MANUAL

INL/EXT-15-34123
Revision 1
Printed April 1, 2020

HERON User Manual

Cristian Rabiti, Paul W. Talbot, Andrea Alfonsi

Prepared by
Idaho National Laboratory
Idaho Falls, Idaho 83415

The Idaho National Laboratory is a multiprogram laboratory operated by Battelle Energy Alliance for the United States Department of Energy under DOE Idaho Operations Office. Contract DE-AC07-05ID14517.

Approved for unlimited release.

Issued by the Idaho National Laboratory, operated for the United States Department of Energy by Battelle Energy Alliance.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

INL/EXT-15-34123
Revision 1
Printed April 1, 2020

HERON User Manual

Project Manager:
Cristian Rabiti
Principal Investigator and Technical Leader:
Paul W. Talbot
Main Developers:
Paul W. Talbot
Andrea Alfonsi

Contents

1 Introduction

HERON is a generic software plugin of RAVEN to perform stochastic technoeconomic analysis of grid energy-resource systems with economic drivers. The development targets analysis of electricity and secondary product generation and consumption in regional balancing areas, including flexibility to include arbitrary resources as well as arbitrary resource consumers and producers. HERON is developed to drive optimization via economic drivers such as system cost minimization, profitability, and net present value (NPV) maximization. As a plugin of RAVEN, HERON provides two primary functions: the automatic generation of RAVEN workflows, and models for optimizing high-resolution dispatch of arbitrary systems including resources, resource consumers, and resource producers. HERON leverages the synthetic history training and generation tools, sampling workflows, code Application Programming Interfaces (API), and optimization schemes.

2 Installation and how to run

2.1 Installation

As a plugin of RAVEN, HERON is installed as a submodule. RAVEN maintains up-to-date instructions for plugin installation in its manuals and other documentation. As of this writing, plugins are installed from the command line from within the top level of the RAVEN repository as

```
scripts/install_plugins.py -s heron
```

Note that RAVEN must be installed completely to use HERON and its components. At this writing HERON introduces one additional Python library to the standard set of RAVEN `Python` dependencies: `dill`. The serialization package `dill` enables information transfer between the HERON templating algorithm and dispatching algorithm not possible with the standard library set. This library, along with any future additional libraries required, should be installed correctly via RAVEN's documented installation procedure for including plugins. HERON also requires use of the RAVEN plugintCashFlow , a utility for economic analysis. As of this writing, `CashFlow` is an officially supported plugin for RAVEN and is installed standard when plugins are included in RAVEN installation.

2.2 How to run

The code can be run easily on a `linux/mac` system by defining an alias to the `Python` script '`main.py`' in the `src` folder of HERON.

3 Cases Introduction

HERON typically relies on a test node which informs the algorithm as to how the case has to be processed by using the predefined metrics described in the following sections.

3.1 Case

The <Case> contains the basic parameters needed for a HERON case.

The <Case> node recognizes the following parameters:

- name: ***string, required***, An appropriate user defined name of the case.

The <Case> node recognizes the following subnodes:

- <mode>: ***[min, max, sweep]***, Minimize, maximize or sweep over multiple values of capacities.
- <metric>: ***[NPV, lcoe]***, This metric can be NPV (Net Present Value) and lcoe (levelized cost of energy) used for techno-economic analysis of the power plants.
- <differential>: ***[yes, y, true, t, si, vero, dajie, oui, ja, yao, verum, evet, dogru, 1, on, no, n, false, f, nono, falso, nahh, non, nicht, bu, falsus, hayir, yanlis, 0, off, Yes, Y, True, T, Si, Vero, Dajie, Oui, Ja, Yao, Verum, Evet, Dogru, 1, On, No, N, False, F, Nono, Falso, Nahh, Non, Nicht, Bu, Falsus, Hayir, Yanlis, 0, Off]***, Differential represents the additional cashflow generated when building additional capacities. This value can be either **'True'** or **'False'**.
- <num_arma_samples>: ***integer***, Number of copies of the trained signals.
- <timestep_interval>: ***integer***, Time step interval between two values of signal.
- <history_length>: ***integer***, Total length of one realization of the ARMA signal.
- <economics>: <economics> contains the details of the econometrics computations to be performed by the code.

  The <economics> node recognizes the following subnodes:

  - <ProjectTime>: ***float***, Total length of the project.
  - <DiscountRate>: ***float***, Interest rate required to compute the discounted cashflow (DCF)

  - **<tax>**: ***float***, Taxation rate is a metric which represents the rate at which an individual or corporation is taxed.
  - **<inflation>**: ***float***, Inflation rate is a metric which represents the the rate at which the average price level of a basket of selected goods and services in an economy increases over some period of time.
  - **<verbosity>**: ***integer***, Length of the output argument.

- **<dispatch_increment>**: ***float***, This is the amount of resource to be dispatched in a fixed time interval. The **<dispatch_increment>** node recognizes the following parameters:

  - **resource**: ***string, required***, Resource to be consumed or produced.

4 Economics Introduction

The economics node describes the basic metrics used to compute the key economic parameters for the techno-economic feasibility of the power production and storage units.

4.1 CashFlow

– no description yet –

The `<CashFlow>` node recognizes the following parameters:

- `name`: ***string, required***, – no description yet –
- `type`: ***string, required***, – no description yet –
- `taxable`: ***[yes, y, true, t, si, vero, dajie, oui, ja, yao, verum, evet, dogru, 1, on, no, n, false, f, nono, falso, nahh, non, nicht, bu, falsus, hayir, yanlis, 0, off, Yes, Y, True, T, Si, Vero, Dajie, Oui, Ja, Yao, Verum, Evet, Dogru, 1, On, No, N, False, F, Nono, Falso, Nahh, Non, Nicht, Bu, Falsus, Hayir, Yanlis, 0, Off], required***, – no description yet –
- `inflation`: ***string, required***, – no description yet –
- `mult_target`: ***[yes, y, true, t, si, vero, dajie, oui, ja, yao, verum, evet, dogru, 1, on, no, n, false, f, nono, falso, nahh, non, nicht, bu, falsus, hayir, yanlis, 0, off, Yes, Y, True, T, Si, Vero, Dajie, Oui, Ja, Yao, Verum, Evet, Dogru, 1, On, No, N, False, F, Nono, Falso, Nahh, Non, Nicht, Bu, Falsus, Hayir, Yanlis, 0, Off], required***, – no description yet –
- `period`: ***[hour, year], optional***, – no description yet –

The `<CashFlow>` node recognizes the following subnodes:

- `<driver>`: The node `<driver>` has the following *ValuedParams* options:

  The `<driver>` node recognizes the following subnodes:

  – `<fixed_value>`: ***float***, It can be a fixed value.

  – `<sweep_values>`: ***comma-separated floats***, It can be a value which is to be swept over multiple values.

  – `<opt_bounds>`: ***comma-separated floats***, Opt bounds.

  – `<ARMA>`: ***string***, It can be an **A**uto **R**egressive **M**oving **A**verage value. The `<ARMA>` node recognizes the following parameters:

    - `variable`: ***string, optional***, Variable generated by an ARMA model.
  - `<Function>`: ***string***, It can be a value generated by running a function. The `<Function>` node recognizes the following parameters:
    - `method`: ***string, optional***, The method containing the function.
  - `<variable>`: ***string***, Variable
  - `<growth>`: ***float***, Growth factor required to grow the variable from one year to another. The `<growth>` node recognizes the following parameters:
    - `mode`: ***[linear, exponential], optional***, The growth mode can be linear or exponential.

- `<reference_price>`: The node `<reference_price>` has the following *ValuedParams* options:

  The `<reference_price>` node recognizes the following subnodes:

  - `<fixed_value>`: ***float***, It can be a fixed value.
  - `<sweep_values>`: ***comma-separated floats***, It can be a value which is to be swept over multiple values.
  - `<opt_bounds>`: ***comma-separated floats***, Opt bounds.
  - `<ARMA>`: ***string***, It can be an **A**uto **R**egressive **M**oving **A**verage value. The `<ARMA>` node recognizes the following parameters:
    - `variable`: ***string, optional***, Variable generated by an ARMA model.
  - `<Function>`: ***string***, It can be a value generated by running a function. The `<Function>` node recognizes the following parameters:
    - `method`: ***string, optional***, The method containing the function.
  - `<variable>`: ***string***, Variable
  - `<growth>`: ***float***, Growth factor required to grow the variable from one year to another. The `<growth>` node recognizes the following parameters:
    - `mode`: ***[linear, exponential], optional***, The growth mode can be linear or exponential.

- `<reference_driver>`: The node `<reference_driver>` has the following *ValuedParams* options:

  The `<reference_driver>` node recognizes the following subnodes:

  - `<fixed_value>`: ***float***, It can be a fixed value.
  - `<sweep_values>`: ***comma-separated floats***, It can be a value which is to be swept over multiple values.
  - `<opt_bounds>`: ***comma-separated floats***, Opt bounds.

  - **<ARMA>**: ***string***, It can be an **A**uto **R**egressive **M**oving **A**verage value. The **<ARMA>** node recognizes the following parameters:
    - **variable**: ***string, optional***, Variable generated by an ARMA model.
  - **<Function>**: ***string***, It can be a value generated by running a function. The **<Function>** node recognizes the following parameters:
    - **method**: ***string, optional***, The method containing the function.
  - **<variable>**: ***string***, Variable
  - **<growth>**: ***float***, Growth factor required to grow the variable from one year to another. The **<growth>** node recognizes the following parameters:
    - **mode**: ***[linear, exponential], optional***, The growth mode can be linear or exponential.

- **<scaling_factor_x>**: The node **<scaling_factor_x>** has the following *Valued-Params* options:

  The **<scaling_factor_x>** node recognizes the following subnodes:

  - **<fixed_value>**: ***float***, It can be a fixed value.
  - **<sweep_values>**: ***comma-separated floats***, It can be a value which is to be swept over multiple values.
  - **<opt_bounds>**: ***comma-separated floats***, Opt bounds.
  - **<ARMA>**: ***string***, It can be an **A**uto **R**egressive **M**oving **A**verage value. The **<ARMA>** node recognizes the following parameters:
    - **variable**: ***string, optional***, Variable generated by an ARMA model.
  - **<Function>**: ***string***, It can be a value generated by running a function. The **<Function>** node recognizes the following parameters:
    - **method**: ***string, optional***, The method containing the function.
  - **<variable>**: ***string***, Variable
  - **<growth>**: ***float***, Growth factor required to grow the variable from one year to another. The **<growth>** node recognizes the following parameters:
    - **mode**: ***[linear, exponential], optional***, The growth mode can be linear or exponential.

- **<depreciate>**: ***integer***, – no description yet –

4.2 Component

The **<component>** represents the component which is a part or element of a larger whole, use to produce, consume one source of energy and produce another.

The `<Component>` node recognizes the following parameters:

- `name`: ***string, required***, Name of the component

The `<Component>` node recognizes the following subnodes:

- `<produces>`: Produces a resource by consuming any generic fuel. The `<produces>` node recognizes the following parameters:
  - `resource`: ***comma-separated strings, required***, Resource to be consumed or produced.
  - `dispatch`: ***[independent, dependent, fixed], optional***, Amount to be dispatched.

  The `<produces>` node recognizes the following subnodes:
  - `<capacity>`: The node `<capacity>` has the following *ValuedParams* options: The `<capacity>` node recognizes the following parameters:
    - `resource`: ***string, optional***, Resources to be consumed or produced.

    The `<capacity>` node recognizes the following subnodes:
    - `<fixed_value>`: ***float***, It can be a fixed value.
    - `<sweep_values>`: ***comma-separated floats***, It can be a value which is to be swept over multiple values.
    - `<opt_bounds>`: ***comma-separated floats***, Opt bounds.
    - `<ARMA>`: ***string***, It can be an **A**uto **R**egressive **M**oving **A**verage value. The `<ARMA>` node recognizes the following parameters:
      - `variable`: ***string, optional***, Variable generated by an ARMA model.
    - `<Function>`: ***string***, It can be a value generated by running a function. The `<Function>` node recognizes the following parameters:
      - `method`: ***string, optional***, The method containing the function.
    - `<variable>`: ***string***, Variable
    - `<growth>`: ***float***, Growth factor required to grow the variable from one year to another. The `<growth>` node recognizes the following parameters:
      - `mode`: ***[linear, exponential], optional***, The growth mode can be linear or exponential.
  - `<minimum>`: The node `<minimum>` has the following *ValuedParams* options: The `<minimum>` node recognizes the following parameters:
    - `resource`: ***string, optional***, – no description yet –

    The `<minimum>` node recognizes the following subnodes:
    - `<fixed_value>`: ***float***, It can be a fixed value.

    - **`<sweep_values>`**: ***comma-separated floats***, It can be a value which is to be swept over multiple values.
    - **`<opt_bounds>`**: ***comma-separated floats***, Opt bounds.
    - **`<ARMA>`**: ***string***, It can be an **A**uto **R**egressive **M**oving **A**verage value. The **`<ARMA>`** node recognizes the following parameters:
      - `variable`: ***string, optional***, Variable generated by an ARMA model.
    - **`<Function>`**: ***string***, It can be a value generated by running a function. The **`<Function>`** node recognizes the following parameters:
      - `method`: ***string, optional***, The method containing the function.
    - **`<variable>`**: ***string***, Variable
    - **`<growth>`**: ***float***, Growth factor required to grow the variable from one year to another. The **`<growth>`** node recognizes the following parameters:
      - `mode`: ***[linear, exponential], optional***, The growth mode can be linear or exponential.
  - **`<consumes>`**: ***comma-separated strings***, The producer can either produce or consume a resource. If the producer is a consumer it must be accompnied with a transfer function to convert one source of energy to another.
  - **`<transfer>`**: The node **`<transfer>`** has the following *ValuedParams* options:

    The **`<transfer>`** node recognizes the following subnodes:

    - **`<fixed_value>`**: ***float***, It can be a fixed value.
    - **`<sweep_values>`**: ***comma-separated floats***, It can be a value which is to be swept over multiple values.
    - **`<opt_bounds>`**: ***comma-separated floats***, Opt bounds.
    - **`<ARMA>`**: ***string***, It can be an **A**uto **R**egressive **M**oving **A**verage value. The **`<ARMA>`** node recognizes the following parameters:
      - `variable`: ***string, optional***, Variable generated by an ARMA model.
    - **`<Function>`**: ***string***, It can be a value generated by running a function. The **`<Function>`** node recognizes the following parameters:
      - `method`: ***string, optional***, The method containing the function.
    - **`<variable>`**: ***string***, Variable
    - **`<growth>`**: ***float***, Growth factor required to grow the variable from one year to another. The **`<growth>`** node recognizes the following parameters:
      - `mode`: ***[linear, exponential], optional***, The growth mode can be linear or exponential.

- **`<stores>`**: Stores the energy in a battery. The **`<stores>`** node recognizes the following parameters:

  - `resource`: ***comma-separated strings, required***, Resource to be consumed or produced.

- `dispatch`: ***[independent, dependent, fixed], optional***, Amount to be dispatched.

The `<stores>` node recognizes the following subnodes:

- `<capacity>`: The node `<capacity>` has the following *ValuedParams* options: The `<capacity>` node recognizes the following parameters:
  - `resource`: ***string, optional***, Resources to be consumed or produced.

  The `<capacity>` node recognizes the following subnodes:
  - `<fixed_value>`: ***float***, It can be a fixed value.
  - `<sweep_values>`: ***comma-separated floats***, It can be a value which is to be swept over multiple values.
  - `<opt_bounds>`: ***comma-separated floats***, Opt bounds.
  - `<ARMA>`: ***string***, It can be an **A**uto **R**egressive **M**oving **A**verage value. The `<ARMA>` node recognizes the following parameters:
    - `variable`: ***string, optional***, Variable generated by an ARMA model.
  - `<Function>`: ***string***, It can be a value generated by running a function. The `<Function>` node recognizes the following parameters:
    - `method`: ***string, optional***, The method containing the function.
  - `<variable>`: ***string***, Variable
  - `<growth>`: ***float***, Growth factor required to grow the variable from one year to another. The `<growth>` node recognizes the following parameters:
    - `mode`: ***[linear, exponential], optional***, The growth mode can be linear or exponential.
- `<minimum>`: The node `<minimum>` has the following *ValuedParams* options: The `<minimum>` node recognizes the following parameters:
  - `resource`: ***string, optional***, – no description yet –

  The `<minimum>` node recognizes the following subnodes:
  - `<fixed_value>`: ***float***, It can be a fixed value.
  - `<sweep_values>`: ***comma-separated floats***, It can be a value which is to be swept over multiple values.
  - `<opt_bounds>`: ***comma-separated floats***, Opt bounds.
  - `<ARMA>`: ***string***, It can be an **A**uto **R**egressive **M**oving **A**verage value. The `<ARMA>` node recognizes the following parameters:
    - `variable`: ***string, optional***, Variable generated by an ARMA model.
  - `<Function>`: ***string***, It can be a value generated by running a function. The `<Function>` node recognizes the following parameters:
    - `method`: ***string, optional***, The method containing the function.
  - `<variable>`: ***string***, Variable

    - `<growth>`: ***float***, Growth factor required to grow the variable from one year to another. The `<growth>` node recognizes the following parameters:
        - `mode`: ***[linear, exponential], optional***, The growth mode can be linear or exponential.

- `<rate>`: The node `<rate>` has the following *ValuedParams* options:

  The `<rate>` node recognizes the following subnodes:

    - `<fixed_value>`: ***float***, It can be a fixed value.
    - `<sweep_values>`: ***comma-separated floats***, It can be a value which is to be swept over multiple values.
    - `<opt_bounds>`: ***comma-separated floats***, Opt bounds.
    - `<ARMA>`: ***string***, It can be an **A**uto **R**egressive **M**oving **A**verage value. The `<ARMA>` node recognizes the following parameters:
        - `variable`: ***string, optional***, Variable generated by an ARMA model.
    - `<Function>`: ***string***, It can be a value generated by running a function. The `<Function>` node recognizes the following parameters:
        - `method`: ***string, optional***, The method containing the function.
    - `<variable>`: ***string***, Variable
    - `<growth>`: ***float***, Growth factor required to grow the variable from one year to another. The `<growth>` node recognizes the following parameters:
        - `mode`: ***[linear, exponential], optional***, The growth mode can be linear or exponential.

- `<initial_stored>`: The node `<initial_stored>` has the following *ValuedParams* options:

  The `<initial_stored>` node recognizes the following subnodes:

    - `<fixed_value>`: ***float***, It can be a fixed value.
    - `<sweep_values>`: ***comma-separated floats***, It can be a value which is to be swept over multiple values.
    - `<opt_bounds>`: ***comma-separated floats***, Opt bounds.
    - `<ARMA>`: ***string***, It can be an **A**uto **R**egressive **M**oving **A**verage value. The `<ARMA>` node recognizes the following parameters:
        - `variable`: ***string, optional***, Variable generated by an ARMA model.
    - `<Function>`: ***string***, It can be a value generated by running a function. The `<Function>` node recognizes the following parameters:
        - `method`: ***string, optional***, The method containing the function.
    - `<variable>`: ***string***, Variable
    - `<growth>`: ***float***, Growth factor required to grow the variable from one year to another. The `<growth>` node recognizes the following parameters:
        - `mode`: ***[linear, exponential], optional***, The growth mode can be linear or exponential.

- **<demands>**: Demands a resource which it consumes. The **<demands>** node recognizes the following parameters:

  - resource: ***comma-separated strings, required***, Resource to be consumed or produced.
  - dispatch: ***[independent, dependent, fixed], optional***, Amount to be dispatched.

  The **<demands>** node recognizes the following subnodes:

  - **<capacity>**: The node **<capacity>** has the following *ValuedParams* options: The **<capacity>** node recognizes the following parameters:

    - resource: ***string, optional***, Resources to be consumed or produced.

    The **<capacity>** node recognizes the following subnodes:

    - **<fixed_value>**: ***float***, It can be a fixed value.
    - **<sweep_values>**: ***comma-separated floats***, It can be a value which is to be swept over multiple values.
    - **<opt_bounds>**: ***comma-separated floats***, Opt bounds.
    - **<ARMA>**: ***string***, It can be an **A**uto **R**egressive **M**oving **A**verage value. The **<ARMA>** node recognizes the following parameters:
      - variable: ***string, optional***, Variable generated by an ARMA model.
    - **<Function>**: ***string***, It can be a value generated by running a function. The **<Function>** node recognizes the following parameters:
      - method: ***string, optional***, The method containing the function.
    - **<variable>**: ***string***, Variable
    - **<growth>**: ***float***, Growth factor required to grow the variable from one year to another. The **<growth>** node recognizes the following parameters:
      - mode: ***[linear, exponential], optional***, The growth mode can be linear or exponential.

  - **<minimum>**: The node **<minimum>** has the following *ValuedParams* options: The **<minimum>** node recognizes the following parameters:

    - resource: ***string, optional***, – no description yet –

    The **<minimum>** node recognizes the following subnodes:

    - **<fixed_value>**: ***float***, It can be a fixed value.
    - **<sweep_values>**: ***comma-separated floats***, It can be a value which is to be swept over multiple values.
    - **<opt_bounds>**: ***comma-separated floats***, Opt bounds.
    - **<ARMA>**: ***string***, It can be an **A**uto **R**egressive **M**oving **A**verage value. The **<ARMA>** node recognizes the following parameters:
      - variable: ***string, optional***, Variable generated by an ARMA model.

    - <Function>: ***string***, It can be a value generated by running a function. The <Function> node recognizes the following parameters:
      - method: ***string, optional***, The method containing the function.
    - <variable>: ***string***, Variable
    - <growth>: ***float***, Growth factor required to grow the variable from one year to another. The <growth> node recognizes the following parameters:
      - mode: ***[linear, exponential], optional***, The growth mode can be linear or exponential.
  - <penalty>: The node <penalty> has the following *ValuedParams* options:

    The <penalty> node recognizes the following subnodes:
    - <fixed_value>: ***float***, It can be a fixed value.
    - <sweep_values>: ***comma-separated floats***, It can be a value which is to be swept over multiple values.
    - <opt_bounds>: ***comma-separated floats***, Opt bounds.
    - <ARMA>: ***string***, It can be an **A**uto **R**egressive **M**oving **A**verage value. The <ARMA> node recognizes the following parameters:
      - variable: ***string, optional***, Variable generated by an ARMA model.
    - <Function>: ***string***, It can be a value generated by running a function. The <Function> node recognizes the following parameters:
      - method: ***string, optional***, The method containing the function.
    - <variable>: ***string***, Variable
    - <growth>: ***float***, Growth factor required to grow the variable from one year to another. The <growth> node recognizes the following parameters:
      - mode: ***[linear, exponential], optional***, The growth mode can be linear or exponential.

- <economics>: The <Economics> contains the attributes required to compute the key economic metrics

  The <economics> node recognizes the following subnodes:
  - <lifetime>: ***integer***, – no description yet –
  - <CashFlow>: – no description yet – The <CashFlow> node recognizes the following parameters:
    - name: ***string, required***, – no description yet –
    - type: ***string, required***, – no description yet –
    - taxable: ***[yes, y, true, t, si, vero, dajie, oui, ja, yao, verum, evet, dogru, 1, on, no, n, false, f, nono, falso, nahh, non, nicht, bu, falsus, hayir, yanlis, 0, off, Yes, Y, True, T, Si, Vero, Dajie, Oui, Ja, Yao, Verum, Evet, Dogru, 1, On, No, N, False, F, Nono, Falso, Nahh, Non, Nicht, Bu, Falsus, Hayir, Yanlis, 0, Off], required***, – no description yet –

  - `inflation`: ***string, required***, – no description yet –
  - `mult_target`: ***[yes, y, true, t, si, vero, dajie, oui, ja, yao, verum, evet, dogru, 1, on, no, n, false, f, nono, falso, nahh, non, nicht, bu, falsus, hayir, yanlis, 0, off, Yes, Y, True, T, Si, Vero, Dajie, Oui, Ja, Yao, Verum, Evet, Dogru, 1, On, No, N, False, F, Nono, Falso, Nahh, Non, Nicht, Bu, Falsus, Hayir, Yanlis, 0, Off], required***, – no description yet –
  - `period`: ***[hour, year], optional***, – no description yet –

The `<CashFlow>` node recognizes the following subnodes:

- `<driver>`: The node `<driver>` has the following *ValuedParams* options:
  The `<driver>` node recognizes the following subnodes:
  - `<fixed_value>`: ***float***, It can be a fixed value.
  - `<sweep_values>`: ***comma-separated floats***, It can be a value which is to be swept over multiple values.
  - `<opt_bounds>`: ***comma-separated floats***, Opt bounds.
  - `<ARMA>`: ***string***, It can be an **A**uto **R**egressive **M**oving **A**verage value. The `<ARMA>` node recognizes the following parameters:
  - `variable`: ***string, optional***, Variable generated by an ARMA model.
  - `<Function>`: ***string***, It can be a value generated by running a function. The `<Function>` node recognizes the following parameters:
  - `method`: ***string, optional***, The method containing the function.
  - `<variable>`: ***string***, Variable
  - `<growth>`: ***float***, Growth factor required to grow the variable from one year to another. The `<growth>` node recognizes the following parameters:
  - `mode`: ***[linear, exponential], optional***, The growth mode can be linear or exponential.
- `<reference_price>`: The node `<reference_price>` has the following *ValuedParams* options:
  The `<reference_price>` node recognizes the following subnodes:
  - `<fixed_value>`: ***float***, It can be a fixed value.
  - `<sweep_values>`: ***comma-separated floats***, It can be a value which is to be swept over multiple values.
  - `<opt_bounds>`: ***comma-separated floats***, Opt bounds.
  - `<ARMA>`: ***string***, It can be an **A**uto **R**egressive **M**oving **A**verage value. The `<ARMA>` node recognizes the following parameters:
  - `variable`: ***string, optional***, Variable generated by an ARMA model.
  - `<Function>`: ***string***, It can be a value generated by running a function. The `<Function>` node recognizes the following parameters:

- `method`: ***string, optional***, The method containing the function.

- `<variable>`: ***string***, Variable

- `<growth>`: ***float***, Growth factor required to grow the variable from one year to another. The `<growth>` node recognizes the following parameters:

- `mode`: ***[linear, exponential], optional***, The growth mode can be linear or exponential.

* `<reference_driver>`: The node `<reference_driver>` has the following *ValuedParams* options:
  The `<reference_driver>` node recognizes the following subnodes:
  - `<fixed_value>`: ***float***, It can be a fixed value.
  - `<sweep_values>`: ***comma-separated floats***, It can be a value which is to be swept over multiple values.
  - `<opt_bounds>`: ***comma-separated floats***, Opt bounds.
  - `<ARMA>`: ***string***, It can be an **A**uto **R**egressive **M**oving **A**verage value. The `<ARMA>` node recognizes the following parameters:
  - `variable`: ***string, optional***, Variable generated by an ARMA model.
  - `<Function>`: ***string***, It can be a value generated by running a function. The `<Function>` node recognizes the following parameters:
  - `method`: ***string, optional***, The method containing the function.
  - `<variable>`: ***string***, Variable
  - `<growth>`: ***float***, Growth factor required to grow the variable from one year to another. The `<growth>` node recognizes the following parameters:
  - `mode`: ***[linear, exponential], optional***, The growth mode can be linear or exponential.
* `<scaling_factor_x>`: The node `<scaling_factor_x>` has the following *ValuedParams* options:
  The `<scaling_factor_x>` node recognizes the following subnodes:
  - `<fixed_value>`: ***float***, It can be a fixed value.
  - `<sweep_values>`: ***comma-separated floats***, It can be a value which is to be swept over multiple values.
  - `<opt_bounds>`: ***comma-separated floats***, Opt bounds.
  - `<ARMA>`: ***string***, It can be an **A**uto **R**egressive **M**oving **A**verage value. The `<ARMA>` node recognizes the following parameters:
  - `variable`: ***string, optional***, Variable generated by an ARMA model.
  - `<Function>`: ***string***, It can be a value generated by running a function. The `<Function>` node recognizes the following parameters:
  - `method`: ***string, optional***, The method containing the function.

Idaho National Laboratory