute-h2k.rb
         path: C:\HTAP
         ChoiceFile: .\HOT2000.choices
         OptionFile: .\HOT2000.options
         Base model: Not specified. Using archetype specified in .choice file
         HOT2000 source folder:
         HOT2000 run folder:

 Reading available options (.\HOT2000.options)...  done.
```

```
 Reading user-defined choices (.\HOT2000.choices)... done.

 Validating choices and options... done.

 Creating a copying of HOT2000 executable directory below master... 197 File(s) copied

 Creating a copy of HOT2000 model file for optimization work...   1 file(s) copied.
 (File C:\HTAP\Arch-1-NZEH-detached-2-story.h2k created.)

 Invoking HOT2000 (PID 7504)... Hot2000 (PID: 7504) finished with exit status 0
 The run was successful (2.22 seconds)!

 Copying results.        1 file(s) copied.

 Parsing results... done

---------------------- SIMULATION RESULTS --------------------------------

 Peak Heating Load (W): 11131.9
 Peak Cooling Load (W): 10914.9

 Energy Consumption:

  34.6 ( Space Heating, GJ )
  6.0 ( Hot Water, GJ )
  2.4 ( Ventilator Electrical, GJ )
  2.1 ( Space Cooling, GJ )
  31.5 ( Appliances + Lights + Plugs + outdoor, GJ )
 ------------------------------------------------------
  76.7 ( H2K Gross energy use GJ )

 Total processing time: 5.88 seconds (H2K run: 2.22 seconds)
```

Substitute-h2k.rb produces a summary output named SubstituePL-output.txt; an example follows.

```
Energy-Total-GJ   =  86.7
Ref-En-Total-GJ   =  0.0
Util-Bill-gross   =  2160.49
Util-PV-revenue   =  0.0
Util-Bill-Net     =  2160.49
Util-Bill-Elec    =  1992.49
Util-Bill-Gas     =  168.0
Util-Bill-Prop    =  0.0
Util-Bill-Oil     =  0.0
Util-Bill-Wood    =  0.0
Energy-PV-kWh     =  0
Gross-HeatLoss-GJ =  172
Energy-HeatingGJ  =  55.1
AuxEnergyReq-HeatingGJ = 107.7
Energy-CoolingGJ  =  3.9
Energy-VentGJ     =  1.4
Energy-DHWGJ      =  5.8
Energy-PlugGJ     =  24.3
EnergyEleckWh     =  25404.4
EnergyGasM3       =  0.0
EnergyOil_l       =  0.0
EnergyProp_L      =  0.0
```

```
EnergyWood_cord    =  0.0
Upgrade-cost       =  1185.0
SimplePaybackYrs   =  2207.9
PEAK-Heating-W     =  20793.0
PEAK-Cooling-W     =  19403.6
PV-size-kW         =  0.0
ERS-Value          =  0.0
NumTries           =  1.0
LapsedTime         =  3.33
```

In addition to this summary output, comprehensive HOT2000 output is located in the edited .h2k file (in this example, `C:\HTAP\KelownaHouse.h2k`).

Application — running BATCH analysis

Depreciated: GENOPT batch runs and optimization

To use HTAP as part of genopt, the following software are needed:

- Java: https://java.com/en/download/
- GenOpt: https://simulationresearch.lbl.gov/GO/

HTAP uses the GenOpt package to implement batch runs and optimization. Lawrence Berkeley National Laboratory publishes GenOpt at https://simulationresearch.lbl.gov/GO/. Most of the HTAP's optimization workflow uses standard GenOpt input files and features — these are documented here: https://simulationresearch.lbl.gov/GO/download/manual-3-1-1.pdf

To run a simple batch simulation, use the following java command:

```
C:\HTAP> java -classpath "C:\Program Files\genopt\genopt.jar" genopt.GenOpt
         Genopt-H2K-INI.GO-ini
```

Most of this command contains keywords and paths that java needs to locate the correct GenOpt package:

- The text `-classpath "C:\Program Files\genopt\genopt.jar"` points to the folder where GenOpt is installed.
- The text genopt.GenOpt refers to the genopt class that java will load
- The text `Genopt-H2K-INI.GO-ini` refers to the Genopt initialization file, which instructs GenOpt on where to find other key input files, how to invoke the substiture-h2k.rb script , and how to parse substiture-h2k.rb output.

Configuring batch and optimization runs usually requires changes to the initialization file (`Genopt-H2K-INI.GO-ini`), and the command file (`Genopt-H2K-CMD.GO-cmd`).

The initialization (.GO-ini) file

The .GO-ini file provides information that GenOpt needs to locate other input files, to start the substitute-h2k.rb script, and to parse the output. Generally, users will leave most of these inputs unchanged. But two parts of this file are commonly edited – the `CallParameter` and the `Command` definitions.

The `CallParameter` section defines the command-line arguments that should be passed to substitute-h2k.rb:

```
   // Simulator command:
   CallParameter {
```

```
      Prefix = "Ruby substitute-h2k.rb";1
      Suffix = " -v -c GenOpt-picked-these-choices.GO-tmp -o HOT2000.options";
   }
```

The `suffix` command includes the name and path of the options file to be used (here HOT2000.options). If a different options file is to be used with updated performance or cost data, this path should be adjusted accordingly.

The `Command` section defines the location of the GenOpt command file:

```
// Optimization configuration.
Optimization{
   Files{
      Command {
         File1 = "Genopt-H2K-CMD.GO-cmd";

      }
   }
}
```

Users may define different command files for specific analysis tasks; this path and file name should be updated accordingly.

The command (.GO-cmd) file

The command file defines how each attribute should be varied during a GenOpt batch run, and the strategy that should be used to examine the solution space. The following is an excerpt of the vary section:

```
Vary{

   // ============================================
   // Parameters that configure the simulation
   // ============================================

   // Location
   Parameter{
      Name = GOtag:Opt-Location;
      Ini  = 1;
      Values = " ABBOTSFORD, PrinceGeorge";
   }

   // ================================================
   // Parameters that change the building design
   // ================================================

   // Archetype definition
   Parameter{
      Name = GOtag:Opt-Archetype;
      Ini  = 1;
      Values = "NZEH-Arch-1";
   }

   // Setting the ACH in the AIM-2 input file
   Parameter{
      Name = GOtag:Opt-ACH;
      Ini  = 1;
      Values = "ACH_2_5, ACH_1_75, ACH_1_5, ACH_1_25";
   }
```

In this example, the command file instructs GenOpt to alternate the location between Abbotsford and Prince George, and the air-tightness between values of 2.5, 1.75. 1.5 and 1.25 ACH.

The command file specifies the algorithm that GenOpt will use to explore the solution space: three algorithms are commonly used:

- **Parametric:** GenOpt evaluates the sensitivity of the model to the proposed changes by varying each parameter one at a time
- **Mesh:** GenOpt evaluates all combinations of parameters in the CMD file to fully explore the solution space
- **Optimization:** GenOpt uses an optimization algorithm to efficiently explore the solution space and narrow in on solutions that satisfy a specific criteria (e.g. lowest cost or most efficient) without evaluating every combination.

These algorithms are specified in the `Algorithm` section of the command file. For a parametric run, the section is set as follows:

```
Algorithm{
  Main = Parametric;
  StopAtError = true;
}
```

For a mesh run, the algorithm section is set as follows:

```
Algorithm{
  Main = Mesh;
  StopAtError = true;
}
```

While GenOpt provides a number of different optimization algorithms, only one – Particle Swarm Optimization with Initial Weight (PSOIW) is compatible with the fully discrete approach used in HTAP. That algorithm is defined as follows:

```
Algorithm{
  Main                      = PSOIW;
  NeighborhoodTopology      = vonNeumann;
  NeighborhoodSize          = 24;         // Disregarded for vonNeumann topology
  NumberOfParticle          = 23;
  NumberOfGeneration        = 2000;
  Seed                      = 628;
  CognitiveAcceleration     = 2;          // 0 < CognitiveAcceleration
  SocialAcceleration        = 3;          // 0 < SocialAcceleration
  MaxVelocityGainContinuous = 1.0;
  MaxVelocityDiscrete       = 1.0;        // 0 < MaxVelocityDiscrete
  InitialInertiaWeight      = 1.2;        // 0 < InitialInertiaWeight
  FinalInertiaWeight        = 0;          // 0 < FinalInertiaWeight
}
```

The GenOpt documentation provides more information on configuring the PSOIW algorithm.

Running GenOpt

Running GenOpt produces the following:

```
PS C:\HTAP> java -classpath "C:\Program Files\genopt\genopt.jar" genopt.GenOpt Genopt-H2K-INI.GO-ini

_____
```

```
GenOpt(R) 3.1.0, December 8, 2011

GenOpt Copyright (c) 1998-2011, The Regents of the
University of California, through Lawrence Berkeley
National Laboratory (subject to receipt of any
required approvals from the U.S. Dept. of Energy).
All rights reserved.

The development of GenOpt is supported by
the U.S. Department of Energy (DOE),
the Swiss Academy of Engineering Sciences (SATW),
the Swiss National Energy Fund (NEFF), and
the Swiss National Science Foundation (SNSF).

Developed by
Lawrence Berkeley National Laboratory
http://simulationresearch.lbl.gov
Michael Wetter, MWetter@lbl.gov

_______________________________________________________________
Assigning 4 threads for simulations.
Require 24 function evaluations.

   ----------------------------------------
                  ^
                  |

        A lot of output appears here

                  |
                  V
   ----------------------------------------

Simulation 24: SimplePaybackYrs = 3299.3
Simulation 24: EnergyTotal      = 126.0
Simulation 24: UtilBillNoPVRevenueDoll  = 2919.2
Simulation 24: UtilRevenuePVDoll        = 0.0
Simulation 24: UtilBillNetDoll  = 2919.23
Simulation 24: UtilCostElecDoll = 1708.54
Simulation 24: UtilCostGasDoll  = 1210.69
Simulation 24: UtilCostPropaneDoll      = 0.0
Simulation 24: UtilCostOilDoll  = 0.0
Simulation 24: EnergyPVkWh      = 0.0
Simulation 24: EnergyHeatingGJ  = 74.4
Simulation 24: EnergyCoolingGJ  = 1.9
Simulation 24: EnergyVentGJ     = 2.4
Simulation 24: EnergyDHWGJ      = 20.4
Simulation 24: EnergyPlugGJ     = 31.5
Simulation 24: FuelEleckWh      = 16327.3
Simulation 24: FuelNaturalGasM3 = 1990.4
Simulation 24: FuelOilL = 0.0
Simulation 24: UpgradeCostDoll  = 9501.0
Simulation 24: PVSizekW = 0.0
Simulation 24: PEAKHeatingW     = 12225.5
Simulation 24: ERS-Value        = 0.0
Simulation 24: NumTries = 1.0
GenOpt completed successfully.
PS C:\HTAP>
```

Processing output

GenOpt writes output to a text file named OuptutListingAll.txt. HTAP includes a ruby script, recover-results.rb to convert these into a .csv format:

```
PS C:\HTAP> .\recover-results.rb -l
Recovering results from TempResultsBatch1.txt
Recovered 24 lines from 1 files.
Results written to file CloudResultsAllData.csv
PS C:\HTAP>
```

recover-results.rb will produce a file called CloudResultsAllData.csv, which contains the GenOpt output.

Troubleshooting

TODO: Write this section .

Contents of the HTAP directory

6	Applications	Contains examples, and files associated with NRCan projects[1]
6	Archetypes	Contains HOT2000 archetype files (.h2k), as well as fuel and construction libraries
6	Doc	Contains documentation on HTAP
6	GenerateChoiceFiles	Contains a perl script[2] that configure HTAP to run a series of scenarios defined in a .csv file.
	Genopt-GENERIC-CONFIG.GO-config	GenOpt file defining configuration options
	Genopt-H2K-CMD.GO-cmd	GenOpt file defining how GenOpt should manipulate HTAP inputs during optimization
	Genopt-H2K-INI.GO-ini	GenOpt file that controls how GenOpt invokes HTAP and how HTAP output should be parsed
	HOT2000.choices	Sample .choice file that defines the parameters for a single HTAP simulation
	HOT2000.options	Sample .options file that defines the valid options for each HTAP parameter.
	HTAP-Template.choices	Template used to by GenOpt to generate files that can be read by HTAP.
	HTAP-Template-MakeChoices.choices	Template used to by GenOpt to generate files that can be read by when running archetypes defined in a .csv file

[1] These may be implemented as a separate git submodule in the future.
[2] Future work will re-write this perl script in ruby.

	recover-results.rb	Ruby script that converts GenOpt output into .csv format
	runH2K.rb	Ruby script that invokes HOT2000.
	start-substitute.rb	Wrapper to substitute-h2k.rb for use when running archetypes defined in a .csv file
	substitute-h2k.rb	Pre- and post-processor used to invoke h2k simulations