# flexvalue

*Release 2.0.8*

**Recurve Analytics, Inc.**

**Feb 01, 2024**

# CONTENTS

CHAPTER

ONE

# SUBMODULES

## 1.1 flexvalue.__version__

## 1.2 flexvalue.cli

### 1.2.1 flexvalue

```
flexvalue [OPTIONS] COMMAND [ARGS]...
```

### get-results

```
flexvalue get-results [OPTIONS]
```

#### Options

**--project-info-file** <project_info_file>

Filepath to the project information file that is used to calculate results

**--database-type** <database_type>

One of 'postgresql', 'bigquery', or 'sqlite'

**--host** <host>

The host for the postgresql database to which you are connecting.

**--port** <port>

The port for the postgresql database to which you are connecting.

**--user** <user>

The user for the postgresql database to which you are connecting.

**--password** <password>

The password for the postgresql database to which you are connecting.

**--database** <database>

The database for the postgresql database to which you are connecting.

**--elec-av-costs-table** <elec_av_costs_table>

Used when –database-type is bigquery. Specifies the electric avoided costs table. Must specify the dataset and the google project (if different than the –project argument).

**--elec-load-shape-table** <elec_load_shape_table>

Used when –database-type is bigquery. Specifies the electric load shape table. Must specify the dataset and the google project (if different than the –project argument).

**--gas-av-costs-table** <gas_av_costs_table>

Used when –database-type is bigquery. Specifies the gas avoided costs table. Must specify the dataset and the google project (if different than the –project argument).

**--therms-profiles-table** <therms_profiles_table>

Used when –database-type is bigquery. Specifies the therms profiles table. Must specify the dataset and the google project (if different than the –project argument).

**--project-info-table** <project_info_table>

Used when –database-type is bigquery. Specifies the table containing the project information. Must specify the dataset and the google project (if different than the –project argument).

**--metered-load-shape-table** <metered_load_shape_table>

Used when –database-type is bigquery. Specifies the metered load shape table. Must specify the dataset and the google project (if different than the –project argument).

**--project** <project>

Used when –database-type is bigquery. Specifies the google project.

**`--output-table`** `<output_table>`

The database table to write output to. This table gets overwritten (not appended to). Must specify the dataset and the google project (if different than the –project argument).

**`--separate-output-tables`**

Create two output tables, one for electric benefits and one for gas.

**`--electric-output-table`** `<electric_output_table>`

The database table to write electric output to, when –separate-output-tables=True. This table gets overwritten (not appended to). Must specify the dataset and the google project (if different than the –project argument).

**`--gas-output-table`** `<gas_output_table>`

The database table to write gas output to, when –separate-output-tables=True. This table gets overwritten (not appended to). Must specify the dataset and the google project (if different than the –project argument).

**`--config-file`** `<config_file>`

Path to the toml configuration file.

**`--elec-av-costs-file`** `<elec_av_costs_file>`

Filepath to the electric avoided costs. Used when –database-type is not BigQuery to load this data into the database from a file.

**`--gas-av-costs-file`** `<gas_av_costs_file>`

Filepath to the gas avoided costs. Used when –database-type is not BigQuery to load this data into the database from a file.

**`--elec-load-shape-file`** `<elec_load_shape_file>`

Filepath to the hourly electric load shape file. Used when –database-type is not BigQuery to load this data into the database from a file.

**`--therms-profiles-file`** `<therms_profiles_file>`

Filepath to the therms profiles file. Used when –database-type is not BigQuery to load this data into the database from a file.

**`--metered-load-shape-file`** `<metered_load_shape_file>`

Filepath to the hourly metered load shape file. Used when –database-type is not BigQuery to load this data into the database from a file.

**`--aggregation-columns`** `<aggregation_columns>`

Comma-separated list of field names on which to aggregate the query.

**`--reset-elec-load-shape`**

Reset the data in the electric load shape table. This restores it to its contents prior to running FLEXvalue.

**`--reset-elec-av-costs`**

Reset the data in the electric avoided costs table. This restores it to its contents prior to running FLEXvalue.

**`--reset-therms-profiles`**

Reset the data in the therms profiles table. This restores it to its contents prior to running FLEXvalue.

**`--reset-gas-av-costs`**

Reset the data in the gas avoided costs table. This restores it to its contents prior to running FLEXvalue.

**`--process-elec-load-shape`**

Process (load/transform) the electric load shape data.

**`--process-elec-av-costs`**

Process (load/transform) the electric avoided costs data.

**`--process-therms-profiles`**

Process (load/transform) the therms profiles table.

**`--process-gas-av-costs`**

Process (load/transform) the gas avoided costs data.

**`--process-metered-load-shape`**

Process (load/transform) the metered load shape data.

**`--elec-components`** `<elec_components>`

Comma-separated list of electric avoided cost component field names

**`--gas-components`** `<gas_components>`

Comma-separated list of electric avoided cost component field names

**`--elec-addl-fields`** `<elec_addl_fields>`

Comma-separated list of additional fields from electric data to include in output

**`--gas-addl-fields`** `<gas_addl_fields>`

Comma-separated list of additional fields from gas data to include in output

**`--use-value-curve-name-for-join`**

Specifies that the ACC and project info tables you are using have multiple curves in them, and that FLEXvalue should join based on the curve names.

## 1.3 flexvalue.db

### 1.3.1 Functions

- *DBManager*: The base class for managing database interactions.
- *PostgresqlManager*: Implementation of DBManager for PostgreSQL databases.
- *BigQueryManager*: Implementation of DBManager for Google BigQuery databases.

**class** flexvalue.db.**DBManager**(*fv_config:* FLEXValueConfig)

**class** flexvalue.db.**PostgresqlManager**(*DBManager*)

**class** flexvalue.db.**BigQueryManager**(*DBManager*)

## 1.4 `flexvalue.config`

### 1.4.1 Functions

- *`FLEXValueConfig`*: A dataclass to contain the configuration for FLEXvalue.

**class** `flexvalue.config.`**`FLEXValueConfig`**(*database_type: str*, *host: str = None*, *port: int = None*, *user: str = None*, *password: str = None*, *database: str = None*, *project: str = None*, *elec_load_shape_file: str = None*, *elec_av_costs_file: str = None*, *therms_profiles_file: str = None*, *gas_av_costs_file: str = None*, *project_info_file: str = None*, *metered_load_shape_file: str = None*, *elec_av_costs_table: str = None*, *elec_load_shape_table: str = None*, *therms_profiles_table: str = None*, *gas_av_costs_table: str = None*, *project_info_table: str = None*, *metered_load_shape_table: str = None*, *output_file: str = None*, *output_table: str = None*, *electric_output_table: str = None*, *gas_output_table: str = None*, *aggregation_columns: List[str] = <factory>*, *process_elec_load_shape: bool = False*, *process_elec_av_costs: bool = False*, *process_therms_profiles: bool = False*, *process_gas_av_costs: bool = False*, *process_metered_load_shape: bool = False*, *reset_elec_load_shape: bool = False*, *reset_elec_av_costs: bool = False*, *reset_therms_profiles: bool = False*, *reset_gas_av_costs: bool = False*, *elec_components: List[str] = <factory>*, *gas_components: List[str] = <factory>*, *elec_addl_fields: List[str] = <factory>*, *gas_addl_fields: List[str] = <factory>*, *separate_output_tables: bool = False*, *use_value_curve_name_for_join: bool = False*)

CHAPTER

TWO

# FLEXVALUE™

Read The Docs Link

Read The Docs PDF Link

Github Link

This library provides aggregators, program administrators, utilities, and regulators a pathway to consistently and transparently gauge the value of their projects, portfolios, and programs. Whereas the first version of FLEXvalue was limited to California, this version allows for user-provided avoided cost as well as load data. It defaults to using the CPUC's published avoided cost data to enable market actors to assess demand flexibility value from either pre-defined or custom/measured load shapes. FLEXvalue accepts user-defined 8,760 hourly savings profiles or deemed load shapes that are part of the Database for Energy Efficiency Resources (DEER). See the user_inputs section below for more information. FLEXvalue currently computes Total Resource Cost (TRC) and Program Administrator Cost (PAC) test results. See the California Standard Practice Manual for more information on cost-effectiveness tests.

FLEXvalue supports BigQuery and PostgreSQL as data stores. For more information, see *Data stores*.

Information on usage can be found below in *Usage*.

## 2.1 Avoided Cost Data

A separate series of python scripts were used to compile the avoided cost data from a source XLSX file available through the CPUC's website.

### 2.1.1 deer_load_shapes

The DEER load shapes are normalized 8,760 hourly savings profiles that correspond to different end-use sectors and technologies (residential HVAC or commercial lighting for example). A full list is provided in reference . Annual deemed MWh savings values are typically assigned to a specific DEER load shape. The resulting 8,760 hourly savings values are then multiplied by the hourly electric avoided costs to produce the electric cost effectiveness benefits.

Recurve has made a few changes to column names to incorporate residential and commercial naming conventions but otherwise the format of the DEER load shapes does not need to be updated. Additional formatting is needed because the electric avoided costs begin on a day of the week that does not align with the DEER load shapes. Recurve has conducted extensive testing and has found that the DEER load shapes need to be shifted by -2 days in order to provide the best alignment with the CPUC's existing Cost Effectiveness Tool.

There are different versions of the DEER load shapes for each of the four California IOUs. The specific DEER load shape selected will depend on the combination of user input program administrator and DEER load shape name. It is not known at this point if a different shift will need to be incorporated upon release of the next electric avoided cost calculator.

### 2.1.2 acc_electricity

The electric avoided cost calculator compiles hourly marginal utility avoided costs for electric savings. Costs are provided for ten different cost components and are projected forward through 2050 for the non-adjusted acc versions and through 2043 for the kicker adjusted one. Avoided costs are distinct for each utility service territory (PG&E, SCE, SDG&E, and SoCalGas) and Climate Zone combination. The electric avoided cost calculator also contains hourly marginal greenhouse gas emissions data, which are also forecasted to 2050.

The electric avoided cost calculator can be downloaded as a .xlsb file from here.

The electric avoided cost calculator is a macro-driven Excel file, so several data extraction and transformation steps were done to combine all calculator results into a single table.

### 2.1.3 acc_gas

The gas avoided cost calculator compiles monthly marginal utility avoided costs for gas savings. Costs are provided for four different components and are projected forward through 2050. Avoided costs are distinct for each utility service territory (PG&E, SCE, SDG&E, and SoCalGas). Avoided costs are also somewhat different for distinct end use categories.

The gas avoided cost calculator can be downloaded as a .xlsb file from here.

The gas avoided cost calculator is a macro-driven Excel file, so several data extraction and transformation steps were done to combine all calculator results into a single table.

*Currently the gas avoided costs data uses the same year-month avoided costs (using Utility: PGE, Class: Total Core, End Use: Small Boiler, Emission Control: Uncontrolled) for all analysis. Based on testing against the CET, different numerical adjustment factors are applied depending on the program administrator.*

The adjustment factors applied are: - PGE: 0.933918 - SCE/SCG: 0.868483 - SDGE: 0.930421

## 2.2 Inputs

Input files are only used to populate tables in a local database. Input tables are used when BigQuery is the datastore. The input structures (i.e. columns) are the same, however, as discussed below. Note that when "headers" are mentioned those are only applicable to input files.

**Project information:** A header row is not required for this file. The columns are: id, state,utility, region, mwh_savings, therms_savings, elec_load_shape, therms_profile, start_year, start_quarter, units, eul, ntg, discount_rate, admin_cost, measure_cost, incentive_cost **Metered load shapes:** Headers are required for this file. Its columns are: utility, hour_of_year, followed by any number of load shapes, one per column.

There required inputs are described below. These inputs can be CSV files or BigQuery tables (if running on BigQuery).

### 2.2.1 Electric avoided costs

The columns for the electric avoided cost data are as follows:

- **state**: The state in which the data is valid.
- **utility**: The utility for which the avoided costs apply.
- **region**: The region for which this load shape is valid. This has been used for climate zone data, for example; but the meaning of this is left to the purpose of the user.
- **datetime**: The date and time for this row.

- **year**: The year for this row
- **quarter**: The quarter (of the year) for this row.
- **month**: The month for this row.
- **hour_of_day**: The hour of day for this row.
- **hour_of_year**: The hour of year for this row. From 0 to 8759.
- **energy**: This column and all the following except “total” are the components of the avoided costs for this hour.
- **losses**:
- **ancillary_services**:
- **capacity**:
- **transmission**:
- **distribution**:
- **cap_and_trade**:
- **ghg_adder**:
- **ghg_rebalancing**:
- **methane_leakage**:
- **total**: This is the total avoided cost for this hour. It is the sum of the values in the component columns.
- **marginal_ghg**:
- **ghg_adder_rebalancing**:
- **value_curve_name**: The name of the value curve representing the avoided costs for a given hour. This may be null if only using a single value curve.

### 2.2.2 Gas Avoided Costs

The columns for the gas avoided cost data are as follows:

- **state**: The state in which the data is valid.
- **utility**: The utility for which the avoided costs apply.
- **region**: The region for which this load shape is valid. This has been used for climate zone data, for example; but the meaning of this is left to the purpose of the user.
- **year**: The year for this row
- **quarter**: The quarter (of the year) for this row.
- **month**: The month for this row.
- **market**: This column and the following, except “total”, are the components of the avoided costs for this hour.
- **t_d**:
- **environment**:
- **btm_methane**:
- **total**: This is the total avoided cost for this hour. It is the sum of the values in the component columns.
- **upstream_methane**:
- **marginal_ghg**:

- **value_curve_name**: The name of the value curve representing the avoided costs for a given hour. This may be null if only using a single value curve.

### 2.2.3 Electric Load Shapes

The electric load shape data has a variable number of columns; after the fixed columns, there are N columns with each representing a given load shape. The load shape data always has exactly 8760 rows.

- **state**: The state for which the data is valid.
- **utility**: Which utility to use when joining with the project_info/user_input data for a given project.
- **region**: The region for which this load shape is valid. This has been used for climate zone data, for example; but the meaning of this is left to the purpose of the user.
- **quarter**: The quarter of the year for this row.
- **month**: The month for this row.
- **hour_of_day**: The hour of day (0 to 23) for this row.
- **hour_of_year**: The hour of the year (0 to 8759) for this row.
- **load_shape_1**: The header/name of this column is matched with the load shape name in the project_info/user_inputs data.
- **load_shape_2**:
- …:
- **load_shape_n**:

### 2.2.4 Therms Profiles

- **state**: The state for which the data is valid.
- **utility**: Which utility to use when joining with the project_info/user_input data for a given project.
- **region**: The region for which this load shape is valid. This has been used for climate zone data, for example; but the meaning of this is left to the purpose of the user.
- **quarter**: The quarter of the year for this row.
- **month**: The month for this row.
- **therms_profile_1**: The header/name of this columns is matched with the therms profile name in the project_info/user_inputs data.
- **therms_profile_2**: The header/name of this columns is matched with the therms profile name in the project_info/user_inputs data.
- **…**:
- **therms_profile_n**: The header/name of this columns is matched with the therms profile name in the project_info/user_inputs data.

### 2.2.5 User Inputs (aka project_info)

Recurve has prepared an example user_inputs file that can be downloaded and used.

The *user_inputs* CSV requires the following columns:

- **ID**: A unique identifier used to reference this measure, project, or portfolio
- **load_shape**: the name of the load shape to use (either referencing a column in the *metered_load_shape* file or an available DEER load shape)
- **start_year**: The year to start with when using avoided costs data
- **start_quarter**: The quarter to start with when using avoided costs data
- **utility**: Which uility to filter by when loading avoided costs data
- **climate_zone**: Which climate zone to filter by when loading avoided costs data
- **units**: Multiplier of the therms_savings and mwh_savings (this multiplier is not applied to any of the costs)
- **eul**: Effective Useful Life (EUL): the expected duration in years that the load impacts persist
- **ntg**: Net-to-gross ratio
- **discount_rate**: The quarterly discount rate to be applied within the net present value calculation
- **admin**: The administrative costs assigned to the given measure, project, or portfolio
- **measure**: The measure costs assigned to the given measure, project, or portfolio
- **incentive**: The incentive costs assigned to the given measure, project, or portfolio
- **therms_profile**: Indicates the season in which therms savings are achieved, can be one of [‘annual’, ‘summer’, ‘winter’]
- **therms_savings**: The first year gas gross savings in Therms
- **mwh_savings**: The first year electricity gross savings in MWh (used to scale the load shape savings data if using custom load shape)
- **value_curve_name**: The name of the value curve that the project should be matched to. This may be null if only using a single value curve.

### 2.2.6 Metered Load Shapes

The *metered_load_shape* CSV requires the following columns:

- **hour_of_year**: Hour of the year (should be one row for each of 0-8759)
- **meter_id1**: the savings values (in MWh), with the column name as a reference in the *load_shape* column of the *user_inputs* table (if that measure/project/portfolio has an electricity savings profile associated with meter_id1
- **meter_id2**: the savings values (in MWh), with the column name as a reference in the *load_shape* column of the *user_inputs* table (if that measure/project/portfolio has an electricity savings profile associated with meter_id2
- . . .
- **meter_id_n**: the savings values (in MWh), with the column name as a reference in the *load_shape* column of the *user_inputs* table (if that measure/project/portfolio has an electricity savings profile associated with meter_id_n

### 2.2.7 Metered Load Shapes

If the user-defined load shape is normalized (the sum of values across all 8,760 hours is 1) then the user should input the annual MWh savings value in the user_inputs file. If the user-defined load shape is not normalized (the sum of values across all 8,760 hours equals the annual MWh savings) the user should enter 1 in for the corresponding MWh savings in the user_inputs file.

## 2.3 Outputs

FLEXvalue outputs will include a single table if the **separate_output_tables** variable is set to False. Otherwise, FLEXvalue will output two tables.

The number of rows will depend on the column names passed into the **aggregation_columns** variable. For example, if a user passes "id" into the aggregation_columns variable, the user would expect a row for each project id. However, if the user passes in ["id", "year"], then the user should expect a row for every id and year permutation (in other words, the number of ids * number of years). In this example, the number of years would be dictated by the eul for each project.

Be advised that outputting a single table with electric and gas values aggregated to a time granularity that isn't shared by both meter types (for example, hour_of_year when gas data is only available at a monthly granularity) may result in surprising behavior, so be sure to examine the outputs.

In addition, some outputs may repeat at certain granularities. For example, **trc_costs** will repeat for every row, so summing this column will not deliver the desired result with an aggregation on hour_of_year.

Below are the outputs that can be expected for each table. If you have other input columns that you would like to pass through to the outputs, you can designate these column names in the **elec_addl_fields** and the **gas_addl_fields** variables. Additional avoided cost components can also be displayed in outputs if the column names are passed into the **elec_components** and **gas_components** variables, respectively.

### 2.3.1 Single Combined Table Outputs

- **id**: The id from the inputs.
- **trc_ratio**: The sum of total benefits (at a given level of aggregation) divided by the total trc costs for a given id.
- **pac_ratio**: The sum of total benefits (at a given level of aggregation) divided by the total pac costs for a given id.
- **electric_benefits**: The sum of electric benefits (at a given level of aggregation).
- **gas_benefits**: The sum of gas benefits (at a given level of aggregation).
- **total_benefits**: The sum of total benefits (at a given level of aggregation).
- **trc_costs**: The trc costs for a given id.
- **pac_costs**: The pac costs for a given id.
- **annual_net_mwh_savings**: The annual net (post net-to-gross) electric savings for a given id, in mWh.
- **lifecycle_net_mwh_savings**: The lifecycle net (post net-to-gross) electric savings for a given id, in mWh.
- **annual_net_therms_savings**: The annual net (post net-to-gross) gas savings for a given id, in therms.
- **lifecycle_net_therms_savings**: The lifecycle net (post net-to-gross) gas savings for a given id, in therms.
- **elec_avoided_ghg**: The lifecycle electric ghg savings (post net-to_gross) for a given id, in metric tons.
- **lifecycle_gas_ghg_savings**: The lifecycle gas ghg savings (post net-to_gross) for a given id, in metric tons.

### 2.3.2 Electric Table Outputs

- **id**: The id from the inputs.
- **trc_ratio**: The sum of total benefits (at a given level of aggregation) divided by the total trc costs for a given id.
- **pac_ratio**: The sum of total benefits (at a given level of aggregation) divided by the total pac costs for a given id.
- **electric_benefits**: The sum of electric benefits (at a given level of aggregation).
- **trc_costs**: The trc costs for a given id.
- **pac_costs**: The pac costs for a given id.
- **annual_net_mwh_savings**: The annual net (post net-to-gross) electric savings for a given id, in mWh.
- **lifecycle_net_mwh_savings**: The lifecycle net (post net-to-gross) electric savings for a given id, in mWh.
- **elec_avoided_ghg**: The lifecycle electric ghg savings (post net-to_gross) for a given id, in metric tons.

### 2.3.3 Gas Table Outputs

- **id**: The id from the inputs.
- **gas_benefits**: The sum of gas benefits (at a given level of aggregation).
- **annual_net_therms_savings**: The annual net (post net-to-gross) gas savings for a given id, in therms.
- **lifecycle_net_therms_savings**: The lifecycle net (post net-to-gross) gas savings for a given id, in therms.
- **lifecycle_gas_ghg_savings**: The lifecycle gas ghg savings (post net-to_gross) for a given id, in metric tons.

## 2.4 Installation from Source

### 2.4.1 Docker

```
docker-compose build

# for running the CLI commands
./flexvalue.sh --help
```

### 2.4.2 Local

```
pip install -e .

# for running cli commands
flexvalue --help
```

If you are calling these commands using the repo code and docker, replace *flexvalue* with *./flexvalue.sh*.

**Note** The requirements currently use *psycopg2-binary*. This will work with Docker, but if you want to use FLEXvalue with PostgreSQL locally outside of Docker, you will need to build psycopg2 from source. For more information, including instructions, please see `this page <https://www.psycopg.org/docs/install.html#psycopg-vs-psycopg-binary>`in the psycopg documentation.

## 2.5 Usage

FLEXvalue can be run in one of three ways:

- from the command line, passing in command-line flags that specify the behavior.
- from the command line, passing a command-line flag that points to a TOML config file, the contents of which specify the behavior.
- as a python library. In this case you would create an instance of `FLEXValueRun`, passing the configuration options as arguments to the constructor, then call the `run()` method.

### 2.5.1 Command-line arguments

FLEXValue uses the following command-line arguments. If `--config-file` is passed, the file it specifies is used for configuration and all other arguments are ignored.

- **–project-info-file**: Filepath to the project information file that is used to calculate results
- **–database-type**: One of ‘postgresql’, ‘bigquery’, or ‘sqlite’
- **–host**: The host for the postgresql database to which you are connecting.
- **–port**: The port for the postgresql database to which you are connecting.
- **–user**: The user for the postgresql database to which you are connecting.
- **–password**: The password for the postgresql database to which you are connecting.
- **–database**: The database for the postgresql database to which you are connecting.
- **–elec-av-costs-table**: Used when –database-type is bigquery. Specifies the electric avoided costs table. This table gets overwritten (not appended to). Must specify the dataset and the google project (if different than the –project argument).
- **–elec-load-shape-table**: Used when –database-type is bigquery. Specifies the electric load shape table. This table gets overwritten (not appended to). Must specify the dataset and the google project (if different than the –project argument).
- **–gas-av-costs-table**: Used when –database-type is bigquery. Specifies the gas avoided costs table. This table gets overwritten (not appended to). Must specify the dataset and the google project (if different than the –project argument).
- **–therms-profiles-table**: Used when –database-type is bigquery. Specifies the therms profiles table. This table gets overwritten (not appended to). Must specify the dataset and the google project (if different than the –project argument).
- **–project-info-table**: Used when –database-type is bigquery. Specifies the table containing the project information. This table gets overwritten (not appended to). Must specify the dataset and the google project (if different than the –project argument).
- **–project**: Used when –database-type is bigquery. Specifies the google project.
- **–output-table**: The database table to write output to. This table gets overwritten (not appended to). Must specify the dataset and the google project (if different than the –project argument).
- **–electric-output-table**: The database table to write electric output to, when separate_output=True. This table gets overwritten (not appended to). Must specify the dataset and the google project (if different than the –project argument).

- **–gas-output-table**: The database table to write gas output to, when separate_output=True. This table gets overwritten (not appended to). Must specify the dataset and the google project (if different than the –project argument).
- **–config-file**: Path to the toml configuration file.
- **–elec-av-costs-file**: Filepath to the electric avoided costs. Used when –database-type is not BigQuery to load this data into the database from a file.
- **–gas-av-costs-file**: Filepath to the gas avoided costs. Used when –database-type is not BigQuery to load this data into the database from a file.
- **–elec-load-shape-file**: Filepath to the hourly electric load shape file. Used when –database-type is not BigQuery to load this data into the database from a file.
- **–therms-profiles-file**: Filepath to the therms profiles file. Used when –database-type is not BigQuery to load this data into the database from a file.
- **–aggregation-columns**: Comma-separated list of field names on which to aggregate the query.
- **–reset-elec-load-shape**: Reset the data in the electric load shape table. This restores it to its contents prior to running FLEXvalue.,
- **–reset-elec-av-costs**: Reset the data in the electric avoided costs table. This restores it to its contents prior to running FLEXvalue.,
- **–reset-therms-profiles**: Reset the data in the therms profiles table. This restores it to its contents prior to running FLEXvalue.,
- **–reset-gas-av-costs**: Reset the data in the gas avoided costs table. This restores it to its contents prior to running FLEXvalue.,
- **–process-elec-load-shape**: Process (load/transform) the electric load shape data.,
- **–process-elec-av-costs**: Process (load/transform) the electric avoided costs data.,
- **–process-therms-profiles**: Process (load/transform) the therms profiles table.,
- **–process-gas-av-costs**: Process (load/transform) the gas avoided costs data.,
- **–elec-components**: Comma-separated list of electric avoided cost component field names,
- **–gas-components**: Comma-separated list of electric avoided cost component field names,
- **–elec-addl-fields**: Comma-separated list of additional fields from electric data to include in output,
- **–gas-addl-fields**: Comma-separated list of additional fields from gas data to include in output.
- **–use-value-curve-name-for-join**: Indicates that the project_info table and the electric avoided costs table use the value curve name. Defaults to false. See below for more information.

### 2.5.2 Config file

The config file uses the same arguments, but they look a litle different; hyphens are replaced with underscores, and the leading – is removed. The config file is in *TOML <https://toml.io/en/>*. Note that the [run] and [database] section headers are structural, not comments - the elements listed below those headers may not be in the other section or FLEXvalue will not run correctly (if at all). Basically, the database-related information (connection information, auth information) is in the [database] section and information related to the execution (whether to reset tables, whether to load files, the aggregation columns, etc.) are in the [run] section

Here is an example config.toml file if you are connecting to BigQuery:

```
[run]
output_table = "output_table"
process_elec_load_shape = false
process_elec_av_costs = false
process_therms_profiles = false
process_gas_av_costs = false
aggregation_columns = ["id", "hour_of_year", "year"]
reset_elec_load_shape = false
reset_elec_av_costs = false
reset_therms_profiles = false
reset_gas_av_costs = false
elec_components = ["energy", "losses", "ancillary_services", "capacity", "transmission",
↪"distribution"]
gas_components = ["market", "t_d", "environment", "btm_methane", "upstream_methane"]
separate_output_tables = True
electric_output_table = "example_dataset.hourly_electric_output"
gas_output_table = "example_dataset.hourly_gas_output"
use_value_curve_name_for_join = False

[database]
#credentials = ""
database_type = "bigquery"
project = "oeem-avdcosts-platform"
elec_av_costs_table = "example_dataset.full_ca_avoided_costs_2020acc_copy"
elec_load_shape_table = "example_dataset.ca_hourly_electric_load_shapes_horizontal_copy"
therms_profiles_table = "example_dataset.ca_monthly_therms_load_profiles_copy"
gas_av_costs_table = "example_dataset.full_ca_avoided_costs_2020acc_gas_copy"
project_info_table = "example_dataset.example_user_inputs_380"
```

Here is an example config file if you are connecting to Postgresql:

```
[run]
project_info_file = "example_user_inputs.csv"
elec_load_shape_file = "ca_hourly_electric_load_shapes.csv"
elec_av_costs_file = "full_ca_avoided_costs_2020acc.csv"
therms_profiles_file = "ca_monthly_therms_load_profiles.csv"
gas_av_costs_file = "full_ca_avoided_costs_2020acc_gas.csv"
output_table = "output_table"
process_elec_load_shape = false
process_elec_av_costs = false
process_therms_profiles = false
process_gas_av_costs = false
aggregation_columns = ["id", "hour_of_year", "year"]
reset_elec_load_shape = false
reset_elec_av_costs = false
reset_therms_profiles = false
reset_gas_av_costs = false
elec_addl_fields = ["hour_of_year", "utility", "region", "month", "quarter", "hour_of_day
↪", "discount"]
gas_addl_fields = ["total", "month", "quarter"]
use_value_curve_name_for_join = False

[database]
```

(continues on next page)

(continued from previous page)

```
database_type = "postgresql"
host = "postgresql"
port = 5432
user = "postgres"
password = "mypassword"
database = "postgres"
```

### 2.5.3 As a python library

Here's an example of calling FLEXvalue directly from python:

```
flex_value_run = FlexValueRun(
      database_type="bigquery",
      project="my-google-project",
      therms_profiles_table="my_target_dataset.ca_monthly_therms_load_profiles_copy",
      gas_av_costs_table="my_avoided_cost_project.my_avoided_cost_dataset.full_ca_
↪avoided_costs_2020acc_gas_copy",
      elec_av_costs_table="my_avoided_cost_project.my_avoided_cost_dataset.full_ca_
↪avoided_costs_2020acc_copy",
      elec_load_shape_table="my_target_dataset.ca_hourly_electric_load_shapes_horizontal_
↪copy",
      metered_load_shape_table="my_source_dataset.example_metered_load_shape",
      reset_elec_load_shape=True,
      process_elec_load_shape=True,
      process_metered_load_shape=True,
      project_info_table="example_project_info",
      output_table="example_output_table",
      aggregation_columns=["id"],
      separate_output_tables=False
      use_value_curve_name_for_join = False
  )
flex_value_run.run()
```

If the "process_X" flag is set, FLEXvalue will prepare that data for use in its main calculation. The meaning of "prepare" in this case depends on which database type you are using; if you are using BigQuery, it will do the following:

- Create new tables for the therms profiles and electric load shapes. These will then be populated from the relevant tables specified in the config
- Update the gas avoided costs table by adding and subsequently populating a timestamp column.
- The electric avoided cost table is not touched - this is a no-op, as the available avoided cost data is already in a format that FLEXvalue can use.

If you are using a relational database, the tables will be created if they don't exist, and then populated based on the data in the input files. The input files are csv files and have the columns described below. Header rows are required for the electrical load shape and therms profiles files. FLEXvalue attempts to determine the presence of header rows and will skip one if found and not needed. Note that FLEXvalue will NOT look up columns by the values in the header row - it's strictly positional.

If the "use_value_curve_name_for_join" flag is set to True, FLEXvalue will include the "value_curve_name" during the "join" step when matching project savings to the avoided cost curve value. This enables users to include multiple value curves in the same avoided costs table if desired. If this flag is set to False, FLEXvalue will ignore the "value_curve_name" columns in both the project inputs and the avoided cost tables.

## 2.6 Data stores

When using PostgreSQL, you must provide the following information:

- database_type - this must be set to "postgresql"
- host
- port
- user
- password
- database

The docker file included in this repository uses the default PostgreSQL 15.1 image. If you look in the docker-compose.yml file you can see the values to provide for those flags.

When using Google BigQuery, you must provide the following information:

- database_type - this must be set to "bigquery"
- project - this is the Google project for your input data

All parameters for tables (e.g. project_info_table, output_table) must include the dataset in the table name. If the dataset is in a different project than the one specified by the `project` parameter, you must include that in the table name as well.

Authentication is based on workload identity management.

## 2.7 License

This project is licensed under Apache 2.0.

### 2.7.1 Other resources

- MAINTAINERS: an ordered list of project maintainers.
- CHARTER: open source project charter.
- CODE_OF_CONDUCT: Code of conduct for contributors

# PYTHON MODULE INDEX

# INDEX